

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ

ДЕРЖАВНИЙ ЗАКЛАД «ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА»

Кваліфікаційна наукова праця
на правах рукопису

МАРКОВА Оксана Миколаївна

УДК [372.8::519.7]:378.4+004.77

ДИСЕРТАЦІЯ

ХМАРНІ ТЕХНОЛОГІЇ ЯК ЗАСІБ НАВЧАННЯ ОСНОВ МАТЕМАТИЧНОЇ ІНФОРМАТИКИ СТУДЕНТІВ ТЕХНІЧНИХ УНІВЕРСИТЕТІВ

13.00.10 – інформаційно-комунікаційні технології в освіті

01 – Освіта/Педагогіка

Подається на здобуття наукового ступеня кандидата педагогічних наук
Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ О. М. Маркова

Науковий керівник – Семеріков Сергій Олексійович, доктор педагогічних
наук, професор

Кривий Ріг – 2018

АНОТАЦІЯ

Маркова О. М. Хмарні технології як засіб навчання основ математичної інформатики студентів технічних університетів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата педагогічних наук за спеціальністю 13.00.10 – інформаційно-комунікаційні технології в освіті (01 – Освіта/Педагогіка). – Криворізький державний педагогічний університет Міністерства освіти і науки України. – Кривий Ріг, 2018.

Під час дослідження наукової проблеми використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів отримано такі основні результати: на підставі аналізу наукової літератури розкриті теоретичні засади навчання основ математичної інформатики студентів технічних університетів; досліджені можливості застосування засобів хмарних технологій у підготовці фахівців з інформаційних технологій; теоретично обґрунтовані методичні засади використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів; дібрані та класифіковані засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів; розроблена методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів та експериментально перевірена її ефективність.

За результатами аналізу наукової літератури, нормативно-законодавчих документів з'ясовано, що забезпечення інноваційного розвитку України можливе на основі комплексного використання в професійній діяльності та професійній підготовці фахівців з інформаційних технологій хмарних технологій і технологій математичної інформатики. Доведено ключову роль математичної інформатики в зменшенні розриву між швидкоплинними змінами в галузі ІКТ та професійною підготовкою фахівців з інформаційних технологій за рахунок застосування технології комп'ютерного моделювання та відповідних програмних засобів навчання, насамперед – систем комп'ютерної математики.

Визначено компетентність з математичної інформатики майбутніх фахівців з інформаційних технологій як особистісно-професійне утворення, яке включає систему знань, умінь, навичок, досвід навчально-дослідницької діяльності з математичної інформатики та позитивне ціннісне ставлення до неї й виявляється в готовності та здатності до модифікації наявних і розробки нових інформаційних технологій на основі моделей і методів математичної інформатики.

Ключове поняття дослідження – хмарні технології (хмарні ІКТ) – визначено як сукупність методів, засобів і прийомів, використовуваних для збирання, систематизації, зберігання та опрацювання на віддалених серверах, передавання через мережу та подання через клієнтську програму всеможливих повідомлень і даних. Відповідно під хмарними технологіями навчання в дослідженні розуміються такі ІКТ навчання, що передбачають використання мережних ІКТ із централізованим мережним зберіганням та опрацюванням даних (виконання програм), за якого користувач виступає клієнтом (користувачем послуг), а «хмара» – сервером (постачальником послуг). Витоки хмарних технологій навчання містяться в застосуванні концепції «комп'ютерної послуги» до освітнього процесу, зокрема наданні місця для зберігання електронних освітніх ресурсів та мобільного доступу до них.

Аналіз вітчизняного та зарубіжного досвіду використання хмарних технологій у підготовці ІТ-фахівців надав можливість обґрунтувати доцільність застосування в навчанні інформатичних дисциплін таких моделей надання хмарних послуг: «програмне забезпечення як послуга», «платформа як послуга» та «інфраструктура як послуга» на основі інформатичної технології віртуальних машин та педагогічної технології дистанційного навчання.

З'ясовано, що розвиток хмарних технологій навчання відзначався неперервністю, взаємозумовленістю та інноваційністю й зумовлений імплементацією положень концепції «комп'ютерної послуги» в організацію освітнього процесу через надання місця для зберігання електронних освітніх ресурсів та мобільного доступу до них.

Доведено, що однією з явних переваг використання хмарних технологій у підготовці майбутніх ІТ-фахівців у технічних університетах є можливість застосування сучасних засобів паралельного програмування як основи хмарних технологій.

З урахуванням теоретично обґрунтованих методичних засад використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів визначено: зміст навчання спецкурсу «Основи математичної інформатики» (змістові модулі «Теорія алгоритмів», «Методи обчислень», «Теорія кодування», «Основи криптографії») для майбутніх фахівців з інформаційних технологій, цілі та технологію навчання, що об'єднує форми організації освітнього процесу й методи навчання (лекції, лабораторні заняття, самостійна робота студентів, навчальні конференції, консультації, індивідуальні заняття, навчально-дослідницькі проекти, контрольні заходи), а також засоби навчання основ математичної інформатики, провідними з яких є засоби хмарних технологій.

У дисертації виокремлено засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів як сукупність хмаро зорієнтованих електронних освітніх ресурсів, що застосовуються для інформаційно-процесуального забезпечення виконання дидактичних завдань або їхніх фрагментів та спрямовані на реалізацію цілей навчання основ математичної інформатики і сприяють формуванню компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій.

Класифіковано засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів на 11 основних (хмаро зорієнтовані програмно-методичні матеріали, тестові системи, тренажери, практикуми, дидактичні демонстраційні матеріали, навчальні посібники та підручники, електронні довідники, системи підтримки навчання, навчальні лабораторії, системи комп'ютерної математики, середовища програмування) та 10 допоміжних (хмаро зорієнтовані додаткові науково-навчальні матеріали, електронна пошта, засоби аудіо- та відеозв'язку, операційні системи, засоби

зберігання даних, текстові процесори, табличні процесори, засоби підготовки презентацій, системи управління базами даних, додаткові хмаро зорієнтовані компоненти).

Доведено, що найбільш значущим засобом хмарних технологій навчання основ математичної інформатики є універсальні хмаро зорієнтовані системи комп'ютерної математики, такі як CoCalc, що інтегрують більшість виокремлених засобів.

Методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів складається із цільового (формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій), змістового (навчання основ математичної інформатики) і технологічного (хмаро зорієнтовані засоби ІКТ, методи та форми їх використання в навчанні математичної інформатики) блоків.

Експериментальна перевірка розробленої методики у формі паралельного педагогічного експерименту та результати статистичного опрацювання отриманих даних підтвердили припущення щодо відсутності в розподілі студентів контрольної та експериментальної груп суттєвих розбіжностей на початку експерименту за рівнями навчальних досягнень ($\chi^2_{\text{емп}} = 3,89$) та засвідчили наявність значущих на рівні 0,01 розбіжностей за рівнями навчальних досягнень ($\chi^2_{\text{емп}} = 20,06$) після завершення педагогічного експерименту, що підтвердило гіпотезу дослідження.

Наукова новизна та теоретичне значення одержаних результатів полягає в тому, що: вперше теоретично обґрунтовано та розроблено методичні засади використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів, дібрано та класифіковано засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів; удосконалено форми, методи та засоби навчання математичної інформатики студентів закладів вищої освіти шляхом розробки та впровадження хмаро зорієнтованих засобів навчання; дістали подальшого розвитку теорія та методика застосування програмних засобів

інформатизації освіти, проектування та розвитку хмаро зорієнтованого навчального середовища, зміст методичних систем навчання інформатичних дисциплін та інформаційно-комунікаційних технологій у різних галузях освіти.

Практичне значення одержаних результатів полягає в тому, що розроблено та впроваджено в навчально-виховний процес закладів вищої освіти: методику використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів, що складається із цільового (формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій), змістового (навчання основ математичної інформатики) та технологічного (хмаро зорієнтовані засоби ІКТ, методи та форми їх використання в навчанні математичної інформатики) блоків; програму спецкурсу «Основи математичної інформатики» для студентів спеціальності 123 «Комп'ютерна інженерія»; хмаро зорієнтовані програмно-методичні матеріали навчання основ математичної інформатики (режим доступу: <http://site.mathinfo.ccjournals.eu/>).

Виконане дослідження не вичерпує всіх аспектів аналізованої проблеми. Подальші наукові пошуки її розв'язання доцільні за такими напрямками: проектування хмаро зорієнтованого середовища навчання майбутніх фахівців з комп'ютерної інженерії; тенденції розвитку хмарних технологій у професійній підготовці та перепідготовці ІТ-фахівців; методика формування дослідницьких компетентностей майбутніх фахівців з інженерії програмного забезпечення засобами хмарних технологій.

Результати дослідження можуть бути використані в системі професійної підготовки майбутніх фахівців з інформаційних технологій у закладах вищої освіти, у системі післядипломної освіти, у самоосвітній діяльності студентів.

Ключові слова: хмарні технології, хмаро зорієнтовані засоби навчання, математична інформатика, студенти технічних університетів, методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

Markova O. M. Cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical universities. – Qualifying scientific work on the rights of manuscript.

Thesis for the degree of candidate of pedagogical science, in specialty 13.00.10 – Information and Communication Technologies in Education (01 – Education/Pedagogics). – Kryvyi Rih State Pedagogical University of the Ministry of Education and Science of Ukraine. – Kryvyi Rih, 2018.

During research of the scientific problem of using cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical universities, the following main results were obtained: based on the analysis of the scientific literature, the theoretical foundations of the learning of the foundations of mathematical informatics for students of technical universities were discovered and disclosed; the possibility of using cloud technology in the training of future IT specialists are explored; the methodical foundations of using cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical universities have are theoretically grounded; the tools of cloud technologies for the learning of the foundations of mathematical informatics for students of technical universities are selected and systemized; methodic of using cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical universities was developed and experimentally tested its effectiveness.

According to the results of the analysis of scientific literature, normative-legislative documents it was clarified that the provision of innovative development of Ukraine is possible on the basis of integrated use of cloud technologies and technologies of mathematical informatics in professional activity and professional training of IT specialists. The key role of mathematical informatics in the reduction of the gap between transient changes in the field of ICT and the professional training of IT specialists has been demonstrated through the use of computer simulation technology and related software tools, primarily computer mathematics systems.

The competence in mathematical informatics of future IT specialists as a personality-professional formation is determined, which includes the system of

knowledge, skills, experience of educational and research activity in mathematical informatics and positive value attitude towards it, and is manifested in readiness and ability to modify existing and development new information technologies based on models and methods of mathematical informatics.

The key concept of the research – cloud technologies (cloud ICT) – was defined as a set of methods, tools and techniques used to collect, organize, store and process on remote servers, transmission over the network and presentation through the client program of all possible messages and data. Accordingly, cloud learning technologies are such learning ICT that involves the use of networked ICTs with centralized network storage and processing of data (execution of programs), in which the user acts as a client (user of services), and the cloud is a server (service provider). The origins of cloud learning technologies are contained in the application of the concept of computer services to the educational process, in particular, the provision of space for the storage of electronic educational resources and mobile access to them.

An analysis of the domestic and foreign experience of using cloud technologies in the training of IT specialists provided an opportunity to substantiate the feasibility of using such models of cloud services as “Software as a service”, “Platform as a service” and “Infrastructure as a service” on the basis of informatics technology of virtual machines and pedagogical technology of distance learning.

It was revealed that the development of cloud technologies of training was characterized by continuity, interdependence and innovation, and due to the implementation of the provisions of the concept of “computer service” in organizing the educational process through the provision of space for the storage of electronic educational resources and mobile access to them.

It is shown that one of the obvious advantages of using cloud technologies in the training of future IT specialists in technical universities is the possibility of using modern tools of parallel programming as the basis of cloud technologies.

Taking into account the theoretically grounded methodical principles of using cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical universities, were identified: the content of learning the

special course “Foundations of mathematical informatics” (content modules “Theory of algorithms”, “Numerical methods”, “Coding theory”, “Fundamentals of cryptography”) for future IT professionals, goals and learning technology, combining the forms of organization of educational process and teaching methods (lectures, laboratory classes, independent students’ work, study conferences, consultations, individual classes, research projects, control activities), as well as the learning tools of the foundations of mathematical informatics, the leading of which are the cloud technologies tools.

The dissertation outlines the cloud technologies tools for the learning of the foundations of mathematical informatics for the students of technical universities as a set of cloud-oriented electronic educational resources that are used for informational and procedural maintenance of the implementation of didactic tasks or their fragments and aimed at the realization of the purposes of studying the foundations of mathematical informatics.

The tools of cloud technologies for the learning of the foundations of mathematical informatics for students of technical universities are systemized, of which 11 core assets (cloud-oriented program and methodic materials, test systems, simulators, workshops, didactic demonstration materials, tutorials and textbooks, electronic guides, learning management systems, learning laboratories, computer mathematics systems, programming environments) and 10 – auxiliary (cloud-based additional scientific and learning materials, e-mail, audio and video communication tools, operating systems, data storage devices, word processors, electronic spreadsheets, presentation tools, database management systems, additional cloud-based components).

It is proved that the most important tool of cloud technologies for the training of the foundations of mathematical informatics are universal cloud-oriented computer mathematics systems, such as CoCalc, which integrate most of the separated tools.

The methodic of using cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical universities consists of a target block (the formation of competences in mathematical informatics of future IT

professionals), content block (learning of the foundations of mathematical informatics) and technological block (cloud-based ICT tools, methods and forms of their use in learning mathematical informatics).

The experimental verification of the developed methodic in the form of a parallel pedagogical experiment and the results of the statistical processing of the its data confirmed the assumption that there were no significant differences in the distribution of students of the control and experimental groups at the beginning of the experiment by the levels of academic achievement ($\chi^2_{\text{emp}} = 3,89$) and showed the presence of significant at level 0,01 differences by the levels of academic achievement ($\chi^2_{\text{emp}} = 20,06$) after the completion of the pedagogical experiment, which confirmed the hypothesis of the study.

The scientific novelty and theoretical significance of the obtained results is that for the first time the methodical foundations of using cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical universities have are theoretically grounded and developed, the tools of cloud technologies for the learning of the foundations of mathematical informatics for students of technical universities are selected and systemized; the forms, methods and learning tools of the mathematical informatics for students of technical universities have been improved through the development and implementation of cloud-oriented teaching aids; the theory and methodic of using software tools for informatization of education, design and development a cloud-based learning environment, the content of methodical systems of learning informatics and ICT in various branches of education got further development.

The practical significance of the results is that next components of the educational process of institutions of higher education has been developed and implemented: methodic of using cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical, consisting of the target (formation of competence in mathematical informatics of future IT specialists), content (training of the foundations of mathematical informatics) and technological (cloud-based ICT tools, methods and forms of their use in the training of

mathematical informatics) blocks; program of the special course “Fundamentals of Mathematical Informatics” for students of specialty 123 “Computer Engineering”; cloud-based program and methodical materials for the study of the foundations of mathematical informatics (access mode: <http://site.mathinfo.ccjournals.eu/>).

The study does not exhaust all aspects of the problem under consideration. Further scientific researches of its solution are expedient in following directions: designing of the cloud-based learning environment of future specialists in computer engineering; trends in the development of cloud technologies in the training and retraining of IT specialists; a methodic of developing the research competencies of future specialists in software engineering using cloud technologies.

The research results can be used in the system of professional training of future IT specialists in higher education institutions, in the system of postgraduate education, in self-education of students.

Keywords: cloud technologies, cloud-based learning tools, mathematical informatics, students of technical universities, methodic of using cloud technologies as a learning tool of the foundations of mathematical informatics for students of technical universities.

Список публікацій здобувача за темою дисертації

1. Туравініна О. М. Математика-0 для педагогічних ВНЗ: вступ до булевої алгебри / Наталія Василенко, Володимир Петров, Оксана Туравініна // Математика в школі. – 2008. – № 3. – С. 16-21.

2. Туравініна О. М. Методології навчання за напрямком «Комп’ютерна інженерія» на базі Криворізького технічного університету / А. І. Купін, В. А. Чубаров, О. М. Туравініна // Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія № 5. Педагогічні науки: реалії та перспективи. – Випуск 17 : збірник наукових праць / За ред. В. Д. Сиротюка. – К. : Вид-во НПУ імені М. П. Драгоманова, 2009. – С. 102-108.

3. Туравініна О. М. Математична інформатика у системі фундаменталізації навчання студентів технічних університетів /

О. М. Туравініна // Збірник наукових праць Кам'янець-Подільського національного університету імені Івана Огієнка. Серія педагогічна / [редкол. : П. С. Атаманчук (голова, наук. ред. та ін.)]. – Кам'янець-Подільський : Кам'янець-Подільський національний університет ім. І. Огієнка, 2012. – Випуск 18 : Інновації в навчанні фізики: навчальний та міжнародний досвід. – С. 189-191.

4. Маркова О. М. Хмарні технології навчання: витoki [Електронний ресурс] / Маркова Оксана Миколаївна, Семеріков Сергій Олександрович, Стрюк Андрій Миколайович // Інформаційні технології і засоби навчання. – 2015. – Том 46, № 2. – С. 29-44. – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/download/1234/913>.

5. Маркова О. М. Модель методичної системи та цілі навчання основ математичної інформатики студентів технічних університетів / Маркова О. М. // Вісник Черкаського університету. Серія Педагогічні науки. – 2016. – № 7. – С. 36-42.

6. Маркова О. М. Моделі використання хмарних технологій у підготовці ІТ-фахівців / Маркова О. М. // Науковий часопис НПУ імені М. П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання : зб. наук. праць / Редрада. – К. : НПУ імені М. П. Драгоманова, 2016. – № 18 (25). – С. 85-94.

7. Markova O. M. The tools of cloud technology for learning of fundamentals of mathematical informatics for students of technical universities [Electronic resource] / Oksana M. Markova // Cloud Technologies in Education : Proceedings of the 5th Workshop on Cloud Technologies in Education (CTE 2017). Kryvyi Rih, Ukraine, April 28, 2017 / Edited by : Serhiy O. Semerikov, Mariya P. Shyshkina. – P. 27-33. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2168). – Access mode : <http://ceur-ws.org/Vol-2168/paper5.pdf>.

8. Markova O. M. CoCalc as a Learning Tool for Neural Network Simulation in the Special Course “Foundations of Mathematic Informatics” [Electronic resource] / Oksana Markova, Serhiy Semerikov, Maiia Popel // ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer 2018 :

Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops (ICTERI, 2018). Kyiv, Ukraine, May 14-17, 2018 / Edited by : Vadim Ermolayev, Mari Carmen Suárez-Figueroa, Vitaliy Yakovyna, Vyacheslav Kharchenko, Vitaliy Kobets, Hennadiy Kravtsov, Vladimir Peschanenko, Yaroslav Prytula, Mykola Nikitchenko, Aleksander Spivakovsky. – P. 388-403. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2104). – Access mode : http://ceur-ws.org/Vol-2104/paper_204.pdf.

9. Туравініна О. М. Застосування інформаційних технологій для підготовки спеціалістів за напрямом «Комп'ютерна інженерія» / А. І. Купін, В. А. Чубаров, О. М. Туравініна // Новітні комп'ютерні технології : матеріали VII Міжнародної науково-технічної конференції : Київ-Севастополь, 15-18 вересня 2009 р. – К. : Міністерство регіонального розвитку та будівництва України, 2009. – С. 101-102.

10. Туравініна О. М. Застосування мобільних засобів навчання математичної інформатики в курсі фізики / О. М. Туравініна // Тези доповідей VII Всеукраїнської науково-практичної конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2010) : Черкаси, 4-6 травня 2010 р. – У 2-х томах. – Черкаси : ЧДТУ, 2010. – Т. 2. – С. 74.

11. Туравініна О. М. До питання про розробку методики навчання математичної інформатики у технічному ВНЗ / О. М. Туравініна, І. О. Теплицький, І. І. Ліннік // Новітні комп'ютерні технології : матеріали VIII Міжнародної науково-технічної конференції : Київ-Севастополь, 14-17 вересня 2010 р. – К. : Міністерство регіонального розвитку та будівництва України, 2010. – С. 168-169.

12. Туравініна О. М. Засоби хмарних обчислень у комп'ютерній інженерії / С. А. Бурма, О. М. Туравініна // Комп'ютерні інтелектуальні системи та мережі : матеріали IV Всеукраїнської конференції аспірантів, студентів та молодих вчених (23-25 березня 2011 р.) / Міністерство освіти і науки України, Криворізький технічний університет, Кафедра комп'ютерних систем та мереж.

– Кривий Ріг : Видавництво Криворізького технічного університету, 2011. – С. 33-34.

13. Туравініна О. М. Amazon EC2 як платформа для організації хмарних обчислень / О. М. Туравініна, А. М. Стрюк, Н. В. Рашевська, К. І. Словак // Новітні комп'ютерні технології : матеріали ІХ Міжнародної науково-технічної конференції : Київ-Севастополь, 13-16 вересня 2011 р. – К. : Міністерство регіонального розвитку та будівництва України, 2011. – С. 187-188.

14. Туравініна О. М. Використання хмарних обчислень у комбінованому навчанні системного програмування / А. М. Стрюк, О. М. Туравініна // Тези доповідей Міжнародної науково-практичної конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2012) : Черкаси, 25-27 квітня 2012 р. – У 2 т. – Черкаси : ЧДТУ, 2012. – Т. 2. – С. 96-97.

15. Туравініна О. М. Хмарні технології навчання студентів / О. М. Туравініна // Новітні комп'ютерні технології : матеріали Х Міжнародної науково-технічної конференції : Севастополь, 11-14 вересня 2012 р. – К. : Міністерство регіонального розвитку та будівництва України, 2012. – С. 119-121.

16. Туравініна О. М. Зміст навчання основ математичної інформатики студентів технічних університетів / О. М. Туравініна, С. О. Семеріков // Матеріали міжнародної науково-методичної конференції «Розвиток інтелектуальних умінь і творчих здібностей учнів та студентів у процесі навчання дисциплін природничо-математичного циклу» (ІТМ*плюс - 2012), м. Суми, 6-7 грудня 2012 р. – Суми : Вид-во СумДПУ ім. А. С. Макаренка, 2012. – С. 142-145.

17. Туравініна О. М. Хмарні технології навчання у системі інформаційно-комунікаційних технологій навчального призначення / О. М. Туравініна // Хмарні технології в освіті : матеріали Всеукраїнського науково-методичного Інтернет-семінару (Кривий Ріг – Київ – Черкаси – Харків, 21 грудня 2012 р.). – Кривий Ріг : Видавничий відділ КМІ, 2012. – С. 9.

18. Туравініна О. М. Sagemaths як хмарний засіб реалізації основних

чисельних методів / Формус Ю. В., Туравініна О. М. // Комп'ютерні інтелектуальні системи та мережі : матеріали VII Всеукраїнської науково-практичної WEB конференції аспірантів, студентів та молодих вчених (25-27 березня 2014 р.) / Міністерство освіти і науки України, ДВНЗ «Криворізький національний університет», Кафедра комп'ютерних систем та мереж. – Кривий Ріг : Видавництво Криворізького національного університету, 2014. – С. 119-120.

19. Маркова О. М. Історичні аспекти розвитку хмарних технологій / Маркова О. М. // Тези доповідей науково-практичного семінару «Хмарні технології в сучасному університеті» (ХТСУ-2015) : Черкаси, 24 березня 2015 р. / Міністерство освіти і науки України, Черкаський державний технологічний університет, Інститут інформаційних технологій і засобів навчання НАПН України, Криворізький національний університет, Національний педагогічний університет імені М. П. Драгоманова, Черкаський національний університет імені Богдана Хмельницького. – Черкаси : ЧДТУ, 2015. – С. 29-30.

20. Маркова О. М. Хмарні обчислення і технології: тенденції розвитку / Халимова Б. А., Маркова О. М. // Комп'ютерні інтелектуальні системи та мережі : матеріали IX Всеукраїнської науково-практичної WEB конференції аспірантів, студентів та молодих вчених (22-24 березня 2016 р.) / Міністерство освіти і науки України, ДВНЗ «Криворізький національний університет», Кафедра комп'ютерних систем та мереж. – Кривий Ріг : Видавничий центр ДВНЗ «Криворізький національний університет», 2016. – С. 25-26.

21. Маркова О. М. Хмарні технології навчання: спроба визначення / О. М. Маркова // Новітні комп'ютерні технології. – Кривий Ріг : ДВНЗ «Криворізький національний університет». – 2014. – Том XII : спецвипуск «Хмарні технології в освіті». – С. 244-248.

22. Маркова О. М. Теоретичні засади навчання основ математичної інформатики студентів технічних університетів з використанням хмарних технологій / О. М. Маркова // Новітні комп'ютерні технології. – Кривий Ріг : ДВНЗ «Криворізький національний університет». – 2016. – Том XIV. – С. 63-64.

23. Маркова О. М. Загальна структура засобів хмарних технологій навчання основ математичної інформатики / Оксана Миколаївна Маркова // Новітні комп'ютерні технології. – Кривий Ріг : ДВНЗ «Криворізький національний університет». – 2017. – Том XV. – С. 231-234.

24. Маркова О. М. SageMathCloud як засіб хмарних технологій комп'ютерно-орієнтованого навчання математичних та інформатичних дисциплін / Шокалюк С. В., Маркова О. М., Семеріков С. О. // Моделювання в освіті: Стан. Проблеми. Перспективи : монографія / Міністерство освіти і науки України, ДВНЗ «Криворізький державний педагогічний університет», Черкаський національний університет імені Богдана Хмельницького, Південноукраїнський національний педагогічний університет імені К. Д. Ушинського ; за заг. ред. Соловйова В. М. – Черкаси : Брама, видавець Вовчок О.Ю., 2017. – С. 130-142.

ЗМІСТ

АНОТАЦІЯ.....	2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	19
ВСТУП.....	20
РОЗДІЛ 1 ПРОБЛЕМА ВИКОРИСТАННЯ ХМАРНИХ ТЕХНОЛОГІЙ ЯК ЗАСОБУ НАВЧАННЯ ОСНОВ МАТЕМАТИЧНОЇ ІНФОРМАТИКИ СТУДЕНТІВ ТЕХНІЧНИХ УНІВЕРСИТЕТІВ.....	32
1.1 Математична інформатика у професійній підготовці студентів технічних університетів.....	32
1.2 Поняття про хмарні технології навчання.....	52
1.3 Досвід використання хмарних технологій у підготовці фахівців з інформаційних технологій	70
Висновки до розділу 1	98
РОЗДІЛ 2 МЕТОДИЧНІ ЗАСАДИ ВИКОРИСТАННЯ ХМАРНИХ ТЕХНОЛОГІЙ ЯК ЗАСОБУ НАВЧАННЯ ОСНОВ МАТЕМАТИЧНОЇ ІНФОРМАТИКИ СТУДЕНТІВ ТЕХНІЧНИХ УНІВЕРСИТЕТІВ	101
2.1 Цілі та зміст навчання основ математичної інформатики студентів технічних університетів.....	101
2.2 Форми організації освітнього процесу, методи та засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів.....	114
2.3 Методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.....	127
2.4 Організація, проведення і результати педагогічного експерименту	213
Висновки до розділу 2	224
ВИСНОВКИ	228
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	232
ДОДАТКИ.....	267
Додаток А Статистичні відомості про підготовку фахівців з інформаційних	

технологій	268
Додаток Б Робоча навчальна програма зі спецкурсу «Математична інформатика»	271
Додаток В Доведення правильності алгоритму Евкліда у системі Аxiom....	288
Додаток Г Демонстрація методу Гауса для розв'язання систем лінійних алгебраїчних рівнянь	307
Додаток Д Список публікацій за темою дисертації та відомості про апробацію результатів дисертації.....	313
Додаток Е Список вищих навчальних закладів та установ, у яких упроваджено результати дослідження	320

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Web-СКМ	Web-система комп'ютерної математики
ЗВО	заклад вищої освіти
ІКТ	інформаційно-комунікаційні технології
ІКТН	інформаційно-комунікаційні технології навчання
ІТ	інформаційні технології
КСМ	комп'ютерні системи та мережі
НСД	найбільший спільний дільник
ХЗ ЕОР	хмаро зорієнтовані електронні освітні ресурси
ЮНЕСКО	Організація Об'єднаних Націй з питань освіти, науки і культури

ВСТУП

Актуальність дослідження. Системним викликом для наукової спільноти є четверта промислова революція (Індустрія 4.0), актуалізована засновником Всесвітнього економічного форуму К. Швабом (K. Schwab) [258]. Визначальною характеристикою Індустрії 4.0 є еволюція та конвергенція нано-, біо-, інформаційних і когнітивних технологій для забезпечення якісних перетворень в економічній, соціальній, культурній та гуманітарній сферах. Від тих, хто професійно займається розробкою та впровадженням технологій шостого технологічного укладу, значною мірою залежить, чи зможе наша країна осідлати хвилю інновацій Індустрії 4.0. Тому пріоритетним напрямом модернізації вищої технічної освіти України є широке впровадження інформаційно-комунікаційних технологій (ІКТ) для формування професійно компетентної особистості інженера, здатного до забезпечення випереджального інноваційного розвитку країни.

Згідно з Постановою Кабінету Міністрів України «Деякі питання визначення середньострокових пріоритетних напрямів інноваційної діяльності загальнодержавного рівня на 2017-2021 роки» (2016) [107], одним із найбільш державно та суспільно значущих напрямів інноваційної діяльності є розвиток сучасних ІКТ та робототехніки, зокрема хмарних технологій, комп'ютерних навчальних систем і технологій математичної інформатики – інтелектуального моделювання, інформаційної безпеки, довгострокового зберігання даних та управління «великими даними», систем штучного інтелекту.

У Постанові Кабінету Міністрів України «Деякі питання визначення середньострокових пріоритетних напрямів інноваційної діяльності загальнодержавного рівня на 2017-2021 роки» (2017) [108] уточнено, що ці напрями разом з інтелектуальними веб-технологіями та хмарними обчисленнями є основою і для формування та визначення тематики наукових досліджень і науково-технічних (експериментальних) розробок, і для формування державного замовлення на підготовку фахівців з інформаційних технологій у технічних університетах.

У проєкті Європейської Комісії «ІКТ-компетентності для хмарних технологій, кібербезпеки та «зелених» ІКТ» (e-Skills for Cloud Computing, Cyber-security and Green IT) [26] наголошено на необхідності модернізації підготовки фахівців з метою подолання розриву між високою швидкістю змін ІКТ та відповідними компетентностями випускників на основі комплексного використання хмарних технологій.

Питанням інформатизації освіти присвячено праці українських та зарубіжних дослідників В. Ю. Бикова [87], К. Дж. Бонка (Curtis J. Bonk) [10], Ш. Ву (S. Wu) [77], А. М. Гуржія [106], А. Джайн (Anjali Jain) [41], М. І. Жалдака [118], В. М. Кухаренка [22], Н. В. Морзе [185], В. В. Олійника [189], Л. Ф. Панченко [193], Б. Раденкович (Božidar Radenković) [63], З. С. Сейдаметової [212], С. О. Семерікова [134], О. В. Співаковського [225], О. М. Спіріна [87], Ю. В. Триуса [238], П. Цао (P. Cao) [65], А. В. Яцишин [269] та ін.

Аналіз проблем ІТ-освіти, розробка теоретичних і методичних аспектів навчання математичної інформатики знайшли відображення в наукових розвідках Ф. Л. Бауера (Friedrich Ludwig "Fritz" Bauer) [8], В. Ю. Бикова [86], Н. О. Бугаєць [91], Е. Бьоргера (Egon Börger) [11], В. М. Глушкова [104], Г. Гооза (Gerhard Goos) [84], Е. В. Дейкстри (Edsger Wybe Dijkstra) [24], М. І. Жалдака [115], Т. П. Кобильника [138], В. С. Круглика [150], О. С. Меняйленка [172], Х. Д. Міллса (Harlan D. Mills) [156], Н. В. Морзе [184], А. Дж. Перліса (Alan Jay Perlis) [60], А. О. Ричкової [209], З. С. Сейдаметової [211], О. В. Співаковського [70], Ю. В. Триуса [237], Ч. Е. Р. Хоора (Charles Antony Richard Hoare) [37], С. М. Яшанова [270].

Дослідженню стану та перспектив використання хмарних технологій у закладах освіти присвячено роботи В. Ю. Бикова [235], К. В. Болгової [89], К. В. Власенко [96], І. С. Войтовича [216], О. О. Жугастрова [74], В. П. Іваннікова [123], А. М. Кобиліна [135], А. В. Колесникова [142], О. М. Кривоноса [149], С. Г. Литвинової [158], Н. В. Морзе [183], Л. Ф. Панченко [194], С. О. Семерікова [173], Л. Е. Соколової [223],

О. М. Спіріна [179], А. М. Стрюка [227], Ю. В. Триуса [238], М. П. Шишкіної [263]. Зарубіжний досвід використання хмарних технологій у навчанні майбутніх ІТ-фахівців репрезентують студії Е. А. Альдахіль (Eman A. Aldakheel) [4], К. Н. Булла (Christopher Neil Bull) [14], М. Вонга (Minjuan Wang) [75], В. Діаз (Veronica Diaz) [23], П. К. Паула (Prantosh Kr. Paul) [58], Х. Раджая (Hassan Rajaei) [64], Я. Соммервілля (Ian Sommerville) [69], У. Стейна (William Stein) [72].

У дисертаціях останніх років представлено різні аспекти впровадження хмарних технологій у навчально-виховний процес закладів освіти. Так, хмарні технології розглядалися як засіб: формування професійних компетентностей вчителя математики (М. В. Попель [198]), ІКТ-компетентностей майбутніх учителів інформатики (В. Г. Шевченко [259]), економістів та менеджерів (Л. С. Галкіна [102]), самоосвітньої компетентності майбутніх фахівців з інформаційних технологій (Т. В. Волошина [99]), компетентності в галузі застосування інструментальних засобів розробки інформаційних систем (М. В. Ступіна [229]), дослідницьких компетентностей старшокласників у процесі профільного навчання фізики (О. В. Мерзликін [174]); управління документообігом у закладах післядипломної педагогічної освіти (С. П. Касьян [131]); хмаро зорієнтовані середовища досліджували С. В. Палій [192] (інформаційно-організаційне середовище довшівської підготовки студентів), С. Г. Литвинова [159] (навчальне середовище загальноосвітнього навчального закладу), М. П. Шишкіна [262] (освітньо-наукове середовище закладу вищої освіти); методики навчання із використанням хмарних технологій розробляли Г. А. Алексанян [78] (математика), Н. В. Скриннік [221] (українська література), М. В. Хомутенко [250] (фізика), О. В. Коротун [146] та С. Н. Сейтвелієва [217] (інформатика). Проте питання використання хмарних технологій у навчанні математичної інформатики студентів технічних спеціальностей не ставало предметом окремого дослідження.

Вивчення нормативно-правових документів, теоретичних напрацювань проблеми дослідження та досвіду практичного впровадження хмарних

технологій у навчально-виховний процес закладів освіти, зокрема технічних університетів, дозволило виокремити такі *суперечності* між: потребою у фундаменталізації професійної підготовки майбутніх ІТ-фахівців та швидкою зміною інформаційних технологій у їхній професійній діяльності; компетентністю з математичної інформатики як основи інноваційної діяльності в галузі інформаційних технологій та нерозробленістю способів її систематичного формування в студентів технічних університетів; значним потенціалом використання хмарних технологій у навчанні основ математичної інформатики майбутніх ІТ-фахівців та нерозробленістю методики їхнього використання як засобу навчання основ математичної інформатики студентів технічних університетів.

Актуальність проблеми використання хмарних технологій у закладах вищої освіти, її недостатня теоретико-методична розробленість, виокремлені суперечності та велика практична значущість зумовили вибір теми дослідження: **«Хмарні технології як засіб навчання основ математичної інформатики студентів технічних університетів».**

Зв'язок роботи з науковими програмами, планами, темами. Дисертацію виконано в межах комплексної теми «Теоретико-методичні основи використання мобільних інформаційно-комунікаційних технологій у навчанні» (ДР № 0116U001867) відповідно до плану роботи спільної науково-дослідної лабораторії з питань використання хмарних технологій в освіті ДВНЗ «Криворізький національний університет» та Інституту інформаційних технологій і засобів навчання НАПН України (м. Київ). Тему затверджено на засіданні Вченої ради ДВНЗ «Криворізький національний університет» (протокол № 2 від 31 жовтня 2017 року).

Об'єкт дослідження – упровадження хмарних технологій в освітній процес технічних університетів.

Предмет дослідження – методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

Мета дослідження – теоретичне обґрунтування, розробка й експериментальна перевірка методики використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

Гіпотеза дослідження полягає в припущенні, що методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів сприятиме підвищенню рівня їхніх навчальних досягнень за умови:

– теоретичного обґрунтування методичних засад використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів;

– здійснення добору та класифікації засобів хмарних технологій навчання основ математичної інформатики студентів технічних університетів;

– упровадження технологій комбінованого та мультистратегійного навчання;

– розробки хмаро зорієнтованих навчально-методичних матеріалів.

Відповідно до мети й гіпотези визначено такі основні **завдання дослідження**:

1. На підставі аналізу наукової літератури розкрити теоретичні засади навчання основ математичної інформатики студентів технічних університетів.

2. Дослідити можливості застосування засобів хмарних технологій у підготовці фахівців з інформаційних технологій.

3. Теоретично обґрунтувати методичні засади використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

4. Дібрати та класифікувати засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів.

5. Розробити та експериментально перевірити ефективність методики використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

Для розв'язання поставлених завдань застосовано такі **методи дослідження**: *теоретичні* – аналіз, узагальнення, систематизація науково-методичної та психолого-педагогічної літератури з актуальних проблем дослідження, чинних галузевих стандартів вищої освіти, навчальних програм, підручників і навчальних посібників, монографій, дисертацій, статей і матеріалів науково-методичних конференцій з проблеми дослідження, з питань інформатики та методики її навчання у вищій технічній школі, інформаційно-комунікаційних технологій в освіті, фундаменталізації інформатичної освіти, застосування хмарних технологій у навчанні з метою визначення теоретико-методичних засад дисертаційної роботи, розробки методики використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів; моделювання для визначення методичних засад використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів;

емпіричні – діагностичні (цілеспрямоване педагогічне спостереження, бесіди з роботодавцями, викладачами та студентами, аналіз досвіду роботи викладачів), ретроспективний аналіз особистого педагогічного досвіду для визначення провідних засобів хмарних технологій навчання майбутніх ІТ-фахівців;

експериментальні (констатувальний та формувальний етапи педагогічного експерименту) – з метою апробації розробленої методики використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів та експериментальної перевірки ефективності впровадження основних положень дослідження в практику підготовки майбутніх ІТ-фахівців;

методи математичної статистики для визначення статистичної значущості отриманих у ході експерименту результатів.

Теоретико-методологічні засади дослідження становлять філософські положення про єдність теорії та практики, взаємозумовленість та взаємозв'язок об'єктивних і суб'єктивних чинників формування особистості; концептуальні

ідеї філософії освіти (В. П. Андрющенко [79], Б. С. Гершунський [103], В. Г. Кремень [196], В. С. Курило [147]); теоретичні засади організації навчального процесу в закладах вищої освіти (Г. О. Атанов [81], В. І. Загвязинський [120], Т. М. Десятов [80], І. М. Дичківська [109], А. В. Коржуєв [208], В. А. Попков [199], Д. В. Чернілевський [253]), зокрема вищих технічних закладах (Т. О. Дмитренко [110], М. М. Зіновкіна [122], Е. В. Лузік [160], О. Ф. Меняєв [171]); ключові положення щодо фундаменталізації навчання студентів технічних університетів (С. А. Баляєва [83], С. Я. Казанцев [129], В. В. Кондратьєв [143], І. В. Левченко [155], О. Г. Мордкович [181], С. О. Семеріков [215], О. І. Суригін [231], А. Д. Суханов [232], О. О. Цапко [251]), навчання математичної інформатики (В. М. Глушков [104], М. І. Жалдак [116], Д. Е. Кнут (Donald Ervin Knuth) [43], Т. П. Кобильник [136], Ю. В. Триус [144]), використання хмарних технологій (Т. Гранц (Timothy Grance) [53], М. Грінбергер (Martin Greenberger) [35], Д. Е. Ірвін (David E. Irwin) [40], Дж. Маккарті (John McCarthy) [28], Дж. Маккендрік (Joe McKendrick) [51], П. Мелл (Peter Mell) [53], А. О. Манн (Alan O. Mann) [47], Д. Ф. Паркхілл (Douglas F. Parkhill) [27]), моделювання систем навчання та освіти (В. А. Байдак [82], В. Ю. Биков [86], В. В. Докучаєва [111], В. В. Лаптев [154], А. М. Пишкало [205], Л. О. Черних [255], Ч. Купісевич (Czesław Eugeniusz Kupisiewicz) [151]); наукові положення теорії та методики використання ІКТ в освіті (В. Ю. Биков [86], К. В. Власенко [97], Ю. В. Горошко [105], М. І. Жалдак [118], М. Ю. Кадемія [128], С. Г. Литвинова [157], О. С. Меняйленко [172], Н. В. Морзе [182], Л. Ф. Панченко [193], Ю. С. Рамський [206], Н. В. Рашевська [134], С. О. Семеріков [265], К. І. Словак [222], О. М. Спірін [261], Ю. В. Триус [237], С. В. Шокалюк [266]); теорія і практика впровадження інноваційних технологій у закладах вищої освіти (В. П. Беспалько [85], Н. П. Волкова [98], Г. К. Селевко [213], С. О. Сисоєва [219], С. Я. Харченко [220], Д. В. Чернілевський [254]), зокрема використання хмарних ІКТ у навчально-виховному процесі закладів вищої освіти (Е. А. Альдахіль [4], В. Ю. Биков [235], Т. А. Вакалюк [93], К. В. Власенко [96], І. С. Войтович [216],

Т. В. Волошина [99], В. Діаз [23], В. П. Іванніков [123], О. В. Коротун [146], Н. В. Морзе [185], М. В. Попель [197], Х. Раджай [64], З. С. Сейдаметова [188], С. О. Семеріков [167], О. М. Спірін [261], А. М. Стрюк [226], Ю. В. Триус [236], М. Є. Федосін [247], М. П. Шишкіна [261]).

Наукова новизна та теоретичне значення одержаних результатів полягає в тому, що: *вперше* теоретично обґрунтовано та розроблено методичні засади використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів, дібрано та класифіковано засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів; *удосконалено* форми, методи та засоби навчання математичної інформатики студентів закладів вищої освіти шляхом розробки та впровадження хмаро зорієнтованих засобів навчання; *дістали подальшого розвитку* теорія та методика застосування програмних засобів інформатизації освіти, проектування та розвитку хмаро зорієнтованого навчального середовища, зміст методичних систем навчання інформатичних дисциплін та інформаційно-комунікаційних технологій у різних галузях освіти.

Практичне значення одержаних результатів полягає в тому, що розроблено та впроваджено в навчально-виховний процес закладів вищої освіти: методику використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів, що складається із цільового (формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій), змістового (навчання основ математичної інформатики) та технологічного (хмаро зорієнтовані засоби ІКТ, методи та форми їх використання в навчанні математичної інформатики) блоків; програму спецкурсу «Основи математичної інформатики» для студентів спеціальності 123 «Комп'ютерна інженерія»; хмаро зорієнтовані програмно-методичні матеріали навчання основ математичної інформатики (режим доступу: <http://site.mathinfo.ccjournals.eu/>).

Результати дослідження можуть бути використані в системі професійної підготовки майбутніх фахівців з інформаційних технологій у закладах вищої освіти, у системі післядипломної освіти, у самоосвітній діяльності студентів.

Дослідження здійснювалися впродовж 2003-2018 рр. і охоплювало три **етапи науково-педагогічного пошуку**.

На аналітико-констатувальному етапі (2003-2007 рр.) була розроблена програма дослідження, що включала визначення вихідних теоретичних позицій, цілі експериментальної роботи і визначення завдань, конкретизацію об'єкту і предмету дослідження, виділення етапів і визначення термінів роботи. Була проаналізована науково-методична література з інформаційно-комунікаційних технологій, досвід навчання інформатичних дисциплін студентів різних спеціальностей, що надало можливість визначити актуальність дослідження та сформулювати його гіпотезу. Розроблялися, перевірялися і удосконалювалися програми навчання; вивчалися сучасні вітчизняні і зарубіжні методи використання інформаційно-комунікаційних технологій у підготовці фахівців; проводився теоретичний аналіз вітчизняної та зарубіжної психолого-педагогічної літератури для з'ясування ступеня вивченості і розробленості проблеми та констатувальний етап педагогічного експерименту.

На проектувально-пошуковому етапі (2008-2014 рр.) були розроблені методичні засади навчання основ математичної інформатики студентів технічних університетів засобами хмарних технологій, програмно-методичне забезпечення навчання основ математичної інформатики та методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

На формувально-узагальнювальному етапі (2015-2018 рр.) проведено формувальний етап педагогічного експерименту; проаналізовано, опрацьовано та узагальнено одержані результати експериментальної роботи; сформульовані загальні висновки та визначено перспективи подальшого вивчення проблеми; виконано оформлення рукопису дисертації.

Експериментальною базою дослідження на різних етапах педагогічного експерименту виступали Криворізький металургійний факультет Національної металургійної академії України, Криворізький технічний університет, Запорізький інститут економіки та інформаційних технологій, ДВНЗ «Криворізький національний університет». Загальна кількість учасників експерименту – 227 студентів.

Вірогідність результатів дослідження обумовлена: теоретичною обґрунтованістю вихідних положень дослідження; застосуванням комплексу методів педагогічного дослідження, адекватних його предмету, меті та завданням; методично обґрунтованим використанням програмного забезпечення; різнобічною апробацією основних положень дисертації; результатами статистичного опрацювання педагогічного експерименту та впровадженням компонентів методики використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

Результати дослідження **впроваджено** в освітній процес Криворізького металургійного факультету Національної металургійної академії України (довідка № 793 від 02.12.2011 р.), Криворізького технічного університету (довідка № 216/2 від 22.11.2011 р.), Запорізького інституту економіки та інформаційних технологій (довідка № 28 від 02.03.2012 р.), ДВНЗ «Криворізький національний університет» (довідка № 01/10-01/2018 від 04.05.2018 р.), Криворізького коледжу Національного авіаційного університету, Дніпровського національного університету імені Олеся Гончара (довідка від 10.05.2018 р.), Житомирського державного технологічного університету (довідка № 44-22.07/1407 від 10.09.2018 р.).

Особистий внесок здобувача в працях, опублікованих у співавторстві, полягає в аналізі змісту підготовки бакалаврів комп'ютерної інженерії (А. І. Купін [152; 153], В. А. Чубаров [152; 153]) та засобів хмарних технологій (Н. В. Рашевська [239], К. І. Словак [239], А. М. Стрюк [239]), визначенні змісту навчання основ математичної інформатики (Н. А. Василенко [94],

В. В. Петров [94], С. О. Семеріков [242]) та шляхів підвищення ефективності освітнього процесу засобами хмарних технологій (А. М. Стрюк [226]), розробці методичних основ проектування та використання засобів хмарних технологій (С. А. Бурма [92], І. І. Ліннік [240], М. В. Попель [48], С. О. Семеріков [48; 264], І. О. Теплицький [240], Ю. В. Формус [248], С. В. Шокалюк [264]), дефініції хмарних технологій навчання та визначення їхніх витоків (С. О. Семеріков [167; 245; 249], А. М. Стрюк [167; 245; 249], Б. А. Халимова [167; 245; 249]).

Апробація результатів дослідження. Основні положення, висновки та результати дослідження доповідалися й обговорювалися на науково-практичних конференціях різного рівня: *Міжнародних* – «Науково-методичні засади управління якістю освіти в педагогічних вищих навчальних закладах» (Київ, 2009), «Новітні комп'ютерні технології» (Севастополь, 2009-2012), «Інформаційні технології в освіті, науці і техніці» (Черкаси, 2012), «Розвиток інтелектуальних умінь і творчих здібностей учнів та студентів у процесі навчання дисциплін природничо-математичного циклу» (Суми, 2012), Workshop on Cloud Technologies in Education (Кривий Ріг, 2012, 2014, 2017), International Conference on ICT in Education, Research, and Industrial Applications: Integration, Harmonization and Knowledge Transfer (Київ, 2018); *Всеукраїнських* – «Інформаційні технології в освіті, науці і техніці» (Черкаси, 2010), «Комп'ютерні інтелектуальні системи та мережі» (Кривий Ріг, 2011, 2014, 2016), «Хмарні технології в сучасному університеті» (Черкаси, 2015), «Комп'ютерне моделювання та інформаційні технології в освіті» (Кривий Ріг, 2016, 2017); засіданнях науково-методичних семінарів кафедри комп'ютерних систем та мереж ДВНЗ «Криворізький національний університет» і спільної науково-дослідної лабораторії з питань використання хмарних технологій в освіті ДВНЗ «Криворізький національний університет» та Інституту інформаційних технологій і засобів навчання НАПН України (м. Київ).

Публікації. Основні результати дослідження відображено у 24 працях, із яких 11 – одноосібні; 6 статей опубліковано в наукових фахових виданнях України, з них три – у виданнях, що включено до міжнародних наукометричних

баз; 2 статті опубліковані в зарубіжних наукових фахових виданнях, включених до міжнародних наукометричних баз.

Структура та обсяг дисертації. Дисертація складається з анотації, переліку умовних позначень, вступу, двох розділів та висновків до них, загальних висновків, списку використаних джерел (270 найменувань, серед яких 76 – англійською мовою), 6 додатків на 60 сторінках. Робота містить 11 таблиць та 62 рисунки. Загальний обсяг дисертації – 327 сторінок.

РОЗДІЛ 1

ПРОБЛЕМА ВИКОРИСТАННЯ ХМАРНИХ ТЕХНОЛОГІЙ ЯК ЗАСОБУ НАВЧАННЯ ОСНОВ МАТЕМАТИЧНОЇ ІНФОРМАТИКИ СТУДЕНТІВ ТЕХНІЧНИХ УНІВЕРСИТЕТІВ

1.1 Математична інформатика у професійній підготовці студентів технічних університетів

Як зазначає М. І. Жалдак, «інформатика, як і будь-яка фундаментальна наукова дисципліна, має вивчати закони природи, всеможливі інформаційні процеси і відповідні технології, тому фундаментальні теоретичні положення, філософські, методологічні основи інформатики, зокрема елементи інформології, які остаточно з'ясовані як теоретично, так і експериментально, швидше за все не будуть змінюватись, або ж еволюціонуватимуть разом з розвитком відповідних теорій» [117, с. 11].

Підкреслюючи роль математики в системі інформатичної підготовки, М. І. Жалдак наводить цікавий факт, що в книзі відомого у всьому світі інформатика академіка В. М. Глушкова [104] з дванадцяти розділів лише один присвячений програмуванню, всі інші присвячені різним розділам математики: «Очевидно, хто не вміє розв'язувати математичні задачі (зокрема з дискретної математики), той не може бути хорошим програмістом. Адже навчання інформатики, як ніякого іншого предмету, формує вміння аналізувати різноманітні явища оточуючого світу, виробляє логічне й синтетичне мислення, здатності до евристичних пошуків, творчості, обґрунтування чи спростування різноманітних гіпотез, аргументованих висновків стосовно досліджуваних явищ і причинно-наслідкових зв'язків між ними» [117, с. 14].

Прогностичний характер роботи В. М. Глушкова найяскравіше проявляється у наступній цитаті, в якій легко упізнаються найсучасніші засоби ІКТ: «Безпаперова інформатика розвивається виключно швидкими темпами... вже недалекий той день, коли зникнуть звичайні книги, газети і журнали. Натомість кожна людина буде носити з собою «електронний» блокнот, який

представляє собою комбінацію плоского дисплея з мініатюрним радіопередавачем. Набираючи на клавіатурі цього «блокнота» потрібний код, можна (перебуваючи в будь-якому місці на нашій планеті), викликати з гігантських комп'ютерних баз даних, пов'язаних в мережі, будь-які тексти, зображення (у тому числі й динамічні), які й замінять не тільки сучасні книги, журнали і газети, а й сучасні телевізори» [104, с. 539].

Проте, як зазначає М. І. Жалдак, «оскільки теоретичні основи інформатики і особливо інформаційні технології бурхливо розвиваються, швидко витісняючи застаріваючі і застарілі технології і окремі теоретичні положення, то запропонувати більш-менш сталий зміст навчання в період становлення самої інформатики як науки досить не просто, особливо якщо прив'язувати його до якихось конкретних складових інформаційних технологій, їх апаратних і програмних складових. ... Вихід із такого становища полягає швидше за все в фундаменталізації курсу інформатики, ... включення до змісту навчання в основному загальних як теоретичних, так і технологічних положень, з демонстрацією їх, звичайно, на конкретних прикладах» [117, с. 15].

А. І. Субетто трактує фундаменталізацію освіти як процес формування «фундаментально-знаннєвого» каркасу знань особистості (ядра системи знань особистості), що визначає найважливіші знаннєві компоненти, з яких складається картина світу на особистісному рівні. Вона забезпечує основні функції – орієнтації, прогнозування, планування, проектування, управління майбутнім, взаємодії з людьми, а також забезпечує потенціал особистості до самонавчання в рамках технології «неперервної освіти» і, відповідно, потенціал особистості, професійну адаптивність у мінливому світі. Найважливішими аспектами (напрямами) фундаменталізації освіти за А. І. Субетто є:

- математизація знань, причому через залучення нових розділів, наприклад;
- формування єдиної наукової картини світу;
- уведення нових навчальних дисциплін з принципово нових перспективних напрямів [230].

На думку А. Д. Суханова, «найважливішим компонентом нової освітньої парадигми є концепція фундаменталізації, що трактує фундаментальність як категорію якості освіти і освіченості» [232, с. 18]. У розумінні дослідника в рамках парадигми відбувається перегляд орієнтирів з примату прагматичних знань на розвиток загальної культури і наукових форм мислення; з історичного контексту становлення наукового знання на сучасні уявлення про структуру і цілісний зміст системи наук. За новою освітньою парадигмою орієнтири у формуванні знань спрямовуються на інтереси особистості у цілковитій відповідності до тенденцій розвитку соціуму. Фундаментальність освіти є основою розвитку наукової компетентності, орієнтованої на усвідомлення глибинних, сутнісних підстав і зв'язків між різноманітними явищами і процесами навколишнього світу, оскільки саме фундаменталізація освіти є основою примноження інтелектуального потенціалу суспільства, гуманізації і соціалізації праці. На думку А. Д. Суханова, освіту можна вважати фундаментальною, якщо вона становить процес нелінійної взаємодії з інтелектуальним середовищем, коли особистість сприймає її для збагачення власного внутрішнього світу і завдяки цьому стає здатною для примноження потенціалу самого середовища.

Під фундаментальними науковими знаннями А. Д. Суханов розумів стрижневі, системотвірні, методологічно значущі уявлення, що підіймаються до джерел розуміння, до первісних сутностей.

Л. С. Йолгіна, здійснивши філософське осмислення фундаменталізації освіти, дійшла висновку, що фундаменталізація освіти неможлива без урахування потреб сталого розвитку суспільства. Дослідник вважає, що практична реалізація сучасної фундаменталізації освіти, яка не відміняє необхідність засвоєння фундаментальних принципів і законів науки, потребує освоєння теорії та методології предмету, що вивчається, а також знання основних законів сучасного природознавства та світової культури. Доцільним, на думку Л. С. Йолгіної, є уведення в науковий оборот поняття «закон – умова», яке з практичної точки зору поєднає філософсько-світоглядні та науково-

теоретичні знання в освітній сфері, покаже актуальність синтезу інтелектуального та духовного досвіду у рамках освіти [113].

О. М. Новіков поняття фундаменталізації освіти визначив як «поглиблення теоретичної загальноосвітньої, загальнонаукової, загальнопрофесійної підготовки учнів і студентів, у професійній же школі ще й розширення профілю їх професійної підготовки на відміну від вузькоспеціалізованої» [187, с. 68], виділяючи наступні основні тенденції фундаменталізації професійної освіти:

- 1) стабілізація ядра змісту загальної середньої освіти;
- 2) формування та розвиток ключових та загальнопрофесійних компетентностей;
- 3) посилення фундаментальної складової професійної освіти;
- 4) перехід до підготовки фахівців широкого профілю;
- 5) підвищення ролі університетської освіти;
- 6) модульна будова змісту освіти;
- 7) посилення наукового потенціалу навчальних закладів.

С. Я. Казанцев основою фундаменталізації вищої освіти вважає забезпечення умов розвитку у студентів системного, цілісного теоретико-методологічного знання, наукового, дослідницького, креативного стилю мислення, діяльності та спілкування [129].

За М. В. Булановою-Топорковою, фундаменталізація освіти сприяє формуванню творчого інженерного мислення, чіткого уявлення про місце своєї професії в системі загальнолюдських знань і практики. Фундаменталізацію вищої освіти дослідник визначила як системне і всеохоплююче збагачення навчального процесу фундаментальними знаннями і методами творчого мислення, виробленими фундаментальними науками [195].

О. О. Цапко на основі розгляду проблеми фундаменталізації освіти у технічному університеті запропонувала системно-концептуальну модель освітнього феномена технічного університету. Дослідник стверджує, що фундаменталізація університетської технічної освіти є стратегією, яка формує

норми нового гуманізму: глобальну етику і глобальну відповідальність [251].

О. І. Суригін у змісті фундаментальної політехнічної університетської освіти виділив наступні 5 компонентів: математичний, інформаційний, природничий (фізико-хімічний, хіміко-біологічний), філософський, гуманітарний [231].

Питанню фундаменталізації навчання у вищих технічних навчальних закладах присвячене дисертаційне дослідження С. А. Баляєвої, в якому стверджується, що фундаменталізація загальнонаукової підготовки у технічних ЗВО можлива, якщо:

- 1) навчання студентів є особистісно орієнтованим;
- 2) зміст навчання орієнтований на методологічно значущі, інваріантні знання із тривалим терміном життя;
- 3) реалізовані принципи єдності фундаменталізації та професіоналізації, інтеграції соціогуманітарної, культурологічної, природничої та спеціальної складових, інформатизації та комп'ютеризації загальнонаукової підготовки, цілісності навчального процесу, творчого розвитку, саморозвитку та самоосвіти особистості;
- 4) реалізовані психолого-педагогічні умови: посилення у змісті професійної освіти абстрактних, теоретичних, прогностичних, проектних компонентів загальнотехнічного знання; проектування циклів загальнонаукових дисциплін, що узагальнено та адекватно відображають фундаментальні ідеї, логіку та структуру відповідних наук з сучасних позицій, цілеспрямована структурно-змістова перебудова навчальних курсів на основі поєднання онтологічних, спеціально-наукових та дидактичних ідей, що піднімають їх статус до рівня фундаментальних; забезпечення цілісності освіти шляхом інтеграції окремих циклів через трансдисциплінарні комунікації та пограничні області знань і культури; розширення гуманітарної складової освіти як основи її особистісної та соціально-професійної направленості [83, с. 7-8].

М. О. Читалін наголошував, що фундаменталізацію професійної освіти необхідно розглядати невід'ємно від її професіоналізації [256], оскільки

професіоналізація та фундаменталізація знаходяться в діалектичній єдності і разом визначають якість професійної освіти на всіх її рівнях: загальному (гуманітарно-економічні та природничо-математичні дисципліни), особливому (загальнопрофесійні дисципліни) та частковому (спеціальні професійні дисципліни).

В. В. Кондратьєв, розглядаючи фундаменталізацію підготовки фахівців у технічних університетах на основі неперервної математичної підготовки, свій вибір пояснив тим, що предметною областю математики є вся дійсність, оскільки не існує жодної області матерії, у якій би не проявлялись закономірності, що вивчаються математикою. Дослідник стверджує, що в умовах технічного університету фундаменталізація навчання реалізується через:

- формування ядра системи інваріантних методологічно важливих знань;
- посилення фундаментальної складової змісту навчання;
- неперервну математичну підготовку як засіб фундаменталізації навчання технічних дисциплін на основі системно-модельного підходу [143, с. 19].

І. В. Левченко фундаменталізацію навчання інформатики визначила як діяльність, спрямовану на підвищення якості фундаментальної підготовки студента, його системотвірних і інваріантних (відносно технологій, конкретних деталей, думки людей тощо) знань і умінь в галузі інформатики, які дозволяють забезпечити загально інтелектуальний і емоційно-моральний розвиток студентів, формувати якості мислення, необхідні для повноцінної діяльності в інформаційному суспільстві, для динамічної адаптації людини до цього суспільства, для збагачення власного внутрішнього світу, для формування внутрішньої потреби у неперервному саморозвитку і самоосвіті [155, с. 17].

З. О. Решетова фундаменталізацію освіти розглянула на основі єдності її фундаментального і професійного аспектів. Одним із напрямів фундаменталізації освіти є фундаменталізація навчальної дисципліни, тобто пошук такого її змісту, який найкраще презентує основи даної конкретної науки. Причому процес фундаменталізації пов'язується з підвищенням наукового

рівня в основному технічних дисциплін – з їх математизацією і кібернетизацією, ергономізацією тощо, тобто з посиленням їх взаємозв'язку з природничо-науковими, соціально-філософськими і економічними галузями знань [207].

С. В. Іванов та М. Г. Кітов, аналізуючи взаємозв'язок філософії та фундаменталізації вищої освіти, обґрунтували думку, що цей процес має три рівні. Перший – поглиблене вивчення базових дисциплін для кожної спеціальності, другий – математизація освіти, що розглядається з позицій профілізації її викладання, третій – філософсько-методологічний, пов'язаний із критичним переосмисленням наукою історично застарілої методологічної парадигми [125]. Автори [125] наголошують, що фундаменталізація і профілізація навчального процесу не можуть бути ефективними без опори на широку гуманізацію освіти.

М. М. Ковтонюк фундаменталізацію професійної та загальнопрофесійної підготовки майбутнього вчителя відносить до категорії якості освіти та освіченості особистості і визначає як «процес багатоваріантної (нелінійної) взаємодії суб'єктів освітнього простору, результатом якого є формування у них універсальних базових, загальнопрофесійних і професійних компетентностей, узагальнених способів мислення, готовності особистості до майбутньої професійної діяльності, саморозвитку та самонавчання «впродовж всього життя» [141, с. 38].

С. І. Калінін під фундаменталізацією математичної освіти у вищій школі розуміє «напрямок її модернізації, що передбачає забезпечення урахування та використання у процесі глибокого і ґрунтового вивчення студентами математичних дисциплін нових наукових досліджень і досягнень у відповідних галузях математики, створення оптимальних умов для виховання у студентів гнучкого наукового мислення, а також застосування в освітньому процесі сучасних досягнень методики навчання математики як наукової області» [130, с. 39]. Таке трактування фундаменталізації математичної освіти передбачає: зниження частки репродуктивних підходів у навчанні студентів; їх знайомство із сучасними математичними дослідженнями, використання викладачем в

процесі навчання його власних фундаментальних досліджень; освоєння студентами науково-інформаційної бази та залучення їх до реальної науково-дослідної роботи. Реалізація даних положень сприятиме активному і глибокому оволодінню студентами математичними методами дослідження, формуванню у них внутрішньої потреби в саморозвитку та самоосвіті, переходу від екстенсивно-інформаційної освіти до інтенсивно-продуктивної. Крім того, дане трактування фундаменталізації математичної підготовки студентів передбачає в практиці їх навчання опору на методологічну складову методики навчання математики. Навчання має проводитися в рамках відповідної методичної системи з урахуванням складових зовнішнього середовища останньою і спиратися на діяльнісний підхід [130, с. 17-18]. При цьому фундаменталізація впливає на всі компоненти методичної системи: цілі навчання, зміст навчання, методи, форми організації та засоби навчання [130, с. 39].

На основі вищевикладеного пропонуємо трактування *фундаменталізації навчання студентів технічних університетів* як процесу зміни змісту навчання на основі:

- 1) виділення фундаментальної та технологічної складової змісту навчання;
- 2) математизації фундаментальної складової;
- 3) стабілізації технологічної складової на основі перспективних напрямів розвитку науки та технології.

Розглянемо більш детально перераховані напрями змін.

Як зазначив О. Г. Мордкович, принцип фундаментальності відображає необхідність солідної, але не відірваної від потреб майбутньої професії математичної підготовки [181].

Відомо, що теоретичний розділ будь-якої науки базується на математичних методах дослідження. Теоретична інформатика серед усіх інших напрямків включає математичну інформатику, оскільки математична інформатика – область інформатики, в якій досліджуються загальні закони інформаційної взаємодії довільно узятих об'єктів реальної дійсності за допомогою математичних засобів і методів.

За Т. П. Кобильником, навчання математичної інформатики сприяє суттєвому підвищенню рівня математичної та інформаційної культури, пізнавальної активності і самостійності студентів інформатичних спеціальностей, що позитивно відображається на якості їхніх знань та вмінь, їхньому інтелектуальному розвитку, рівні професійної підготовки. При цьому фундаментальність навчання математичної інформатики може бути досягнута за допомогою поєднання в змісті навчання теорії, абстракції і реалізації [140]. За такого підходу математичну інформатику можна вважати фундаменталізованою навчальною дисципліною, у якій стабілізація технологічної складової виконана на основі виділення класів програмних засобів її навчання, насамперед – систем комп'ютерної математики (СКМ).

В. В. Єфименко визначив комп'ютерну математику як «галузь інформатики, в якій вивчаються проблеми розробки, впровадження та використання інформаційних технологій розв'язування математичних задач» [114, с. 8] та навчальну дисципліну, «в процесі навчання якої формуються інформатично-математичні знання, а також вивчаються можливості використання сучасних інформаційно-комунікаційних технологій розв'язування прикладних задач стосовно дослідження різноманітних процесів і явищ на основі математичного моделювання та застосування теорій і методів, зокрема чисельних, для дослідження таких моделей, зокрема на основі імітаційних моделей для імітації перебігу досліджуваних процесів і проявів явищ, що вивчаються, методів відшукування оптимальних варіантів управління різноманітними, зокрема виробничими, процесами і т. д.» [114, с. 9]. У процесі навчання комп'ютерної математики формуються основні поняття про сучасні методи дослідження різноманітних об'єктів, побудови відповідних математичних чи інформаційних моделей та їх дослідження за допомогою засобів сучасних ІКТ; студенти ознайомлюються з можливостями створення прикладних програм за допомогою засобів програмування, вбудованих у системи комп'ютерної математики; у майбутніх учителів формуються знання і вміння використання систем комп'ютерної математики, для аналізу

математичних моделей процесів і явищ з найрізноманітніших галузей знань і діяльності людей.

Навчання математичної інформатики та комп'ютерної математики сприяє формуванню у студентів наукового світобачення, теоретичного мислення, що є ознакою фундаментальності професійної освіти. При цьому відкриваються нові можливості фундаменталізації інформатичної освіти через оволодіння цілісним баченням різноманітних явищ і процесів в сучасному світі, формування наукового світогляду, що відкриває шлях до гармонійного поєднання природничих, зокрема інформатичних та математичних, і гуманітарних знань.

Серед провідних напрямів фундаменталізації освіти у вищій школі С. О. Семеріков виділив математизацію змісту навчання та інформаційне моделювання [215], що відповідають другій та третій складовій фундаменталізації навчання студентів технічних університетів в умовах використання СКМ.

Таким чином, фундаменталізація навчання студентів технічних університетів вимагає проектування змісту навчання математичної інформатики на основі виокремлених напрямів.

Кафедрою інформаційних систем Київського національного університету ім. Т. Г. Шевченка наукові дослідження в області математичної інформатики проводяться у межах таких дисциплін: дискретна математика; теорія алгоритмів; аналіз складності алгоритмів; програмні логіки; верифікація та оптимізація програм; мережі Петрі; категоріальний аналіз; математична лінгвістика; теорія автоматів; комп'ютерна алгебра; узагальнене програмування; лямбда-числення [132].

На кафедрі математичної інформатики Київського національного університету ім. Т. Г. Шевченка з 1980 р. працює наукова школа математичної інформатики під керівництвом А. В. Анісімова, у якій виконуються дослідження з таких напрямів: рекурсивно-паралельні перетворювачі інформації та їх використання; дискретні процеси опрацювання даних; системи паралельного програмування; локальні алгоритми на дискретних структурах;

алгоритми асоціативно-локального пошуку в семантичних мережах; комп'ютерна лінгвістика; квантові алгоритми; проблематика штучного інтелекту; криптографія; захист даних у комп'ютерних мережах; комп'ютерна обробка надвеликих чисел; інтелектуальні бази знань; розподілені обчислення; методи стиснення даних; завадостійке кодування; обробка надвеликих даних; дистрибутивні алгоритми [169].

У Гомельському державному університеті імені Франциска Скорини математична інформатика є спеціалізацією до спеціальності «Математика» [224] на факультеті, що веде підготовку майбутніх фахівців з інформаційних технологій.

Підготовка фахівців з інформаційних технологій у технічних ЗВО України виконувалась у межах галузі знань 0501 «Інформатика та обчислювальна техніка» за напрямом підготовки 6.050102 «Комп'ютерна інженерія» (з 01.09.2015 – з усіх спеціальностей галузі знань «Інформаційні технології»). У додатку А наведено відомості про ліцензійний обсяг, державне замовлення та кількість зарахованих абітурієнтів по денній та заочній формах навчання у 2011 році.

Відповідно до табл. А.1, навіть у 2011 році, що відзначався значним зменшенням кількості випускників середніх шкіл (191 тис. проти 340 тис. у 2010 році [267]) та недобором за багатьма напрямами підготовки, фахівці з інформаційних технологій є суспільно затребуваними: кількість зарахованих абітурієнтів перевищує обсяг державного замовлення.

Загальний ліцензійний обсяг (більше 10 тис.) свідчить про високий рівень зацікавленості ЗВО України у підготовці фахівців з інформаційних технологій. Суспільна та державна значущість підготовки фахівців з інформаційних технологій підкреслюється у Рекомендаціях парламентських слухань на тему: «Створення в Україні сприятливих умов для розвитку індустрії програмного забезпечення» [204]. У п. 3 Рекомендацій вказується, що при прийнятті законів про державний бюджет України необхідно врахувати пріоритетність розвитку ІТ-освіти та формування державного замовлення з урахуванням потреби у

фахівцях ІТ-ринку. У п. 9 наголошується на необхідності сприяти створенню та забезпеченню постійного функціонування, оновлення й вільного доступу до відкритих онлайн-ових засобів програмно-інформаційного та лінгвістичного забезпечення, у тому числі (п. 23) – електронних освітніх ресурсів навчального призначення.

Найбільше перевищення контингенту першокурсників над держзамовленням спостерігається у Національному технічному університеті України «Київський політехнічний інститут» (128 осіб), яким були внесені складові галузевого стандарту вищої освіти України за напрямом підготовки 6.050102 «Комп'ютерна інженерія» [190; 191].

Освітньо-професійна програма бакалавра напряму підготовки 6.050102 (з 01.09.2015 – спеціальності 123) «Комп'ютерна інженерія» передбачає 3 цикли підготовки: цикл гуманітарної та соціально-економічної підготовки; цикл математичної та природничо-наукової підготовки; цикл професійної та практичної підготовки. На рис. 1.1 показано розподіл обсягів нормативної частини освітньо-професійної програми за циклами.

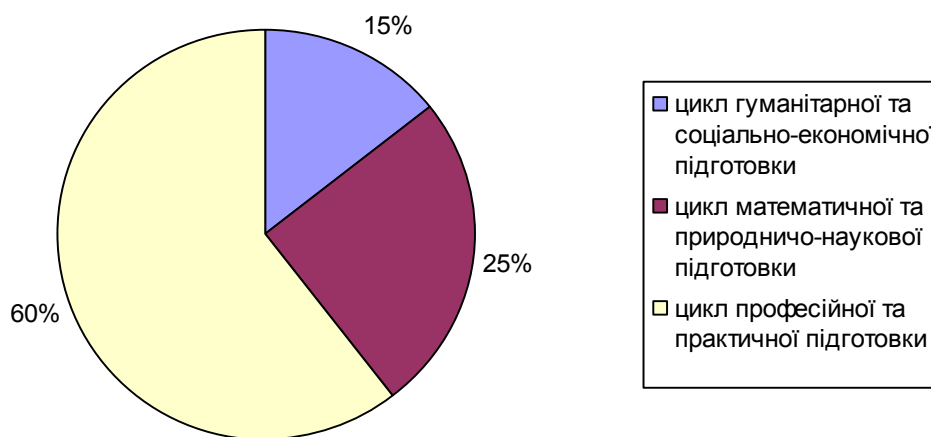


Рис. 1.1. Розподіл обсягів нормативної частини освітньо-професійної програми бакалавра напряму підготовки 6.050102 «Комп'ютерна інженерія» за циклами

У відповідності до визначення математичної інформатики як напряму наукових досліджень, що знаходиться на межі математики та інформатики і, з одного боку, є складовою теоретичної інформатики, де математичні моделі і засоби використовуються для моделювання та дослідження інформаційних

процесів у різних сферах діяльності людини, а, з іншого боку, займається використанням інформаційних систем і технологій для розв'язування складних математичних задач [118, с. 6], з переліку виробничих функцій, типових задач діяльності, умінь та компетенцій майбутнього фахівця з інформаційних технологій були виділено такі, що відповідають типовим завданням діяльності з математичної інформатики (табл. 1.1).

Таблиця 1.1

Виробничі функції, типові задачі діяльності, уміння та компетенції фахівця з інформаційних технологій, що відносяться до математичної інформатики

Шифр та назва типового завдання діяльності	Підшифр та зміст уміння	Шифр компетенції
<i>Дослідницька виробнича функція</i>		
2.ПФ.Д.01. Математичні перетворення та розрахунки	01. Розв'язувати математичні задачі шляхом створення відповідних застосувань	КЗП.01
2.ПФ.Д.03. Розрахунки електричних кіл	04. Розраховувати нелінійне електричне та магнітне коло графічним або чисельним методом	КЗП.03
2.ПФ.Д.05. Чисельні розрахунки	01. Володіти методами та технологіями розробки та оцінювання алгоритмів	КЗП.05
	02. Вибирати та обґрунтовувати методи обчислень стійких до похибок	
	03. Застосовувати методи та алгоритми оптимального розв'язування задач інтерполяції	
	04. Володіти методами обчислення нелінійних рівнянь	
	05. Розв'язувати системи лінійних алгебраїчних рівнянь	
	06. Здійснювати вибір методу інтегрування та аналізу похибок	
	07. Володіти методами обчислень диференціальних рівнянь	
	08. Ставити та вирішувати оптимізаційні задачі	
2.ПФ.Д.06. Аналіз та синтез дискретних об'єктів	06. Володіти теорією кодування	КЗП.06
2.ПФ.Д.07. Розробка електронних схем	02. Виконувати синтез і аналіз одиночних каскадів напівпровідникових пристроїв у відповідності з їх параметрами і параметричними співвідношеннями з урахуванням їх динамічних і статичних характеристик	КЗП.07
	03. Узагальнювати динамічні показники електронних пристроїв, застосовуючи поняття періодичної, перехідної і імпульсної характеристики, розраховувати типові функціональні блоки і вузли аналогових пристроїв	
	04. Розраховувати базові логічні і цифрові елементи	

Продовження таблиці 1.1

Шифр та назва типового завдання діяльності	Підшифр та зміст уміння	Шифр компетенції
<i>Проектувальна виробнича функція</i>		
3.ПФ.Д.02. Синтез комбінаційних схем	01. Формулювати практичні задачі комп'ютерної логіки в термінах алгебри перемикальних функцій, абстрактної та структурної теорії цифрових автоматів	КСП.02
	03. Проводити мінімізацію перемикальних функцій та систем функцій формалізованими та неформалізованими методами	
	04. Отримувати операторні форми перемикальних функцій для різних елементних базисів. Розробляти комбінаційні схеми, оцінювати їх параметри	
	06. Виконувати абстрактний та структурний синтез автоматів з використанням теорії часових функцій та композиції елементарних автоматів	
	07. Аналізувати функції поведінки автоматів і застосовувати способи уникнення збоїв в їх роботі	
	08. Виконувати синтез та аналіз типових вузлів, що застосовуються у комп'ютерах, використовувати для побудови цифрових схем ВІС, що програмуються	
	10. Подавати додатні та від'ємні числа у різних машинних кодах та різних форматах	
3.ПФ.Д.04. Розробка типового вузла і пристрою ТО	05. Розробляти арифметичні пристрої з розподіленою та зосередженою логікою на сучасній елементній базі	КСП.04
	06. Розробляти управляючі пристрої з жорсткою та гнучкою логікою на сучасній елементній базі	
3.ПФ.Е.05. Розробка системних програми	10. Розробляти елементи синтаксичних та семантичних аналізаторів трансляторів	КСП.05
3.ПФ.Е.10. Програмування для паралельних та розподілених комп'ютерних систем	02. Здійснювати побудову паралельного алгоритму і виконувати його аналіз	КСП.10
	05. Виконувати моделювання паралельних обчислень	
3.ПФ.Д.11. Робота базами даних	01. Будувати модель даних концептуального (логічного) рівня – модель «сутність-зв'язок»	КСП.11
	02. Будувати модель даних даталогічного (фізичного) рівня – реляційна модель	
	03. Застосовувати постреляційні моделі даних	
3.ПФ.Е.13. Розробка великих програмних систем	03. Моделювати різні аспекти системи, для якої створюється програмне забезпечення	КСП.13
<i>Технологічна виробнича функція</i>		
3.ПФ.Д.07. Автоматизація проектування ТО	03. Визначати математичну модель для рішення задачі проектування	КСП.07
	05. Визначати методи оптимізації	
<i>Організаційна виробнича функція</i>		
3.ПФ.Д.12. Захист інформації	03. Створювати програмні та апаратні підсистеми криптографічного захисту інформації	КСП.12
	04. Формувати і управляти ключовою інформацією для підсистем аутентифікації	

Із виділених виробничих функцій фахівця з інформаційних технологій, що відносяться до математичної інформатики, 5 – дослідницькі, 6 – проектувальні, 1 – технологічна і 1 – організаційна. Перелік професійних компетенцій фахівця з інформаційних технологій (КЗП – загально професійні, КСП – спеціальні професійні), що відносяться до математичної інформатики та система умінь, яка їх відображає, наведені у табл. 1.2.

Таблиця 1.2

Професійні компетенції фахівця з інформаційних технологій, що відносяться до математичної інформатики, та система умінь, яка їх відображає

№	Шифр та зміст компетенції	Підшифр та зміст уміння
1	КЗП.01. Ґрунтовна підготовка з вищої математики для використання математичного апарату при розв'язанні прикладних і наукових завдань в області комп'ютерної інженерії	ПР.Р.01. Підготовленість до використання існуючих та розроблення нових математичних методів для вирішення задач, пов'язаних з проектуванням та використанням КСМ
2	КЗП.03. Ґрунтовна підготовка з теорії електричних та магнітних кіл	ПР.Р.01. Уміння використовувати методи теорії електричних та магнітних кіл при проектуванні апаратних складових комп'ютерних систем
3	КЗП.05. Знання методів побудови та аналізу ефективних алгоритмів і чисельних методів та уміння їх реалізувати в конкретних застосуваннях	ПР.Р.01. Здатність до аналізу, оцінювання та вибору існуючих алгоритмів, розробки нових алгоритмів, які пов'язані з проектуванням апаратних та програмних компонент КСМ ПР.Р.02. Вміти використовувати чисельні методи при вирішенні задач комп'ютерної інженерії
4	КЗП.06. Знання дискретних структур і вміння застосовувати сучасні методи дискретної математики для аналізу і синтезу складних систем	ПР.Р.01. Уміння застосовувати сучасні методи дискретної математики для аналізу, синтезу та проектування КСМ різного призначення
5	КЗП.07. Ґрунтовна підготовка з комп'ютерної електроніки	ПР.Р.01. Підготовленість до використання відповідних законів електроніки при вирішенні завдань, пов'язаних з проектуванням апаратних засобів КСМ
6	КСП.02. Знання теоретичних (логічних та арифметичних) основ побудови сучасних комп'ютерів	ПР.Р.01. Вміння застосовувати комп'ютерну логіку при проектуванні блоків комп'ютера ПР.Р.02. Вміти застосовувати комп'ютерну арифметику при проектуванні арифметично-логічних пристроїв
7	КСП.04. Знання схемотехнічних основ сучасних комп'ютерів	ПР.Р.01. Вміти розробляти окремі блоки комп'ютерів
8	КСП.05. Знання особливостей системного програмування, володіти методами та засобами розробки елементів системних програм	ПР.Р.01. Уміння створювати системні програми різного призначення та їх окремі елементи

Продовження таблиці 1.2

№	Шифр та зміст компетенції	Підшифр та зміст уміння
9	КСП.07. Знання методів автоматизованого проектування КСМ та їх компонент	ПР.Р.01. Уміння використовувати сучасні комп'ютерні засоби системного, функціонального, конструкторського та технологічного проектування
10	КСП.10. Підготовка в області розробки програмного забезпечення для КС з паралельною і розподіленою архітектурою, володіння засобами сучасних мов та бібліотек паралельного програмування	ПР.Р.01. Уміння розробляти паралельні алгоритми, програмувати процеси, організувати взаємодію процесів, розробляти та налагоджувати паралельні (розподілені) програми
11	КСП.11. Знання сучасних теорій організації баз даних, методів і технологій їх розробки	ПР.Р.01. Уміння проектувати бази даних з різною структурною організацією та призначенням
12	КСП.12. Знання організаційних, технічних, алгоритмічних і інших методів і засобів захисту інформації в КСМ, відповідно законодавству та стандартам в цій області, з сучасними криптосистемами	ПР.Р.01. Уміння застосовувати методи і засоби забезпечення безпеки програм і даних при проектуванні і експлуатації КСМ
13	КСП.13. Знання теоретичних основ і сучасних технологій розробки складних програмних систем (програмної інженерії)	ПР.Р.01. Уміння застосовувати технології та інструментальні засоби проектування і створення програмних систем

Аналіз табл. 1.2 засвідчує, що саме у загально професійних компетенціях реалізуються дослідницькі виробничі функції фахівця з інформаційних технологій, що відносяться до математичної інформатики. Крім того, загально професійні компетенції відносяться не лише до спеціальності 123 «Комп'ютерна інженерія», а й до всієї галузі знань 12 «Інформаційні технології». Це надає можливість зробити висновок про те, що зміст навчання основ математичної інформатики студентів технічних університетів має бути спрямованим на формування їх дослідницьких виробничих функцій.

У відповідності до освітньо-професійної програми бакалавра з напрямку підготовки 6.050102 «Комп'ютерна інженерія», дослідницькі виробничі функції з основ математичної інформатики фахівців з інформаційних технологій формуватимуться у навчальних дисциплінах, наведених у табл. 1.3.

Таблиця 1.3

Навчальні дисципліни, у яких формуються дослідницькі виробничі функції з основ математичної інформатики у фахівців з інформаційних технологій

Шифр уміння	Код і назва дисципліни	Підшифр і назва блоку (розділу) змістовних модулів (тем)	
2.ПФ.Д.01.01	2.01. Вища математика	01. Математичний аналіз	
		02. Диференційні рівняння	
		03. Лінійна алгебра та аналітична геометрія	
2.ПФ.Д.03.01	2.03. Теорія електричних та магнітних кіл	01. Теорія лінійних електричних кіл постійного струму	
2.ПФ.Д.03.02		02. Лінійні електричні кола синусоїдного струму	
2.ПФ.Д.03.03		03. Не синусоїдальні періодичні та перехідні процеси в лінійних електричних колах	
2.ПФ.Д.03.04		04. Електричні кола з розподіленими параметрами та елементи теорії нелінійних кіл	
2.ПФ.Д.05.01	2.05. Алгоритми та методи обчислень	01. Теорія алгоритмів	
2.ПФ.Д.05.02		02. Методи обчислень	
2.ПФ.Д.06.01	2.06. Дискретна математика	01. Теорія множин	
2.ПФ.Д.06.04		04. Теорія графів	
2.ПФ.Д.07.02	2.07. Комп'ютерна електроніка	02. Пристрої цифрової електроніки	
2.ПФ.Д.07.03			
2.ПФ.Д.07.04			
3.ПФ.Д.02.01	3.02. Комп'ютерна логіка	01. Комп'ютерна логіка	
3.ПФ.Д.02.02			
3.ПФ.Д.02.03			
3.ПФ.Д.02.04			
3.ПФ.Д.02.05			
3.ПФ.Д.02.06			
3.ПФ.Д.02.07			
3.ПФ.Д.02.08			
3.ПФ.Д.02.09		02. Комп'ютерна арифметика	
3.ПФ.Д.02.10			
3.ПФ.Д.02.11			
3.ПФ.Д.02.12			
3.ПФ.Д.02.13			
3.ПФ.Д.04.02	3.03. Архітектура комп'ютерів		02. Архітектура процесорів
3.ПФ.Д.04.03			
3.ПФ.Д.04.04		03. Організація пам'яті	
3.ПФ.Д.04.05			
3.ПФ.Д.04.06		04. Організація вводу-виводу	
3.ПФ.Д.04.07		05. Вступ до мікропроцесорної техніки	
3.ПФ.Д.04.01		3.04. Комп'ютерна схемотехніка	01. Схемотехніка типових вузлів і блоків
3.ПФ.Д.04.02	02. Схемотехніка запам'ятовуючих пристроїв		
3.ПФ.Д.04.03	03. Схемотехніка арифметичних пристроїв		
3.ПФ.Д.04.04			
3.ПФ.Д.04.05			

Продовження таблиці 1.3

Шифр уміння	Код і назва дисципліни	Підшифр і назва блоку (розділу) змістовних модулів (тем)	
3.ПФ.Д.04.06		04. Управляючі та комунікаційні засоби	
3.ПФ.Д.04.07			
3.ПФ.Д.04.08		05. Схемотехніка систем на ВПС та НВІС	
3.ПФ.Д.04.09			
3.ПФ.Д.04.10			
3.ПФ.Д.05.03	3.05. Системне програмування	02. Багатомодульні системні програми	
3.ПФ.Д.05.04		07. Основи теорії трансляторів	
3.ПФ.Д.05.10			
3.ПФ.Д.07.03	3.07. Технології проектування комп'ютерних систем	03. Системне проектування	
		04. Операційне проектування	
05. Функціональне проектування			
06. Технічне проектування			
3.ПФ.Д.07.05		07. САПР, що тиражується	
3.ПФ.Д.12.03		3.12. Захист інформації у комп'ютерних системах	04. Симетричні схеми, ключі та системи шифрування
3.ПФ.Д.12.04			05. Асиметричні схеми, ключі та системи шифрування
3.ПФ.Е.13.03	3.13. Інженерія програмного забезпечення	01. Проектування та розробка ПЗ	

Вилучення із табл. 1.3 умінь, що відносяться до спеціальних професійних компетенцій, надає можливість обгрунтовано визначити навчальні дисципліни (табл. 1.4), у процесі навчання яких формується *компетентність з математичної інформатики майбутніх фахівців з інформаційних технологій* – особистісно-професійне утворення, яке включає в себе систему знань, умінь, навичок, досвід навчально-дослідницької діяльності з математичної інформатики та позитивне ціннісне ставлення до неї й виявляється в готовності та здатності до модифікації існуючих і розробки нових інформаційних технологій на основі моделей та методів математичної інформатики. Виходячи з того, що дві з вказаних дисциплін – «Вища математика» і «Теорія електричних та магнітних кіл» відносяться до блоку фундаментальної підготовки, зміст навчання основ математичної інформатики студентів технічних університетів складатимуть такі основні змістові модулі: теорія алгоритмів, методи обчислень, теорія множин, теорія графів, комп'ютерна логіка, комп'ютерна арифметика, схеми шифрування.

Навчальні дисципліни, у яких формується компетентність з математичної інформатики майбутніх фахівців з інформаційних технологій

Назва дисципліни	Назва блоку (розділу) змістовних модулів (тем)
Вища математика	Математичний аналіз
	Диференційні рівняння
	Лінійна алгебра та аналітична геометрія
Теорія електричних та магнітних кіл	Теорія лінійних електричних кіл постійного струму
	Лінійні електричні кола синусоїдного струму
	Не синусоїдальні періодичні та перехідні процеси в лінійних електричних колах
	Електричні кола з розподіленими параметрами та елементи теорії нелінійних кіл
Алгоритми та методи обчислень	Теорія алгоритмів
	Методи обчислень
Дискретна математика	Теорія множин
	Теорія графів
Комп'ютерна логіка	Комп'ютерна логіка
	Комп'ютерна арифметика
Захист інформації у комп'ютерних системах	Симетричні схеми, ключі та системи шифрування
	Асиметричні схеми, ключі та системи шифрування

Як зазначалось вище, Т. П. Кобильником запропоновано основні компоненти зміст навчання математичної інформатики для студентів інформатичних спеціальностей педагогічного університету. На основі загальних закономірностей і принципів, сучасних підходів і концепцій педагогіки та психології вищої школи ним обґрунтовано можливість використання СКМ при вивченні математичної інформатики та деяких математичних дисциплін у педагогічному університеті, розглянуто особистісно-орієнтовані технології навчання (метод проектів, ситуаційне, проблемне, модульно-рейтингове навчання), обґрунтовано доцільність їх використання при навчанні математичної інформатики [137].

Вивчення математичної інформатики у педагогічному ЗВО Т. П. Кобильник пропонує проводити у двох напрямках: у рамках курсу «Системи комп'ютерної математики» та спецкурсу «Математична інформатика» [138]. Вивчення курсу «Системи комп'ютерної математики» можливе із одночасним опануванням методів обчислень. Зміст спецкурсу

«Математична інформатика» для студентів педагогічних університетів складають наступні блоки: системи штучного інтелекту, додаткові розділи чисельних методів (розпізнавання образів), теорія кодування, основи криптографії.

Порівняння змісту навчання математичної інформатики у педагогічних та технічних ЗВО показує (рис. 1.2), що, за винятком окремих розділів систем штучного інтелекту, всі інші складові змісту навчання математичної інформатики у педагогічних ЗВО входять до змісту навчання математичної інформатики у технічних ЗВО, доповнюючись такими блоками: 1) дискретна математика (теорія множин, теорія графів), 2) комп'ютерна логіка (комп'ютерна логіка, комп'ютерна арифметика).

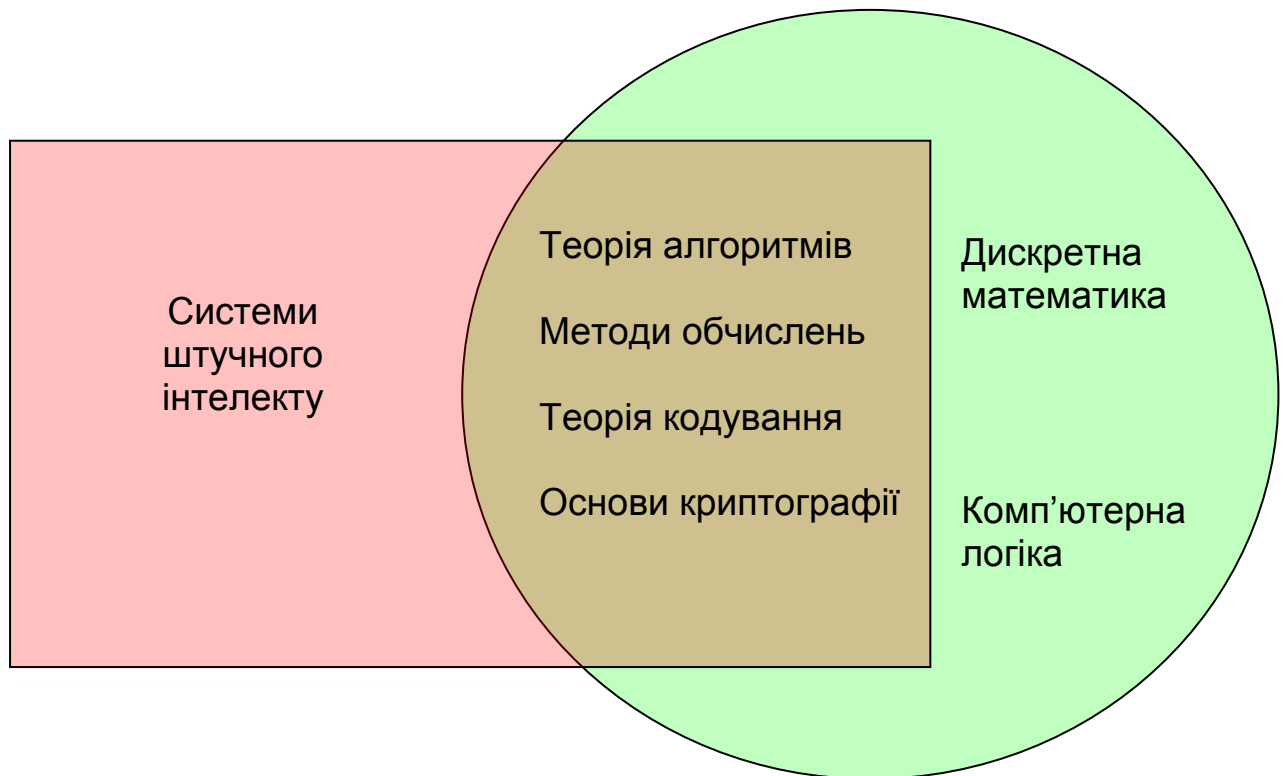





Рис. 1.2. Співвідношення змісту навчання математичної інформатики студентів педагогічних та технічних університетів ( – математична інформатика у педагогічних ЗВО,  – математична інформатика у технічних ЗВО,  – основи математичної інформатики)

Спільний зміст навчання математичної інформатики студентів технічних та педагогічних ЗВО, показаний перетином фігур на рис. 1.2, надалі

позначатимемо як *зміст навчання основ математичної інформатики студентів технічних університетів* (табл. 1.5).

Таблиця 1.5

Зміст навчання основ математичної інформатики студентів технічних університетів

Назва модуля	Зміст модуля
I. Теорія алгоритмів	1. Аналіз алгоритмів 2. Алгоритмічні стратегії 3. Побудова алгоритмів
II. Методи обчислень	1. Комп'ютерне моделювання 2. Задачі лінійної алгебри 3. Задачі нелінійної алгебри 4. Методи наближення функцій 5. Розв'язання диференціальних рівнянь 6. Методи оптимізації 7. Нейронні мережі та розпізнавання образів
III. Теорія кодування	1. Математичні основи теорії кодування 2. Основні поняття теорії завадостійкого кодування 3. Лінійні коди 4. Циклічні коди, їхні властивості 5. Коди Боуза-Чоудхурі-Хоквінгема 6. Коди Ріда-Соломона, згорткові коди
IV. Основи криптографії	1. Криптографічні системи 2. Симетричні криптосистеми 3. Асиметричні криптосистеми 4. Електронні цифрові підписи 5. Управління криптографічними ключами

Таким чином, шляхом виділення фундаментальної складової змісту навчання та її математизації спроектовано зміст навчання основ математичної інформатики студентів технічних університетів. Проте процедура фундаменталізації не буде повною, якщо не виконати виділення технологічної складової змісту навчання та її стабілізацію на основі перспективних інформаційних технологій.

1.2 Поняття про хмарні технології навчання

Теоретичною основою хмарних технологій є концепція «комунальних обчислень» (utility computing), сутність якої у 1961 р. Дж. Маккарті виклав у доповіді, присвяченій сторіччю Массачусетського технологічного інституту

[18], розглянувши комп'ютерні ресурси (обчислювальні, зберігальні та інші) як вимірювані та гнучко дозовані послуги на зразок тих, що надають оператори зв'язку: «Комп'ютерні ресурси можуть бути організовані як комунальні послуги на зразок телефонної системи. ... Кожному абоненту такої послуги необхідно сплачувати лише за спожите, проте він буде мати доступ до усіх мов програмування на великій кількості систем. ... Деякі абоненти також можуть надавати послуги іншим. ... Комунальні обчислення мають стати основою нової та важливої індустрії» [28, с. 2].

Продаж комп'ютерних послуг у 1960-х рр. не був звичайною бізнес-моделлю: як зазначає Т. Хей (Thomas Haigh) [36, с. 8], контракти на встановлення комп'ютерного обладнання у деяких випадках включали програмні послуги та стандартні пакети. Із зростанням інтересу до мережного та віддаленого доступу до комп'ютерів очікувалось, що така бізнес-модель стане звичною. Основою побудови «комунальних комп'ютерів» (computer utilities) вважались системи розподілу часу (коли декілька користувачів отримували доступ у режимі реального часу до одного комп'ютера). За такої моделі тисячі користувачів могли бути абонентами гігантських мереж, використовуючи термінали для доступу до комп'ютерного обладнання та програмного забезпечення, що виконувалось на віддалених комп'ютерах.

Із часом трактування «computer utility» як «комп'ютерної послуги» змінилось із власне послуги, що надається, на її програмне забезпечення. Проте початкове трактування в останнє десятиріччя знову набуває актуальності, тому доцільним є дослідження генезису та еволюції хмарних технологій.

У 1959 році А. О. Манн у статті [47], прогножуючи розвиток менеджменту систем управління (сьогодні відомого як «контролінг»), зокрема, якості послуг, вказував на 4 головні фактори, що спостерігались у розвитку електричних, телефонних, телеграфних та інших систем бізнесу з продажу (постачання послуг):

- 1) розмір інвестицій, необхідних для кожної окремої системи;
- 2) потенційне перекриття та дублювання елементів системи із

збільшенням кількості окремих систем;

3) невід’ємна для електронних систем властивість зменшення вартості опрацювання одиниці із зростанням швидкості, обсягу та потужності;

4) зростання вартості утилізації кожної окремої системи.

Такі системи А. О. Манн називає системами комунальних послуг (public utility), особливостями яких є:

– наголос на максимізацію обсягу послуг за мінімізації прибутку при безперервному та ефективному веденні бізнесу;

– зацікавленість громадськості у постійному детальному нагляді, висока ступінь регулювання відповідності бізнесу суспільним інтересам;

– обслуговування всіх бажаючих за розумною ціною і без дискримінації;

– централізоване постачання послуги;

– наголос на задоволенні потреб користувачів послуги;

– акцент на ініціюванні радикальних бізнес-інновацій;

– виховання відданості працівників службі;

– вплив на послугу зв’язків із громадськістю;

– акцент на наукові дослідження і технологічне лідерство;

– послуга має надаватися за вимогою та приносити прибуток після капітальних вкладень.

До переваг використання комп’ютерних систем як комунальних послуг А. О. Манн відніс наступні:

1) можливість використання розробленого програмного забезпечення на різних комп’ютерних системах, у тому числі – неіснуючих на час розробки програми;

2) зростання кількості комп’ютерних систем покращує надання послуги;

3) вартість послуги зменшується із зростанням розміру та потужності системи;

4) резервна потужність комп’ютерних систем, необхідна для задоволення пікових потреб користувачів, нарощується швидше та дешевше за інші типи систем за рахунок їх модульності;

5) диверсифікація послуги між постачальниками відбувається прозоро для її споживача;

б) додавання нових компонентів комп'ютерної системи збільшує її потужність за меншу вартість порівняно із попередніми компонентами;

7) об'єднання наукових обчислень та опрацювання бізнес-даних у спільній системі загального призначення сприяє економії коштів так само, як і надання різним споживачам стандартизованих послуг.

Крім того, А. О. Манн виділив й деякі історичні передумови для розгляду комп'ютерних послуг як комунальних:

1. Багато із існуючих комунальних послуг розвивались за такою ж схемою, як і комп'ютерний бізнес:

а) спочатку зростаюча кількість користувачів організується у спільні підприємства та кооперативи;

б) далі розпочинається продаж надлишкових потужностей і послуг;

в) нарешті, зростаючі потреби користувачів не можуть бути задоволені швидко та адекватно за місцем вимоги, що приводить до необхідності розвитку постачальника послуги.

У комп'ютерному бізнесі є вимоги, аналогічні вимогам користувачів комунальних послуг: більш високі швидкості, більші обсяги збережуваних даних, краще введення-виведення, підтримка мов загального призначення та спрощеного програмування, менші витрати на заміну комп'ютерного обладнання та його перепрограмування, канали передачі даних більш низької вартості з радикально більшою надійністю та швидкістю, надання високоякісних послуг, а також зниження цін в цілому.

2. Існує безсумнівна зростаюча тенденція до об'єднання комп'ютерних систем з мережами зв'язку, адже значна частина всіх опрацьовуваних даних поступає через канали їх передачі. Таким чином, все більше і більше традиційних постачальників зв'язку надають лінії передачі даних як елементи інтегрованої системи опрацювання даних. Безперечно, ця тенденція буде зростати з боку як різноманітності та масштабу, так і з боку спільної вартості

послуг передавання даних, адже зв'язок сьогодні вже є комунальною послугою.

3. Розповсюдженою практикою в комп'ютерному бізнесі стає оренда, а не продаж обладнання, з супутнім наданням обслуговування, запасних частин, навчання, резервного копіювання, а також численних послуг, включених в орендну плату. Це є ще однією важливою історичною характеристикою комунального бізнесу.

У 1959 році було важко уявити будь-яку перспективу для окремої людини як користувача великомасштабної комп'ютерної системи, проте А. О. Манн вказав, що «такі щоденні вимоги, як опрацювання даних або інформаційний пошук потребують оренди машинного часу» [47, с. 257], тому найбільш перспективною комп'ютерною послугою вважає доступ до комунальних мереж передавання та опрацювання даних, об'єднання яких сьогодні відоме як Інтернет.

Протилежністю «комунальним обчисленням» на основі великих, потужних, швидкісних центрів опрацювання даних, які обслуговують велику кількість клієнтів, А. О. Манн вважав засоби «персональних обчислень» – сучасні персональні комп'ютери, вказуючи, що зростання їх характеристик може призвести до відсутності необхідності у «комунальних обчисленнях», проте сам же й наводить контртезу – зростаюча складність зв'язків та комунікацій змушує об'єднуватись для ефективного опрацювання даних: «Цілком можливо, що розвиток комп'ютерних систем буде подібний практиці, яка існує в електричній та телефонній системах, коли деякі фірми або організації-споживачі закупають та експлуатують свої власні системи, але в той же час користуються послугами інженерних мереж. ... Такий розвиток подій може бути гарним для користувачів» [47, с. 258].

До технологічних проблем «комунальних обчислень» – надання комп'ютерних послуг – А. О. Манн відніс:

- нерозробленість методів високошвидкісного високонадійного зв'язку;
- відсутність стандартизації устаткування, що має працювати тривалий час;

– необхідність стандартизації програм та даних засобами автоматизації програмування мовами загального призначення.

Реалізація запропонованої концепції мала надати такі переваги:

– для організацій: можливість споживання більшої кількості комп'ютерних послуг меншої вартості;

– для постачальників комп'ютерного обладнання: перспективи для збільшення швидкості, ємності, потужності та переходу від друкованих документів до пристроїв зчитування символів і візуальних дисплеїв;

– для розробників програмних систем: засоби уніфікації використовуваних мов програмування, трансляторів тощо;

– для користувачів: рівний доступ до програм та даних, розв'язання проблем немобільності програм та обладнання, застосування сучасних технологій опрацювання даних та інші засоби задоволення їх зростаючих потреб.

Суттєвий внесок у дослідження проблеми комп'ютерної послуги вніс Д. Ф. Паркхілл, огляд однойменної книги якого, виданої 1966 року, виконав американський математик, автор відомої «ділеми ув'язненого» та піонер дослідження операцій і системного аналізу М. М. Флуд (Merrill M. Flood) [27]: «Ця книга є особливо важливою для тих, хто має досвід роботи з комп'ютерними системами розподілу часу, як корисний приклад інформаційних послуг. Автор починає з короткого, але значущого аналізу історії обчислювальної техніки. ... Його історичний аналіз впливу ... ранніх «мульті-» систем допомагає визначити перспективу для більш нових розробок «комп'ютерних послуг». ...

Для пояснення основних характеристик існуючих систем розподілу часу та спільного використання файлів у комп'ютерних системах детально обговорюються дев'ять «ранніх комп'ютерних послуг». ... Автор визначає Computer Sciences Corporation (REMOTRAN) як первісток «публічної комп'ютерної послуги загального призначення», зазначаючи, що «історики можуть коли-небудь розглядати 14 грудня 1964 року, як день, коли

комп'ютерна послуга, нарешті, досягла повноліття» ...

Автор наводить власний аналіз імовірного характеру майбутніх комп'ютерних послуг через екстраполяцію існуючих комп'ютерних технологій (апаратного та програмного забезпечення), а також основних економічних, соціальних, технічних і правових проблем, які необхідно вирішити для швидкого просування комп'ютерних послуг....

Автор книги ... вказує великі переваги МАС-подібних систем [Multiple Access Computer – комп'ютер спільного доступу] для введення даних, редагування збережених файлів та легкого обміну файлами програм та даних між користувачами».

М. М. Флуд не поділяв занепокоєння Д. Ф. Паркхілла щодо можливих негативних наслідків розвитку комп'ютерних послуг: загрози конфіденційності, можливе технологічне безробіття, можливості для політичного контролю і багато інших потенційних небезпек можуть і не здійснитися, якщо будуть укладені угоди з усіма зацікавленими сторонами: державою, бізнесом, абонентами послуг. Сам Д. Ф. Паркхілл власне бачення комп'ютерної послуги реалізував, починаючи з 1969 року, у Міністерстві зв'язку Канади при розробці та впровадженні відеотекс-телетекстової системи Telidon, основна частина якої – комунікаційний протокол передавання даних у телевізійному сигналі (NABTS) – використовувалась до 2014 року в MSN TV.

Серед ранніх дослідників з Массачусетського технологічного інституту, що працювали над згадуваним Д. Ф. Паркхіллом Project MАС, виділимо М. Грінбергера, який був піонером проектування систем розподілу комп'ютерного часу та даних. Поняття «загальнодоступних комп'ютерних послуг» («computing as public utility») [18, с. 236], що належить Дж. Маккарті, з'явилося саме у збірнику «Управління та комп'ютери майбутнього» за редакцією М. Грінбергера. Подальший розвиток – від комп'ютерних послуг до інформаційних – став предметом обговорення у роботі «Комп'ютери завтрашнього дня» [35].

М. Грінбергер зазначав, що економічні та політичні умови, що склались у

1960-х рр., сприяють поширенню комп'ютерних послуг в усіх сферах суспільного життя для розв'язання широкого кола завдань: від рутинних чисельних розрахунків та маніпуляції текстовими даними до автоматичного управління приладами, моделювання динамічних процесів, статистичного аналізу та забезпечення інформаційної діяльності.

Необхідною основою цього було поширення систем розподілу комп'ютерного часу, що надають можливість зменшити накладні витрати, пов'язані з опрацюванням користувацьких програм та даних, їх зберіганням та доставлянням.

Процедура надання комп'ютерних послуг має бути простою та прозорою, як надання послуг електричної мережі:

– по-перше, щоб отримати доступ до електричної мережі, достатньо увімкнути клему на комутаторі або вставити вилку в розетку – так само простим має бути доступ й до комп'ютерної мережі;

– по-друге, була винайдена і вдосконалена велика кількість різних видів електроустаткування, кожне з яких виконує власну функцію і має власне живлення – так само і комп'ютер із ЕОМ загального призначення має еволюціонувати у велику кількість різних видів комп'ютерних пристроїв, пов'язаних комп'ютерною мережею;

– по-третє, електрика є відносно однорідним продуктом, що централізовано виробляється і безперервно надається споживачеві – так само й комп'ютерні послуги мають бути стандартизовані та надаватись споживачеві за вимогою.

Головним різновидом комп'ютерних послуг М. Грінбергер вважав інформаційні послуги загального призначення, особливістю яких є те, що їх виробництво та споживання відбувається у одній мережі та виконується одними засобами, на відміну від доступу до електричної мережі та виробництва електроенергії.

Ураховуючи активний розвиток нинішніх ліній зв'язку, М. Грінбергер вважав малоімовірним, що компанії, які надаватимуть інформаційні послуги,

будуть інвестувати у створення власних мереж зв'язку. Цей прогноз здійснювався протягом майже 50 років, доки провідні світові постачальники інформаційних послуг не розпочали розвиток власних безпроводних мереж зв'язку на основі аеростатів (Google [45]) та безпілотників (Facebook [59]) з метою забезпечення доступу до послуг мешканців сільських, віддалених та необслуговуваних районів.

Забезпечення надання інформаційних послуг може бути державним (сучасними прикладами такого забезпечення є державні Інтернет-провайдери КНДР), державно регульованим (приклад – фільтрація Інтернет-трафіку з метою забезпечення інформаційної безпеки КНР) та вільним (нерегульованим).

Від інформаційних послуг важко або взагалі неможливо відмовитись, проте для їх надання необхідні великі інвестиції у програмне забезпечення послуг. Частина цих витрат можна перекласти на клієнтів, але тоді відповідальність за розробку, обслуговування та модифікацію ядра системи надання послуг необхідно покласти на постачальника послуги.

Найпершою сферою упровадження інформаційних послуг, на думку М. Грінбергера, будуть банківська справа та роздрібна торгівля. Автор фактично описав сучасний стан мобільного та Інтернет-банкінгу, роботу касових терміналів тощо, наголошуючи на тих зручностях, що вони надають. Прогноз щодо поширення віртуальних магазинів та електронних грошей не просто справдився – сьогодні це є провідним напрямом розвитку торгівельно-грошових відносин.

Серед інших прикладів, що обговорюються у роботі – комп'ютерно керовані страхові послуги та фондові ринки («комп'ютер може бути запрограмований для підтримки стабільного та динамічного ринку ... в інтересах суспільства. Кожен інвестор матиме "місце" на комп'ютеризованому обміні, і навіть брокери стануть необов'язковими...» [35]), медико-інформаційні системи, системи централізованого контролю дорожнього трафіку, системи комп'ютерної торгівлі, автоматичні бібліотеки, інтегровані системи управління, навчальні термінали та навчальні мережі, науково-дослідні термінали у

лабораторіях, проектні термінали в інженерних фірмах, комп'ютеризовані спільноти тощо.

Окремий розділ роботи [35] присвячено наданню інформаційних послуг з моделювання як найбільш перспективного підходу до аналізу складних систем і випадкових процесів: «використання моделювання ... системними аналітиками, керівниками, вченими-суспільствознавцями, та іншими фахівцями буде помітно розширюватися, оскільки інформаційна послуга забезпечує легкий доступ до потужних комп'ютерів і систем програмування.

Більшість користувачів [інформаційних послуг з] моделювання не матимуть знання або бажання будувати свої власні моделі... Сприяння в розробці, налагодженні, та перевірці достовірності моделей буде надана он-лайн центром моделювання...

Якщо виключити непередбачені перешкоди, інтерактивний он-лайн комп'ютерний сервіс з надання інформаційних послуг може бути настільки ж звичайним явищем у 2000 році, як телефонний зв'язок сьогодні. У 2000 році людина повинна мати набагато краще розуміння себе і своєї системи, не тому, що вона буде розумнішою за наших сучасників, а тому, що вона навчиться творчо використовувати найпотужніший з винайдених підсилювачів інтелекту.» [35]

Таким чином, вже через п'ять років технологічні проблеми надання комп'ютерних послуг, указані А. О. Манном, відійшли на другий план перед соціальними, визначеними Д. Ф. Паркхіллом та М. Грінбергером: загрози конфіденційності, технологічне безробіття, можливості для політичного контролю і багатьма іншими небезпеками, потенційними у 1960-х р. та реальними у 2010-х. Через 35 років ідеї А. О. Манна щодо надання комп'ютерних послуг як комунальних реалізувались Інтернет-провайдерами, а ще через 15 років ідеї М. Грінбергера стосовно надання спеціалізованих комп'ютерних інформаційних послуг (зокрема, з комп'ютерного моделювання) через мережі зв'язку трансформувались у хмарні технології, основою яких є хмарні обчислення (cloud computing).

У 2014 році А. В. Зверева поняття послуги системної інтеграції у сфері інформаційних технологій розглянула як «сукупність ІТ-послуг по створенню та супроводу (післяпродажному обслуговуванню) ІТ-інфраструктури підприємства, яка включає в себе центри опрацювання даних, мережі, робочі місця, прикладне програмне забезпечення та корпоративні інформаційні системи, а також систему інформаційної безпеки» та класифікувала дану послугу за наступними основними ознаками:

- за територіальною ознакою: послуги локальні (обмежені підприємством) або територіально розподілені;
- за масштабом виконуваного проекту: створення комплексного інформаційного середовища або автоматизація окремих видів діяльності організації в рамках вже наявного інформаційного середовища;
- за наявності / відсутності аутсорсингу: побудова інформаційного простору на базі власної ІТ-інфраструктури або різні види оренди ІТ-інфраструктури та супутніх послуг;
- за місцем надання послуги: послуга призначена для об'єктів, розташованих у великих містах, регіональних центрах, невеликих містах і малих населених пунктах [121, с. 12-14].

За даними ознаками послуга системної інтеграції у сфері інформаційних технологій є спеціалізованою комп'ютерною інформаційною послугою, що надається засобами хмарних обчислень, перше повідомлення про які, за даними Google Ngram Viewer [33], з'явилося у грудні 2007 року (рис. 1.3).

Авторами першого повідомлення про хмарні обчислення «Google and IBM Donate 1,600 Computers to 'Cloud' Project» (рис. 1.4) наголошувалось про існуючу можливість узгодженого та ефективного вирішення спільної проблеми великою кількістю комп'ютерів, у зв'язку з чим ряд компаній та університетів США виділили кошти на створення і застосування «хмарних» обчислювальних проєктів: «За суттю, хмари є кластерами комп'ютерів – чисельністю від декількох десятків до декількох тисяч – що опрацьовують дані одночасно. Google та IBM щойно пожертвували 1600 комп'ютерів для використання з цією

метою у трьох університетах. Один з них, Університет штату Меріленд – планує використовувати хмару для перекладу складних іншомовних текстів. Студенти створять програмне забезпечення для переваг хмарного комп'ютера. Учасники вірять, що така підготовка необхідна для того, щоб упоратись зі зростаючим обсягом даних, що мають бути опрацьовані» [32].

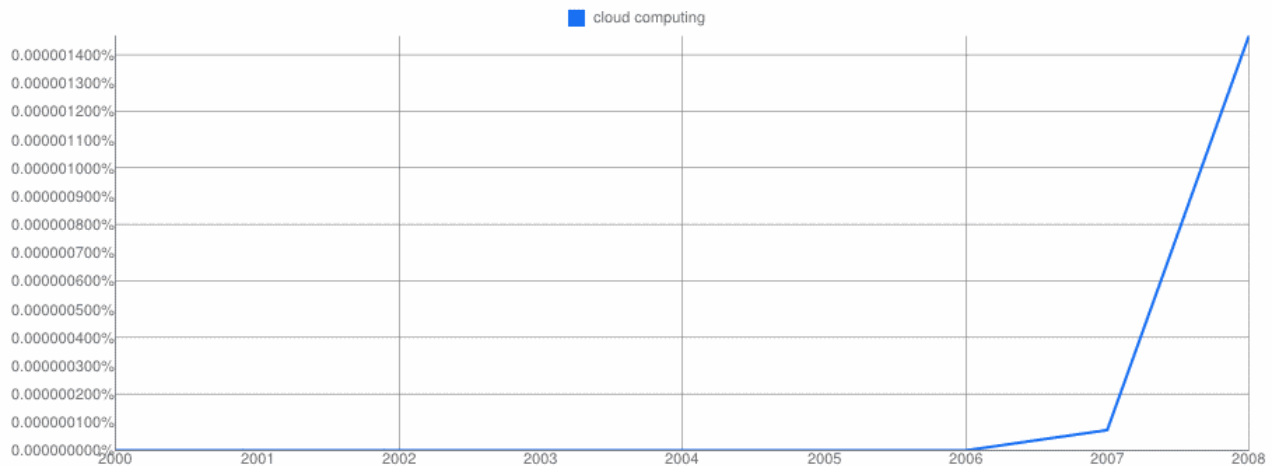


Рис. 1.3. Результати пошуку у системі Google Ngram Viewer

Google and IBM Donate 1,600 Computers to 'Cloud' Project

It stands to reason you would be able to tackle a task more effectively with many computers than you could with one—the challenge is getting the many to work in concert. That's why a number of U.S companies and universities are devoting resources to the creation and application of "cloud" computing projects. Clouds are essentially clusters of computers—numbering from dozens to thousands—that process data simultaneously.

Google and IBM recently donated 1,600 computers to be used for this purpose to three universities. One, the University of Maryland, plans to use its cloud to translate difficult foreign language texts. Students there will write the software to take advantage of the cloud computer. Participants believe such training is essential for keeping pace with the growing amount of data needing to be processed.

Рис. 1.4. Перше повідомлення про хмарні обчислення [32]

У дисертації Д. Е. Ірвіна, присвяченій архітектурі операційних систем для інфраструктури мережного серверу, поняття «cloud computing» ототожнено з «utility computing» [40, с. 38].

Р. М. Шор (R. M. Shor) вказав, що організація хмарних обчислень потребує доступу до Інтернет, серверів, високошвидкісних мереж, дискового простору, баз даних та програмного забезпечення загального призначення [66, с. 3]. На думку Д. Сігле (Del Siegle), використання хмарних обчислень надає студентам і викладачам основні переваги розглянутих раніше комп'ютерних (інформаційних) послуг, зокрема – можливість використовувати програми без встановлення їх на свої комп'ютери, а також забезпечує доступ до збережених файлів з будь-якого комп'ютера, підключеного до Інтернет. Як сучасна «комп'ютерна утиліта», хмарні технології забезпечують: більш ефективні обчислення за рахунок централізованого зберігання, опрацювання та високої пропускної здатності, одночасну роботу над проектом великої кількості користувачів незалежно від їх місцезнаходження. Термін «хмарні обчислення» використовується тому, що послуги і зберігання надаються через Інтернет (або хмару) [67].

Незважаючи на наступність концепцій інформаційної послуги та хмарних обчислень, забезпечення останніх стимулювало розвиток ряду нових професій, таких як архітектор бізнес та інформаційних технологій, фахівець з хмарної міграції, інженер-програміст хмаро зорієнтованого програмного забезпечення, експерт з безпеки даних, мережний та комунікаційний аналітик [66, с. 16].

П. Мелл та Т. Гранц визначили хмарні обчислення як модель надання, за необхідності, повсюдного та зручного мережного доступу до спільно використовуваних налаштовуваних обчислювальних ресурсів, які можуть бути швидко надані та вивільнені з мінімальними зусиллями з управління або із взаємодії з постачальником послуг (сервіс-провайдером) [53, с. 2].

Дане трактування хмарних обчислень більш відоме як «визначення NIST» (The NIST Definition of Cloud Computing). Згідно цього визначення, хмарна модель підтримує високу доступність послуг (сервісів) і описується п'ятьма

основними характеристиками:

1) *самообслуговування на вимогу* (on-demand self-service), за якого споживач хмарних послуг може самостійно отримати доступ до надаваних обчислювальних ресурсів в односторонньому порядку в міру потреби, автоматично, без необхідності взаємодії з співробітниками кожного постачальника послуг;

2) *широкий мережний доступ* (broad network access), за якого запрошені обчислювальні ресурси доступні в мережі через стандартні механізми для різних платформ, тонких і товстих клієнтів (мобільних телефонів, планшетів, ноутбуків, робочих станцій і т. п.);

3) *об'єднання ресурсів в пули* (resource pooling), за якого обчислювальні ресурси провайдера об'єднуються в пули для обслуговування різних споживачів в моделі множинної оренди, з можливістю динамічного призначення і перепризначення різних фізичних і віртуальних ресурсів за запитами споживачів. Немає необхідності в тому, щоб споживач знав точне місце розташування ресурсів, проте можна вказати їх місцезнаходження на більш високому рівні абстракції (наприклад, країна, регіон чи центр опрацювання даних). Прикладами такого роду ресурсів можуть бути системи зберігання, обчислювальні потужності, пам'ять, пропускна здатність мережі;

4) *швидка еластичність* (rapid elasticity), за якої ресурси можуть бути еластично виділені і вивільнені, в деяких випадках автоматично, для швидкого масштабування пропорційно до попиту. Для споживача можливості надання ресурсів бачаться як необмежені, тобто вони можуть бути надані в будь-якій кількості та в будь-який час;

5) *вимірюваність послуги* (measured service). Хмарні системи автоматично контролюють і оптимізують використання ресурсу, вимірюючи його на певному рівні абстракції, відповідному типу послуги (наприклад, обсягу зберігання, обчислювальної потужності, смуги пропускання і активних сеансів роботи користувачів). Використані ресурси можна відстежувати і контролювати, що забезпечує прозорість як для постачальника, так і для

споживача послуги.

Автори [53] пропонують три моделі надання хмарних послуг:

а) *програмне забезпечення як послуга* – Cloud Software as a Service (SaaS), за якої постачальник послуг надає споживачеві програмні засоби, що виконуються в хмарній інфраструктурі. Програми мають бути доступні з різних клієнтських пристроїв через інтерфейс тонкого клієнта, такий як браузер. Споживач не контролює саму хмарну інфраструктуру (мережі, сервери, операційні системи, системи зберігання, деякі специфічні для програм можливості тощо), в якій виконується програма. У ряді випадків споживачеві може бути надана можливість доступу до деяких користувальницьких налаштувань;

б) *платформа як послуга* – Cloud Platform as a Service (PaaS), за якої споживачеві надаються можливості для розгортання на хмарній інфраструктурі створених споживачем або придбаних програм, що розробляються з використанням підтримуваних постачальником послуг інструментів і мов програмування;

в) *інфраструктура як послуга* – Cloud Infrastructure as a Service (IaaS), за якої споживачеві надаються засоби опрацювання даних, зберігання, мережного доступу та інших базових обчислювальних ресурсів, на яких споживач може розгорнути і виконувати довільне програмне забезпечення, включаючи операційні системи та програми. Споживач не контролює саму хмарну інфраструктуру, але може контролювати операційні системи, засоби зберігання, розміщені програми і, можливо, мати обмежений контроль над деякими мережними компонентними.

Доступ до хмари регулюється чотирма моделями розгортання:

– *приватна хмара* (private cloud), за якої хмарна інфраструктура функціонує цілком в цілях обслуговування однієї організації. Інфраструктура може управлятися самою організацією, третьою стороною або будь-якою їх комбінацією та може існувати як на стороні споживача, так і у зовнішнього постачальника послуги;

– *хмара співтовариства* (community cloud), за якої хмарна інфраструктура використовується обмеженим співтовариством споживачів з декількох організацій, що мають спільні принципи (наприклад, місію, вимоги до безпеки, правила, вимоги). Інфраструктура може управлятися самими організаціями, третьою стороною або будь-якою їх комбінацією та може існувати як на стороні споживача, так і у зовнішнього постачальника послуги;

– *публічна хмара* (public cloud), за якої хмарна інфраструктура створена в якості загальнодоступної. Така інфраструктура розгортається постачальником хмарних послуг і може належати та управлятися фірмами, навчальними чи урядовими організаціями або будь-якою їх комбінацією;

– *гібридна хмара* (hybrid cloud), за якої хмарна інфраструктура є композицією двох чи більше хмар (приватних, публічних або хмар співтовариства), що залишаються унікальними сутностями, але об'єднуються через стандартизовані або пропрієтарні технології, що забезпечують перенесення даних і програм між хмарами.

Сучасним розвитком моделі SaaS є DaaS (Desktop as Service), за якої постачальник послуг надає споживачеві доступ до віртуального екрану програмного засобу (робочого стола тощо), що виконуються в хмарній інфраструктурі.

Визначення та моделі NIST у значній мірі базуються на матеріалах «Білої книги хмарних технологій» [46] Syntec informatique (Франція), у якій вперше виконано порівняння класичної та хмарних моделей надання комп'ютерних послуг (рис. 1.5). Класична модель надання комп'ютерних послуг відзначається тим, що споживач послуги повністю управляє інфраструктурою, у той час як хмарні моделі розташовані на рис. 1.5 у порядку зменшення керованості з боку споживача.

Як стверджує Д. Сігле, «економічно доведено, що хмарні обчислення економлять час і гроші. Як недорогий та освітньо корисний ресурс, ці технології будуть корисні як для школярів, так і для студентів, оскільки надають можливість пізнання різних комп'ютерних досягнень та здобуття

навичок роботи у спільному навчальному середовищі. Так роботу, розпочату над проектом у начальному закладі, можна продовжити вдома, якщо здійснити передачу файлів, завантаживши спільне програмне забезпечення» [67, с. 42].

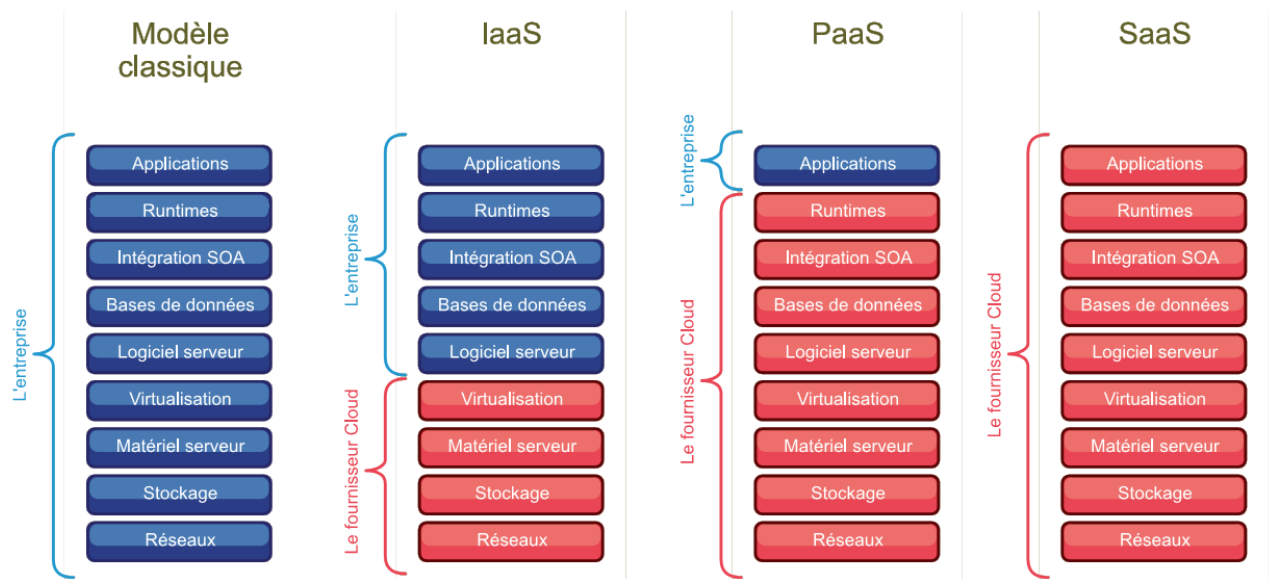


Рис. 1.5. Класичні та хмарні моделі надання комп'ютерних послуг [46, с. 6]

Таким чином, розвиток концепції комп'ютерних послуг за останні 50 років відзначався неперервністю: термінальні послуги 1960-х викликали до життя мережні операційні системи 1970-х, у яких сформувалась мережна культура 1980-х, що набула поширення у гіпертекстовому Інтернет 1990-х, який перетворився на джерело надання послуг у 2000-х та компонент соціального життя – у 2010-х. У той же час розвиток засобів хмарних технологій є діалектичним: у 2010-х, як і в 1980-х рр., нові термінальні комп'ютерні пристрої (смартфони, фаблети тощо) все більше набувають характеристик персональних комп'ютерів із мережним доступом, що можуть об'єднуватись у власні мережі за p2p-технологіями (Bluetooth, WiFi Direct тощо).

Для визначення хмарних технологій навчання скористаємося еталонним трактуванням М. І. Жалдака *інформаційно-комунікаційних технологій* як сукупності «методів, засобів і прийомів, використовуваних для збирання, систематизації, зберігання, опрацювання, передавання, подання всеможливих повідомлень і даних» [118, с. 8].

Тоді *хмарні технології* (хмарні інформаційно-комунікаційні технології) як різновид ІКТ визначимо як сукупність методів, засобів і прийомів, використовуваних для збирання, систематизації, зберігання та опрацювання на віддалених серверах, передавання через мережу та подання через клієнтську програму всеможливих повідомлень і даних.

За В. Ю. Биковим, педагогічна технологія – це структура організації часової і просторової взаємодії складових педагогічної системи і компонент навчальної системи, яка побудована відповідно до цілей навчання і виховання, змісту навчання та обраних методів навчання і виховання [86, с. 386]. Множину педагогічних технологій, що використовуються у процесі навчання, називають *технологіями навчання*.

За В. Ю. Биковим, *інформаційно-комунікаційні технології навчання* – це «комп'ютерно орієнтована складова педагогічної технології, яка відображає деяку формалізовану модель певного компоненту змісту навчання і методики його подання у навчальному процесі, яка представлена в цьому процесі педагогічними програмними засобами і яка передбачає використання комп'ютера, комп'ютерно орієнтованих засобів навчання і комп'ютерних комунікаційних мереж для розв'язування дидактичних завдань або їх фрагментів» [86, с. 141].

Отже, враховуючи, що хмарні технології є підмножиною інформаційно-комунікаційних технологій, а ІКТ навчання є підмножиною технологій навчання, під *хмарними технологіями навчання* будемо розуміти такі ІКТ навчання, що передбачають використання хмарних ІКТ. Останні спрощено можуть бути визначені як мережні ІКТ, що передбачають централізоване мережне зберігання та опрацювання даних (виконання програм), за якого користувач виступає клієнтом (користувачем послуг), а «хмара» – сервером (постачальником послуг).

На рис. 1.6 наведено співвідношення змісту понять «технологія навчання», «ІКТ», «хмарні технології», «ІКТ навчання» та «хмарні технології навчання».

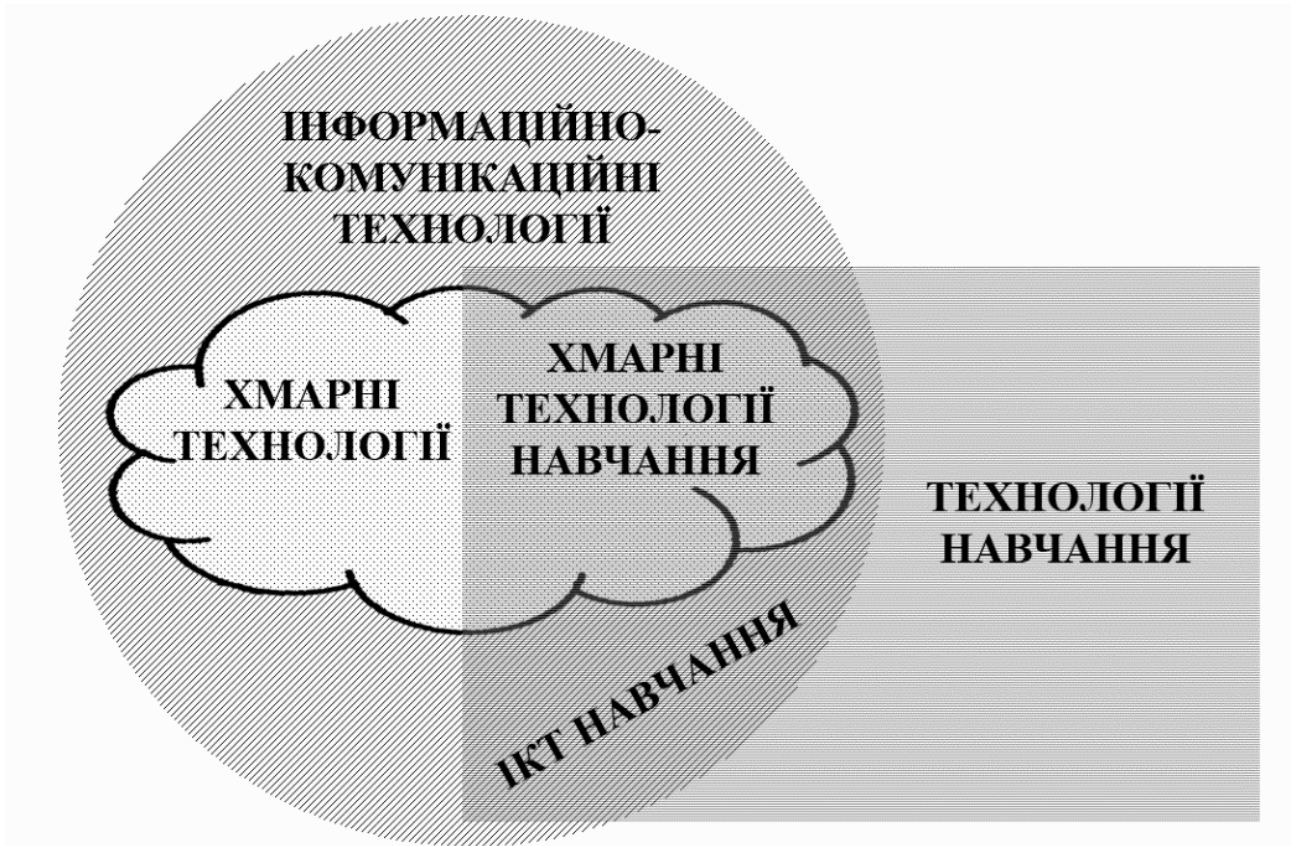







Рис. 1.6. Співвідношення технологій навчання, інформаційно-комунікаційних технологій, хмарних технологій, ІКТ навчання та хмарних технологій навчання

( – технології навчання,  – ІКТ,  – ІКТ навчання,  – хмарні технології,  – хмарні технології навчання)

1.3 Досвід використання хмарних технологій у підготовці фахівців з інформаційних технологій

Сучасний рівень навчання математичної інформатики майбутніх фахівців у галузі інформаційних технологій вимагає використання таких технологій, які б у повній мірі відповідали вимогам сучасності. Так, Я. Соммервілл у [69] окреслює проблему навчання хмарних технологій з точки зору програмної інженерії, розглядаючи три етапи такого навчання:

1) сенсibilізація (*Sensitisation*) – ознайомчий рівень, на якому студентам надається можливість ознайомитись зі змістом навчання (на цьому рівні не передбачається отримання студентами практичних навичок та теоретичних

знань);

2) практика (*Practice*) – рівень отримання практичних навичок, що характеризується активним використанням студентами хмарних послуг AWS чи Microsoft Azure;

3) принципи (*Principles*) – рівень засвоєння студентами фундаментальних принципів хмарних технологій, на якому студенти отримують розуміння про існування загальних, а не конкретних знань, необхідних для розуміння нових систем та послуг.

К. В. Болгова, розглядаючи використання хмарних обчислень для розробки віртуальних лабораторій, що використовують методи чисельного моделювання для відтворення досліджуваних процесів і явищ при неможливості доступу до реального експериментального устаткування, пропонує наступні моделі автоматизації освітніх процесів:

– IaaS: характеризується віртуалізацією обчислювальної інфраструктури ЗВО з подальшим її наданням різним підрозділам для вирішення власних завдань (у тому числі установки спеціалізованих програм для розгортання електронних освітніх ресурсів з доступом через Інтернет);

– PaaS: орієнтована на надання віртуальних ресурсів з уже встановленими обчислювальними пакетами, що забезпечують моделювання та доступ до даних;

– SaaS: традиційна модель надання доступу до програмного забезпечення як до веб-додатку, що забезпечує можливість використання електронних освітніх ресурсів;

– Data as a Service: допоміжна модель, орієнтована на використання хмарних сховищ для колективного доступу до масивів даних, що застосовуються при роботі з електронними освітніми ресурсами;

– HaaS (Hardware as a Service): специфічна модель для організації віртуальних лабораторій на основі не тільки комп'ютерного моделювання, а й віддаленого доступу до реальних інформаційно-вимірювальних систем або інших технічних засобів [89, с. 6-7].

Дослідник зазначає, що у світлі розвитку мультидисциплінарних

напрямів наукових досліджень і відповідних освітніх програм окремої уваги заслуговує модель організації хмарних обчислень AaaS (Application as a Service), яка забезпечує розробку і використання композитних програм– сукупності взаємодіючих хмарних сервісів, орієнтованих на вирішення загального завдання. Такі програми створюються за рахунок зв'язування існуючих програмних пакетів у складний програмний комплекс, що виконується на розподілених ресурсах хмарного середовища. Однією з найбільш прогресивних технологій роботи з композитними програмами є концепція, орієнтована на розвиток інтелектуальних технологій підтримки життєвого циклу проблемно-орієнтованих середовищ розподілених обчислень з використанням моделі AaaS. Зокрема, вона забезпечує інтелектуальну підтримку робочих процесів користувача на основі експертних знань, відчужуваних безпосередньо від розробників предметно-орієнтованих сервісів, навколо яких формується віртуальне професійне співтовариство.

Характеризуючи таке предметно-орієнтоване хмарне середовище, М. Є. Федосін виділяє три типи технологій забезпечення дослідницької діяльності:

- грід, що надають уніфікований доступ до обчислювального обладнання як до єдиної платформи;
- хмарні обчислення;
- веб-лабораторії на основі технологій Web 2.0, спрямовані на організацію предметно-орієнтованих наукових співтовариств з наданням користувачам розвинених засобів комунікації та взаємодії [247, с. 8].

Основна відмінність між грід та хмарними обчисленнями, на думку Є. М. Найнга, полягає у їх використанні: грід переважно використовуються для розв'язання задач за обмежений проміжок часу, а хмарні обчислення орієнтовані на надання довготривалої послуги [112, с. 7]. Дослідник робить висновок про те, що грід та хмарні обчислення доповнюють один одного.

Г. А. Алексанян вказує, що застосування хмарних технологій дозволяє більш ефективно організувати самостійну діяльність за рахунок мобільності,

доступності й зручності використання на будь-якому пристрої з доступом до Інтернет [78, с. 5]. Упровадження хмарних технологій у навчальний процес вищої та середньої школи, на його думку, забезпечує:

- ефективне використання навчальних площ (відпадає необхідність виділення окремих та спеціально обладнаних приміщень під традиційні комп'ютерні аудиторії);

- можливість швидкого створення, адаптування і тиражування електронних освітніх ресурсів;

- кардинальне скорочення витрат, необхідних на створення і підтримку комп'ютерних аудиторій;

- можливість для студентів здійснювати зворотний зв'язок із викладачем шляхом оцінки та коментування пропонованих їм освітніх сервісів;

- мобільність студентів через можливість навчатися в будь-який час і в будь-якому місці, де є доступ до Інтернет;

- гарантію ліцензійності програмного забезпечення, використовуваного у процесі навчання, та скорочення витрат шляхом створення функціонально еквівалентних освітніх сервісів на базі програмного забезпечення з відкритим кодом;

- мінімізацію кількості необхідних ліцензій за рахунок їх централізованого використання;

- можливість централізованого адміністрування програмних та інформаційних ресурсів, що використовуються у навчальному процесі [78, с. 45-46].

Е. А. Альдахіль основою застосування хмарних технологій у навчанні вважає дистанційне навчання [4, с. 21], під яким у вітчизняній нормативній базі «розуміється індивідуалізований процес набуття знань, умінь, навичок і способів пізнавальної діяльності людини, який відбувається в основному за опосередкованої взаємодії віддалених один від одного учасників навчального процесу у спеціалізованому середовищі, яке функціонує на базі сучасних психолого-педагогічних та інформаційно-комунікаційних технологій» [203].

Основною метою дистанційного навчання є надання освітніх послуг шляхом застосування у навчанні сучасних ІКТ за певними освітніми або освітньо-кваліфікаційними рівнями відповідно до державних стандартів освіти.

Автори [203], характеризуючи одне із уживаних понять – «веб-ресурси навчальних дисциплін (програм)» (у тому числі дистанційні курси), – вказують, що вони, зокрема, мають бути доступні через Інтернет за допомогою веб-браузера. Останній виступає у якості клієнтської частини програмного забезпечення, за допомогою якого надається послуга у вигляді дистанційного навчального курсу. Веб-ресурси навчальних дисциплін (програм), що необхідні для забезпечення дистанційного навчання, можуть містити:

- методичні рекомендації щодо їх використання, послідовності виконання завдань, особливостей контролю тощо;
- документи планування навчального процесу (навчальні програми, навчально-тематичні плани, розклади занять);
- відео– та аудіозаписи лекцій, семінарів тощо;
- мультимедійні лекційні матеріали;
- термінологічні словники;
- практичні завдання із методичними рекомендаціями щодо їх виконання;
- віртуальні лабораторні роботи із методичними рекомендаціями щодо їх виконання;
- віртуальні тренажери із методичними рекомендаціями щодо їх використання;
- пакети тестових завдань для проведення контрольних заходів, тестування із автоматизованою перевіркою результатів, тестування із перевіркою викладачем;
- ділові ігри із методичними рекомендаціями щодо їх використання;
- електронні бібліотеки чи посилання на них;
- бібліографії;
- дистанційний курс, що об'єднує зазначені вище веб-ресурси навчальної дисципліни (програми) єдиним педагогічним сценарієм;

– інші ресурси навчального призначення [203].

У роботі Т. М. Шалкіної [257] розглянуто інноваційне інформаційно-предметне середовище підготовки майбутніх інженерів-програмістів як сукупність педагогічних, інформаційно-комунікативних, матеріально-технічних компонентів, необхідних для організації навчальної діяльності студентів з формування професіональних знань і умінь у виділеній предметній галузі в процесі розв'язання професійно-орієнтованих задач. Дослідник наголошує на тому, що інформаційно-предметне середовище стає фактором підготовки майбутніх інженерів-програмістів при реалізації наступних умов:

– наявність автоматизованого навчального середовища і сукупності навчально-методичних матеріалів для організації навчальної діяльності студента;

– включення студента у професійну діяльність шляхом запровадження в навчальний процес професійно-орієнтованих задач;

– орієнтація студента на самостійну, пошукову, науково-дослідницьку діяльність, використовуючи ресурси мережі Інтернет, різні електронні і друковані носії для пошуку матеріалу у виділеній предметній галузі, аналізу отриманої інформації, обґрунтування вибору засобів вирішення задачі.

Остання умова розвинена А. О. Ричковою [209] з позицій особистісно-діяльнісного підходу у напрямі формування професійної самостійності майбутніх інженерів-програмістів засобами дистанційних технологій навчання. Зокрема, дослідником визначені дидактичні, психолого-педагогічні і організаційно-комунікативні можливості використання дистанційних технологій навчання та розроблена відповідна методика.

У хмаро зорієнтованому навчальному середовищі дистанційне навчання може бути реалізоване у віртуальному класі (середовищі), в якому навчальна взаємодія студентів та викладача опосередковується хмарними ІКТ. Е. А. Альдахіль зазначає, що із розвитком Web 2.0 все більшої важливості у віртуальному навчальному середовищі набуває соціальна взаємодія [4, с. 22].

Е. А. Альдахіль пропонує виділити два основні типи віртуальних класів,

що розрізняються за режимом взаємодії суб'єктів навчання – асинхронні та синхронні. Проте у більшості випадків чітке доцільними є не їх розмежування, а їх комбінування: так, А. М. Стрюк вказує, що, якщо розглядати форми організації навчання як синхронні (спільна спеціально організована навчальна діяльність у визначений час у визначеному місці – наприклад, у традиційному аудиторному навчанні) та асинхронні (індивідуальна навчальна діяльність, що має бути виконана за певний час – наприклад, у традиційному заочному навчанні), то навіть за традиційної організації навчального процесу комбінування форм організації навчання стає необхідним. Така синхронна форма, як лекція, може супроводжуватися як фронтальною (синхронною), так і груповою (синхронно-асинхронною), і навіть індивідуальною (асинхронною) лабораторною роботою. Більшою мірою це стосується активності навчальної діяльності, класифікацію якої можна подати відповідно зі співвідношенням синхронних та асинхронних часових проміжків, оскільки чим вище рівень самостійності навчальної діяльності, тим вищий ступінь її асинхронності. При цьому найвищий ступінь асинхронності (та, відповідно, найбільшу частину самостійної роботи) мають дистанційне та мобільне навчання [228, с. 20].

Розглядаючи тенденції розвитку комбінованого навчання, К. Дж. Бонк та Ч. Р. Грем (Charles R. Graham) [10] вказують, що рівень використання засобів ІКТ у найближчому майбутньому вже не буде фактором вибору форми навчання, а враховуючи, що частка самостійної роботи у процесі підготовки бакалаврів комп'ютерної інженерії вже є досить значною, доцільним є розгляд комбінованого навчання у вимірі «мобільність ІКТ – синхронність». Дослідники Мічиганського віртуального університету на чолі з Дж. Уотсоном (John Watson) вважають, що у процесі комбінованого навчання із зростанням частки самостійної роботи змінюються ролі викладача, студента та технології навчання через розширення меж використання онлайн навчання, збільшення гнучкості планування занять та консультаційної підтримки, перехід до мобільного навчання [76, с. 12].

Дж. Адамс (Jean Adams) вказує, що комбіноване навчання інтегрує

синхронні та асинхронні комунікаційні технології, формальне та неформальне навчання, друковані та електронні навчальні матеріали, онлайн та оффлайн фасилітацію, забезпечуючи умови для створення якісних інтерактивних навчальних матеріалів для самонавчання та неперервної підтримки процесу навчання [2]. Аналізуючи вимоги до системи підтримки комбінованого навчання системного програмування бакалаврів програмної інженерії, А. М. Стрюк у якості важливого засобу організації спільної роботи викладача та студентів вказує віртуальні класи та хмарні технології [228, с. 100].

Комбіноване навчання реалізує чотири системні принципи відкритої освіти, сформульованих В. Ю. Биковим: мобільності учнів і вчителів; рівного доступу до освітніх систем; надання якісної освіти; принцип формування структури та реалізації освітніх послуг [214, с. 55-56]. Стосовно комбінованого навчання В. Ю. Биков зазначає, що «на практиці в межах єдиного навчального плану і навчально-виховного процесу різні форми здобуття освіти поєднуються між собою у тих чи інших пропорціях... Такий підхід надає можливість раціонально організувати навчально-виховний процес (з точки зору досягнення високої якості освіти), використовуючи такі форми навчання на тих або інших його етапах, які виявляються найбільш доцільними з педагогічної точки зору на кожному з них і дозволяють найкращим чином поєднати і збалансувати наявні можливості навчального закладу (насамперед, врахувати обмеження ресурсного характеру) і можливості (бажання) тих, хто навчається» [86, с. 99]. Мобільне навчання С. О. Семеріков визначає як «підхід до навчання, при якому на основі мобільних електронних пристроїв створюється мобільне освітнє середовище, де студенти можуть використовувати їх у якості засобу доступу до навчальних матеріалів, розміщених в Інтернеті, будь-де та будь-коли» [214, с. 119]. Навчальне середовище комбінованого навчання Е. А. Альдахіль відносить до хмаро зорієнтованих, пропонуючи для побудови такого середовища застосовувати три моделі надання хмарних послуг: SaaS, IaaS, PaaS [4, с. 31].

З. С. Сейдаметова, розглядаючи методичну систему рівневої підготовки інженерів-програмістів [211], відзначає необхідність фундаменталізації

навчання дисциплін блоку професійно-орієнтованої та практичної підготовки з одночасним «професійним тюнінгом» (налаштування на професію). Значне місце у цьому, на думку дослідника, має посідати «енкаридж» – дружнє професійне програмно-методичне оточення, що є складовою хмарно орієнтованого середовища вищого навчального закладу.

У табл. 1.6 наведено порівняння засобів технологій дистанційного навчання та засобів хмарних технологій за технічними характеристиками.

Таблиця 1.6

Порівняння засобів технологій дистанційного навчання та хмарних технологій за технічними характеристиками

Параметр	Засоби технологій дистанційного навчання	Засоби хмарних технологій
Безпека даних	Висока	Низька або середня
Безпека програмного забезпечення	Залежить від навчального закладу	Залежить від постачальника хмарних послуг
Вартість апаратного та програмного забезпечення	Середня або висока	Низька
Відновлення після збоїв	Ручне	Автоматичне
Доступність системи за збою одного з серверів	Відсутня	Наявна
Доступність системи незалежно від часових або просторових обмежень	Так	Так
Збереження даних та виконання обчислень на комп'ютері користувача	Обов'язково	Не обов'язково
Масштабованість навчальних курсів	Обмежена	Необмежена
Мобільність програмного забезпечення	Так	Так
Можливість інтеграції різних платформ	Низька	Висока
Потреба в обслуговуванні системи при зміні апаратного чи програмного забезпечення	Наявна	Відсутня
Раціональне використання ресурсів	Не завжди	Завжди
Управління змістом навчання	Вимагає певних зусиль	Вимагає незначних зусиль
Фізичне розташування навчальних матеріалів	Відоме явно	Приховане

Використання хмарних технологій у підготовці ІТ-фахівців надає ряд переваг:

– для студентів:

- 1) повсюдна доступність необхідних електронних освітніх ресурсів;
- 2) мобільність програм та даних;

3) відсутність суттєвих програмно-апаратних обмежень на використовувані ресурси;

4) опанування хмарних технологій як провідних для ІТ-галузі;

5) відсутність необхідності адміністрування програмного забезпечення для досягнення найвищої продуктивності при використанні систем програмування та ін.;

6) можливість проведення неруйнівних експериментів у віртуалізованому програмно-апаратному середовищі;

– для викладачів та співробітників:

7) можливість використання гнучких (у термінах Amazon – еластичних) хмаро зорієнтованих ресурсів (зокрема, з метою розробки завдань різного рівня складності та ресурсоспоживання);

8) можливість уніфікації програмного забезпечення у Web-орієнтованих операційних системах (наприклад, eyeOS);

9) зниження витрат на адміністрування та утримання ІТ-інфраструктури;

10) широкі можливості моделювання комп'ютерних систем та мереж;

11) можливість простого зберігання та багаторазового використання віртуальних лабораторій;

– для адміністраторів комп'ютерних систем та мереж:

12) зниження витрат на ліцензування програмного забезпечення;

13) зняття обмежень на використовуване апаратне та програмне забезпечення завдяки технології віртуалізації;

14) можливість обслуговування потенційно необмеженої кількості студентів;

15) спрощення та уніфікація технічного обслуговування у хмарі.

Переваги використання хмарних технологій для різних категорій учасників навчального процесу подано на рис. 1.7.

Таким чином, для студентів хмарні технології виступають насамперед засобом підвищення їх професійних компетентностей, для викладачів та співробітників – засобом підвищення ефективності навчання без додаткового

навантаження на них, для адміністраторів комп'ютерних систем та мереж – засобом уніфікації та спрощення задач адміністрування. Для ЗВО у цілому використання хмарних технологій приводить до вагомого зниження витрат на обслуговування, оновлення програмного забезпечення та ліцензування, апаратне конфігурування, забезпечення потужності і зменшення площі інфраструктури, що надає можливість зменшити видатки, не впливаючи на якість надання освітніх послуг.



Рис. 1.7. Переваги використання хмарних технологій для різних категорій учасників навчального процесу

Використання хмарних технологій накладає ряд обмежень.

У першу чергу, це *забезпечення конфіденційності та безпеки*: постачальники хмарних послуг надають їх через центри опрацювання даних, розташовані в різних країнах, у яких може бути законодавчо обумовлена можливість доступу до приватних даних користувачів. Крім того, при зберіганні даних у публічній хмарі користувач не має фізичного доступу до серверів із своїми даними, і до них потенційно може бути наданий несанкціонований доступ. Для підвищення безпеки постачальники хмарних послуг використовують різні методи. Наприклад, Google визначає такі напрями

забезпечення конфіденційності та безпеки даних [34]:

– корпоративна культура: перевірка анкетних даних співробітників, навчання всіх співробітників заходам безпеки, інформаційні заходи про безпеку та конфіденційність, окремі підрозділи для забезпечення конфіденційності та безпеки, внутрішній аудит дотримання безпеки, співпраця із фахівцями у галузі безпеки;

– експлуатаційна безпека: управління уразливостями, запобігання виконання шкідливого програмного забезпечення, моніторинг, розслідування інцидентів;

– технологічна безпека: новітні центри опрацювання даних, спеціальне серверне апаратне та програмне забезпечення, моніторинг обладнання, захищена глобальна мережа, захист даних при передаванні, незначна затримка та постійна доступність обладнання і послуг;

– незалежна стороння сертифікація;

– домовленість про використання даних: даними володіють клієнти хмарної платформи, дані не скануються та не передаються третім особам, дані фізично видаляються постачальником протягом 180 днів після видалення клієнтом;

– обмеження доступу до даних: логічна ізоляція даних користувачів хмарної платформи, аудит авторизації та доступу, розподіл прав доступу користувачем, доступ правоохоронних органів до даних клієнта лише за запитом, контроль дотримання безпеки та конфіденційності з боку сторонніх постачальників хмарних послуг.

Друге обмеження – це *швидкість передавання даних через мережу*: так як дані передаються через Інтернет-з'єднання, важливо обрати відповідного постачальника послуг Інтернет та обладнання, щоб уникнути затримки передавання даних. До основних шляхів зменшення мережних затримок відносяться:

– зменшення затримок на кожному вузлі мережі;

– мінімізація кількості проміжних мережеских вузлів;

- усунення перевантаження мережі;
- зменшення затримок, пов'язаних із транспортним протоколом [6, с. 2].

Третім обмеженням є *сумісність*: кожен постачальник хмарних послуг має власну унікальну архітектуру або реалізацію. Наприклад, Google використовує для зберігання даних BigTable – розподілену базу даних на основі Google File System, що не поширюється та не використовується за межами Google, в той час як Facebook використовує Apache Cassandra – відкриту розподілену систему управління базами даних, що використовується такими компаніями, як Cisco, IBM, Cloudkick, Reddit, Digg, Rackspace і Twitter.

Четверте обмеження – *прив'язка до постачальника хмарних послуг* – виникає за умови використання ним несумісних фірмових форматів даних, обміну повідомленнями тощо. Для запобігання такої прив'язки Дж. Маккендрік рекомендує використовувати хмарні послуги на основі промислових стандартів, а не на основі фірмових технологій постачальника послуг [51]:

1) *міжплатформенні інтерфейси* на основі WS-I (насамперед HTTP та JSON, включаючи засоби аутентифікації і управління доступом, такі як OAuth) забезпечують основну частину реальної мобільності хмарних послуг;

2) *програмно-платформенні інтерфейси* повинні бути незалежними від постачальника хмарних послуг, специфікації стандарту хмарної платформи для виконання програм повинні бути незалежним від мови, а самі інтерфейси мають бути розроблені для найбільш широко використовуваних мов програмування;

3) *специфікація послуг* має бути орієнтована на сприйняття людиною, охоплювати всі функціональні і нефункціональні характеристики, підтримувати WS-I разом із WSDL;

4) *інтерфейси управління послугами* мають бути стандартизовані для масового поширення хмарних послуг як доступного комерційного продукту, тому їх необхідно модифікувати відповідно до нових стандартів, що розробляються DMTF, OASIS, OGF, SNIA та OpenStack;

5) *моделі даних* мають бути чітко визначені з використанням стандартів форматування тексту (таких, як XML, але не лише за його допомогою) та опису схем;

6) *слабкий зв'язок* між компонентами є більш доцільним, ніж сильний: сильно пов'язані компоненти важко інтегрувати, особливо у процесі розвитку хмарної системи;

7) *орієнтація на обслуговування* за відповідною стандартною моделлю надання послуг (IaaS, PaaS, SaaS тощо) надає можливість автоматично урахувати рекомендації щодо слабого зв'язку між компонентами, специфікації послуг та інтерфейсів;

8) *розповсюдження програмного забезпечення* через відповідні Інтернет-магазини має бути стандартизоване та налаштовуване відповідно до вимог законодавства;

9) *REST-орієнтований доступ до інформаційних ресурсів*, зокрема через виклики за протоколом HTTP (GET, POST, PUT, DELETE тощо) має використовуватись замість віддаленого виклику процедур (за RPC-подібними протоколами);

10) *формати образів віртуальних машин* повинні задовольняти вимогам відкритого формату віртуалізації (OVF DMTF) – це надає можливість переносити образи віртуальних машин між різними хмарними інфраструктурами як одного постачальника хмарних послуг, так й різних.

П'яте обмеження – *ясність угод про рівень обслуговування*, у тому числі щодо таких питань, як гарантована безвідмовна робота і зберігання даних, якщо користувач вирішить змінити постачальника хмарних послуг. Європейською Комісією у межах ініціативи «Європа–2020» розроблені рекомендації щодо стандартизації таких угод, що включають відповідні принципи: нейтральність до технології та бізнес-моделі, повсюдна застосовність, однозначні визначення, порівнювані цілі рівня обслуговування, забезпечення відповідності, наявність стандартів та керівництв для різних типів клієнтів, визначення суттєвих характеристик хмарних послуг, доказовість ефективності послуг та ін. [15,

с. 5-9].

Шосте обмеження – *масштабованість зберігання* – може бути розв’язане за допомогою концепції еластичності (надання додаткових ресурсів на вимогу), проте на сьогодні в автоматичному режимі це поки що неможливо.

Проведений аналіз освітньо-професійних програм підготовки фахівців з інформаційних технологій в Україні надав можливість визначити моделі надання хмарних послуг, які доцільно використовувати у процесі навчання нормативних навчальних дисциплін (табл. 1.7).

Таблиця 1.7

Моделі надання хмарних послуг, які доцільно використовувати у процесі навчання нормативних навчальних дисциплін циклів математичної, природничо-наукової й професійної та практичної підготовки майбутніх фахівців з інформаційних технологій

Модель	Навчальні дисципліни
SaaS	Вища математика Теорія ймовірності та математична статистика Алгоритми та методи обчислень Дискретна математика Екологія Комп’ютерна логіка Організація баз даних
PaaS	Фізика Теорія електричних та магнітних кіл Комп’ютерна електроніка Програмування Комп’ютерна схемотехніка Паралельні та розподілені обчислення Інженерія програмного забезпечення
IaaS	Архітектура комп’ютерів Системне програмування Системне програмне забезпечення Технології проектування комп’ютерних систем Комп’ютерні системи Комп’ютерні мережі Захист інформації у комп’ютерних системах

У табл. 1.7 вказано найменший рівень моделі розгортання, який може бути використаний при навчанні відповідної дисципліни.

Для навчальної дисципліни «Вища математика» може бути застосована

одна із Web-орієнтованих систем комп'ютерної математики або комплекс засобів ІКТ навчання вищої математики (лекційні демонстрації, динамічні моделі, тренажери та навчальні експертні системи), до яких надається спільний Web-доступ. К. І. Словак [222] теоретично обґрунтована та експериментально перевірена доцільність використання у навчанні вищої математики мобільних математичних середовищ – відкритого модульного мережного мобільного інформаційно-обчислювального програмного забезпечення, що надає користувачу (викладачу, студенту) можливість мобільного доступу до інформаційних ресурсів математичного і навчального призначення, створюючи умови для організації повного циклу навчання (зберігання та подання навчальних матеріалів; проведення навчальних математичних досліджень; підтримка індивідуальної та колективної роботи; оцінювання навчальних досягнень тощо) та інтеграції аудиторної і позааудиторної роботи у безперервний процес навчання.

До визначальних характеристик мобільного математичного середовища відносяться мобільність доступу (виконуваність на широкому спектрі комп'ютерних пристроїв, що надає можливість залучити у якості засобів навчання нетбуки, планшетні комп'ютери та смартфони), мобільність програмного забезпечення (можливість перенесення середовища на різні програмно-апаратні платформи без суттєвої модифікації), мережність (зберігання математичних об'єктів на мережних серверах, що надає можливість уніфікувати доступ до них як в аудиторії, так і за її межами), відкритість (можливість зміни інформаційної та обчислювальної складової середовища), модульність (можливість додавання, заміни та вилучення компонентів середовища), об'єктна орієнтованість (можливість прототипування, створення, модифікації, наслідування, інкапсуляції математичних об'єктів), можливість природного застосування ефективних педагогічних технологій організації спільної роботи над навчальними проектами у навчальних спільнотах.

Основними складовими авторського мобільного математичного середовища «Вища математика» є обчислювальне ядро (математичний пакет) та

інформаційне забезпечення, що містить методичні та додаткові інформаційні матеріали. У роботі показано, що в якості обчислювального ядра мобільного математичного середовища доцільно обрати Web-СКМ Sage, що надає можливість: в рамках єдиного середовища реалізувати основні типи програмних засобів (лекційні демонстрації, динамічні моделі, тренажери, навчальні експертні системи), використання яких спрямоване на активізацію навчальної (у тому числі самостійної) діяльності студентів; автоматизувати обчислювальний процес розв'язування задач прикладної спрямованості, зосередившись на побудові моделі та інтерпретації результатів обчислювального експерименту. Враховуючи, що інформаційне забезпечення, яке входить до складу мобільного математичного середовища, є предметно-орієнтованим, К. І. Словак розглядає клас мобільних математичних середовищ, що мають однакове обчислювальне ядро та варіативне інформаційне забезпечення (рис. 1.8). Так, заміна методичної складової інформаційного забезпечення у авторському мобільному математичному середовищі «Вища математика» надає можливість створювати нові середовища з предметів фізико-математичного циклу.

Мінімальний рівень моделі надання хмарних послуг, що необхідний для реалізації такого середовища – SaaS, за яким може бути отриманий доступ до Web-СКМ Sage. Для підтримки консольного інтерфейсу до такого середовища необхідною є модель PaaS. І, нарешті, самостійне розгортання середовища може бути автоматизоване через віртуалізацію комп'ютера із встановленою операційною системою та всіма компонентами середовища за моделлю IaaS (саме у такому вигляді розроблене К. І. Словак середовище поширюється через Інтернет).

С. О. Семеріков наводить приклад використання моделі SaaS у навчанні з використанням онлайн-інтегрованих середовищ програмування, вказуючи, що більшість Інтернет-IDE має досить специфічний характер (на відміну від IDE загального призначення, таких як Eclipse): IDE є не скільки послугою, що надається, скільки інструментом для користувачів, які використовують інші

послуги.

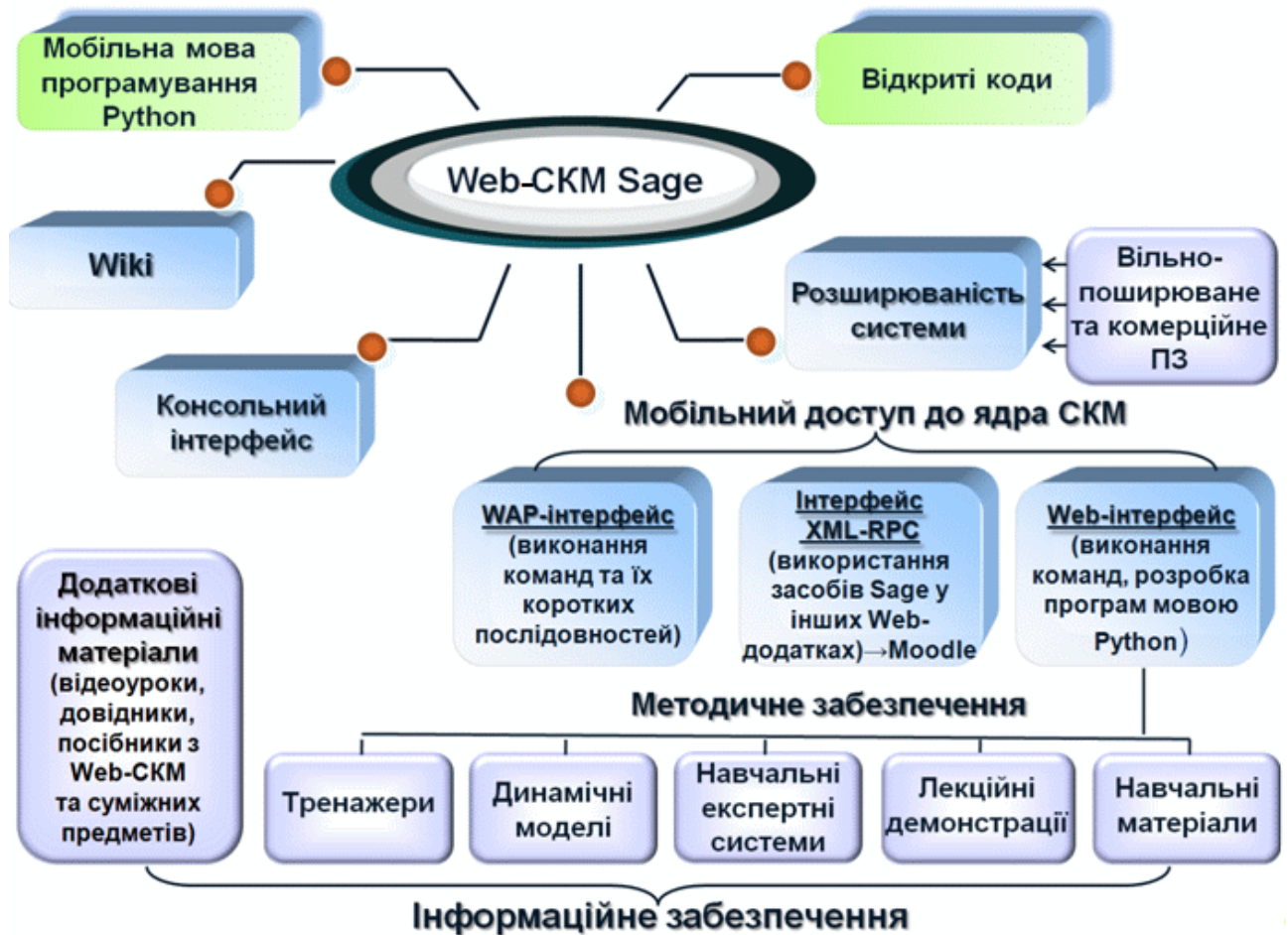


Рис. 1.8. Архітектура мобільного математичного середовища на основі Web-CKM Sage (за [222, с. 9])

Прикладами таких послуг є Coghead, ZohoCreator, BungeeBuilder, MicrosoftPopFly і YahooPipes. Всі ці сервіси є пропріетарними, в деяких з них використовуються свої власні мови, і вимагається розміщення всіх послуг виключно на їх серверах. Проте існує кілька служб, які мають більш загальний характер, та базуються на стандартних мовах. Наприклад, у Heroku застосовується мова Ruby і він може бути використаний для розробки та розгортання Ruby-додатків. Cloud9 надає можливість створення програм більш ніж 40 мовами програмування, у тому числі C#, C/C++, Clojure, CoffeeScript, ColdFusion, Groovy, Java, Javascript, Lua, OCaml, PHP, Perl, Python, Ruby, Scala. Онлайн-IDE є гарним інструментом для залучення до програмування: його використання надає користувачеві можливість швидко почати кодування і

миттєво отримати результати. Хоча онлайн-IDE не є повноцінними інтегрованими середовищами, робота з ними дає уявлення про те, як в цілому може виглядати середовище програмування. Безпосередніми перевагами онлайн-IDE є відсутність необхідності у їх інсталяції та миттєве розгортання проекту. Крім того, їх застосування відкриває нові можливості для обміну навчальними матеріалами та співробітництва [214, с. 264-269].

Таким чином, при використанні онлайн-IDE надається послуга розробки програмного забезпечення за моделлю SaaS. У той же час самі онлайн-IDE виступають відповідними платформами, доступ до яких надається за моделлю PaaS. Останнє є важливим для підготовки ІТ-фахівців, оскільки створює умови для відпрацювання реальних навичок (зокрема, адміністрування комп'ютерних систем та мереж) у віртуалізованому середовищі. Яскравим прикладом такого середовища є образ віртуальної машини (Virtual Appliances), що поширюється через VMware Solution Exchange – Virtual Education Laboratory (VEL) виробництва iNetwork, Inc. Даний продукт надає послугу віддаленого комп'ютера для навчальних експериментів у галузі інформаційних технологій. Хмаро зорієнтована інфраструктура VEL побудована на платформі VMware, а надійність її роботи забезпечують кілька центрів опрацювання даних.

Використовуючи VEL, викладачі можуть створювати власні мережні конфігурації у якості лабораторних завдань для студентів, яким надається можливість експериментувати з різними операційними системами без необхідності фізичної присутності у комп'ютерному класі. Для майбутніх фахівців із комп'ютерної інженерії VEL надає можливість опановувати методи моніторингу мережі, виявлення проникнень, розробки мобільних програм для смартфонів на базі Android, забезпечення бездротової та мобільної безпеки, аудиту загроз та ін. [74]. Отже, при застосуванні віртуальних машин у підготовці ІТ-фахівців надаються комп'ютерні послуги за моделлю IaaS.

Вибір моделі SaaS для навчання основ баз даних обумовлений мінімальними вимогами до програмного забезпечення навчальної дисципліни. Ураховуючи, що доступ до реляційних баз даних традиційно забезпечується

запитами мовою SQL, мінімально необхідне програмне забезпечення повинно надавати можливості виконання запитів мовою SQL та перегляду результатів виконання, створення баз даних, перегляд, створення та модифікація структури таблиць, заповнення та редагування даних, пошук даних за шаблонами, їх імпорт та експорт, адміністрування серверу баз даних та ін. через Web-інтерфейс (рис. 1.9).

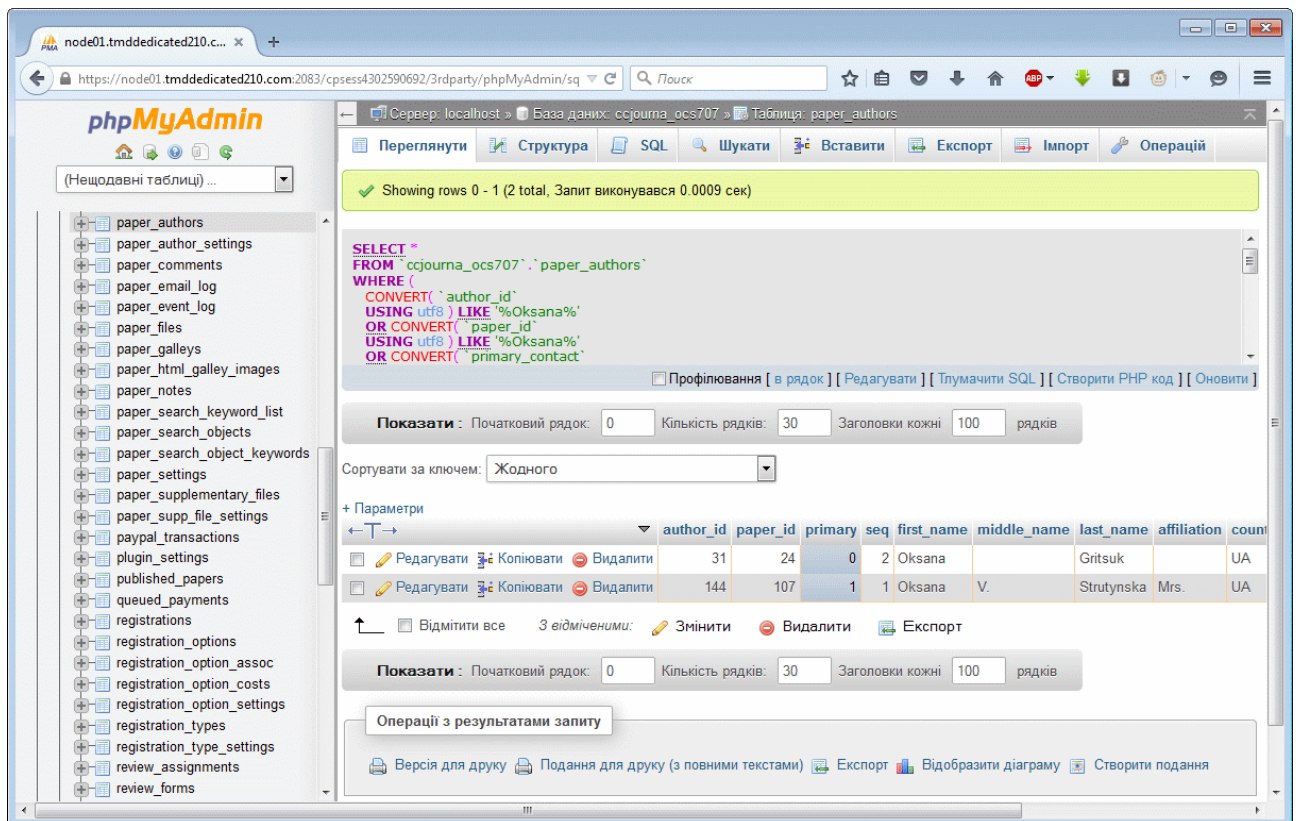


Рис. 1.9. phpMyAdmin – SaaS-засіб навчання основ баз даних

Указаним вимогам у повній мірі відповідають phpMyAdmin [21] та частина хмарної платформи Google – Cloud SQL.

Х. Раджай та Е. А. Альдахіль [64] наводять приклад використання хмарних технологій за моделлю PaaS у процесі навчання систем управління баз даних. Студентам надається можливість створювати власні бази даних, зв'язувати бази даних, що розташовані на різних серверах, отримувати дані за допомогою мови SQL. Автори [64] пропонують застосовувати для цього IBM Cloud і Windows Azure. Так, серед налаштованих образів віртуальних машин від IBM є образи СУБД DB2. Студенти створюють власні бази даних, обираючи

і доповнюючи образ DB2. Після початкового налаштування DB2, студенти можуть підвищити рівень безпеки шляхом створення приватних і публічних ключів та визначення різних рівнів доступу до бази даних (власник, адміністратор, користувач). Після цього студенти можуть піднятися до рівня SaaS, використовуючи Web-браузер для створення запитів мовою SQL через інтерфейс phpMyAdmin або створений власноруч. Крім того, вони можуть використовувати віддалений доступ з власних комп'ютерів за допомогою відкритих і закритих ключів за моделлю DaaS.

Два із чотирьох головних компонентів хмарної платформи Windows Azure (рис. 1.10) пов'язані з базами даних: послуги Storage (доступу до таблиць, неструктурованих даних великого обсягу, файлів та запитів) та Azure SQL Database (Azure Search, Document DB, Redis Cache та StorSimple) надають можливість доступу до нереляційних та реляційних СУБД відповідно.

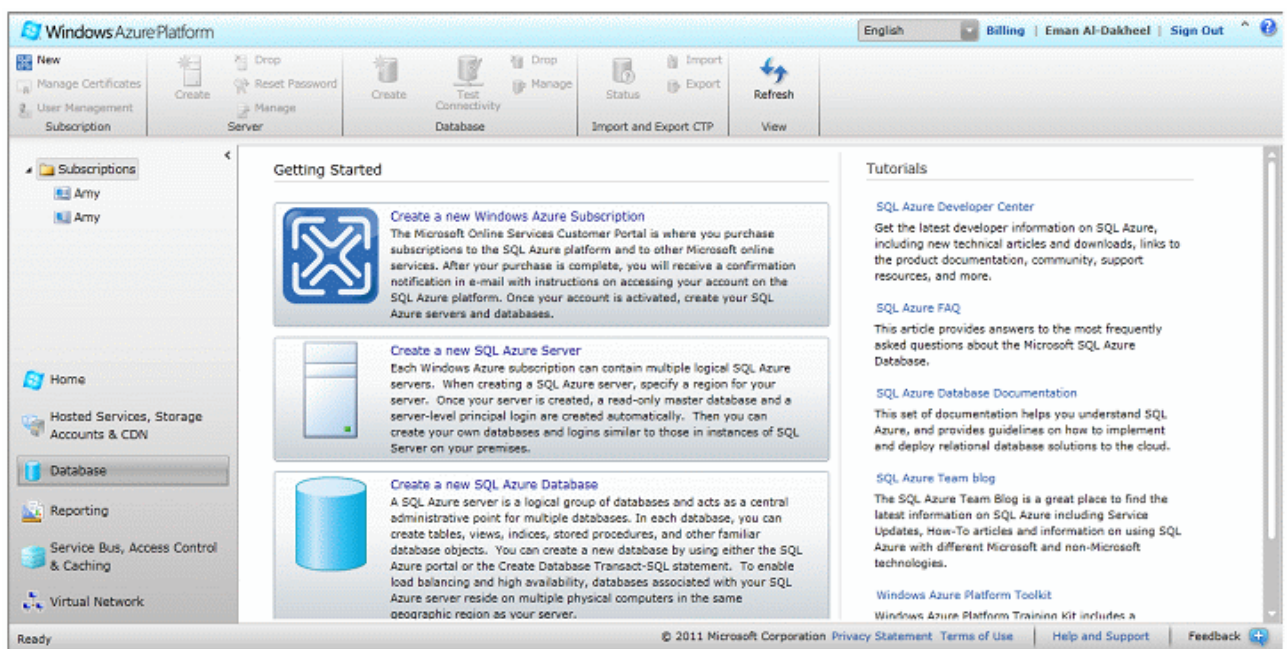


Рис. 1.10. Можливості із роботи з хмарними СУБД, що надає Windows Azure

Навчання операційних систем – одне із традиційних застосувань віртуальних машин на основі IaaS: середовища віртуалізації, що включає в себе сервери, програмне забезпечення та мережне обладнання. Для доступу до IaaS клієнти мають сплатити вартість обраної послуги, що визначається споживанням таких комп'ютерних ресурсів, як час роботи у операційній

системі, час використання системи опрацювання даних, дисковий простір, мережний трафік.

Однією із найбільш розвинених інфраструктур є Amazon Web Services – інфраструктура хмарних обчислень Amazon (Amazon Web Services (AWS)), що надає можливість вільного вибору операційної системи, моделі програмування, конфігурації обчислювальної системи. Послуги, що надає AWS, спрямовані на полегшення управління простими (Amazon SimpleDB) та реляційними базами даних (Amazon RDS), запитами (Amazon SQS), платіжами (Amazon FPS), зберігання (Amazon S3) та доставляння даних (Amazon CloudFront), віртуалізацію (Amazon EC2), обмін повідомленнями у хмарі (Amazon SNS) та між хмарою і приватною мережею організації (Amazon VPC).

Центральним компонентом AWS є Amazon EC2 (Amazon Elastic Compute Cloud), що використовує Amazon Machine Image – образ віртуальної машини, що містить операційну систему (Linux, Windows та ін.) та необхідне користувачеві хмарних послуг програмне забезпечення. Для використання образу у EC2 його файлова система образу стискається, шифрується, підписується цифровим підписом та ділиться на 10-мегабайтні частини, що завантажуються на сервер Amazon S3 для зберігання.

«Еластичність» сервісу EC2 забезпечується: 1) оплатою лише за час активності послуги; 2) урахуванням географічного положення клієнта та серверів. Еластичний обчислювальний блок (Elastic Compute Unit (ECU)) у Amazon EC2 є абстракцією комп'ютерних ресурсів, що відповідає моделям 2007 року процесорів Opteron та Xeon з тактовою частотою 1.0-1.2 ГГц.

EC2 використовує монітор віртуальних машин Xen, кожна з яких працює на віртуальному приватному сервері. Найпростішими серед стандартних типів віртуальних машин EC2 є мінімальні (tiny) конфігурації t1.micro та t2.micro, що надають 640 Мб та 1 Гб ОЗП відповідно.

Починаючи з грудня 2010 р., Amazon надає новим користувачам безкоштовний ресурсний кредит у розмірі 750 год. на місяць роботи конфігурації t2.micro під управлінням Linux або Windows та 30 Гб дискового

простору, що може бути використаний протягом року. Після року використання або вичерпання кредиту користувач переходить на платне обслуговування (оплати здійснюється лише за фактично спожиті ресурси).

Для моніторингу роботи хмари Amazon надає послугу Cloud Watch, за допомогою якої користувачі контролюють використання процесора, диску та мережі. При використанні послуги Amazon's Auto Scaling feature за нестачі системних ресурсів вони автоматично додаються, що забезпечує безперервну роботу віртуальної машини у хмарі.

Для початку роботи у EC2 необхідно звернутися за адресою <http://aws.amazon.com/ec2/> та зареєструватися у якості користувача (рис. 1.11).

The screenshot displays the AWS Management Console interface. At the top, there is a navigation bar with 'AWS Services Edit' and the user name 'Oksana Markova'. The main content area is titled 'Amazon Web Services' and is divided into several columns of service categories:

- Compute:** EC2 (Virtual Servers in the Cloud), EC2 Container Service (Run and Manage Docker Containers), Elastic Beanstalk (Run and Manage Web Apps), Lambda (Run Code in Response to Events).
- Storage & Content Delivery:** S3 (Scalable Storage in the Cloud), CloudFront (Global Content Delivery Network), Elastic File System (Fully Managed File System for EC2), Glacier (Archive Storage in the Cloud), Import/Export Snowball (Large Scale Data Transport), Storage Gateway (Integrates On-Premises IT Environments with Cloud Storage).
- Database:** RDS (Managed Relational Database Service), DynamoDB (Predictable and Scalable NoSQL Data Store), ElastiCache (In-Memory Cache), Redshift (Managed Petabyte-Scale Data Warehouse Service).
- Networking:** VPC (Isolated Cloud Resources), Direct Connect (Dedicated Network Connection to AWS), Route 53 (Scalable DNS and Domain Name Registration).
- Developer Tools:** CodeCommit (Store Code in Private Git Repositories), CodeDeploy (Automate Code Deployments), CodePipeline (Release Software using Continuous Delivery).
- Management Tools:** CloudWatch (Monitor Resources and Applications), CloudFormation (Create and Manage Resources with Templates), CloudTrail (Track User Activity and API Usage), Config (Track Resource Inventory and Changes), OpsWorks (Automate Operations with Chef), Service Catalog (Create and Use Standardized Products), Trusted Advisor (Optimize Performance and Security).
- Security & Identity:** Identity & Access Management (Manage User Access and Encryption Keys), Directory Service (Host and Manage Active Directory), Inspector (Analyze Application Security), WAF (Filter Malicious Web Traffic).
- Analytics:** EMR (Managed Hadoop Framework), Data Pipeline (Orchestration for Data-Driven Workflows), Elasticsearch Service (Run and Scale Elasticsearch Clusters), Kinesis (Work with Real-time Streaming data), Machine Learning (Build Smart Applications Quickly and Easily).
- Internet of Things:** AWS IoT (Connect Devices to the cloud).
- Mobile Services:** Mobile Hub (Build, Test, and Monitor Mobile apps), Cognito (User Identity and App Data Synchronization), Device Farm (Test Android, Fire OS, and iOS apps on real devices in the Cloud), Mobile Analytics (Collect, View and Export App Analytics), SNS (Push Notification Service).
- Application Services:** API Gateway (Build, Deploy and Manage APIs), AppStream (Low Latency Application Streaming), CloudSearch (Managed Search Service), Elastic Transcoder (Easy-to-use Scalable Media Transcoding), SES (Email Sending Service), SQS (Message Queue Service), SWF (Workflow Service for Coordinating Application Components).
- Enterprise Applications:** WorkSpaces (Desktops in the Cloud), WorkDocs (Secure Enterprise Storage and Sharing Service), WorkMail (Secure Email and Calendaring Service).

On the right side of the console, there is a 'Resource Groups' section with a 'Create a Group' button and an 'Additional Resources' section with links for 'Getting Started', 'AWS Console', 'AWS Marketplace', and 'AWS re:Invent'. At the bottom right, there is a 'Service Health' section with a green checkmark and the text 'All services operational'.

Рис. 1.11. Консоль управління AWS

Наступним кроком є завантаження власного образу операційної системи або вибір одного із запропонованих безкоштовних образів – наприклад, Amazon Linux AMI (AMI – Amazon Machine Image) 2015.09.1, що включає набір інструментів AWS, інтерпретатори мов Python, Perl, Ruby, Java, СУБД MySQL, PostgreSQL та ін.

AWS забезпечує захищений доступ до операційної системи з використанням криптографії з відкритим ключем (рис. 1.12) через SSH (рис. 1.13).

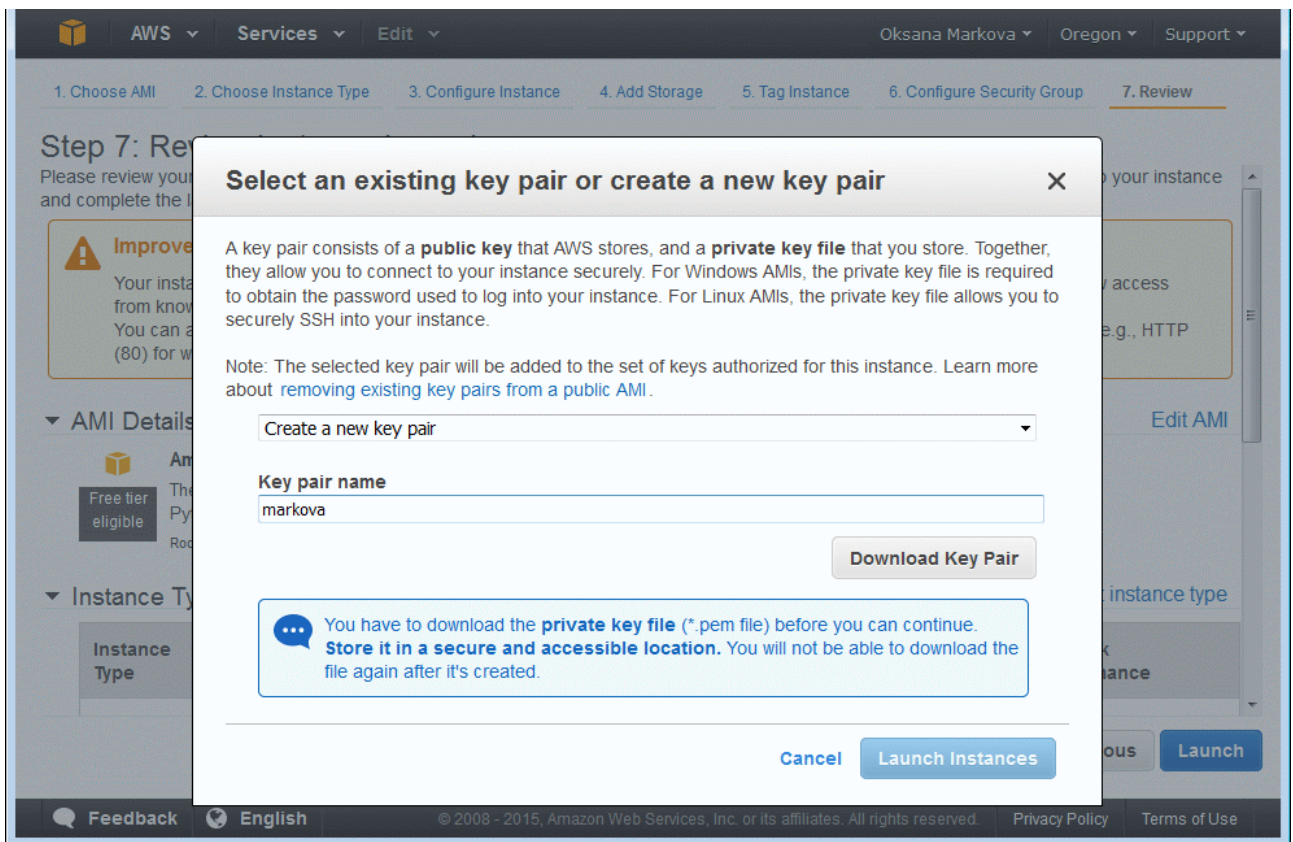


Рис. 1.12. Генерування пари ключів для доступу до встановленого образу операційної системи

Як зауважують Х. Раджай та Е. А. Альдахіль, курс «Операційні системи» є одним з найбільш вигідних курсів для майбутніх фахівців у галузі комп'ютерних технологій, оскільки більшість постачальників хмарних послуг пропонують різні образи операційних систем. Студенти можуть виконувати різноманітні вправ і завдання з програмування на наявних образах операційних систем. Крім того, вони можуть створювати свої власні операційні системи у

вигляді власного образу віртуальної машини у хмарі. Завдяки використанню технологій віртуалізації, ніякої шкоди не буде заподіяно, якщо студентська версія буде втрачена (на відміну від шкоди для реальної машини, що трапляється в результаті надання студентам засобів системного адміністрування) [64, с. 10]. Таким чином, застосування віртуальних машин в курсі «Операційні системи» створює умови для набуття студентами професійних компетентностей на високому рівні.

```

ec2-user@ip-172-31-27-37:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Mon Nov 16 14:41:09 2015 from 85.159.2.142

  _ | _ | _ )
  _ | ( _ - /   Amazon Linux AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/
[ec2-user@ip-172-31-27-37 ~]$ who am i
ec2-user pts/0          2015-11-16 14:42 (85.159.2.142)
[ec2-user@ip-172-31-27-37 ~]$ ls /
bin      dev      lib      lost+found  opt      run      srv      usr
boot    etc      lib64    media      proc     sbin     sys      var
cgroup  home    local    mnt        root     selinux  tmp
[ec2-user@ip-172-31-27-37 ~]$ uname
Linux
[ec2-user@ip-172-31-27-37 ~]$ uname -a
Linux ip-172-31-27-37 4.1.10-17.31.amzn1.x86_64 #1 SMP Sat Oct 24 01:31:37 UTC 2
015 x86_64 x86_64 x86_64 GNU/Linux
[ec2-user@ip-172-31-27-37 ~]$

```

Рис. 1.13. Доступ до налаштованого образу операційної системи через захищене з'єднання

Студенти можуть вивчати «поведінку» різних алгоритмів планування часу в операційних системах з використанням будь-якої мови програмування, віртуальної пам'яті, управління пристроями тощо. За традиційного підходу це можливо лише за допомогою моделювання. Цікавий приклад такої віртуальної машини як складової системи «Агапа» розроблений А. М. Стрюком.

Розроблений ним модуль для проведення віртуальних лабораторних робіт з системного програмування надає можливість продемонструвати покрокове

виконання програм центральним процесором. Інтерфейс модуля (рис. 1.14) імітує роботу програми-налагоджувача. В окремих вікнах робочої області відображаються: вихідний текст програми у вигляді шістнадцяткових кодів та мнемонічних команд асемблеру; вміст поточного сегменту оперативної пам'яті у шістнадцяткових кодах та символах ASCII; вміст основних регістрів процесору; стан регістру прапорів; вміст програмного стеку.

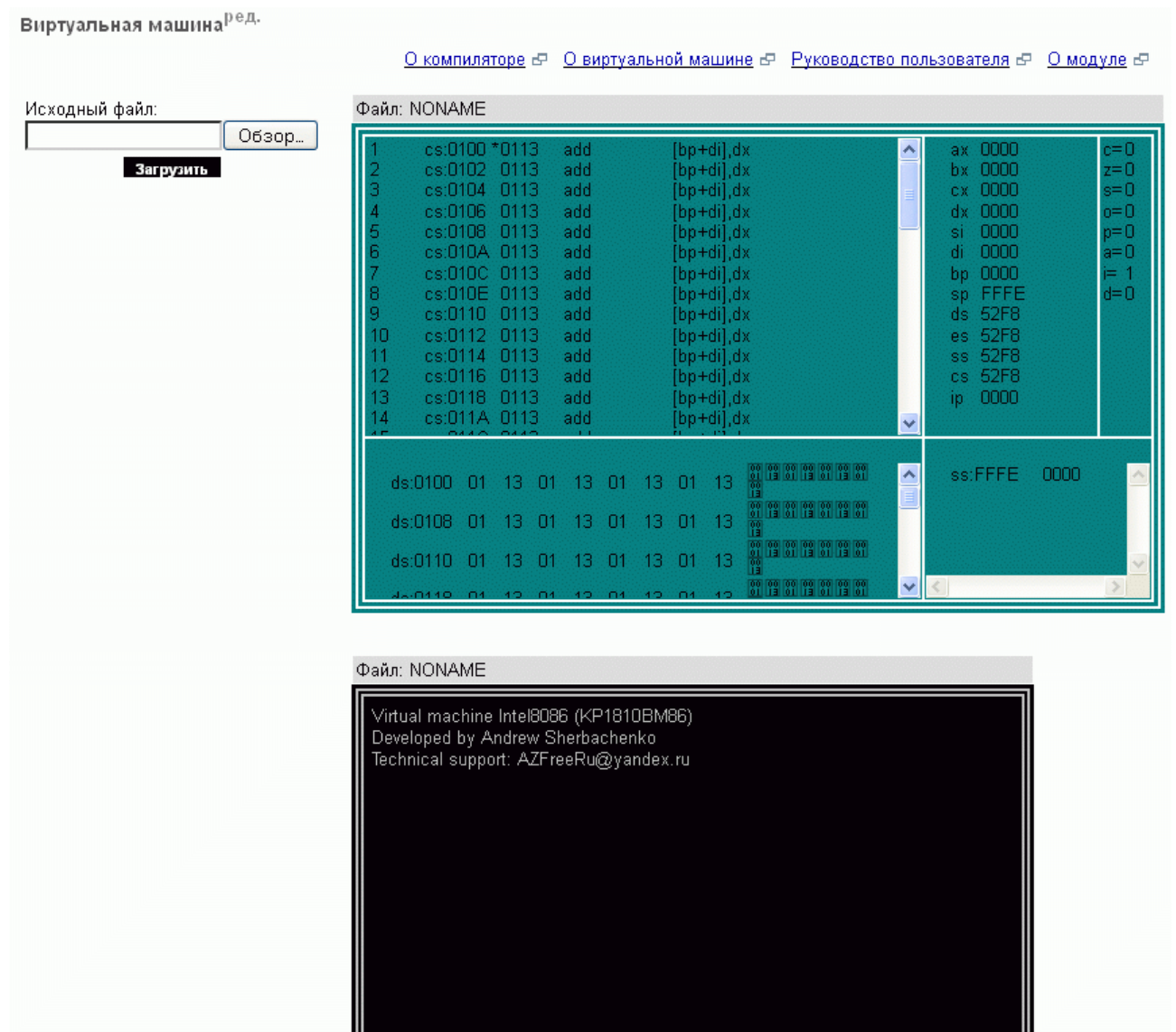


Рис. 1.14. Модуль проведення віртуальних лабораторних робіт з системного програмування

Можливість завантажувати у модуль вихідні тексти програм суттєво розширює можливості використання модуля у проведенні віртуальних лабораторних робіт з дисципліни «Системне програмування». Модуль

використовується викладачем під час роботи зі студентами для наочної демонстрації роботи прикладів, що містять фрагменти програм або алгоритмічних структур [228, с. 143-144].

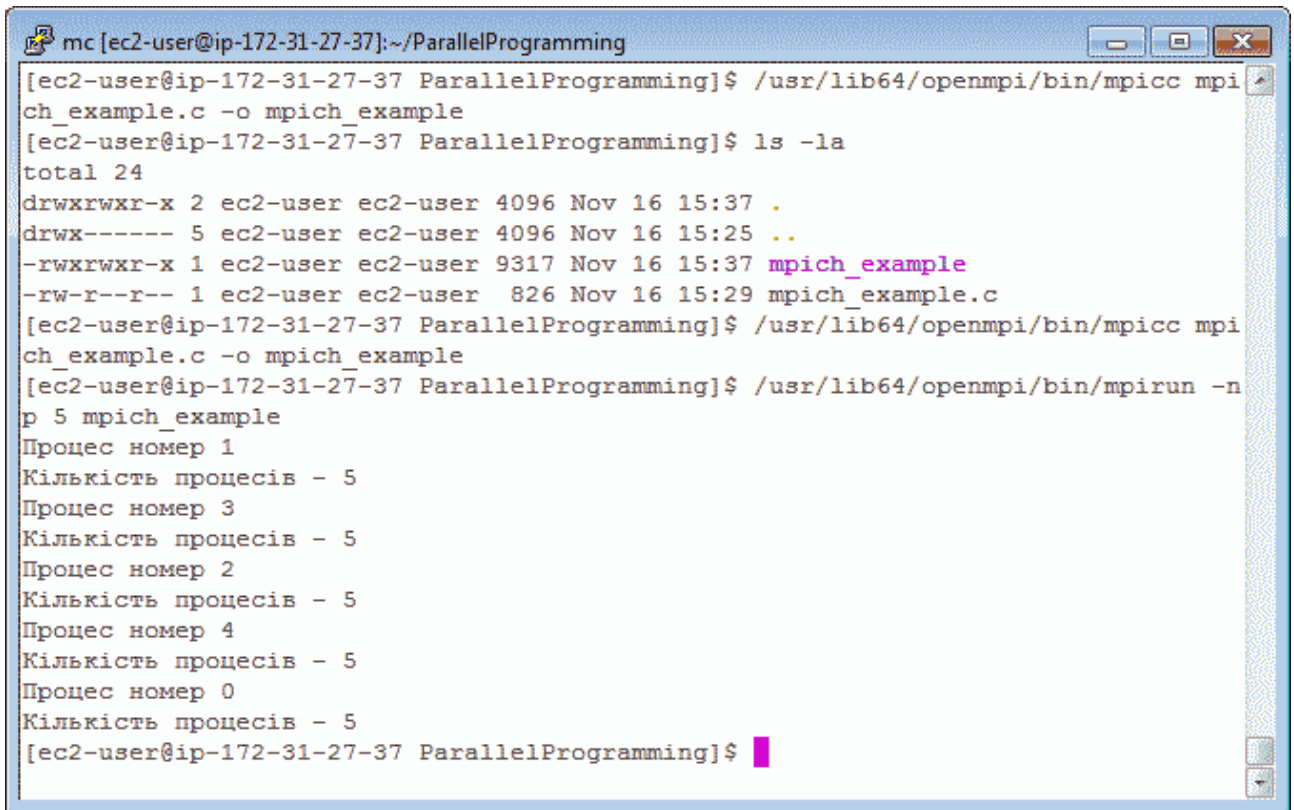
Однією з явних переваг використання хмарних технологій у підготовці майбутніх ІТ-фахівців є можливість використання сучасних технологій паралельного програмування, яке виступає фундаментом для високопродуктивних обчислень, які, у свою чергу, є основою хмарних технологій. Використання у курсі «Операційні системи» віртуальних машин, розміщених у хмарі, надає можливість демонстрації як платформозалежних, так й мобільних технологій паралельного програмування.

Кластерні, ґрид- та інші системи високопродуктивних обчислень традиційно використовуються для розв'язання задач моделювання, які є одним із наріжних каменів інформатики у цілому та математичної інформатики зокрема.

Для того, щоб навчити студентів основних технологій паралельних обчислень та парадигм паралельного програмування, доцільним є використання засобів хмарних технологій для виконання лабораторних робіт, на яких студенти повинні набути компетентності у таких питаннях, як масштабованість та продуктивність обчислювальних систем, обмеження на паралелізм, накладні витрати на синхронізацію, розподіл та балансування навантаження обчислювальних модулів тощо. Традиційно набуття компетентностей з паралельного програмування відбуваються на лабораторних роботах, що виконуються на локальному кластері або корпоративній хмарі. Використовуючи високопродуктивні розподілені хмарні сервери, що надають Amazon та інші провідні постачальники хмарних послуг, можна поліпшити розуміння студентами питань, пов'язаних із розподіленими хмарними системами.

Однією із ефективних технологій паралельного програмування є стандарт Message Passing Interface (MPI) та його реалізація у відповідних бібліотеках програмування, що надають можливість створення багатопоточних програм,

використання спільної пам'яті та ін. через механізм передавання повідомлень. На рис. 1.15 показано результати роботи програми-прикладу використання MPI у операційній системі Amazon Linux AMI із встановленою бібліотекою OpenMPI, що об'єднала технології та ресурси багатьох інших проектів (FT-MPI, LA-MPI, LAM/MPI й PASCX-MPI) та використовується на більшості суперкомп'ютерів, що входять до TOP500.



```

mc [ec2-user@ip-172-31-27-37]:~/ParallelProgramming
[ec2-user@ip-172-31-27-37 ParallelProgramming]$ /usr/lib64/openmpi/bin/mpicc mpi
ch_example.c -o mpich_example
[ec2-user@ip-172-31-27-37 ParallelProgramming]$ ls -la
total 24
drwxrwxr-x 2 ec2-user ec2-user 4096 Nov 16 15:37 .
drwx----- 5 ec2-user ec2-user 4096 Nov 16 15:25 ..
-rwxrwxr-x 1 ec2-user ec2-user 9317 Nov 16 15:37 mpich_example
-rw-r--r-- 1 ec2-user ec2-user 826 Nov 16 15:29 mpich_example.c
[ec2-user@ip-172-31-27-37 ParallelProgramming]$ /usr/lib64/openmpi/bin/mpicc mpi
ch_example.c -o mpich_example
[ec2-user@ip-172-31-27-37 ParallelProgramming]$ /usr/lib64/openmpi/bin/mpirun -n
p 5 mpich_example
Процес номер 1
Кількість процесів - 5
Процес номер 3
Кількість процесів - 5
Процес номер 2
Кількість процесів - 5
Процес номер 4
Кількість процесів - 5
Процес номер 0
Кількість процесів - 5
[ec2-user@ip-172-31-27-37 ParallelProgramming]$

```

Рис. 1.15. Приклад роботи паралельної програми у хмарному середовищі

Таким чином, застосування хмарних технологій у навчанні паралельного програмування із використанням бібліотеки OpenMPI надає студентам можливість випробувати, налагоджувати і розгортати власні програми на масштабованих розподілених системах для того, щоб виявити приховані помилки та інші проблеми, що виникають у процесі паралельного програмування.

Незважаючи на значний успішний досвід використання різних засобів хмарних технологій у підготовці фахівців з інформаційних технологій, залишається відкритим питання методичного обґрунтування їх використання як

засобу навчання основ математичної інформатики студентів технічних університетів.

Висновки до розділу 1

У процесі вивчення проблеми використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів були отримані такі основні результати та висновки:

1. Оскільки ІКТ і теоретичні основи інформатики бурхливо розвиваються, швидко витісняючи застарілі технології й окремі теоретичні положення, то для забезпечення випереджальної ІКТ-підготовки студентів технічних університетів необхідною є фундаменталізація їхнього навчання шляхом виділення фундаментального та технологічного складників змісту навчання, математизації фундаментального та стабілізації технологічного складників на основі перспективних напрямів інноваційного розвитку науки та технологій.

2. Провідним напрямом фундаменталізації підготовки майбутніх фахівців у галузі ІКТ є впровадження у загальнопрофесійну підготовку моделей та методів математичної інформатики – фундаменталізованої навчальної дисципліни, у якій стабілізація технологічної складової виконана на основі виділення класів програмних засобів її навчання, насамперед – систем комп'ютерної математики. Показано ключову роль математичної інформатики в зменшенні розриву між високою швидкістю змін у галузі ІКТ та професійною підготовкою фахівців з інформаційних технологій за рахунок застосування технології комп'ютерного моделювання та відповідних програмних засобів навчання, насамперед систем комп'ютерної математики.

3. Зміст навчання основ математичної інформатики студентів технічних університетів складають змістові модулі «Теорія алгоритмів», «Методи обчислень», «Теорія множин», «Теорія графів», «Комп'ютерна логіка», «Комп'ютерна арифметика», «Схеми шифрування» навчальних дисциплін, у яких формується компетентність з математичної інформатики майбутніх фахівців з інформаційних технологій – особистісно-професійне утворення, яке

включає систему знань, умінь, навичок, досвід навчально-дослідницької діяльності з математичної інформатики та позитивне ціннісне ставлення до неї й виявляється в готовності та здатності до модифікації наявних і розробки нових інформаційних технологій на основі моделей та методів математичної інформатики.

4. Хмарні технології (хмарні ІКТ) як різновид ІКТ можна визначити як сукупність методів, засобів і прийомів, використовуваних для збирання, систематизації, зберігання та опрацювання на віддалених серверах, передавання через мережу та подання через клієнтську програму всеможливих повідомлень і даних. Витоки хмарних технологій навчання містяться у застосуванні концепції комп'ютерної послуги до навчального процесу, зокрема, надання місця для зберігання електронних освітніх ресурсів та мобільного доступу до них. З'ясовано, що розвиток концепції комп'ютерних послуг за останні 50 років відзначався неперервністю, взаємозумовленістю та інноваційністю. Так, термінальні послуги 1960-х викликали до життя мережні операційні системи 1970-х, у яких сформувалась мережна культура 1980-х, що набула поширення в гіпертекстовому Інтернеті 1990-х, який перетворився на джерело надання послуг у 2000-х та компонент соціального життя – у 2010-х. Водночас розвиток засобів хмарних технологій є діалектичним: у 2010-х, як і в 1980-х рр., нові термінальні комп'ютерні пристрої (смартфони, фаблети тощо) все більше набувають характеристик персональних комп'ютерів із мережним доступом, що можуть об'єднуватись у власні мережі за p2p-технологіями (Bluetooth, WiFi Direct тощо).

5. Ураховуючи, що хмарні технології є підмножиною інформаційно-комунікаційних технологій, а ІКТ навчання є підмножиною технологій навчання, під хмарними технологіями навчання в дослідженні розуміємо такі ІКТ навчання, що передбачають використання мережних ІКТ із централізованим мережним зберіганням та опрацюванням даних (виконання програм), коли користувач виступає клієнтом (користувачем послуг), а „хмара” – сервером (постачальником послуг).

6. За результатами вивчення досвіду використання хмарних технологій у підготовці IT-фахівців визначено доцільність застосування в навчанні інформатичних дисциплін таких моделей надання хмарних послуг: «програмне забезпечення як послуга», «платформа як послуга» та «інфраструктура як послуга» на основі інформатичної технології віртуальних машин та педагогічної технології дистанційного навчання. Використання хмарних технологій у підготовці IT-фахівців має низку переваг:

– для студентів: повсюдна доступність необхідних електронних освітніх ресурсів; мобільність програм та даних; відсутність суттєвих програмно-апаратних обмежень на використовувані ресурси; опанування хмарних технологій як провідних для IT-галузі; відсутність необхідності адміністрування програмного забезпечення для досягнення найвищої продуктивності при використанні систем програмування та ін.; можливість проведення неруйнівних експериментів у віртуалізованому програмно-апаратному середовищі;

– для викладачів та співробітників: можливість використання гнучких хмаро зорієнтованих ресурсів (зокрема з метою розробки завдань різного рівня складності та ресурсоспоживання); можливість уніфікації програмного забезпечення у Web-орієнтованих операційних системах; зниження витрат на адміністрування та утримання IT-інфраструктури; широкі можливості моделювання комп'ютерних систем і мереж; можливість простого зберігання та багаторазового використання віртуальних лабораторій;

– для адміністраторів комп'ютерних систем та мереж: зниження витрат на ліцензування програмного забезпечення; зняття обмежень на використовуване апаратне та програмне забезпечення завдяки технології віртуалізації; можливість обслуговування потенційно необмеженої кількості студентів; спрощення та уніфікація технічного обслуговування в хмарі.

Основні результати першого розділу опубліковано у роботах [94; 152; 153; 163; 167; 168; 242; 243; 244; 245; 249].

РОЗДІЛ 2

МЕТОДИЧНІ ЗАСАДИ ВИКОРИСТАННЯ ХМАРНИХ ТЕХНОЛОГІЙ ЯК ЗАСОБУ НАВЧАННЯ ОСНОВ МАТЕМАТИЧНОЇ ІНФОРМАТИКИ СТУДЕНТІВ ТЕХНІЧНИХ УНІВЕРСИТЕТІВ

2.1 Цілі та зміст навчання основ математичної інформатики студентів технічних університетів

Теоретичне обґрунтування методичних засад процесу використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів є необхідною основою для подальшої розробки відповідної методики використання. Тому актуальним є аналіз складників методичної системи, виявлення найбільш слабких місць і проблем, що здатні помітно погіршити її якості і без подолання яких неможливий її подальший розвиток.

У традиційній п'ятикомпонентній моделі, запропонованій А. М. Пишкало [205], використовується системний підхід стосовно компонентів процесу навчання (всі компоненти утворюють єдине ціле із визначеними внутрішніми зв'язками). Згідно з цією моделлю, методична система навчання – це сукупність ієрархічно пов'язаних компонентів: цілей навчання, змісту, методів, засобів і форм організації навчання (рис. 2.1). Функціонування системи підпорядковано закономірностям, що пов'язані з внутрішньою будовою самої системи, коли зміна однієї чи декількох її компонентів призведе до зміни всієї системи.

До методичних засад явно чи неявно включаються принципи навчання. У моделі А. М. Пишкало, на відміну від моделі Ч. Купісевича, вони є неявними: дослідник вказує, що співвіднесення принципів із цілями, змістом, засобами або процесом навчання призводить до втрати ними ознак загальних норм дидактичної діяльності [151, с. 147-152]. У зв'язку з цим принципи навчання при побудові методичних систем ураховуються (табл. 2.1), але до відповідних моделей не включаються.

В. В. Лаптев, Н. І. Рижова та М. В. Швецький об'єднують методи, форми

організації, засоби та зміст навчання у методичне забезпечення, виводячи цілі навчання за межі методичної системи та доповнюючи її очікуваними результатами навчання і трьома технологіями: відбору методів, форм та засобів навчання; відбору змісту навчання; встановлення зв'язків між елементами системи [154, с. 109-110].

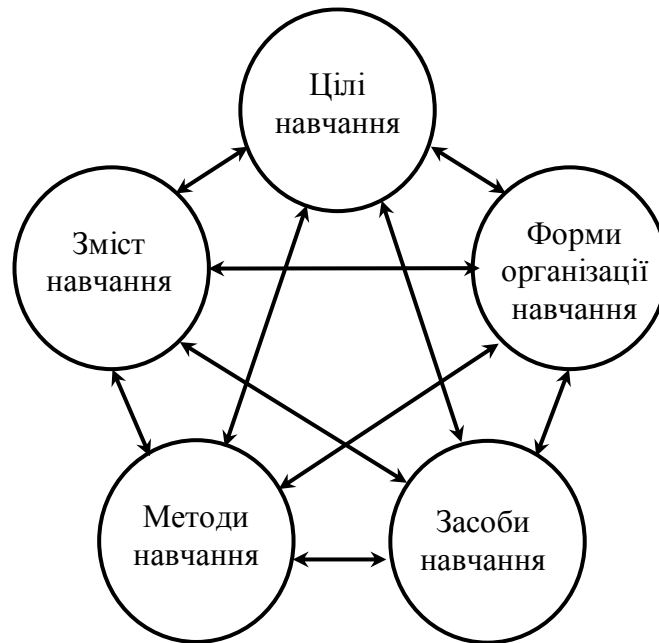


Рис. 2.1. Структура методичної системи навчання (за А. М. Пишкало)

Таблиця 2.1

Співвідношення основних компонентів навчального процесу та принципів навчання (за В. А. Байдаком [82, с. 31])

Основні компоненти навчального процесу	Принципи навчання
Задачі навчання	Принцип спрямованості навчання на рішення у взаємозв'язку задач освіти, виховання та загального розвитку тих, хто навчається
Зміст навчання	Принципи: науковості навчання; зв'язку навчання з життям, практикою; систематичності та послідовності навчання; доступності
Методи навчання та відповідні їм засоби	Принципи: наочності навчання; свідомості та активності тих, хто навчається, за керівної ролі викладача; поєднання різних методів, а також засобів навчання у залежності від задач та змісту навчання
Форми організації навчання	Принцип поєднання різноманітних форм навчання в залежності від задач, змісту та методів навчання
Умови навчання	Принцип створення необхідних умов для навчання
Результати навчання	Принцип міцності, усвідомленості та дієвості результатів навчання, виховання та розвитку

Розглядаючи сукупність таких компонентів, як методи, форми організації

та засоби навчання, услід за Л. О. Черних вважаємо, що вони утворюють певну підсистему єдиної системи, що називають *технологією навчання* [255]. Схематичне зображення структури системи навчання з виділеною пунктиром технологічною підсистемою подано на рис. 2.2.

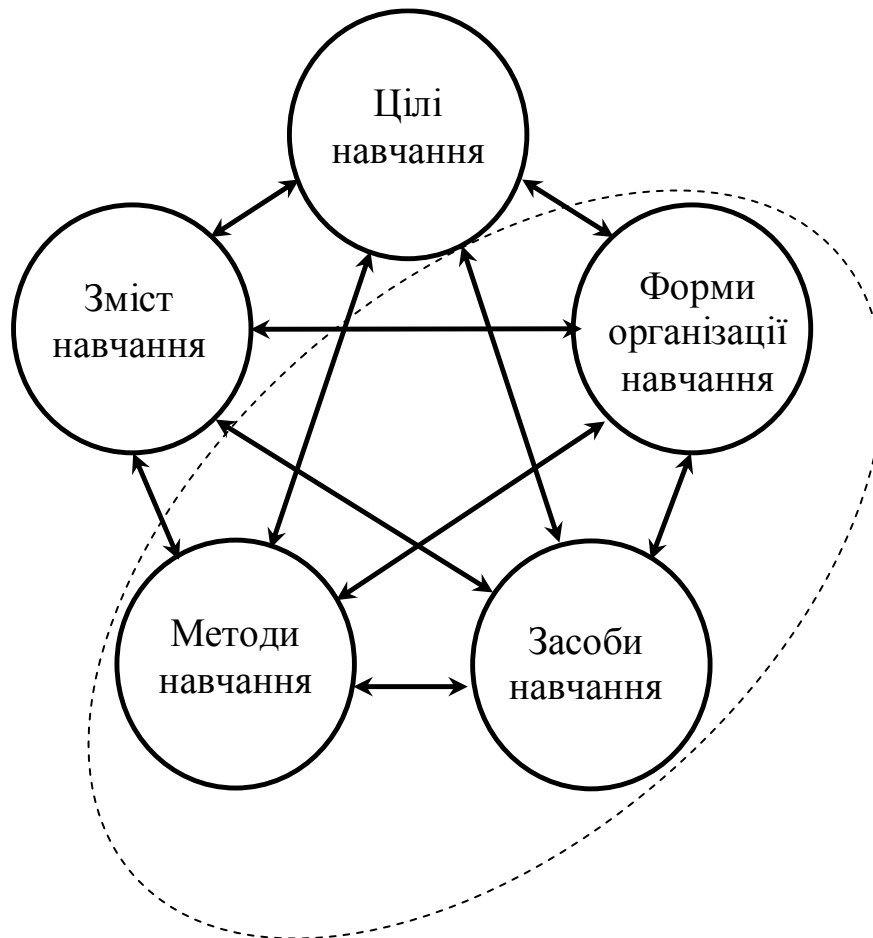


Рис. 2.2. Структура методичної системи навчання з виділеною підсистемою «технологія навчання» (за Л. О. Черних)

Виокремлення технології навчання зумовлено суттєво більш тісними зв'язками між її компонентами: адже «підсумком теоретичного узагальнення педагогічного та методичного матеріалу» [205, с. 42] була структура методичної системи, у якій цілі та зміст навчання впливали на технологічні складові, як це показано у [82, с. 25]. Ураховуючи, що саме технологія навчання має найбільший вплив на відповідну методику використання хмарних технологій, надалі особливу увагу буде приділено засобам хмарних технологій навчання основ математичної інформатики студентів технічних університетів.

«Можна говорити про те, що поява принципово нових засобів навчання, що якісно змінюють можливості передавання інформації і розширюють можливості організації навчального процесу, приводить до перегляду змісту, форм і методів навчання і може опосередковано позначитися на цілях навчання» [200, с. 7]. Це зауваження майже на 10 років випередило появу комп'ютерів у масовій школі, але з позицій сьогодення можна стверджувати, що в ньому сконцентровані всі основні ідеї створення й обґрунтування методичних засад використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів: використання ІКТ у цілому та хмарних технологій зокрема як засобу навчання в значній мірі обумовлює цілі, зміст, методи й форми організації навчання в сучасному технічному університеті.

Виділення засобів хмарних технологій навчання основ математичної інформатики вимагає побудови технології навчання, зумовлюючи вибір відповідних хмаро зорієнтованих форм організації та методів навчання. З іншого боку, теорія, методи та засоби хмарних технологій суттєво впливають на основний зміст навчання та його цілі. Таким чином, теорія, методи та засоби хмарних технологій є основою побудови методичної системи навчання основ математичної інформатики студентів технічних університетів, як це показано на рис. 2.3.

При обґрунтуванні методичних засад процесу використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів було необхідно:

- урахувати професійну спрямованість підготовки майбутніх фахівців з інформаційних технологій шляхом відбору змісту навчання та професійно орієнтованих засобів хмарних технологій;

- забезпечити фундаментальність навчання шляхом системного опанування та застосування моделей і методів математичної інформатики;

- спрогнозувати результати педагогічного впливу, передбачаючи, які компетентності мають бути сформовані у процесі навчання основ математичної

інформатики та як будуть використовуватись засоби хмарних технологій у подальшій професійній підготовці.

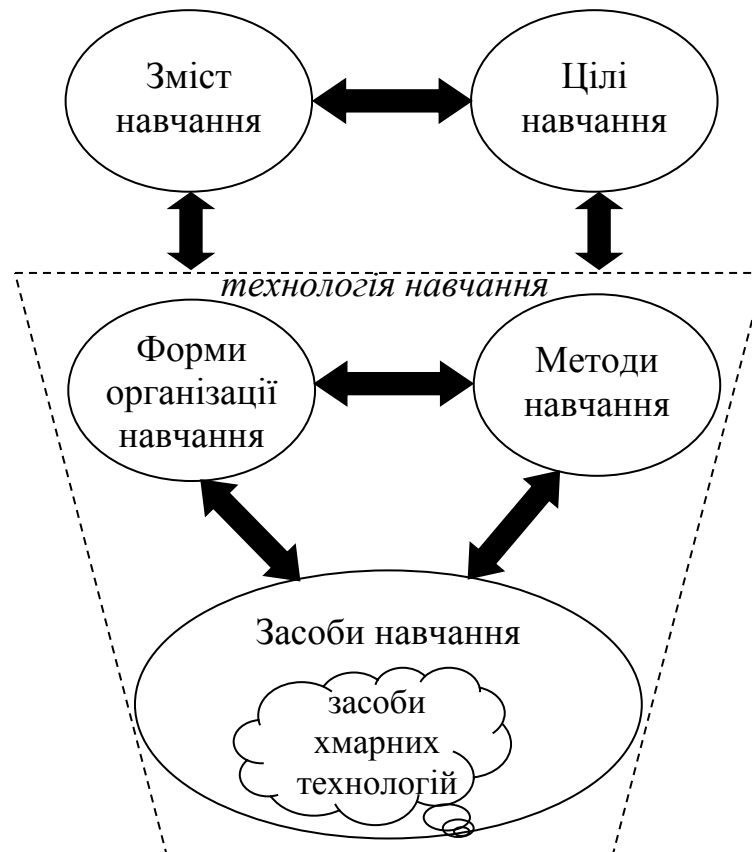


Рис. 2.3. Структура методичної системи навчання основ математичної інформатики студентів технічних університетів

Отже, виходячи з визначеної структури, було виділено цільовий, змістовий та технологічний компоненти методичних засад процесу використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

Мета (ціль) навчання – ідеалізоване передбачення кінцевих результатів навчання; те, до чого прагнуть учасники навчального процесу – студенти і викладачі. Виділяють три основні групи взаємопов'язаних цілей: 1) освітня – формування у студентів наукових знань, загальнонавчальних і спеціальних умінь та навичок; 2) розвивальна – розвиток мови, мислення, пам'яті, здібностей, рухової та сенсорної систем тощо; 3) виховна – формування світогляду, моралі, естетичної культури тощо.

Загальні цілі навчання інформатики визначаються специфікою її внеску в

розв'язання основних задач загальної освіти людини: формування основ наукового світогляду, що полягає у розумінні сучасної картини світу, зміні змісту та характеру діяльності людини відповідно до розвитку нових інформаційних технологій; розвиток мислення тих, хто навчається, зокрема теоретичного, творчого та мислення нового типу (операційного) у процесі вивчення інформатики з використанням комп'ютерної техніки; підготовка тих, хто навчається, до практичної діяльності та продовження освіти, оскільки саме вивчення інформатики впливає на формування комп'ютерної грамотності та інформаційної культури, що уможливорює існування людини у сучасному інформаційному суспільстві [154, с. 268].

У процесі здобуття загальної середньої освіти елементи математичної інформатики реалізуються насамперед в інформаційно-комунікаційному компоненті відповідного стандарту через формування в учнів навичок і вмінь проводити основні операції з інформаційними об'єктами, зокрема:

- створювати інформаційні об'єкти, фіксувати, записувати, спостерігати за ними і вимірювати їх, зокрема, в рамках реалізації індивідуальних і колективних проектів;

- висувати і перевіряти нескладні гіпотези навчально-пізнавального характеру, створювати, вивчати та використовувати інформаційні об'єкти;

- вивчати, аналізувати інформаційні процеси, що відбуваються у живій природі, суспільстві та техніці;

- одержувати уявлення про основи управління, прийняття рішень, основні принципи роботи засобів інформаційних технологій;

- виявляти та аналізувати інформаційні процеси в технічних, біологічних і соціальних системах;

- будувати і використовувати інформаційні моделі, а також засоби опису та моделювання явищ і процесів [202].

Загальною метою навчання інформатики студентів технічних університетів є:

- 1) формування уявлень про ідеї та методи дискретної математики як

форми опису та подання об'єктів навколишнього світу;

2) формування та розвиток операційного стилю мислення майбутніх ІТ-фахівців (вміння формалізувати задачу; виділити в ній логічно самостійні частини; визначити взаємозв'язок цих частин; спроектувати рішення за допомогою спадної чи висхідної технологій; верифікувати результат);

3) формування уявлення про методологію інформатики (обчислювальний експеримент) як основний метод пізнання в галузі інформатики.

Виходячи з даного трактування, *цілі навчання* спецкурсу «Основи математичної інформатики» визначимо через формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій.

Цілі навчання спецкурсу «Основи математичної інформатики» конкретизовані у таких *завданнях*:

– ознайомлення зі структурами даних і алгоритмами, які є фундаментом сучасної методології розробки програм;

– вивчення методів розв'язування інженерних та наукових задач з використанням чисельних методів;

– ознайомлення з основними принципами кодування і модуляції сигналів у процесі передавання даних, опрацювання сигналів, збільшення перешкодозахищеності при передаванні даних по каналах зв'язку;

– формування умінь описувати основні методи реєстрації сигналів, декодування і виявлення помилок за допомогою різних коригуючих кодів;

– вивчення основ алгоритмічних аспектів теорії чисел та їх застосування в сучасній криптографії;

– опанування хмарних технологій для практичної реалізації основних методів математичної інформатики.

У робочій навчальній програмі (додаток Б) завдання навчання спецкурсу «Основи математичної інформатики» конкретизуються у вимогах до знань та вмінь, за якими визначається *зміст навчання* – «педагогічно обґрунтовані, логічно впорядковані та текстуально зафіксовані в навчальній програмі наукові відомості про матеріал, що підлягає вивченню, які подані в згорнутому вигляді

й визначають зміст навчальної діяльності педагогів і пізнавальної діяльності тих, хто навчається з метою оволодіння усіма компонентами змісту професійної освіти відповідного рівня та профілю» [154, с. 155].

Модель змісту навчання спецкурсу «Основи математичної інформатики» складається із двох блоків:

1) основний блок змісту навчання містить опис досліджуваного математичного об'єкта, що розкриває його будову та властивості, та діяльнiсну частину – алгоритми та їх реалізацію;

2) допоміжний блок змісту навчання містить питання історії математичної інформатики та хмарних технологій, а також допоміжні розділи математичної інформатики.

Зміст навчання включає в себе:

а) навчальні елементи із теоретичного (базисного) матеріалу, який є сукупністю основних понять конкретної предметної галузі знань та зв'язків між ними;

б) задачний матеріал, який визначається у відповідності з теоретичним матеріалом та призначається для вироблення у тих, хто навчається, конкретних умінь та навичок для відповідної галузі знань;

в) міжпредметні та внутрішньо предметні взаємозв'язки між навчальними елементами [154, с. 156].

Ступінь засвоєння змісту навчання можна оцінити за результатами навчання, що, згідно Закону України «Про вищу освіту» є «сукупність знань, умінь, навичок, інших компетентностей, набутих особою у процесі навчання за певною освітньо-професійною, освітньо-науковою програмою, які можна ідентифікувати, кількісно оцінити та виміряти» [201, стаття 1].

Визначення змісту навчання за спецкурсом «Основи математичної інформатики» необхідно здійснювати з урахуванням принципів, спільних для будь-яких навчальних дисциплін, так і властивих насамперед для інформатичних:

1. *Принцип єдності освітньої, розвивальної та виховної функцій навчання*

передбачає, що навчання спрямоване на досягнення цілей різнобічного розвитку особистості, на формування знань, умінь, навичок, її моральних і естетичних якостей, що є основою вибору життєвих ідеалів і соціальної поведінки [148, с. 713]. Реалізація цього принципу визначається метою навчання спецкурсу «Основи математичної інформатики», що передбачає формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій.

2. *Принцип науковості змісту і методів навчання* основ математичної інформатики відображає їх зв'язок із математичною інформатикою як наукою. Принцип науковості передбачає розкриття наукових явищ і фактів у взаємозв'язках і стосунках, застосування в організації діяльності студентів методів проблемного навчання, дослідницьких методів.

3. *Принцип систематичності та послідовності* передбачає опору на зміст навчання, моделі та методи тих навчальних дисциплін, вивчення яких передують спецкурсу – «Вища математика», «Теорія електричних та магнітних кіл», «Комп'ютерна логіка», та подальше використання засобів хмарних технологій та методів математичної інформатики у навчанні дисциплін «Алгоритми та методи обчислень», «Дискретна математика», «Комп'ютерна електроніка», «Архітектура комп'ютерів», «Комп'ютерна схемотехніка», «Системне програмування», «Технології проектування комп'ютерних систем», «Захист інформації у комп'ютерних системах», «Інженерія програмного забезпечення» та при виконанні курсових і дипломних робіт.

4. *Принцип міцності знань* означає ґрунтовність засвоєння знань, умінь і навичок, стійке закріплення набутого у пам'яті, вільне відтворення й застосування його на практиці. Міцне засвоєння знань досягається всім ходом навчання. В ході підготовки до ознайомлення з новим матеріалом відповідним чином спрямовуються інтереси, формується позитивне ставлення до навчання. Реалізується через комплекс загально дидактичних методів навчання (пояснювально-ілюстративний, репродуктивний, евристичний та дослідницький), хмарних засобів ІКТ навчання та форм організації навчання

(навчально-дослідницький проект).

5. *Принцип наочності* реалізується через застосування у навчанні основ математичної інформатики хмаро зорієнтованих засобів наочності, насамперед експериментальних (досліди, комп'ютерне моделювання), символічних та графічних (графіки, схеми, формули).

6. *Принцип зв'язку навчання з практикою* реалізується прикладною та професійною спрямованістю навчання хмарних технологій та основ математичної інформатики студентів технічних університетів.

7. *Принцип відповідності навчальним цілям.* Цілі навчання основ математичної інформатики визначаються, виходячи із загальних цілей вищої освіти – формування професійних компетентностей майбутніх фахівців з інформаційних технологій.

8. *Принцип фундаментальності,* реалізація якого передбачає: опанування методологічно важливих та інваріантних знань, що мають довгий термін життя, необхідних для професійної діяльності ІТ-фахівця; тісний зв'язок ІТ-освіти з професійною практичною діяльністю; розвиток творчої і пізнавальної активності та самостійності студентів; модернізацію системи професійної підготовки з урахуванням перспектив розвитку «економіки знань» та інформаційного суспільства; системність засвоєння інформатичних дисциплін на основі глибокого розуміння сучасного стану та існуючих проблем інформатики [214, с. 86].

9. *Принцип відкритості.* Цей принцип передбачає можливість корекції змісту спецкурсу залежно від спеціальності без порушення цілісності фундаментального ядра дисципліни.

10. *Принцип сучасності.* Швидкий розвиток хмарних технологій вимагає регулярного перегляду навчальної програми спецкурсу з метою модернізації застарілих компонентів.

11. *Принцип перспективності.* Цей принцип передбачає формування у студентів готовності до подальшого навчання протягом усього життя, що надасть їм можливість бути здатними до вирішення професійних проблем у

майбутньому.

12. *Принцип доступності та принцип індивідуалізації* навчання передбачають урахування індивідуальних особливостей тих, хто навчається, пов'язане з необхідністю добору викладачем таких способів, методів і засобів навчання, які б сприяли оптимальному засвоєнню ними навчального матеріалу відповідно до певного етапу їх розумового, морально-соціального та фізичного розвитку. Реалізація цього принципу у навчанні вимагає додержання таких правил: у навчанні необхідно переходити від легкого до більш складного, від відомого до нового, невідомого; враховувати різницю у рівнях просування у процесі навчання окремих студентів. Забезпечення цього принципу в спецкурсі досягається як через послідовність подання навчальних матеріалів, так й через застосування системи навчальних впливів, що підтримується відповідними засобами навчання.

13. *Принцип свідомості й активності* реалізується через розкриття та усвідомлення студентами цілей навчання спецкурсу, планування та організацію власної роботи за ним, усвідомлення особистісної значущості результатів навчання [148, с. 713-714].

14. При формуванні змісту важливо визначити місце засобів хмарних технологій. Для цього доцільно включити їх до змісту навчання, реалізуючи виділений Г. О. Михаліним *принцип диференційованої фундаментальності* [177, с. 12], згідно якого фундаментальна підготовка повинна бути диференційованою не тільки метою, а й засобом підготовки.

На думку Т. П. Кобильника, основну увагу в навчанні математичної інформатики слід акцентувати на методологічних аспектах та математичному апараті інформатики, які утворюють основу великої кількості науково-технічних та соціально-економічних інформаційних технологій, що реально використовуються в теоретичних дослідженнях та практичній діяльності. При вивченні математичної інформатики основна увага приділяється питанням, які пов'язані з формальними системами, моделями, алгоритмічними засобами та теорією програмування, кодування та організацією систем [140].

Зміст спецкурсу містить сукупність двох взаємопов'язаних складових: теоретичної та практичної. Теоретична складова спрямована на формування у студентів уявлень про структури даних і алгоритми, які є фундаментом сучасної методології розробки програм; методи розв'язування інженерних та наукових задач з використанням чисельних методів; основні принципи кодування і модуляції сигналів у процесі передавання даних, опрацювання сигналів, збільшення перешкодозахищеності при передаванні даних по каналах зв'язку; основні методи реєстрації сигналів, декодування і виявлення помилок за допомогою різних коригуючих кодів; алгоритмічні аспекти теорії чисел та їх застосування в сучасній криптографії.

Практичний аспект пов'язується з набуттям умінь аналізувати, оцінювати та обирати існуючі алгоритми; використовувати методи та технології розробки та оцінювання алгоритмів; розробляти нові алгоритми, які пов'язані з проектуванням апаратних та програмних компонентів комп'ютерних систем і мереж; використовувати існуючі та розроблювати нові математичні методи для вирішення задач, пов'язаних з проектуванням та використанням комп'ютерних систем і мереж; вибирати методи обчислень, стійкі до похибок; розв'язувати лінійні та нелінійні алгебраїчні рівняння та їх системи; застосовувати методи інтерполяції та апроксимації; здійснювати вибір методу інтегрування диференціального рівняння; ставити та вирішувати оптимізаційні задачі; застосувати теорему Колмогорова для апроксимації довільної функції тришаровою нейронною мережею; будувати кільця за заданим модулем; застосовувати методи завадостійкого кодування для відновлення даних при їх пошкодженні; будувати блочні шифри; будувати лінійні коди та двійкові коди Боуза-Чоудхурі-Хоквінгема; будувати генеруючий та перевірючий поліноми для кодування та декодування циклічних кодів; застосовувати коди Ріда-Соломона для передавання даних у комп'ютерних мережах; застосовувати методи і засоби забезпечення безпеки програм і даних при проектуванні і експлуатації комп'ютерних систем і мереж; враховувати вимоги до систем захисту даних; створювати програмні та апаратні підсистеми криптографічного

захисту даних; використовувати алгоритм RSA та системи цифрового підпису для передавання даних у комп'ютерних мережах; формувати і управляти ключовою інформацією для підсистем аутентифікації; використовувати хмарні технології для практичної реалізації основних методів математичної інформатики.

У загальній структурі спецкурсу «Основи математичної інформатики» обсяг лабораторних занять має співвідноситись із обсягом теоретичних як 1:1.

Зміст спецкурсу складають 4 змістові модулі.

У *першому змістовому модулі* «Теорія алгоритмів» розглядаються базові поняття та методи, що відносяться до аналізу алгоритмів (машина з довільним доступом до пам'яті; аналіз алгоритму сортування методом включення; порівняння функцій), алгоритмічних стратегій (асимптотичний аналіз верхньої та середньої оцінок складності алгоритмів; порівняння найкращих, середніх і найгірших оцінок; O -, o -, ω - та θ -нотації; емпіричні вимірювання ефективності алгоритмів; накладні витрати алгоритмів за часом і пам'яттю; рекурентні співвідношення та аналіз рекурсивних алгоритмів; порівняння алгоритмів; вплив структур даних і особливостей реалізації на ефективність алгоритмів; методи розробки алгоритмів), побудови алгоритмів (значення, класифікація та характеристики сортувань при реалізації алгоритмів; прості сортування, їх переваги та недоліки; складні сортування, їх переваги та недоліки; порівняння простих та складних сортувань).

У *другому змістовому модулі* «Методи обчислень» розглядаються основи комп'ютерного моделювання (поняття про моделі та моделювання; властивості та класифікація моделей; особливості комп'ютерного моделювання; особливості імітаційного статистичного моделювання), задачі лінійної та нелінійної алгебри, методи наближення функцій (інтерполяція; апроксимація), методи розв'язання звичайних диференціальних рівнянь I порядку, основні методи оптимізації (метод випадкового пошуку; метод дихотомії; метод «золотого перетину»; метод Фібоначчі; симплексний пошук), нейронні мережі та задача розпізнавання образів (математична модель нейрона; застосування

теорема Колмогорова для апроксимації довільної функції тришаровою нейронною мережею).

У *третьому змістовому модулі* «Теорія кодування» розглянуто математичні основи теорії кодування, основні поняття теорії завадостійкого кодування, лінійні коди, циклічні коди, коди Боуза-Чоудхурі-Хоквінгема, коди Ріда-Соломона, згорткові коди.

У *четвертому змістовому модулі* «Основи криптографії» розглядаються основні криптографічні системи (симетричні та асиметричні) та їх застосування для розв'язання задач управління криптографічними ключами та електронними цифровими підписами.

Підсумковий контроль знань за спецкурсом – залік за результатами поточного та модульного контролів і захисту індивідуальних навчально-дослідних проектів нейронних мереж. Вибір останніх обумовлений тим, що, по-перше, у їх основу покладено фундаментальний математичний апарат, по-друге, нейромережеве моделювання є одним із найсучасніших напрямів досліджень у галузі математичної інформатики, і по-третє, отримані у процесі моделювання результати можуть бути застосовані у всіх змістових модулях спецкурсу.

2.2 Форми організації освітнього процесу, методи та засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів

Закон України «Про вищу освіту» [201] визначає освітній процес як інтелектуальну, творчу діяльність у сфері вищої освіти і науки, що провадиться у закладі вищої освіти через систему науково-методичних і педагогічних заходів та спрямована на передачу, засвоєння, примноження і використання знань, умінь та інших компетентностей у осіб, які навчаються, а також на формування гармонійно розвиненої особистості.

Стаття 50 Закону [201] розділяє форми організації освітнього процесу та види навчальних занять. Освітній процес у ЗВО здійснюється за такими формами: навчальні заняття; самостійна робота; практична підготовка;

контрольні заходи (заліки, іспити, захист курсових і магістерських робіт та ін.).

Основними видами навчальних занять у ЗВО є: 1) лекція; 2) лабораторне, практичне, семінарське, індивідуальне заняття; 3) консультація.

Також виділяють форми організації науково-дослідної роботи студентів (науково-дослідні гуртки, проблемні групи, об'єднання, школи, студентські наукові товариства) [176].

Методи навчання – способи досягнення навчальної мети, система взаємопов'язаних дій викладача та студентів, які забезпечують засвоєння змісту навчання. Методи навчання визначають як специфічну внутрішню навчальну діяльність, таку як запам'ятовування частини інформації або встановлення зв'язків між окремими частинами інформації та зовнішньою навчальною діяльністю, такими як виділення та анування інформації у зовнішніх поданнях. Коли декілька методів навчання використовуються разом в узгодженому порядку та у цілеспрямований спосіб, вони формують стратегію навчання. Когнітивні навчальні стратегії допомагають ефективно опрацювати інформацію, зберігати її в довгостроковій пам'яті та видобувати її звідти.

Когнітивна навчальна стратегія – це цілеспрямована розумова діяльність, спрямована на підвищення рівня знань та умінь того, хто навчається. Приклади когнітивних навчальних стратегій включають узагальнення, викладення, концептуальне відображення, створення аналогій, удосконалення, під-цілеспрямованість, самодослідження тощо [25, с. 599]. У [61, с. 596] когнітивна навчальна стратегія визначається як послідовність ефективних методів навчання, які використовуються цілеспрямовано та гнучко, все більше автоматизуються, але залишаються свідомо застосовуваними.

Виділяють такі традиційні методи навчання:

– *пояснювально-ілюстративний метод* – метод навчання, пов'язаний із використанням засобів наочності та репродуктивним характером засвоєння знань. Метод також передбачає використання самостійних робіт, задач, вправ, складання графіків, таблиць за наявним зразком тощо;

– *методи проблемного навчання* передбачають створення у навчальному процесі проблемно-пошукових ситуацій (виникають унаслідок взаємодії пізнаючої сторони з пізнавальним об'єктом, завдяки чому виникає пізнавальне протиріччя) та спрямований на розвиток активності, самостійності, творчих здібностей.

За рівнями проблемності методи проблемного навчання поділяють на:

– *проблемний виклад* – полягає в розкритті викладачем шляху дослідження проблеми; показуючи її вирішення від початку до кінця, викладач демонструє еталон проблемного мислення. Метод використовується тоді, коли студенти ще не можуть самостійно розв'язувати проблемні задачі;

– *частково-пошуковий* – полягає у створенні викладачем проблемної ситуації: він сам формулює проблему і залучає студентів до її розв'язання;

– *пошуковий* – викладач створює проблемну ситуацію, формулює проблему, а студенти самостійно її розв'язують;

– *дослідницький* – студенти самі в умовах проблемної ситуації формулюють проблему і самостійно її розв'язують [170, с. 112].

Виходячи із цілей навчання, форми організації та методи навчання основ математичної інформатики повинні бути спрямованими на формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій.

У електронних навчальних середовищах (зокрема, хмаро зорієнтованих) навчальний матеріал часто складається з різних статичних та динамічних словесних (письмовий текст, голосовий запис) та графічних (фотографії, анімації, симуляції) способів подання, які можуть мати різний вплив на навчання.

Так, різні способи подання можуть підкреслювати різні аспекти предметної галузі, які не можуть бути описані за допомогою інших способів подання, або доповнюють один одного таким чином, щоб досягти більш повного подання. Такі комбінації не тільки надають різні навчальні можливості для студентів, а й пред'являють до них підвищені вимоги. Наприклад, студенти

повинні розуміти, що: 1) інформація є закодованою в кожному окремому поданні; 2) кожне її подання пов'язане з предметною галуззю; 3) інформація в одному поданні може бути пов'язана з інформацією в іншому поданні [3, с. 185-187].

Таким чином, студенти не лише повинні навчитися виявляти відповідні компоненти словесних та графічних репрезентації, а й як їх пов'язати один з одним. Якщо репрезентації є динамічними, студенти повинні також навчитися виявляти та співвідносити як просторово, так і тимчасово розділені компоненти. Інтерактивні подання ставлять ще більші вимоги до студентів, оскільки їм потрібно планувати, контролювати та оцінювати їх взаємодію з іншими поданнями.

Одним з підходів до підтримки навчання за допомогою різних статичних та динамічних репрезентацій є проектування електронного середовища. Так, у монографії С. Г. Литвинової виокремлено принципи проектування хмаро зорієнтованого навчального середовища навчального закладу – штучно побудованої системи, яка за допомогою хмарних сервісів забезпечує навчальну мобільність, групову співпрацю суб'єктів навчання (викладачів та студентів) для ефективного, безпечного досягнення дидактичних цілей [158, с. 125]. У цій та інших роботах показано, що проектування електронного навчального середовища сприяє покращенню результатів навчання.

Інший підхід до покращення результатів навчання – це проектування стратегій навчання, що також засноване на теоріях та моделях навчання, що спрямовані на розширення можливостей студентів для ініціації, планування, організації, моніторингу та регулювання власної навчальної діяльності та грамотного розв'язання складних навчальних задач. Це вимагає посилення ролі самостійної роботи студентів на основі проектного підходу, реалізованого в спецкурсі у формі індивідуальних навчально-дослідницьких проектів. Як вважає В. Є. Лук'янченкова [161], слід заохочувати тенденцію послідовної розробки студентами єдиної наукової тематики, що дозволить поглибити і детально дослідити наукову проблему, яку розробляють, накопичити досвід

дослідницької роботи і підвищити свою наукову активність. На думку дослідника, основні цілі науково-дослідної роботи студентів направлені на:

- 1) прищеплення студентам навичок проведення і постановки самостійних наукових досліджень, вироблення творчого підходу до вирішення наукових проблем, що постають перед ними, освоєння методів самостійної роботи з науковою літературою, поглиблення і розвиток знань, їх практичне застосування;
- 2) залучення найбільш здібних студентів до вирішення науково-технічних проблем, що мають безпосередній вихід у народне господарство;
- 3) можливість обґрунтованого вибору студентами наукового напрямку, де б найбільш яскраво могли виявлятися їх творчі здібності;
- 4) виховання вже в стінах ЗВО резерву вчених, дослідників, викладачів, формування потенціалу майбутньої науки;
- 5) прискорення професійного становлення майбутніх фахівців;
- 6) набуття студентами вмінь та навичок роботи в колективі, виховання почуття колективізму й відповідальності;
- 7) сприяння науково-технічному прогресу шляхом участі в розробці актуальних проблем науки, техніки, культури.

Об'єднання різних способів подання даних є основою теорії мультимедійного навчання Р. Е. Майєра (Richard E. Mayer), який виділяє чотири різні види когнітивних процесів: вибір, організація, перетворення та інтеграція даних. Вибрані текстові та графічні дані спочатку опрацьовуються окремо. Далі обрані дані організуються в дві окремі моделі: для словесних даних та графічних. Під час опрацювання даних словесні подання можуть бути перетворені на графічні (наприклад, шляхом побудови розумових образів) і навпаки (наприклад, шляхом внутрішньої вербалізації зображень). Для того, щоб мультимедійне навчання було успішним, обидві моделі повинні бути інтегровані та пов'язані з попередніми знаннями [50].

С. Д. Цвілик [252] вважає, що суттєвою рисою наступності у формах організації освітнього процесу є розвиток і вдосконалення уже визначених форм на кожному етапі навчання зі зростанням практичної спрямованості навчання, внесенням у традиційні форми елементів інноваційних технологій

навчання; встановлення тісного взаємозв'язку предметів різних циклів. Цим створюються значні можливості для реалізації внутрішніх, міжпредметних і міжциклових зв'язків. У студентів формуються знання вищого рівня узагальненості порівняно із знаннями, які набуваються в процесі вивчення окремих дисциплін.

Однією із перспективних способів інтеграції різних методів навчання є мультистратегійне навчання, запропоноване П. Б. Бразділем (Pavel V. Brazdil), яке стосується розробки методів навчання та систем, що інтегрують різні стратегії введення та/або репрезентації у розв'язанні заданої навчальної задачі, яка характеризується трьома компонентами: типом знань, які слід опанувати; вхідними даними, доступними студенту; попередніми (або фоновими) знаннями студента [13, с. 2396].

Завдяки взаємодоповнюваності різних навчальних стратегій, багатостратегічні навчальні підходи мають потенціал для більш широкого кола застосувань, ніж моностратегічні системи. П. Б. Бразділ виділяє п'ять основних підходи до організації мультистратегійного навчання.

I. Поєднання різних стратегій виведення.

За такого підходу передбачається, що застосовувані методи виведення суттєво різняться: індуктивне узагальнення, абдуктивне виведення або дедукція. Перетворення знань за такого підходу називаються трансмутаціями. Абстракції та спеціалізації (конкретизації) змінюють рівень деталізації. Головною проблемою об'єднання різних стратегій у єдине ціле є визначення умов застосування кожної із них.

Даний підхід вимагає побудови інтелектуального агента, включаючи визначення взаємозв'язку між механізмами, що відповідають за набуття знань та їх перетворення з використанням різних стратегій (включаючи індукцію), до яких може звернутися агент, та визначення механізму управління для визначення того, коли який механізм використовується (зокрема, за допомогою методів математичної логіки).

II. Поєднання різних методів навчання у межах однієї когнітивної

навчальної стратегії для однієї та тієї ж мети.

Даний підхід передбачає комбінування моделей для однієї та тієї ж навчальної мети за однаковою навчальною стратегією двома основними способами: 1) для різних наборів даних застосовується один алгоритм навчання; 2) для одного набору даних застосовуються різні алгоритми навчання. Так як механізм виведення та цілі навчання фіксовані, метою підходу є поліпшення певної міри успіху (наприклад, зменшення помилки класифікації).

III. Метод машинного навчання разом із допоміжними обчислювальними процесами.

За даного підходу вимагається, щоб певний метод навчання, який доповнюється одним або декількома допоміжними обчислювальними методами, мав певні переваги над оригінальним методом навчання. Допоміжний обчислювальний метод, як правило, застосовується для конкретної мети (наприклад, подолання певного недоліку методу навчання). Це може бути поєднання символічних та субсимвольних методів, вдосконалення класифікації, заснованої на правилах, з використанням генетичних алгоритмів, застосування кластеризації перед регресією тощо.

IV. Динамічний вибір методу машинного навчання, який використовує багаторівневу інформацію.

Оскільки кількість алгоритмів машинного навчання зростає, виникає проблема у виборі методу для конкретної задачі. Оскільки жоден алгоритм не є універсальним, для управління процесом вибору алгоритму можна використати метадані, які включають насамперед дані стосовно задач, що розглядалися у минулому із застосуванням різних алгоритмів машинного навчання та їх відповідної продуктивності. Крім того, вони включають також деякі характеристики нової задачі.

V. Динамічна багатоступенева обробка, яка використовує багаторівневу інформацію і планування.

Даний підхід можна розглядати як комбінацію двох попередніх. Він

передбачає динамічний добір алгоритмів або методів, але мета полягає не в тому, щоб обрати єдиний метод машинного навчання, а визначити робочий процес – послідовність кроків, на кожному із яких застосовуватиметься певний метод. Останнє досягається використанням онтологій операцій, багаторівневої інформації, що характеризує окремі операції, які у свою чергу використовуються як евристики, що можуть бути використані для управління пошуком.

Мультистратегійне навчання надає можливість застосування кібернетичного підходу до вибору провідної стратегії навчання, а відповідно – провідних форм та методів навчання основ математичної інформатики: лекції, лабораторні заняття, самостійна робота студентів, навчальні конференції, консультації, індивідуальні заняття, навчально-дослідницькі проекти, контрольні заходи.

Таким чином, *основними формами організації освітнього процесу та методами навчання* основ математичної інформатики студентів технічних університетів є лекції, лабораторні заняття, самостійна робота студентів, навчальні конференції, консультації, індивідуальні заняття, навчально-дослідницькі проекти, контрольні заходи, а перспективним засобом розширення можливостей студентів для ініціації, планування, організації, моніторингу та регулювання власної навчальної діяльності та грамотного розв'язання складних навчальних задач є проектування когнітивних навчальних стратегій та їх подальша інтеграція у мультистратегійному навчанні.

Засоби навчання за В. Ю. Биковим [86, с. 395] – це природньо і штучно (спеціально і не спеціально створені) матеріальні та інформаційні об'єкти, які використовуються для матеріально-технічного та інформаційно-процесуального забезпечення навчально-виховного процесу, є важливими складовими навчального середовища, застосовуються учасниками навчально-виховного процесу для досягнення педагогічних цілей відповідно до державних стандартів. Причому, ефективність використання засобів навчання визначається не самим фактом їх використання, а тим, якою мірою вони сприяють

розв'язанню педагогічних задач. Найбільша ефективність використання засобів в тих випадках, коли наочний матеріал виступає не як предмет споглядання, а як засіб розв'язування професійно-практичних задач.

На кожному етапі розвитку психолого-педагогічної науки адекватно розвиваються і засоби навчання, в яких акумулюються і відтворюються науково-технічні, психолого-педагогічні та соціально-економічні досягнення свого часу. Еволюція використання засобів навчання визначається потребами педагогічної практики, а їх розвиток спрямовується на задоволення цих потреб. Засобам навчання завжди притаманна різноманітність форм їх реалізації та методик їх застосування, вони підпорядковуються тій парадигмі освіти, що склалася у суспільстві на даний час.

Зокрема, існуючий рівень науки і техніки формує технологічне та інформаційне середовище, яке використовує людина, впливає на буття людини в оточуючому їй природному і соціальному середовищі, визначає рівень можливостей людини на конкретному етапі науко-технічного прогресу, соціально-економічного розвитку суспільства. Аналіз стану та тенденцій науково-технічного прогресу надає можливість прогнозувати подальший розвиток засобів навчання, що мають формуватися та розвиватися у навчальних закладах, та бути основою такого навчального середовища, яке відповідало би науковим, технологічним і соціальним умовам розвитку суспільства та потребам освіти.

Вплив наукових і технічних досягнень людства на зміст, структуру та організацію процесу навчання опосередковується і має матеріальний вираз у засобах навчання, як знаряддях навчальної діяльності. Вплив використання засобів навчання на результати навчально-виховного процесу найбільш яскраво виявляється у процесі навчання природничо-математичних і технологічних дисциплін [86, с. 396].

Згідно Проекту Положення про електронні освітні ресурси [88], *електронні освітні ресурси* – це «навчальні, наукові, інформаційні, довідкові матеріали і засоби, розроблені в електронній формі і представлені на носіях

будь-якого типу або розміщені в комп'ютерних мережах, які відтворюються за допомогою електронних цифрових технічних засобів і необхідні для ефективної організації навчально-виховного процесу, у частині, що стосується його наповнення якісними навчально-методичними матеріалами» [88, с. 2].

Під *хмаро зорієнтованими електронними освітніми ресурсами (ХЗ ЕОР)* услід за О. В. Мерзликіним розумітимемо вид електронних освітніх ресурсів, що використовуються за хмарною моделлю доступу, а саме – навчальні, наукові, інформаційні, довідкові матеріали та засоби, які розроблені в електронній формі, використовуються за хмарною моделлю доступу, відтворюються за допомогою відповідних електронних цифрових технічних засобів і необхідні для ефективної організації навчально-виховного процесу, у частині, що стосується його наповнення якісними навчально-методичними матеріалами. Тоді до складу хмаро зорієнтованого електронного освітнього ресурсу входять як відповідні засоби ІКТ (програмна складова), так й дані навчального призначення (інформаційна складова) [175, с. 111].

Послугуючись класифікацією електронних освітніх ресурсів, поданою у [88], уведемо відповідне поняття: *хмаро зорієнтовані електронні ресурси навчання основ математичної інформатики* (засоби хмарних технологій навчання основ математичної інформатики) – це сукупність ХЗ ЕОР, що застосовуються для інформаційно-процесуального забезпечення виконання дидактичних завдань або їх фрагментів та спрямовані на реалізацію цілей навчання основ математичної інформатики.

Засоби хмарних технологій навчання основ математичної інформатики поділяються на:

– *хмаро зорієнтовані навчальні посібники* – навчальні електронні видання, що доповнюють підручник і призначені для поширення за хмарною моделлю доступу;

– *хмаро зорієнтовані засоби оцінювання навчальних досягнень* – засоби хмарних технологій, що надають можливість автоматизації процесів визначення рівня навчальних досягнень студентів, призначені для підтримки

процесів оцінювання та самооцінювання у навчанні;

– *хмаро зорієнтовані навчальні лабораторії* – програмні ХЗ ЕОР, що можуть застосовуватись при проведенні лабораторних і практичних занять для здійснення експериментальних досліджень з комп'ютерними моделями.

– *хмаро зорієнтовані електронні довідники* – електронні навчальні видання за хмарною моделлю доступу до коротких наукових і прикладних відомостей довідкового змісту;

– *хмаро зорієнтовані дидактичні демонстраційні матеріали* – ХЗ ЕОР, призначені для наочного подання об'єктів та процесів, що вивчаються;

– *хмаро зорієнтовані середовища моделювання* – хмаро зорієнтовані навчальні лабораторії, призначені для моделювання об'єктів, явищ і процесів, що є предметом вивчення, або надання засобів для побудови і дослідження моделей;

– *хмаро зорієнтовані тренажери* – програмні ХЗ ЕОР, призначені для формування і закріплення умінь та практичних навичок, опанування методів, процедур виконання певних видів навчальної або професійної діяльності, а також для здійснення самопідготовки;

– *хмаро зорієнтовані практикуми* – програмні ХЗ ЕОР, призначені для формування і закріплення умінь та практичних навичок, використання теоретичних знань для розв'язання практичних завдань і вправ;

– *хмаро зорієнтовані предметні середовища* – комплекс взаємопов'язаних програмних ХЗ ЕОР для розв'язання задач певного класу із предметної галузі, що вивчається, призначений для автоматизації дій, що виникають у цій галузі;

– *хмаро зорієнтовані системи комп'ютерної математики* – комплекс програмних ХЗ ЕОР для автоматизації виконання чисельних та аналітичних обчислень;

– *хмаро зорієнтовані середовища програмування* – комплекс взаємопов'язаних програмних ХЗ ЕОР для розробки програмного забезпечення;

– *хмаро зорієнтовані навчально-методичні комплекси* – структурована

сукупність ХЗ ЕОР, що містять навчальні матеріали, призначені для спільного використання у процесі навчання;

– *хмаро зорієнтовані програмно-методичні матеріали* – електронні навчальні видання за хмарною моделлю доступу, що визначають зміст, обсяг, порядок навчання певної дисципліни, її розділу, тем (навчальні програми, плани; плани занять);

– *хмаро зорієнтовані навчально-методичні матеріали*– електронні навчальні видання за хмарною моделлю доступу, що містять матеріали з методики навчання певної дисципліни (її розділу, частини);

– *хмаро зорієнтовані додаткові науково-навчальні матеріали* – інформаційні ресурси за хмарною моделлю доступу, які сприяють доповненню і розширенню уявлень про об'єкти і процеси, що є предметом вивчення;

– *хмаро зорієнтовані тестові системи* – ХЗ ЕОР, що містять стандартизовані тестові завдання та призначені для оцінювання рівня навчальних досягнень тих, хто навчається;

– *хмаро зорієнтовані операційні системи* – комплекс програмно-апаратних засобів для автоматизації самостійного розгортання операційного середовища за моделями PaaS та IaaS через віртуалізацію комп'ютера та операційної системи відповідно із усіма необхідними компонентами середовища для доступу за моделями DaaS та SaaS;

– *хмаро зорієнтовані системи підтримки навчання* – система ХЗ ЕОР для підтримки всіх етапів і компонентів процесу навчання, що надають можливість автоматизації організації навчального процесу через збереження і доставляння навчальних ресурсів та організацію навчальної діяльності, управління навчальним процесом, облік та контроль виконання різних видів навчальних робіт, контроль за використанням навчальних ресурсів, адміністрування окремих студентів та груп, організацію взаємодії з викладачем, звітність тощо;

– *хмаро зорієнтовані комунікаційні засоби* – програмні засоби хмарних технологій для обміну даними у голосовій, текстовій, графічній та ін. формах.

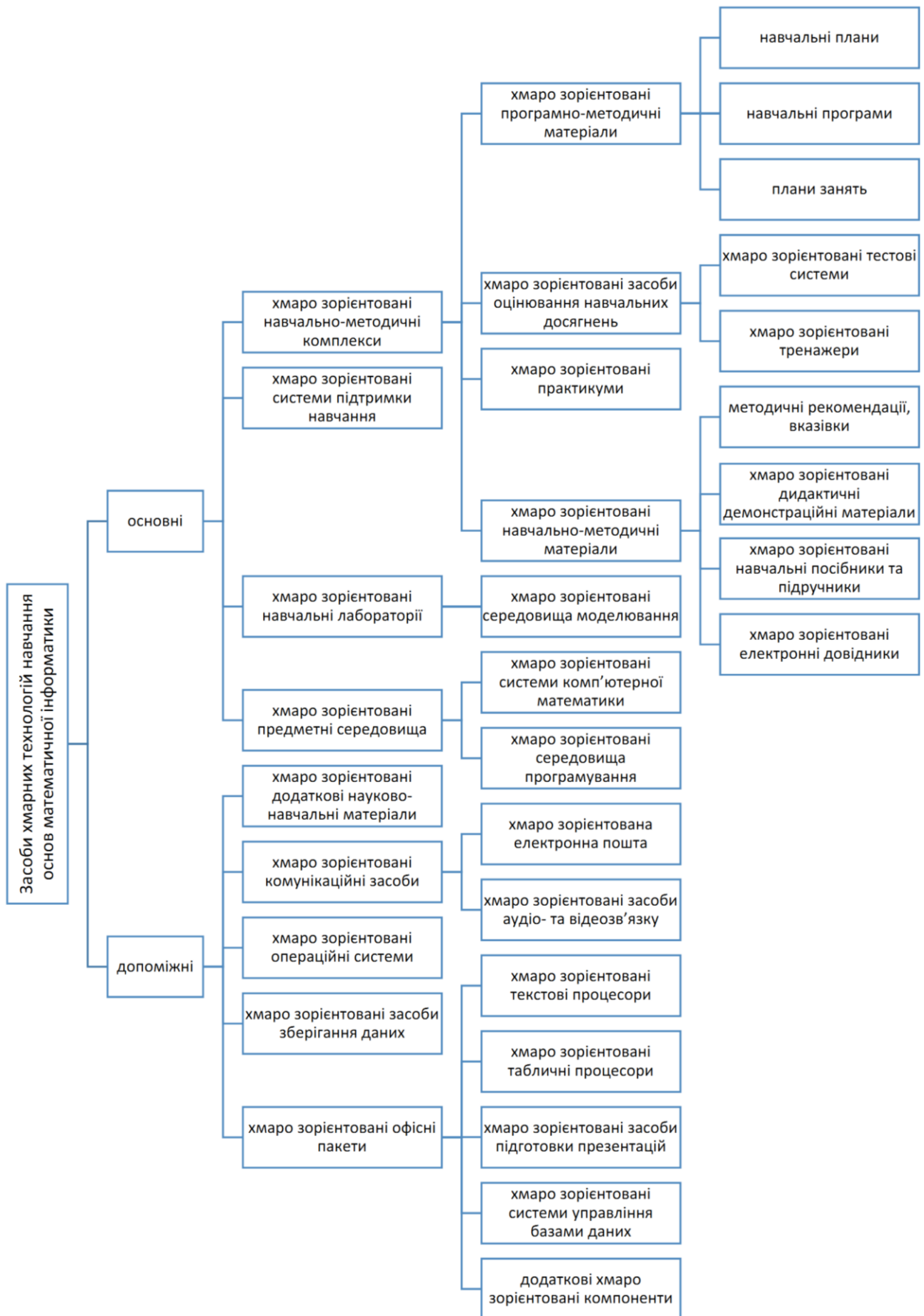


Рис. 2.4. Засоби хмарних технологій навчання основ математичної інформатики

Для підвищення дидактичної ефективності застосовані засоби хмарних технологій навчання основ математичної інформатики використовуються в навчально-виховному процесі спільно з іншими навчально-методичними матеріалами (наприклад, із традиційними підручниками та навчальними посібниками, методичними рекомендаціями для викладачів та студентів тощо), формуючи *хмаро зорієнтовані програмно-методичні комплекси*.

На рис. 2.4 подано загальну структуру засобів хмарних технологій навчання основ математичної інформатики. Приклади таких засобів та методику їх використання у навчанні основ математичної інформатики студентів технічних університетів розглянемо у наступному підрозділі.

2.3 Методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів

Згідно [186], термін «методика» має декілька значень: 1) сукупність взаємозв'язаних способів та прийомів доцільного проведення будь-якої роботи; 2) документ, який описує послідовність методів, правил і засобів виконання роботи; 3) вчення про методи викладання певної науки, предмета.

Згідно В. В. Ягупова, метод – це засіб досягнення будь-якої мети, вирішення конкретного завдання; сукупність прийомів або операцій практичного чи теоретичного засвоєння (пізнання) дійсності, а методика – конкретні принципи, форми та засоби використання методів, за допомогою яких здійснюється більш глибоке пізнання різноманітних педагогічних проблем та їх розв'язання [268]. Методика повинна мати такі характеристики, як правильність, обґрунтованість, економічність, оптимальність та ін.

Методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів складається із трьох основних блоків: цільовий (формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій), змістовий (навчання основ математичної інформатики) та технологічний (хмаро зорієнтовані засоби ІКТ, методи та форми їх використання у навчанні

математичної інформатики). Технологічний блок методики визначає провідний зміст діяльності (індивідуальні та групові навчальні дослідження), форму організації навчання (спецкурс), види діяльності з формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій та відповідні засоби ІКТ.

Розглянемо більш детально технологічний блок розробленої методики.

Для розміщення *хмаро зорієнтованих програмно-методичних матеріалів* доцільно використовувати сайт, який може бути створений з використанням пакету Google Apps for Education. Для отримання безкоштовної підписки на останній необхідно зареєструвати домен освітньої установи та подати відповідну заявку (так, для спецкурсу «Основи математичної інформатики» було виділено домен mathinfo.ccjournals.eu, пов'язаний із ДВНЗ «Криворізький національний університет»).

Розробка сайту виконується у складовій Google Apps for Education – Google Sites. Це надає можливість розміщення, редагування та зберігання до 10 Гб програмно-методичних матеріалів на хмарних серверах Google. Для зручності доступу до сайту його необхідно пов'язати із освітнім доменом, використовуючи записи CNAME. На рис. 2.5 показано фрагмент сайту <http://site.mathinfo.ccjournals.eu/>, що виступає точкою входу до всіх засобів хмарних технологій навчання основ математичної інформатики студентів технічних університетів.

Хмаро зорієнтовані засоби оцінювання навчальних досягнень представлені насамперед *хмаро зорієнтованими тестовими системами*. На сторінці «Засоби оцінювання навчальних досягнень» сайту спецкурсу «Основи математичної інформатики» представлені посилання на деякі з них. Наприклад, у системі Quizful (<http://www.quizful.net>) студентами пропонується пройти тестування із модуля 4 «Основи криптографії» (рис. 2.6). За відсутності розроблених тестів вони можуть бути створені у інших системах (наприклад, <http://www.onlinequizcreator.com> – рис. 2.7).

The screenshot shows a web browser window with the URL `site.mathinfo.ccjournals.eu/korotko-1`. The page title is "Коротко про спецкурс 'Основи математичної інформатики'". The header features the logo "КСІМ" and the text "Математична інформатика". A search bar is present in the top right. The main content area includes a table of contents for the course.

Коротко про спецкурс "Основи математичної інформатики"

У таблиці 1 наведений перелік зміст модулів, що увійшли до спецкурсу "Основи математичної інформатики"

Таблиця 1
Зміст навчання основ математичної інформатики студентів технічних університетів

Назва модуля	Зміст модуля
I. Теорія алгоритмів	1. Аналіз алгоритмів

Рис. 2.5. Сайт спецкурсу «Основи математичної інформатики», розміщеного на хмарних серверах Google

The screenshot shows a quiz result page from `www.quizful.net/testRun/O47jP6f4`. The title is "Тест 'Криптография' - не пройден пройти еще раз". The results show 11 correct answers out of 20 (55%), with 15 more needed. The test date is 18.05.2016 16:48. A question about the algorithm used in the Russian standard GOST 28147-89 is shown, with DES selected as the correct answer.

Результат прохождения те... x +

`www.quizful.net/testRun/O47jP6f4` Поиск

Тест "Криптография" - не пройден пройти еще раз

Правильных ответов: 11 / 20 (55 %) требуется: 15

Дата прохождения теста: 18.05.2016 16:48

✓ Какой алгоритм шифрования стал прообразом для отечественного ГОСТ28147-89?

<input checked="" type="radio"/> DES	313 / 501
<input type="radio"/> DSA	48 / 501
<input type="radio"/> Rijndael	76 / 501
<input type="radio"/> ГОСТ34.11	60 / 501
<input type="radio"/> IDEA	20 / 501

Рис. 2.6. Результаты тестування студента у хмаро зорієнтованій тестовій системі Quizful

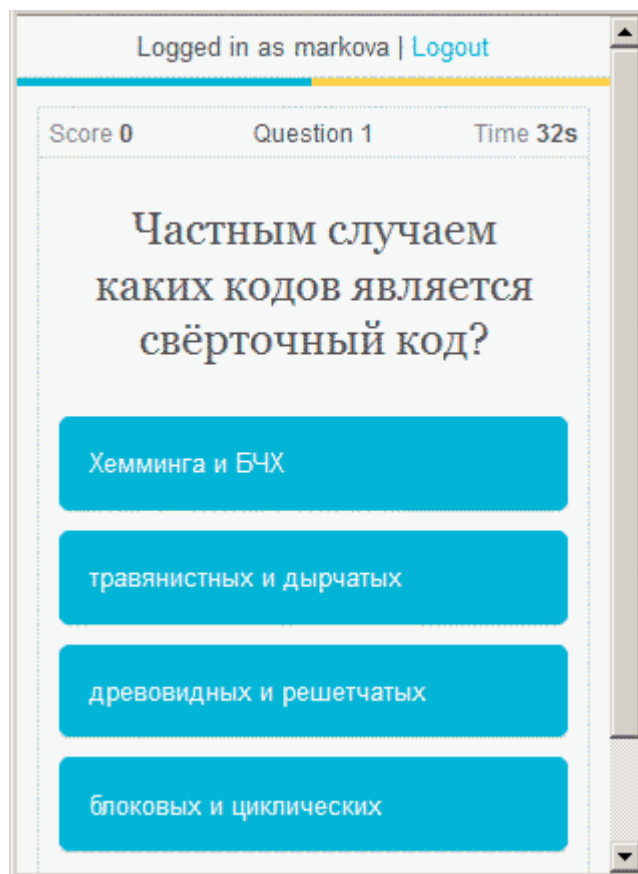


Рис. 2.7. Тестування у хмаро зорієнтованій тестовій системі QuizWorks

Виходячи з того, що розробка сайту спецкурсу ведеться у хмаро зорієнтованому середовищі Google Apps, доцільним є інтеграція хмаро зорієнтованих тестових систем із сайтом. Для цього пропонується користувачам освітнього домену надати можливість використання засобів для розробки тестів, доступних у Google Apps Marketplace (рис. 2.8), шляхом їх вибору для застосування користувачами домену.

Серед доступних у Google Apps Marketplace засобів, які успішно використовуються у навчальному процесі та мають значну кількість позитивних відгуків, відзначимо Flubaroo – хмаро зорієнтовану тестову систему, що надає можливість перевірити тест, створений за допомогою Google Forms (рис. 2.9). Flubaroo надає можливість надіслати результати оцінювання з поясненнями електронною поштою або через Google Drive, проаналізувати середній бал, побудувати гістограму розподілу відповідей та виконати ряд інших дій, що у цілому надають можливість виявляти студентів, які потребують

додаткової допомоги, оперативно визначити питання, що потребують додаткової уваги з боку викладача.

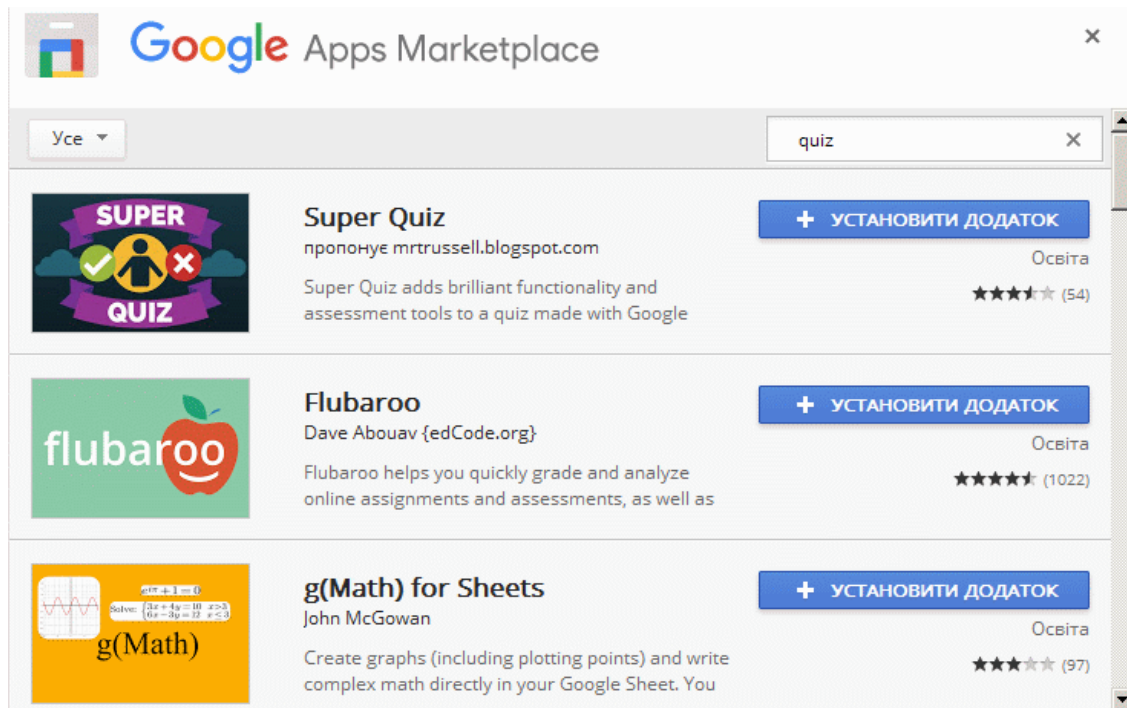


Рис. 2.8. Деякі хмаро зорієнтовані тестові системи, доступні у Google Apps Marketplace

Тестування з основ математичної інформатики

Змістовий модуль 3. Теорія кодування

*Обов'язкове поле

Уведіть Ваші прізвище, ім'я та по батькові *

Купін Андрій Гнатович

Кодування - це *

- відновлення даних
- спотворення даних
- перетворення даних
- перетворення даних у форму, зручну для передавання

НАДІСЛАТИ

Рис. 2.9. Створення тесту в Google Form

Створена форма повинна містити обов'язкове поле для ідентифікації студента та бути пов'язаною з таблицею у Google Sheets, для якої Flubaroo після встановлення стає доповненням. Для проведення оцінювання необхідним є наявність форми з усіма еталонними відповідями (зазвичай, її заповнює викладач).

Оцінювання виконується для даних із результатами тестування, що були збережені у Google Sheets, вибором пункту Доповнення – Flubaroo – Grade Assignment:

- на першому кроці для кожного питання обирається шкала оцінювання (автоматичне – Normal Grading, ручне – Grade by Hand або без оцінювання – Skip Grading) та встановлюється кількість балів, що нараховується за правильну відповідь; питання, що ідентифікує студента, позначається як Identifies Student;

- на другому кроці обирається рядок листа таблиці Student Submissions із правильними відповідями (Answer Key), що надалі виключається з оцінювання;

- останній крок – автоматичне створення нового листа Grades, що містить форматовані результати оцінювання по кожному студенту та статистику проходження тесту.

Super Quiz – інша хмаро зорієнтована тестова система, яка надає можливість проводити тестування в ігровій формі (вікторина). Запитання вікторини представлені формою, створеною в Google Form. Після закінчення вікторини усі відповіді порівнюються із перевірочними ключами та автоматично класифікуються на правильні та неправильні.

К. І. Словак [222] під тренажерами розуміє програми, основне призначення яких полягає у поданні всіх етапів розв'язування математичної задачі. В процесі самостійної роботи тренажери виступають як засіб формування та удосконалення практичних навичок, перевірки досягнутих результатів, розрахований на повторення та закріплення навчального матеріалу. *Хмаро зорієнтовані тренажери* можуть містити елементи тестування, проте головне їх призначення – відпрацювання певних навичок у хмаро зорієнтованому навчальному середовищі. Ураховуючи, що цей процес може

бути достатньо тривалим, розробники тренажерів часто застосовують елементи гейміфікації.

На рис. 2.10 наведено приклад використання хмаро зорієнтованої експертної системи комп'ютерної математики Wolfram|Alpha у якості тренажера.

Wolfram|Alpha Step-by-step Solution

Result: Use prime factorization ▼ Hide hints

Find the least common multiple:
 $\text{lcm}(90, 342)$

Find the prime factorization of each integer:
 The prime factorization of 90 is:
 $90 = 2 \times 3^2 \times 5$

The prime factorization of 342 is:
 $342 = 2 \times 3^2 \times 19$

Find the largest power of each prime factor.
 The largest power of 2 that appears in the prime factorizations is 2^1 .
 The largest power of 3 that appears in the prime factorizations is 3^2 .
 The largest power of 5 that appears in the prime factorizations is 5^1 .
 The largest power of 19 that appears in the prime factorizations is 19^1 .
 Therefore $\text{lcm}(90, 342) = 2^1 \times 3^2 \times 5^1 \times 19^1$:

Answer:
 $\text{lcm}(90, 342) = 1710$




Рис. 2.10. Покрокове знаходження найменшого спільного кратного у Wolfram|Alpha

Одним із найбільш потужних засобів хмарних технологій, що надає можливість реалізації *хмаро зорієнтованих практикумів*, є CoCalc (раніше відомий як SageMathCloud) – хмаро зорієнтований варіант системи

комп'ютерної математики Sage, розміщений на серверах Google. CoCalc – не просто хмаро зорієнтований варіант системи комп'ютерної математики, а й система підтримки навчання математичних та інформатичних дисциплін. Основними компонентами CoCalc є:

1) Sage Worksheets – робочі аркуші Sage, що надають можливість інтерактивного виконання команд Sage або мов програмування та документування, таких як C++ та HTML;

2) «блокноти» IPython (з 2016 року – Jupyter Notebook) – синхронізований сеанс мовою програмування Python, який є частиною бібліотеки для наукових та інженерних обчислень SciPy. CoCalc надає можливість кільком користувачам взаємодіяти через засоби комунікації у «блокнотах» IPython у синхронному та асинхронному режимах;

3) система документообігу мовою LaTeX з повною підтримкою sagetex, bibtex тощо;

4) система резервного копіювання – повне збереження усіх змінених файлів проекту користувача кожні 2 хвилини;

5) система реплікації передбачає збереження кожного проекту в трьох фізично відокремлених центрах опрацювання даних [72].

CoCalc надає можливості:

– інтерактивного вивчення математики, природничих та комп'ютерних наук;

– співпраці з іншими користувачами в режимі реального часу;

– організації навчальних курсів: додавання студентів, створення власних проектів, моніторингу їх розвитку тощо з використанням хмаро зорієнтованих навчально-методичних матеріалів;

– створення та редагування навчальних і наукових текстів засобами LaTeX, Markdown або HTML;

– додавання власних файлів, опрацювання даних, оприлюднення результатів та ін.

Наявність інструменту Besides Sage Worksheets у складі Jupyter Notebooks

надає користувачам повний доступ до класичного Linux-терміналу [16].

Основна робота у CoCalc відбувається у проекті. Користувач може створити будь-яку кількість незалежних проектів – особистих робочих просторів, у яких користувач зберігає ресурси різних типів. Користувач також може запросити інших до співпраці у межах спільного проекту та надати відкритий доступ до файлів або папок.

Кожен проект виконується на сервері CoCalc, де він ділить дисковий простір, центральний процесор та оперативну пам'ять з іншими проектами. Безкоштовний тарифний план передбачає використання лише тих ресурсів сервера, що у поточний момент є вільними. Крім того, якщо проект користувача безкоштовного тарифного плану не використовується протягом декількох тижнів, він переміщується у вторинне сховище з метою вивільнення ресурсів серверу, і його повторний запуск займе суттєво більше часу, ніж у користувача платного тарифного плану.

Учасники проекту можуть об'єднати власні обчислювальні та зберігальні ресурси з метою покращення можливостей проекту в цілому та перерозподілу ресурсів між собою. Організувати спільну роботу з ресурсами проекту CoCalc можна або на рівні окремо взятого ресурсу, зокрема робочого аркушу, або на рівні проекту в цілому.

Відкриття спільного доступу на рівні окремо взятого ресурсу є нічим іншим, як web-оприлюдненням вмісту ресурсу у режимі «лише для читання» для всіх користувачів мережі Інтернет, які мають посилання на даний ресурс. Недоліками такого оприлюднення є те, що користувач-«читач» не має можливості управляти обчисленнями на робочому аркуші, навіть якщо автор використав стандартні елементи управління у ньому. Проте, у разі необхідності, оприлюднений робочий аркуш може бути скопійований або завантажений.

Організація спільної роботи на рівні проекту в цілому можлива як без використання ресурсу типу course, так і за його допомогою. Перший спосіб передбачає підключення до проекту учасників, які матимуть можливість спільно працювати з вже існуючими навчальними ресурсами проекту або

додавати нові, запрошувати інших учасників, спілкуватись за допомогою текстового та/або відео чатів у рамках спільного проекту. Внесок кожного учасника спільного проекту у вирішення його завдань може бути переглянутий на сторінках історії роботи з проектом або на сторінках його резервних копій [197].

Матеріали хмаро зорієнтованого практикуму в CoCalc доцільно оформити у вигляді структурованого набору робочих аркушів Sage, розміщених у каталогах, кожен із яких відповідає певному змістовому модулю (рис. 2.11).

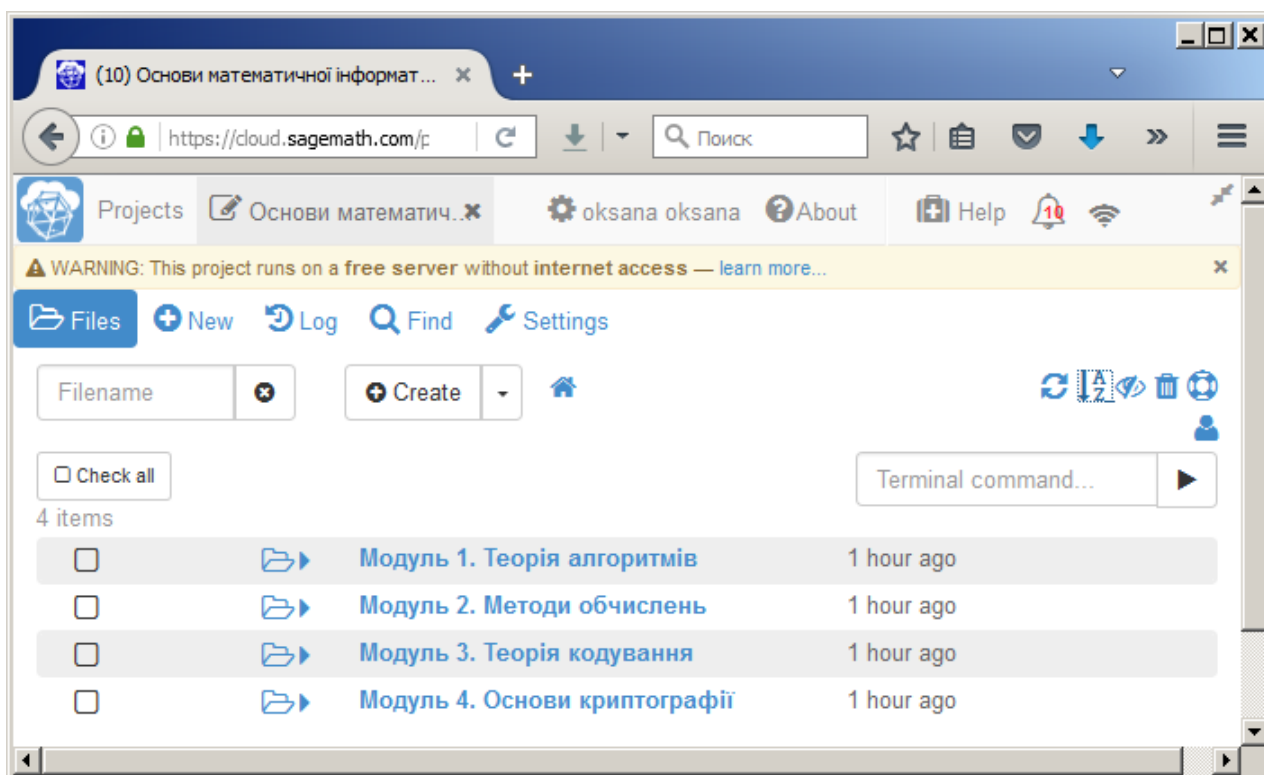


Рис. 2.11. Хмаро зорієнтований практикум з основ математичної інформатики у CoCalc

За кожним модулем передбачено розв'язання ряду практичних завдань і вправ. Наприклад, на лекції з першої теми «Криптографічні системи» модуля «Основи криптографії» передбачено розгляд класичних криптосистем (зокрема, шифру Цезаря) та коротких відомостей про криптоаналіз. До відповідного теоретичного матеріалу доцільно запропонувати наступні завдання:

Завдання 1. Реалізуйте у Sage функції шифрування/дешифрування, які приймають ключ (ціле число від 0 до 25) і рядок. Функція повинна

опрацьовувати лише літери 'a', 'b', ..., 'z' (як у верхньому, так й у нижньому регістрах), будь-які інші символи повинні залишитися незмінними.

```

#алфавіт (літери 'a', 'b', ..., 'z')
en_alphabet = "abcdefghijklmnopqrstuvwxyz"
#перевірка входження літери до алфавіту
def is_alphabetic_char(c):
    return (c.lower() in en_alphabet)
#повертає номер літери в алфавіті
def char_to_num(c):
    return en_alphabet.index(c.lower())
#повертає літеру із заданим номером
def num_to_char(x):
    return en_alphabet[x % 26]
#функція шифрування тексту plaintext ключем k
def CaesarEncrypt(k, plaintext):
    ciphertext = ""
    #для кожної літери
    for j in xrange(len(plaintext)):
        #якщо вона належить алфавіту
        if is_alphabetic_char(plaintext[j]):
            #виконати її циклічний зсув праворуч на ключ k
            c = num_to_char((k + char_to_num(plaintext[j])) % 26)
        #інакше
        else:
            #залишити літеру без змін
            c = plaintext[j]
        #додати перетворену літеру до шифротексту
        ciphertext += c
    #повернути результат
    return ciphertext
#функція дешифрування тексту ciphertext ключем k
def CaesarDecrypt(k, ciphertext):
    plaintext = ""
    #для кожної літери
    for j in xrange(len(ciphertext)):
        #якщо вона належить алфавіту
        if is_alphabetic_char(ciphertext[j]):
            #виконати її циклічний зсув ліворуч на ключ k
            p = num_to_char((char_to_num(ciphertext[j]) - k) % 26)
        #інакше
        else:
            #залишити літеру без змін
            p = ciphertext[j]
        #додати перетворену літеру до відкритого тексту
        plaintext += p
    #повернути результат
    return plaintext
#тестування шифрування
CaesarEncrypt(7, "markova@mathinfo.ccjournals.eu")

    'thyrvch@thaopumv.jjqvbyuhsz.lb'

#тестування дешифрування
CaesarDecrypt(7, "thyrvch@thaopumv.jjqvbyuhsz.lb")

    'markova@mathinfo.ccjournals.eu'

```

Завдання 2. Реалізуйте функцію, яка виконує атаку на шифротекст методом повного перебору. Функція повинна надрукувати список ключів і пов'язаних з ними розшифровок. Функція також повинна приймати у якості додаткового параметру підрядок і друкувати тільки потенційні відкриті тексти, які містять відповідну розшифровку.

```
def BruteForceAttack(ciphertext, keyword=None):
    #перебір усіх можливих ключів
    for k in xrange(26):
        #намагання розшифрувати ключем k
        plaintext = CaesarDecrypt(k, ciphertext)
        #якщо підрядок keyword не заданий або міститься у розшифрованому тексті
        if (None==keyword) or (keyword in plaintext):
            #друкувати ключ та розшифровку
            print "Для ключа", k, " розшифровка буде", plaintext
    return
```

Приклад використання створеної функції:

```
#розшифровка повним перебором
BruteForceAttack("ilkd ifsb jxqebjxqfzxi fkclojxqfzp!")

Для ключа 0 розшифровка буде ilkd ifsb jxqebjxqfzxi fkclojxqfzp!
Для ключа 1 розшифровка буде hkjc hera iwpdaiwpeywh ejbknwpeyo!
Для ключа 2 розшифровка буде gjib gdqz hvoczvhvodxvg diajmhvodxn!
Для ключа 3 розшифровка буде fiha fcpu gunbyguncwuf chzilguncwm!
Для ключа 4 розшифровка буде ehgz ebox ftmaxftmbvte bgyhkftmbvl!
Для ключа 5 розшифровка буде dgfy danw eslzweslausd afxgjeslauk!
Для ключа 6 розшифровка буде cfex czmv drkyvdrkztrc zewfidrkztj!
Для ключа 7 розшифровка буде bedw bylu cqjxucqjysqb ydvehcqjysi!
Для ключа 8 розшифровка буде adcv axkt bpiwtbpixrpa xcudgbpigrh!
Для ключа 9 розшифровка буде zcbu zwjs aohvsaohwqoz wbtcfaohwqg!
Для ключа 10 розшифровка буде ybat yvir zngurzngvnpny vasbezngvpf!
Для ключа 11 розшифровка буде xazs xuhq ymftqymfuomx uzradymfuoe!
Для ключа 12 розшифровка буде wzyr wtgp xlespxletnlw tyqzcxletnd!
Для ключа 13 розшифровка буде vuxq vsfo wkdrowkdsmkv sxpybwkdsnc!
Для ключа 14 розшифровка буде uxwp uren vjcqnvjcrlju rwoxavjcrbl!
Для ключа 15 розшифровка буде twvo tqdm uibpmuibqkit qvnwzuibqka!
Для ключа 16 розшифровка буде svun spcl thaolthapjhs pumvythapjz!
Для ключа 17 розшифровка буде rutm robk sgznksgzoigr otluxsgzoiy!
Для ключа 18 розшифровка буде qtsl qnaj rfymjrfynhfq nsktrwrfynhx!
Для ключа 19 розшифровка буде psrk pmzi qexliqexmgep mrjsvqexmgw!
Для ключа 20 розшифровка буде orqj olyh pdwkhpdwlfdo lqirupdwlfv!
Для ключа 21 розшифровка буде nqpi nkxg ocvjgocvkecn kphqtocvkeu!
Для ключа 22 розшифровка буде mpoh mjwf nbuifnbujdbm jogpsnbujdt!
Для ключа 23 розшифровка буде long live mathematical informatics!
Для ключа 24 розшифровка буде knmf khud lzsgdlzshbzk hmenqlzshbr!
Для ключа 25 розшифровка буде jmle jgtc kyrfckyrGayj gldmpkyrgaq!
```

```
#розшифровка за відомим фрагментом тексту
BruteForceAttack("thyrvch@thaorumv.jjqvbyuhsz.lb", "cc")
```

Завдання 3. Продемонструйте результат функції шифрування із завдання 1 на наступних парах (ключ, відкритий текст):

`k = 16, plaintext = "Caesar non supra grammaticos"`

`k = 15, plaintext = "Veni, vidi, vici"`

`k = 16, plaintext = "The Caesar cipher is easy to decrypt"`

```
k = 16; plaintext = "Caesar non supra grammaticos"
CaesarEncrypt(k, plaintext)
k = 15; plaintext = "Veni, vidi, vici"
CaesarEncrypt(k, plaintext)
k = 16; plaintext = "The Caesar cipher is easy to decrypt"
CaesarEncrypt(k, plaintext)
```

```
'squiqh ded ikfhq whqccqjysei'
'ktsx, kxsx, kxrх'
'jxu squiqh syfxuh yi uqio je tushofj'
```

Завдання 4. Продемонструйте результат функції дешифрування із завдання 1 на наступних парах (ключ, зашифрований текст):

`k = 42, ciphertext = "squiqh ded ikfhq whqccqjysei"`

`k = 42, ciphertext = "jxu squiqh syfxuh yi uqio je tushofj"`

`k = 67, ciphertext = "ktsx, kxsx, kxrх"`

Поясніть вибір ключів, більших за довжину алфавіта.

```
k = 42; ciphertext = "squiqh ded ikfhq whqccqjysei"
CaesarDecrypt(k, ciphertext)
k = 42; ciphertext = "jxu squiqh syfxuh yi uqio je tushofj"
CaesarDecrypt(k, ciphertext)
k = 67; ciphertext = "ktsx, kxsx, kxrх"
CaesarDecrypt(k, ciphertext)
```

```
'caesar non supra grammaticos'
'the caesar cipher is easy to decrypt'
'veni, vidi, vici'
```

Завдання 5. Змініть функції шифрування та дешифрування із завдання 1 у такий спосіб, щоб регістр тексту не змінювався та опрацьовувались кириличні літери. Продемонструйте їх працездатність.

```
#змінений алфавіт
alphabetL="abcdefghijklmnopqrstuvwxyzабвгдеёжзийклмнопрстуфхцчщъыьэюяґеіі"

def is_alphabetic_char(c):
    #без зміни регістру
    return (c in alphabetL)

def char_to_num(c):
    #без зміни регістру
    return alphabetL.index(c)

def num_to_char(x):
    #циклично з урахування кількості символів у алфавіті
```

```

return alphabetL[x % len(alphabetL)]

def CaesarEncrypt(k, plaintext):
    ciphertext = ""
    for j in xrange(len(plaintext)):
        if is_alphabetic_char(plaintext[j]):
            c = num_to_char((k + char_to_num(plaintext[j])) %
                             len(alphabetL))
        else:
            c = plaintext[j]
        ciphertext += c
    return ciphertext

def CaesarDecrypt(k, ciphertext):
    plaintext = ""
    for j in xrange(len(ciphertext)):
        if is_alphabetic_char(ciphertext[j]):
            p = num_to_char((char_to_num(ciphertext[j]) - k) %
                             len(alphabetL))
        else:
            p = ciphertext[j]
        plaintext += p
    return plaintext

#на вхід функції шифрування подаємо UNICODE-рядок
print CaesarEncrypt(10,u"Оксана Миколаївна Маркова")

#на вхід функції дешифрування подаємо UNICODE-рядок
print CaesarDecrypt(10,u"Офьйчй Мтфшхйјлчй Мйгфшлй")
'Офьйчй Мтфшхйјлчй Мйгфшлй'
'Оксана Миколаївна Маркова'

```

Важливим джерелом *хмаро зорієнтованих дидактичних демонстраційних матеріалів* з основ математичної інформатики є Wolfram Demonstration Project, що містить значну кількість демонстрацій з усіх модулів курсу (рис. 2.12). Для їх використання необхідно встановити вільно поширюваний програвач Wolfram CDF Player, інсталяція якого містить плагіни для найбільш поширених веб-браузерів.

Так, при розгляді теми «Лінійні коди» доцільно використати демонстрацію роботи коду Хемінга за посиланням <http://demonstrations.wolfram.com/TheHamming74Code/>. Код Хеммінга (7, 4) є кодом, що може виправити одну помилку, використовуючи 7-бітне кодове слово для передавання чотирьох бітів даних. Відправник обчислює три біта парності для кожного 4-бітного слова даних, збирає дані та біти парності у 7-бітне кодове слово та передає це кодове слово. Отримувач обчислює три біти перевірки парності з прийнятого 7-бітного слова. Якщо помилки при

передаванні не було, всі три контрольні біти парності дорівнюватимуть нулеві. Якщо один біт при передаванні був змінений, значення трьох бітів парності (інтерпретується як 3-бітне двійкове число) буде вказувати на позицію помилки, яка потім може бути усунута.

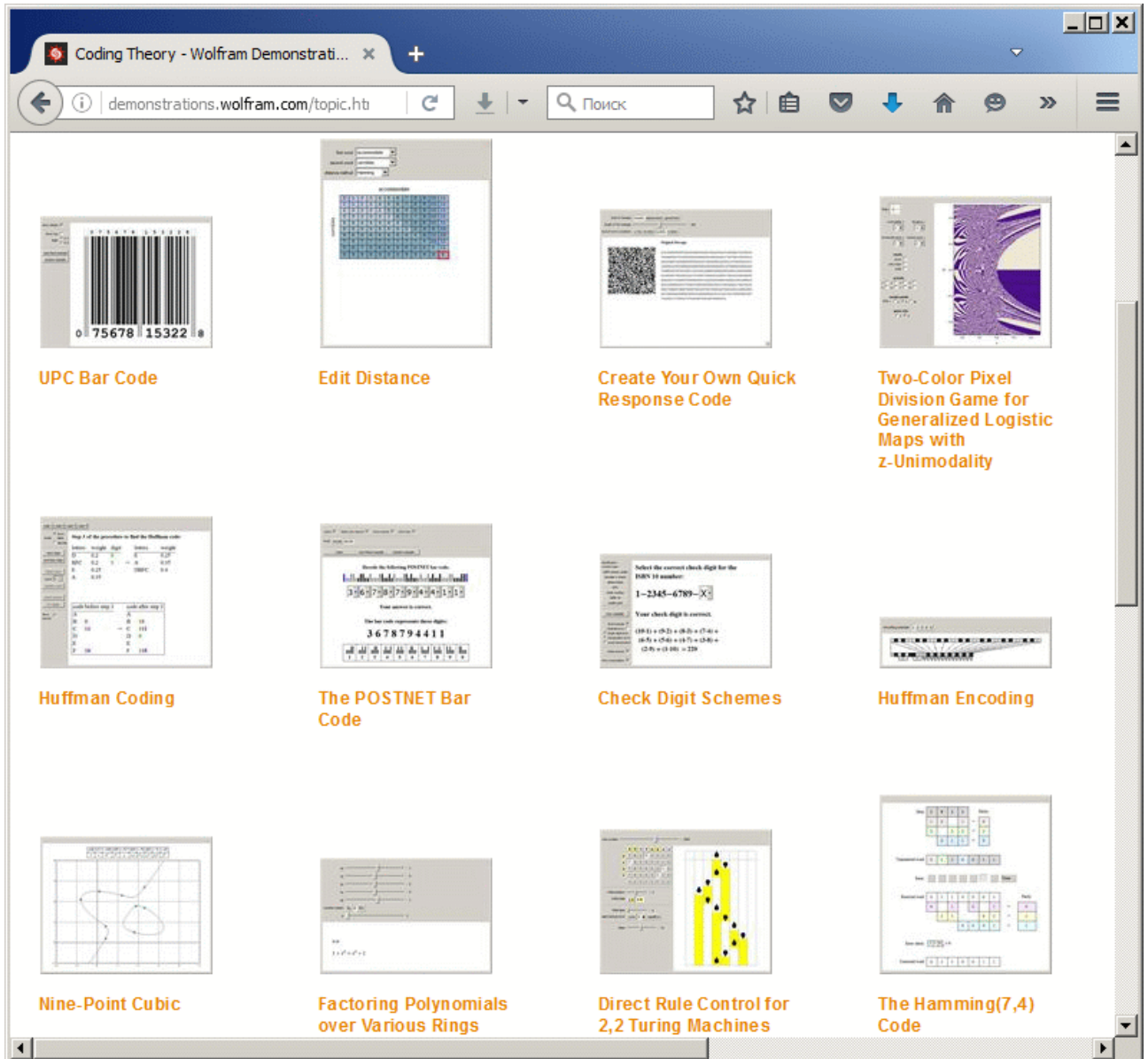


Рис. 2.12. Фрагмент Wolfram Demonstration Project за модулем «Теорія кодування»

Дана демонстрація надає можливість змоделювати таку передачу шляхом встановлення бітів даних, що передаються, у верхній частині вікна (рис. 2.13), та уведення, за бажання, помилки в будь-якому біті слова, що передається.

Хмаро зорієнтовані навчальні посібники можуть бути розроблені з

використанням різних засобів, у тому числі – як файли у форматі CDF (Computable Document Format). Це надає можливість об'єднання даних у різних форматах (текст, відео, аудіо, обчислювальних тощо) у інтерактивний навчальний посібник. Інший спосіб такого об'єднання – використання засобів для створення інтерактивних математичних текстів, що їх надає CoCalc.

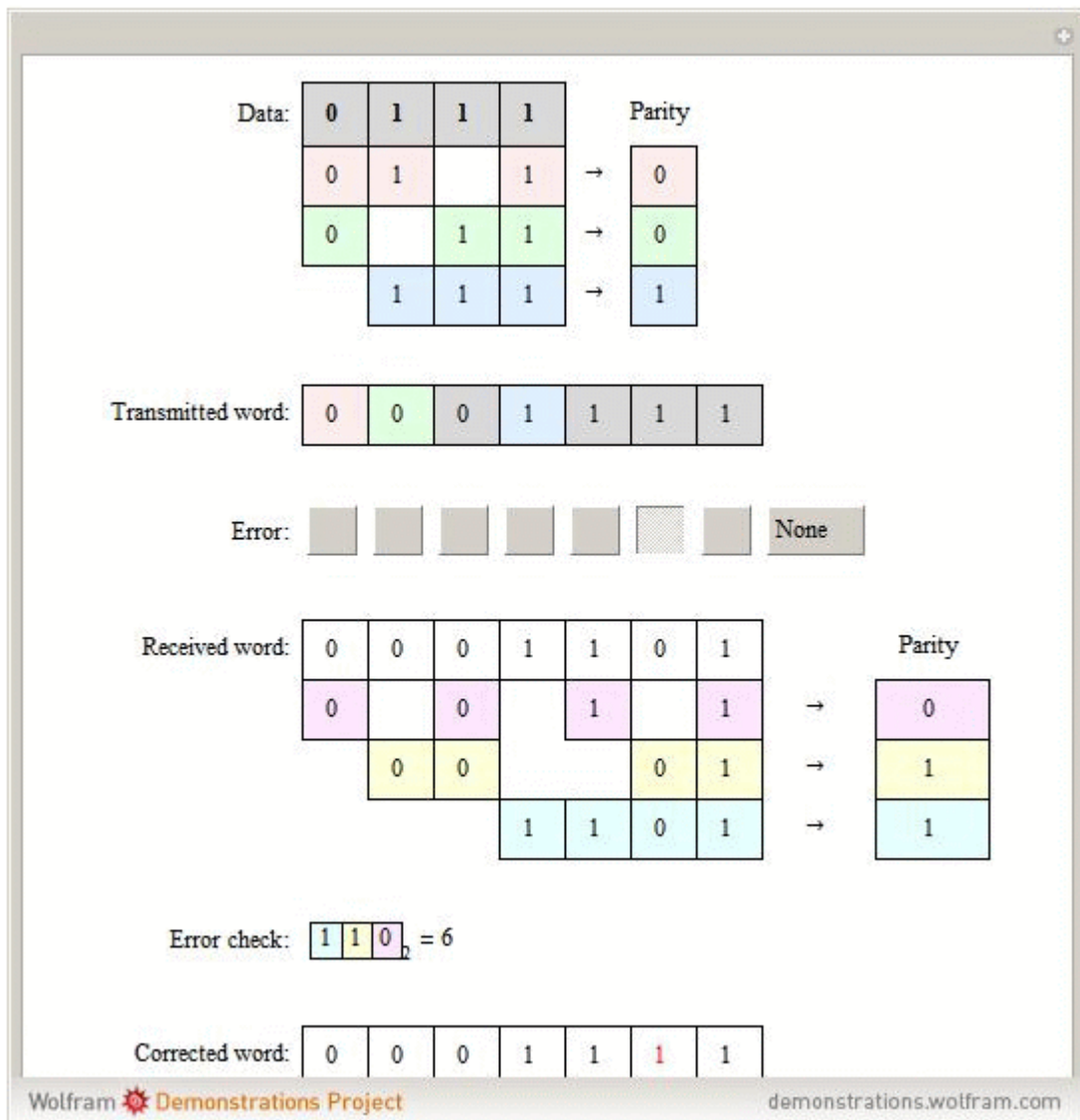


Рис. 2.13. Демонстрація роботи коду Хемінга

Основним засобом створення інтерактивних математичних текстів у CoCalc є система LaTeX, відома своєю надзвичайною стабільністю, здатністю працювати на багатьох комп'ютерних платформах і операційних системах, а також практично повною відсутністю помилок. Номер версії LaTeX сходиться до числа π і зараз дорівнює 3.14159 [55].

SageTeX є пакетом для системи LaTeX, що надає можливість вбудовувати результати обчислень в документи, описані мовою TeX [124]. Для початку використання SageTeX слід вказати `\usepackage{sagetex}` на початку документа.

Для вставки результатів виконання команд мовою Sage у документ мовою TeX застосовується команда `\sage{<код Sage>}`, де <код Sage> – будь-який код мовою Sage. Наприклад, виконання команди `\sage{matrix([[1, 2], [3, 4]])^2}` призведе до появи у тексті її результату мовою LaTeX:

```
\left(\begin{array}{rr}
7 & 10 \\
15 & 22
\end{array}\right)
```

Команда `\sage` може посилатися також на змінні Sage, визначені раніше у поточному документі.

Графічні побудови виконуються за допомогою команди `\sageplot[<опції LaTeX>][<формат>]{<графічні об'єкти>, <ключові аргументи>}`, де:

<опції LaTeX> – будь-який текст, що передається у необов'язкові аргументи, вказані в квадратних дужках, команди `\includegraphics`. Якщо не вказано, буде використовуватися `\width=.75\textwidth`;

<формат> – вказує розширення графічного файлу, у якому Sage намагатиметься зберегти рисунок. Якщо не вказано, буде збережено у форматі EPS або PDF;

<графічні об'єкти> – відповідні графічні команди мовою Sage;

<ключові аргументи> – будь-які параметри виконання попередніх.

Наприклад, команда `\sageplot[angle=30, width=5cm]{plot(sin(x), 0, pi), axes=False}` призведе до появи у тексті її результату – зображення частини синусоїди, нахиленої під кутом 30°.

Якщо у команді була помилка, замість результату її виконання SageTeX вставляє два знаки питання.

Для тривимірної графіки вказання формату є обов'язковим:

```
\sage{var("x y")}
\sageplot[][png]{plot3d(sin(x+y), [x, 0, pi], [y, -pi, 0]), axes=False}
```

Останній приклад можна було б оформити у вигляді фрагменту мовою Sage як блок за допомогою команд `\begin{sageblock}` та `\end{sageblock}`. Наприклад, можна визначити блок:

```
\begin{sageblock}
  var('x')
  f(x) = sin(x) - 1
  g(x) = log(x)
  h(x) = diff(f(x) * g(x), x)
\end{sageblock}
```

Після цього у тексті можна звертатись до описаних у ньому функцій та змінних. Наприклад, «Маємо $h(2) = \text{sage}\{h(2)\}$, де h є похідною добутку f та g ». Виклик `\sage` буде коректно замінений на `sin(1)-1`.

Команда `\sagetexindent` визначає відступ коду Sage у документі LaTeX.

Для розробки україномовних посібників у CoCalc необхідно підключити модулі для підтримки українського правопису та кодування Unicode. Наприклад:

```
\documentclass[a5paper,10pt]{article}
\usepackage[ukrainian]{babel}
\usepackage[utf8]{inputenc}
\usepackage{sagetex}
\title{Завдання до теми "\\Криптографічні системи"}
\author{О. М. Маркова \\
  Кафедра комп'ютерних систем та мереж \\ ДВНЗ "Криворізький національний
  університет"}
\date{2015-2016 н.р.}
\begin{document}
\maketitle
```

На рис. 2.14 показано фрагмент такого посібника, розроблюваного у CoCalc.

Хмаро зорієнтовані електронні довідники з основ математичної інформатики доцільно будувати на основі робочих аркушів Sage, що надають можливість поєднувати подання теоретичних матеріалів із практичними прикладами. На рис. 2.15 подано фрагмент довідкової системи Sage, що містить обчислювальні елементи.

Таким чином, *хмаро зорієнтовані навчально-методичні матеріали* можуть бути створені за допомогою будь-якого доцільного засобу хмарних технологій, проте, урахувавши, що матеріали для основ математичної

інформатики повинні містити відповідну обчислювальну (програмну) складову, для їх створення доцільно використовувати спеціалізовані засоби хмарних технологій, насамперед – *хмаро зорієнтовані предметні середовища*: хмаро зорієнтовані системи комп'ютерної математики та хмаро зорієнтовані середовища програмування.

The screenshot shows the CoCalc web interface with a SageMath notebook. The notebook contains code for a Caesar cipher. The left pane shows the source code, and the right pane shows the rendered output. The code defines a CaesarEncrypt function and a CaesarDecrypt function. The output shows the results of calling these functions with specific keys and messages.

```

58     #якщо вона належить алфавіту
59     if is_alphabetic_char(ciphertext[j]):
60         #виконати її циклічний зсув ліворуч на ключ
61         k
62         p = num_to_char((
63             char_to_num(ciphertext[j]) - k) % 26)
64         #інакше
65         else:
66             #залишити літеру без змін
67             p = ciphertext[j]
68             #додати перетворену літеру до відкритого
69             #тексту
70             plaintext += p
71             #повернути результат
72             return plaintext
73 #тестування шифрування
74 print CaesarEncrypt(7, "markova@mathinfo.ccjournals.eu")
75 #тестування дешифрування
76 print CaesarDecrypt(7, "thyrvch@thaopumv.jjqvbyuhsz.lb")
77 \end{sageblock}
78
79 Виклик \textbf{CaesarEncrypt} повертає результат:\\
80 $\sage{CaesarEncrypt(7, 'markova@mathinfo.ccjournals.eu')}
81 }$
82
83 Виклик \textbf{CaesarDecrypt} повертає результат:\\
84 $\sage{CaesarDecrypt(7, 'thyrvch@thaopumv.jjqvbyuhsz.lb')}
85 }$
86 \end{document}

```

```

#залишити літеру без змін
p = ciphertext[j]
#додати перетворену літеру до відкритого
plaintext += p
#повернути результат
return plaintext

#тестування шифрування
print CaesarEncrypt(7, "markova@mathinfo.ccjournals.eu")

#тестування дешифрування
print CaesarDecrypt(7, "thyrvch@thaopumv.jjqvbyuhsz.lb")

Виклик CaesarEncrypt повертає результат:
thyrvch@thaopumv.jjqvbyuhsz.lb
Виклик CaesarDecrypt повертає результат:
markova@mathinfo.ccjournals.eu

```

Рис. 2.14. Розробка хмаро зорієнтованих навчальних посібників у CoCalc

Як хмаро зорієнтоване предметне середовище, CoCalc у своєму складі містить як системи комп'ютерної математики, так й середовища програмування. Вибір конкретного інструментального засобу здійснюється через прив'язку до типу файлу або командою вибору середовища програмування. На етапі створення нових файлів в домашній директорії проекту користувачу надається можливість обрати мову програмування (рис. 2.16). Відповідно до здійсненого вибору завантажується середовище із вбудованим компілятором (інтерпретатором).

CoCalc підтримує редагування файлів:

– мовами C, C#, C++, CoffeeScript, Clojure, Sql, Eiffel, Ecl, Elm, Fortran, Go,

Haskell, Java, Julia, JavaScript, Lua, Ocaml, PHP, Perl, Python, Ruby, Scala, Scheme, TypeScript;

– систем комп'ютерної математики Pari, Octave, R, Sage.

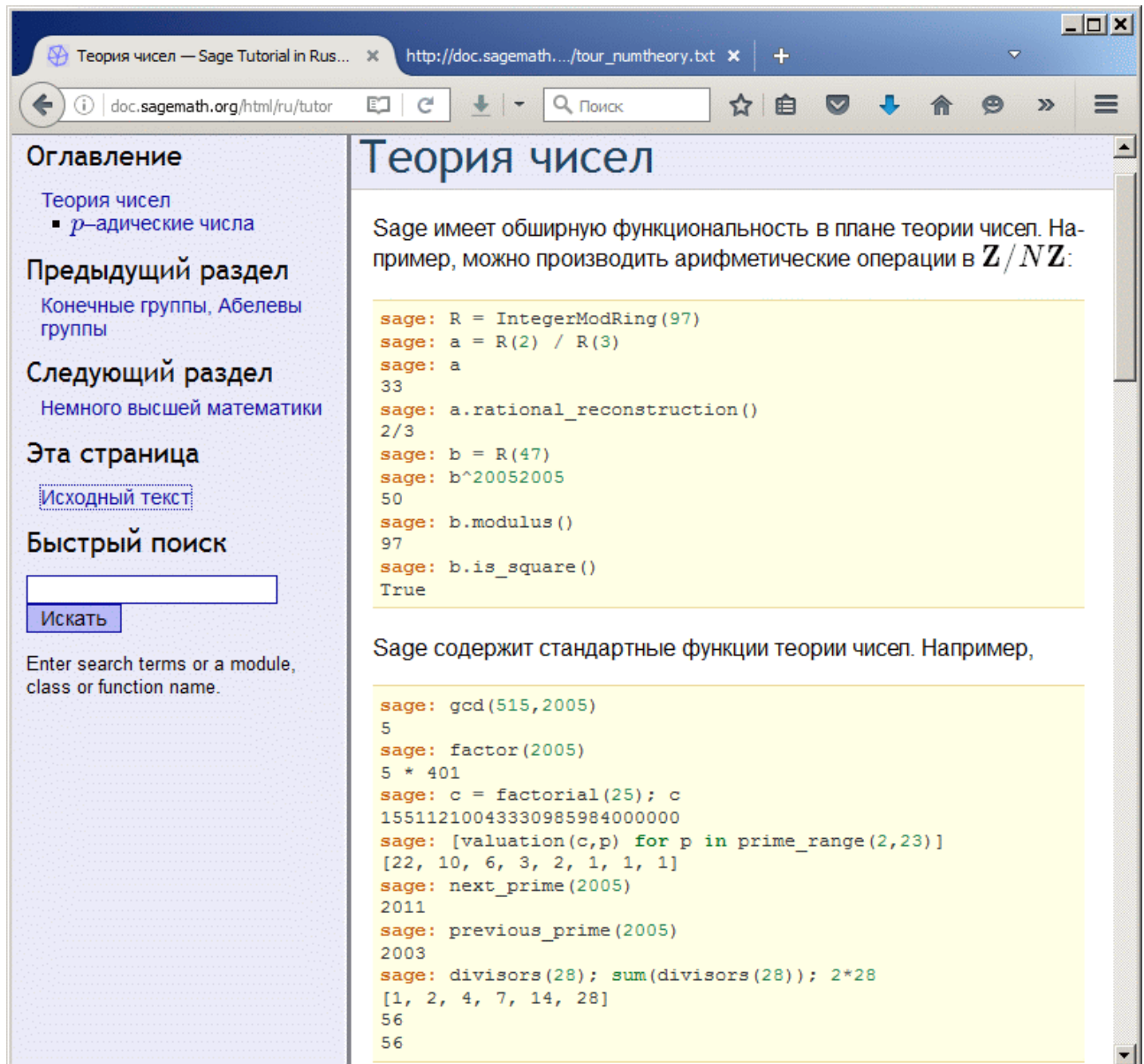


Рис. 2.15. Хмаро зорієнтований електронний довідник як частина довідкової системи CoCalc

Найпростіший спосіб опрацювання файлів, редагування яких відбувалось у середовищі CoCalc – робота з ними у режимі терміналу Linux. Так, на початку вивчення змістового модуля «Основи криптографії» студентам можна запропонувати завдання: описати мовою C++ програму, що виконує кодування заданого у командному рядку файлу кодом Цезаря (рис. 2.17). До інших

параметрів програми відносяться ім'я результуючого файлу та сам код.

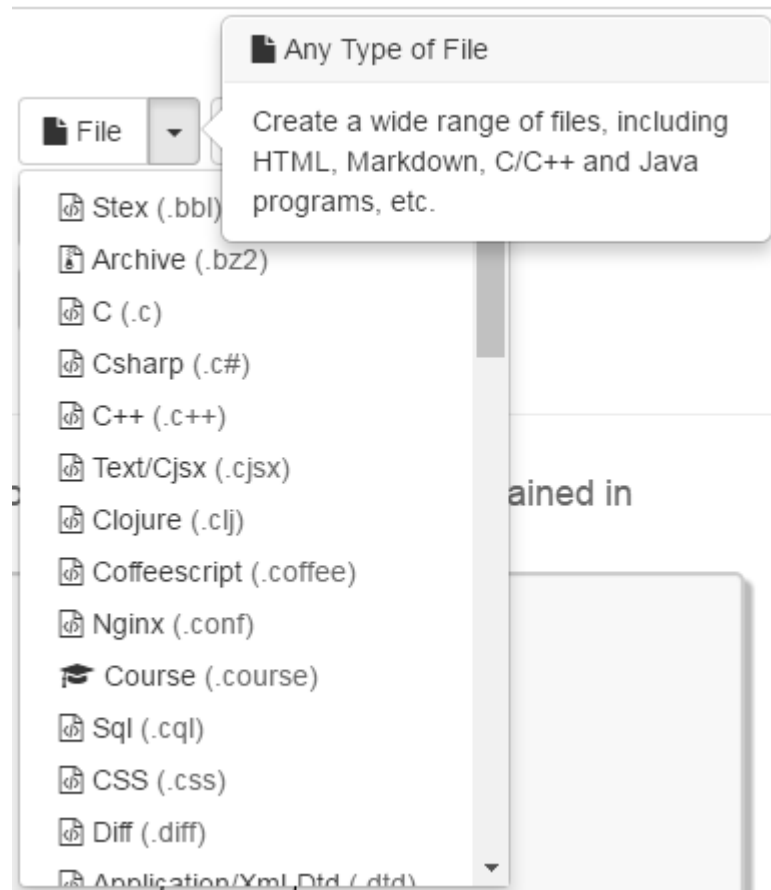


Рис. 2.16. Вибір типу файлу в домашній директорії проекту

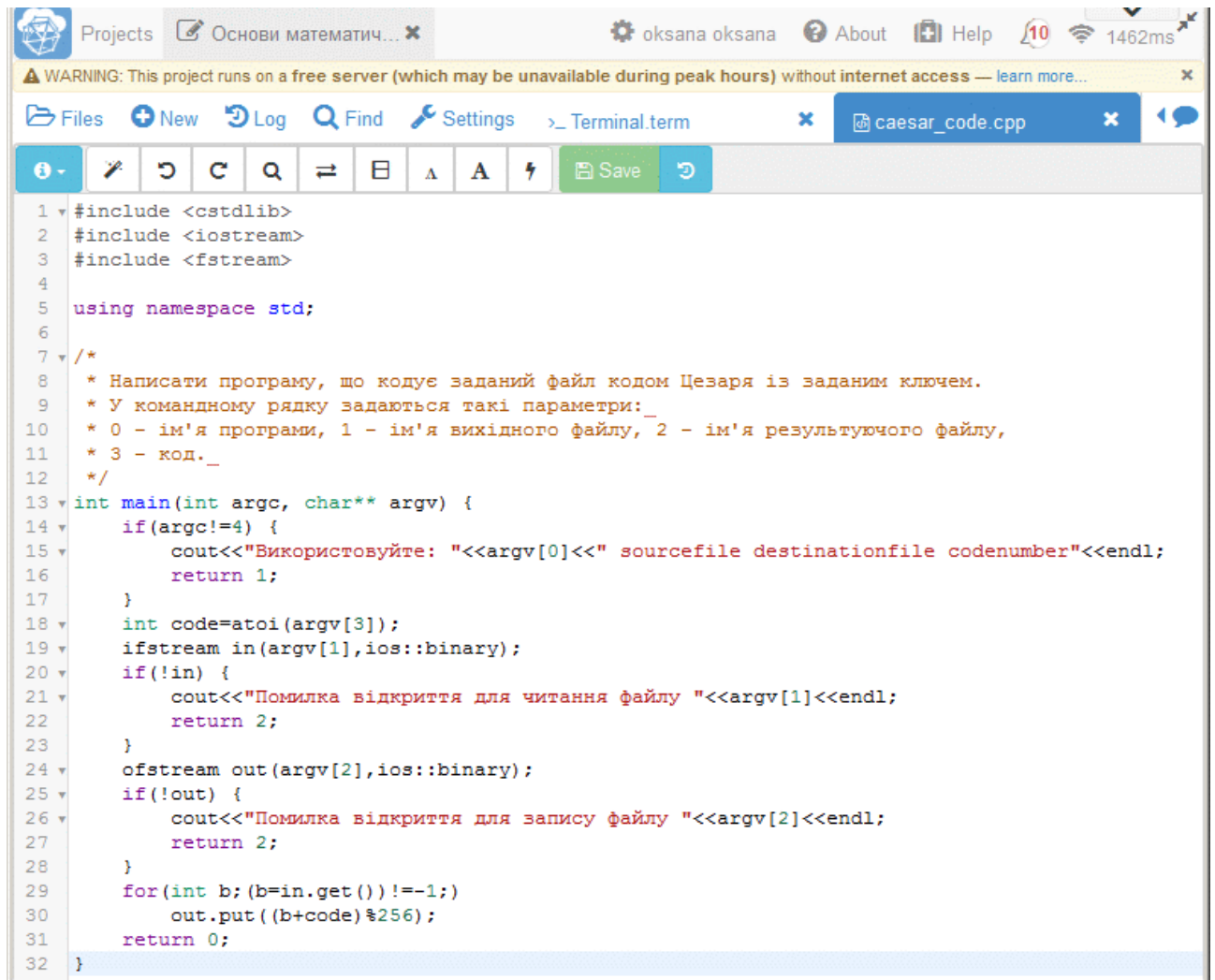
Для тестування розробленої програми її необхідно скомпілювати та виконати. Файли, створені у результаті виконання програми, стають складовою студентського проекту у середовищі CoCalc.

На рис. 2.18 показано термінальний сеанс із тестування розробленої програми. Тут доцільно звернути увагу студентів, що програма для кодування одночасно є й програмою для декодування: ключі кодування та декодування відрізняються лише знаком. Ураховуючи, що використовується ділення за модулем 256, еквівалентним ключу декодування -12 є ключі 244 ($-12+256$), 500 ($-12+512$) тощо.

Інший спосіб виконання програм у середовищі CoCalc – безпосередньо на робочих аркушах Sage. Для цього на початку комірки необхідно вказати одну із так званих «магічних команд» IPython (**%magics** надає повний їх перелік):

%auto – при вказанні на початку комірки виконує її автоматичне

обчислення при першому запуску Sage. Вказання `%auto` надає можливість ініціалізувати функції, змінні, інтерактивні елементи тощо при завантаженні робочого листа.



```

1 #include <cstdlib>
2 #include <iostream>
3 #include <fstream>
4
5 using namespace std;
6
7 /*
8  * Написати програму, що кодує заданий файл кодом Цезаря із заданим ключем.
9  * У командному рядку задаються такі параметри:
10 * 0 - ім'я програми, 1 - ім'я вихідного файлу, 2 - ім'я результуючого файлу,
11 * 3 - код.
12 */
13 int main(int argc, char** argv) {
14     if(argc!=4) {
15         cout<<"Використовуйте: "<<argv[0]<<" sourcefile destinationfile codenumber"<<endl;
16         return 1;
17     }
18     int code=atoi(argv[3]);
19     ifstream in(argv[1],ios::binary);
20     if(!in) {
21         cout<<"Помилка відкриття для читання файлу "<<argv[1]<<endl;
22         return 2;
23     }
24     ofstream out(argv[2],ios::binary);
25     if(!out) {
26         cout<<"Помилка відкриття для запису файлу "<<argv[2]<<endl;
27         return 2;
28     }
29     for(int b;(b=in.get())!=-1;)
30         out.put((b+code)%256);
31     return 0;
32 }

```

Рис. 2.17. Вікно редагування програми мовою C++ у середовищі CoCalc

`%axiom` – виконує команди Ахіом – системи комп'ютерної математики, що складається з інтерпретатора, компілятора та бібліотеки строго типізованих математичних типів. Ахіом – пізня назва системи, відомої раніше як Scratchpad. Розробка системи була розпочата у 1965 році для внутрішніх потреб фірми ІВМ як дослідницька платформа для розробки нових ідей в галузі обчислювальної математики. Друга версія Ахіом (під назвою Scratchpad II) розроблялась з 1977 року. Зміна назви на Ахіом, виконана у 1990 році, була пов'язана з комерціалізацією системи. У 2001 році Ахіом стала вільно поширюваним

є складовою більш загальної проблеми доведення правильності програм).

Мета роботи – довести, що реалізація алгоритму Евкліда знаходження НСД у Ахіом є правильною. Для цього необхідно показати, що для НСД(a, b): НСД є функцією від двох змінних (a, b), що повертає третю змінну c ; a, b є елементами відповідного типу; результат c також правильного типу; НСД завершується знаходженням найбільшого спільного дільника.

НСД у Ахіом є категоріально визначеною над будь-яким евклідовим кільцем – областю цілісності R , для якої визначена евклідова функція (евклідова норма) $d: R \rightarrow \mathbb{N} \cup \{-\infty\}$, причому $d(a) = -\infty \Leftrightarrow a = 0$, і можливе ділення з остачею, за нормою меншою ніж дільник, тобто для будь-яких $a, b \in R, b \neq 0$ є подання $a = bq + r$, для якого $d(r) < d(b)$.

Позначимо $D(a, b)$ множину всіх спільних дільників a та b . Тоді $\text{НСД}(a, b) = \max D(a, b)$.

Тип даних `NonNegativeInteger` успадковує функцію `gcd` з `Integer`, який, у свою чергу, є нащадком `EuclideanDomain`. У останньому міститься реалізація алгоритму Евкліда знаходження НСД:

```
gcd(x, y) ==                                --Алгоритм Евкліда
  x:=unitCanonical x
  y:=unitCanonical y
  while not zero? y repeat
    (x, y) := (y, x rem y)
    y:=unitCanonical y                    -- не впливає на коректність алгоритму, але
                                           -- може покращити продуктивність та гарантує,
                                           -- що gcd(x, y)=gcd(y, x) при
                                           -- canonicalUnitNormal
  x
```

Функція `unitCanonical` належить категорії `IntegralDomain`.

За допомогою `Coq` доводиться наступна функція знаходження НСД:

```
Fixpoint gcd a b :=
  match a with
  | 0 => b
  | S a' => gcd (b mod (S a')) (S a')
  end.
```

Вона транслюється у мову розширень Ахіом – SPAD:

```
GCD(x:NNI, y:NNI) :NNI ==
  zero? x => y
  GCD(y rem x, x)
```

Для доведення правильності алгоритму НСД визначимо:

– ділення: $t/a \Leftrightarrow \exists u(t \cdot u = a)$

– НСД: $\text{НСД}(a, b) = \forall t \max(t/a \wedge t/b)$

– теорему: $(t/a \wedge t/b) \Leftrightarrow t/(a-b) \wedge t/b$

– алгоритм Евкліда: $a > b \Rightarrow \text{НСД}(a, b) = \text{НСД}(a-b, b)$

За визначенням НСД необхідно показати, що

$$\forall t \max(t/a \wedge t/b) = \forall t \max(t/(a-b) \wedge t/b)$$

Це вимагає доведення теореми про те, що $(t/a \wedge t/b) \Leftrightarrow (t/(a-b) \wedge t/b)$.

Нехай t – довільне фіксоване значення. Припустимо, що

$$(t/a \wedge t/b) \tag{2.1}$$

та покажемо, що

$$t/(a-b) \tag{2.2}$$

та

$$t/b \tag{2.3}$$

(2.3) випливає із припущення, а для доведення (2.2), за визначенням ділення, необхідно знайти таке w , що

$$t \cdot w = a - b \tag{2.4}$$

З (2.1), за визначенням ділення, знайдемо такі u та v , що $t \cdot u = a$ та $t \cdot v = b$. Звідси маємо: $a - b = t \cdot u - t \cdot v$. Але $t \cdot u - t \cdot v = t \cdot (u - v)$, тому необхідно знайти $w = u - v$, тобто знайти таке w , що $t \cdot u - t \cdot v = t \cdot w$.

Автор Coq Т. Кокан (Thierry Coquand) побудував його на основі числення конструкцій, яке можна розглядати в якості розширення Каррі-Говардовського ізоморфізму. Основним компонентом числення конструкцій є терми, які будуються за такими правилами:

– **T** є термом (також зветься *Type*);

– **P** є термом (також зветься *Prop*, тип усіх суджень);

– змінні (x, y, \dots) є термами;

– якщо **A** і **B** є термами, то термами також є (A, B) , $(\lambda x : A, B)$, $(\forall x : A, B)$.

Числення конструкцій складаються з п'яти видів об'єктів:

- 1) доведення, які є термами типів, що є твердженнями;
- 2) твердження, відомі також як малі типи;
- 3) предикати, які є функціями, що повертають твердження;
- 4) великі типи, які є типами предикатів. \mathbf{P} є прикладом великого типу;
- 5) сам \mathbf{T} є типом великих типів.

Числення конструкцій надає можливість доводити типізовані судження:

$$x_1 : A_1; x_2 : A_2 \dots \vdash t : B$$

які можуть бути прочитані як імплікація «якщо змінні x_1, x_2, \dots мають типи A_1, A_2, \dots , то терм t має тип B ».

Допустимі судження для числення конструкцій виводяться з набору правил умовиводів. Надалі ми використовуватимемо Γ для позначення послідовності задання типів $x_1 : A_1; x_2 : A_2, \dots$ і \mathbf{K} для позначення або \mathbf{P} , або \mathbf{T} . Запис $A : B : C$ означатимемо « A має тип B , і B має тип C ». Запис $B (x := N)$ означатимемо результат підстановки терму N для змінної x у термі B . Правило виведення записується у вигляді:

$$\frac{\Gamma \vdash A : B}{\Gamma' \vdash C : D}$$

що означає: якщо $\Gamma \vdash A : B$ є правильним судженням, то правильним є й $\Gamma' \vdash C : D$.

Правила умовиводу для числення конструкцій:

- 1) $\frac{}{\Gamma \vdash P : T}$;
- 2) $\frac{\Gamma \vdash A : K}{\Gamma, x : A \vdash x : A}$;
- 3) $\frac{\Gamma, x : A \vdash t : B : K}{\Gamma \vdash (\lambda x : A. t) : (\forall x : A. B) : K}$;
- 4) $\frac{\Gamma \vdash M : (\forall x : A. B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B(x := N)}_A$;
- 5) $\frac{\Gamma \vdash M : A \quad A =_\beta B \quad B : K}{\Gamma \vdash M : B}$.

Визначення логічних операторів:

$$A \Rightarrow B \equiv \forall x : A. B \quad (x \notin B)$$

$$A \wedge B \equiv \forall C : P. (A \Rightarrow B \Rightarrow C) \Rightarrow C$$

$$A \vee B \equiv \forall C : P. (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow C$$

$$\neg A \equiv \forall C : P. (A \Rightarrow C)$$

$$\exists x : A. B \equiv \forall C : P. (\forall x : A. (B \Rightarrow C)) \Rightarrow C$$

Основні типи даних, що використовуються в інформатиці, можуть бути визначені в численні конструкцій у такий спосіб:

а) логічний тип: $\forall A : P. A \Rightarrow A \Rightarrow A$;

б) натуральні числа: $\forall A : P. (A \Rightarrow A) \Rightarrow (A \Rightarrow A)$;

в) добуток $A \times B$: $A \wedge B$;

г) диз'юнктивне об'єднання $A + B$: $A \vee B$.

Повний код Coq для доведення правильності алгоритму Евкліда у системі Ахіом подано у додатку В. Зазначимо, що інтеграція Coq у Ахіом стала можливою лише тому, що Ахіом є строго типізованою системою.

%capture – перехоплення або ігнорування виведення результату шляхом розміщення `%capture` або `%capture` (опціональні аргументи) на початку комірки або на початку рядка. При використанні лише `%capture` стандартні потоки виведення та виведення помилок повністю ігноруються. При вказанні `%capture` з аргументами стандартні потоки виведення та виведення помилок перенаправляються у змінну або файл. Наприклад, при вказанні

```
%capture(stdout='output', stderr=open('error','w'), append=True, echo=True)
```

стандартний потік виведення `stdout` додається (оскільки `append=True`) до глобальної змінної `output`, стандартний потік помилок `stderr` записує у файл `'error'`, а результати обчислення, як і раніше, відображається у CoCalc (`echo=True`).

%coffeescript виконує код мовою CoffeeScript, записується `coffeescript(s=None, once=False)`. Наприклад:

```
%coffeescript console.log 'hi'
```

або

```
coffeescript("console.log 'hi'")
```

Якщо встановити `once=False`, то код буде виконуватися щоразу при завантаженні аркуша, як і з `%auto`.

При виконанні коду у CoCalc додатково визначається функція друку `print`. Код мовою CoffeeScript транслюється у JavaScript та виконується безпосередньо у браузері, тому продуктивність програм, описаних даною мовою, не залежить від обчислювальної потужності хмарного серверу.

У спецкурсі з основ математичної інформатики з використанням даної мови можуть бути розглянуті такі обчислювально ємні задачі, як створення та налаштування нейронної мережі. Враховуючи значний час, необхідний для цього, та важливість теми «Нейронні мережі та розпізнавання образів» для спецкурсу в цілому (як такої, що поєднує обчислювальну та інтелектуальну змістові лінії), студентам пропонується індивідуальне науково-дослідне завдання – розробка штучної нейронної мережі.

Штучна нейронна мережа – це математична модель, а також її програмна та апаратна реалізація, побудовані за принципом функціонування біологічних нейронних мереж – мереж нервових клітин живого організму. Це поняття виникло при вивченні процесів, які відбуваються в мозку, та при намаганні змодельовати ці процеси. Після розробки алгоритмів навчання отримувані моделі стали використовуватися в практичних цілях: в задачах прогнозування, для розпізнавання образів, в задачах керування тощо.

Штучні нейронні мережі являють собою систему з'єднаних між собою за допомогою синапсів штучних нейронів, які взаємодіють. На вхід штучного нейрона поступає деяка множина сигналів, кожний з яких є виходом іншого нейрона. Кожний вхід множиться на ваговий коефіцієнт синапса, всі доданки додаються, визначаючи рівень активації нейрона як скалярний добуток вектору входу на вектор вагових коефіцієнтів. Отримане значення подається на вхід функції активації, яка нормує значення у певному діапазоні: для полярної функції активації – $[0; 1]$, для біполярної – $[-1; +1]$.

Найчастіше використовуються тришарові нейронні мережі, загальна

архітектура яких подана на рис. 2.19.

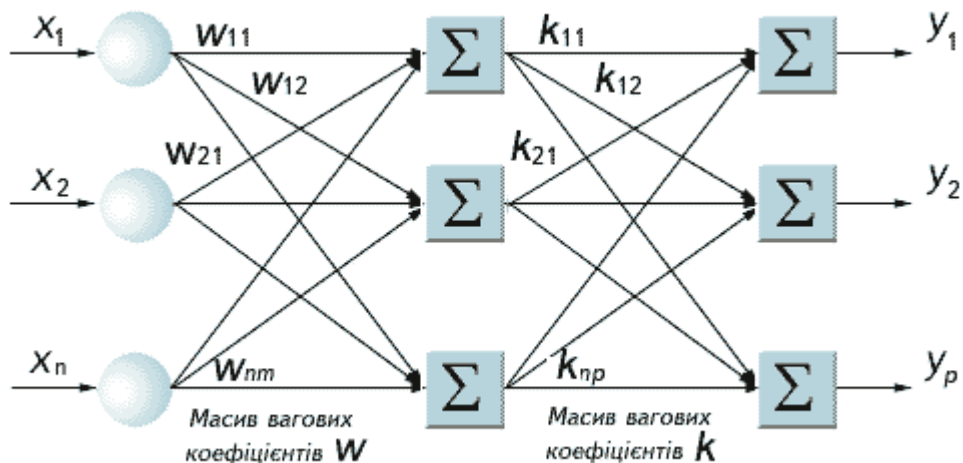


Рис. 2.19. Архітектура тришарової нейронної мережі (за Ф. Уоссерменом (Philip D. Wasserman) [246])

Для розробки нейронної мережі пропонуємо студентам наступний код мовою CoffeeScript:

```
%coffeescript

#Штучна нейронна мережа (на основі коду Phillip Wang -
#https://github.com/lucidrains/coffee-neural-network)

class Synapse #синапс - з'єднує два нейрони
  constructor: (@source_neuron, @dest_neuron)->
    #початкова вага - випадкове значення в межах [-1;+1]
    @weight = @prev_weight = Math.random() * 2 - 1

class TanhGate #тангенціальна функція активації
  calculate: (activation)->
    math.tanh(activation)
  derivative: (output)-> #похідна функції активації
    1 - output * output

class SigmoidGate #сигмоїдальна функція активації
  calculate: (activation)->
    1.0 / (1.0 + Math.exp(-activation))
  derivative: (output)-> #похідна функції активації
    output * (1 - output)

class ReluGate #ступінчаста функція активації
  @LEAKY_CONSTANT = 0.01
  calculate: (activation)->
    if activation < 0 then activation * ReluGate.LEAKY_CONSTANT else activation
  derivative: (output)->
    if output > 0 then 1 else ReluGate.LEAKY_CONSTANT
```

Перша функція активації відповідає гіперболічному тангенсу та є

біполярною, друга – логістичній функції та є полярною. Остання функція активації описує ступінчасту функцію і також відноситься до полярних.

```
class Neuron #штучний нейрон
  @LEARNING_RATE = 0.1 #сталі, що задають швидкість навчання
  @MOMENTUM = 0.05

  constructor: (opts={})->
    gate_class = opts.gate_class || SigmoidGate
    @prev_threshold = @threshold = Math.random() * 2 - 1
    @synapses_in = []
    @synapses_out = []
    @dropped = false
    @output = 0.0
    @error = 0.0
    @gate = new gate_class()

  dropout: ->
    @dropped = true
    @output = 0

  calculate_output: -> #обчислення реакції нейрону
    @dropped = false
    activation = 0
    for s in @synapses_in
      activation += s.weight * s.source_neuron.output
    activation -= @threshold
    @output = @gate.calculate(activation)#

  derivative: ->
    @gate.derivative @output

  output_train: (rate, target)-> #обчислення вагових коефіцієнтів вихідного шару
    @error = (target - @output) * @derivative()
    @update_weights(rate)

  hidden_train: (rate)-> #обчислення вагових коефіцієнтів прихованого шару
    @error = 0.0
    for synapse in @synapses_out
      @error += synapse.prev_weight * synapse.dest_neuron.error
    @error *= @derivative()
    @update_weights(rate)

  update_weights: (rate)->
    for synapse in @synapses_in
      temp_weight = synapse.weight
      synapse.weight += (rate * Neuron.LEARNING_RATE * @error *
        synapse.source_neuron.output) + (Neuron.MOMENTUM * (synapse.weight -
        synapse.prev_weight))
      synapse.prev_weight = temp_weight
    temp_threshold = @threshold
    @threshold += (rate * Neuron.LEARNING_RATE * @error * -1) +
      (Neuron.MOMENTUM * (@threshold - @prev_threshold))
    @prev_threshold = temp_threshold
```

Навчання мережі полягає в знаходженні коефіцієнтів зв'язку між нейронами. У процесі навчання нейронна мережа здатна виявляти складні

залежності між вхідними та вихідними даними, а також здійснювати узагальнення. Це означає, що в разі успішного навчання мережа зможе повернути правильний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних та/або «зашумлених», частково спотворених даних.

```
class NeuralNetwork #нейронна мережа
  @DROPOUT = 0.3

  #параметри конструктору - тип функції активації, кількість нейронів вхідного
  #шару, кількість нейронів прихованих шарів, кількість нейронів вихідного шару
  constructor: (gate_class, input, hidden..., output)->
    opts = {gate_class}
    @input_layer = (new Neuron(opts) for i in [0...input])
    @hidden_layers = for hidden in hidden
      (new Neuron(opts) for i in [0...hidden])
    @output_layer = (new Neuron(opts) for i in [0...output])
    for i in @input_layer
      for h in @hidden_layers[0]
        synapse = new Synapse(i, h)
        i.synapses_out.push synapse
        h.synapses_in.push synapse
    for layer, ind in @hidden_layers
      next_layer = if ind==( @hidden_layers.length-1)
        @output_layer
      else
        @hidden_layers[ind+1]
      for h in layer
        for o in next_layer
          synapse = new Synapse(h, o)
          h.synapses_out.push synapse
          o.synapses_in.push synapse

  train: (input, output)-> #навчання нейронної мережі
    @feed_forward(input)
    for neuron, ind in @output_layer
      neuron.output_train 0.5, output[ind]
    for layer in @hidden_layers by -1
      for neuron in layer
        neuron.hidden_train 0.5

  feed_forward: (input)-> #проходження вхідного сигналу через всі шари мережі
    for n, ind in @input_layer
      n.output = input[ind]
    for layer in @hidden_layers
      for n in layer
        if Math.random() < NeuralNetwork.DROPOUT
          n.dropout()
        else
          n.calculate_output()
    for n in @output_layer
      n.calculate_output()

  current_outputs: -> #результат обчислення на вихідному шарі
```

```

(n.output for n in @output_layer)

#функція клонування даних
clone = (obj) ->
  return obj if obj is null or typeof (obj) isnt "object"
  temp = new obj.constructor()
  for key of obj
    temp[key] = clone(obj[key])
  temp

```

У якості прикладу пропонуємо спочатку розглянути нейронну мережу для логічної функції двох змінних «АБО». Особливістю даного прикладу є те, що на вхід мережі подаються лише два полярні значення – 0 та 1, на виході отримуємо також одне із двох значень.

За наслідком із теореми Колмогорова, для того, щоб тришарова нейронна мережа відтворювала будь-яку функцію багатьох змінних, розмірність прихованого шару повинна бути щонайменше на 1 більше за подвоєну розмірність входу. Для даного прикладу можна редукувати розмірність прихованого шару з 5 до 2 – це пов'язано із тим, що на виході можливі лише два значення:

```

#Створення тришарової нейронної мережі:
#2 нейрони на вході, 2 - на виході та 1 - на прихованому шарі
nn = new NeuralNetwork(SigmoidGate, 2, 2, 1)

#навчальна послідовність складається із пар "вхід - вихід"
pairs = [
  #наприклад, для диз'юнкції
  [[0,0], [0]],
  [[0,1], [1]],
  [[1,0], [1]],
  [[1,1], [1]]
]

#Навчання виконуватимемо до досягнення граничної кількості ітерацій
numiter = 150000

for i in [0...numiter]
  err=0
  for pair in pairs
    nn.train pair[0], pair[1]
    nn.feed_forward pair[0]
    out=nn.current_outputs()
    for k in [0...pair[1].length]
      err+=(out[k]-pair[1][k])*(out[k]-pair[1][k])
  err=Math.sqrt(err/4)
  if i%1000==0
    print "Ітерація ", i, ", помилка = ", err

#Тестування мережі #1
for i in pairs

```

```

nn.feed_forward i[0]
print "Подаємо на вхід ", i[0], ", отримали ", nn.current_outputs(),
", повинно бути ", i[1]

```

Результати тестування нейронної мережі подано на рис. 2.20.

```

...
Итерація 140000 , помилка = 0.00421791796837043
Итерація 141000 , помилка = 0.026060135533211935
Итерація 142000 , помилка = 0.01662893256689042
Итерація 143000 , помилка = 0.28026623651875215
Итерація 144000 , помилка = 0.0010612101620747125
Итерація 145000 , помилка = 0.2743559388471105
Итерація 146000 , помилка = 0.02865800216671595
Итерація 147000 , помилка = 0.027756609296225383
Итерація 148000 , помилка = 0.0009493239126797038
Итерація 149000 , помилка = 0.37406581007087847
Подаємо на вхід [0,0] , отримали [0.0010199996254682498] , повинно бути [0]
Подаємо на вхід [0,1] , отримали [0.9572457742977656] , повинно бути [1]
Подаємо на вхід [1,0] , отримали [0.9999319400884712] , повинно бути [1]
Подаємо на вхід [1,1] , отримали [0.46649109350708023] , повинно бути [1]

```

Рис. 2.20. Тестування нейронної мережі для логічної функції «АБО»

З рис. 2.20 можна побачити, що для навчання мережі обрана кількість ітерацій (150000) є зовеликою – так, обчислювальний процес можна було б зупинити при зменшенні помилки до наперед визначеного значення суттєво раніше. Слід звернути увагу на результати обчислення, коли на вхід мережі подається пара (1; 1) – на відміну від попередніх трьох тестів, отримуємо значення, що суттєво відрізняється від еталонного. Для усунення даної помилки пропонуємо обрати іншу полярну функцію активації – ступінчасту.

Наступний приклад показує, як побудувати нейронну мережу для довільних значень входу та виходу:

```

#Створення тришарової нейронної мережі:
#3 нейрони на вході, 7 - на виході та 1 - на прихованому шарі
nn = new NeuralNetwork(SigmoidGate, 3, 7, 1)

#згенеруємо нову навчальну послідовність
data = []
count = 1000 #кількість пар "вхід - вихід"

for i in [0...count]
  x1 = Math.random()*100-60
  x2 = Math.random()*100-40
  x3 = Math.random()*100-50
  data.push([[x1,x2,x3], [x1 + x2 - x3]])

```

Архітектура мережі відповідає теоремі Колмогорова, але значення, на

яких передбачається її навчання, не є полярними, як це вимагається логістичною функцією. Приведення їх до потрібного діапазону вимагає нормування, для чого необхідно знайти граничні значення для входу та виходу:

```
#пошук граничних значень для входу та виходу
mininput = data[0][0][0]
maxinput = data[0][0][0]
minoutput = data[0][1][0]
maxoutput = data[0][1][0]

for i in [0...count]
  for j in [0...data[i][0].length]
    if data[i][0][j]<mininput
      mininput=data[i][0][j]
    if data[i][0][j]>maxinput
      maxinput=data[i][0][j]
  for j in [0...data[i][1].length]
    if data[i][1][j]<minoutput
      minoutput=data[i][1][j]
    if data[i][1][j]>maxoutput
      maxoutput=data[i][1][j]
```

Навчання мережі виконуємо на нормалізованих даних:

```
#навчальна послідовність, нормована в межах [0; 1]
normdata=clone(data)

for i in [0...count]
  for j in [0...data[i][0].length]
    normdata[i][0][j]=(data[i][0][j]-mininput)/(maxinput-mininput)
  for j in [0...data[i][1].length]
    normdata[i][1][j]=(data[i][1][j]-minoutput)/(maxoutput-minoutput)

#Навчання виконуватимемо до досягнення граничної кількості ітерацій
numiter = 100000

for i in [0...numiter]
  err=0
  for pair in normdata
    nn.train pair[0], pair[1]
    nn.feed_forward pair[0]
    out=nn.current_outputs()
    for k in [0...pair[1].length]
      err+=(out[k]-pair[1][k])*(out[k]-pair[1][k])
  err=Math.sqrt(err/4)
  if i%1000==0
    print "Ітерація ", i, ", помилка = ", err
```

При тестуванні мережі слід виконати зворотну процедуру денормалізації:

```
#Тестування мережі #2
for i in [0...count]
  nn.feed_forward normdata[i][0]
  res=nn.current_outputs()
  print "Подаємо на вхід ", data[i][0], ", отримали ",
    res[0]*(maxoutput-minoutput)+minoutput, ", повинно бути ", data[i][1][0]
```


На рис. 2.21 подано результати тестування.

```

Итерация 96000 , помилка = 1.9951509319586445
Итерация 97000 , помилка = 2.0141378667422836
Итерация 98000 , помилка = 1.9155861398659084
Итерация 99000 , помилка = 1.9405192614965148
Подаємо на вхід [27.699390635066337,59.41341761361207,49.259104071956685] , отримали 36.35087315082319 , повинно
бути 37.85370417672172
Подаємо на вхід [39.00633118239884,-16.21662884665037,-41.22289806481234] , отримали -9.172790623682289 , повинно
бути 64.01260040056081
Подаємо на вхід [27.747141959298915,-26.331914098465703,46.00194152219592] , отримали -5.272461110228235 , повинно
бути -44.58671366136271
Подаємо на вхід [-23.11429081653037,-24.202363779809698,-0.16588587495485285] , отримали -30.680435299350293 ,
повинно бути -47.15076872138522
Подаємо на вхід [-11.79389853611631,23.72835228983797,-19.9927199117171] , отримали -3.1010809642714605 , повинно
бути 31.92717366543336
Подаємо на вхід [27.519457157784757,-0.8885653362011041,35.02090493243577] , отримали 37.646162889115914 , повинно
бути -8.390013110852117
Подаємо на вхід [5.519824002791353,25.860920751786693,35.38865410980026] , отримали 21.525788294005025 , повинно
бути -4.007909355222105
Подаємо на вхід [-59.39481245405906,31.465997970337654,46.95086766798531] , отримали -26.363806615506718 , повинно
бути -74.8796821517067
Подаємо на вхід [16.04025015123746,-3.9688678307271914,3.623055486515561] , отримали 15.30852052831304 , повинно
бути 8.448326833994706
Подаємо на вхід [11.559482975934174,38.75477763182535,-41.237508936767384] , отримали 69.23303789586686 , повинно
бути 91.5517695445269
Подаємо на вхід [-1.8505337686284378,48.565634513181465,21.90470182083193] , отримали 64.1378190908014 , повинно
бути 24.810398923721095
  
```

Рис. 2.21. Результати тестування нейронної мережі для функції додавання

При обговоренні результатів тестування знову доцільно звернути увагу на значення, що суттєво відрізняються від еталонних. Для цього попередньо пригадуємо, що вхідні значення генеруються випадково у наступних діапазонах: $x_1 \in [-60; 40)$, $x_2 \in [-40; 60)$, $x_3 \in [-50; 50)$. Аналіз результатів показує, що, чим ближче вхідні значення до границь діапазону, тим більше відхилення отриманого результату від еталонного. Це надає можливість зробити висновок про межі застосування побудованої мережі, в яких вона зберігає адекватність.

%command – надає можливість виконання «магічної» команди шляхом її уведення в якості параметра або створення власної «магічної» команди шляхом визначення функції, яка приймає в якості вхідних даних рядок та виводить рядок.

%cython – надає можливість використання компілятора C/C++, зокрема, з метою розробки модулів для інтерпретатора Python. Вказання даної «магічної» команди попередньо компілює код, що суттєво збільшує швидкість його виконання. В якості параметра можна вказати ім'я файлу, що

компілюється (обов'язково повинно завершуватись на 'рух'). Результат виконання даної «магічної» команди – файл динамічної бібліотеки (.so / .dll / .dylib), завантажуваний для виконання скомпільованого коду.

Для порівняння швидкості виконання коду у традиційному режимі Sage та режимі Cython доцільно запропонувати багаторазове виконання одного й того ж коду у різних режимах (рис. 2.22). Для цього доцільно використати ще одну «магічну» команду – %time.

```

1 %python
2 def is2pow(n):
3     while n != 0 and n%2 == 0:
4         n = n >> 1
5     return n == 1
6
7 %time [n for n in range(10^6) if is2pow(n)]

```

[1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288]

CPU time: 1.19 s, Wall time: 1.29 s

```

1 %cython
2 def is2powV2(n):
3     while n != 0 and n%2 == 0:
4         n = n >> 1
5     return n == 1

```

Defined is2powV2
Auto-generated code...

```

1 %time [n for n in range(10^6) if is2powV2(n)]

```

[1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288]

CPU time: 0.99 s, Wall time: 0.99 s

Рис. 2.22. Порівняння швидкості виконання коду в режимах Python та Cython

%default, %default_mode – встановлює режим за замовчуванням для обчислень у комірці. При виклику `default_mode()` повертає поточний режим. Наприклад, виклик `default_mode("gp")` встановлює у якості засобу для обчислень Pari/GP, а `default_mode("sage")` чи `default_mode("")` повертає у режим Sage. Еквівалентом першого виклику є

```
%default_mode gp
```

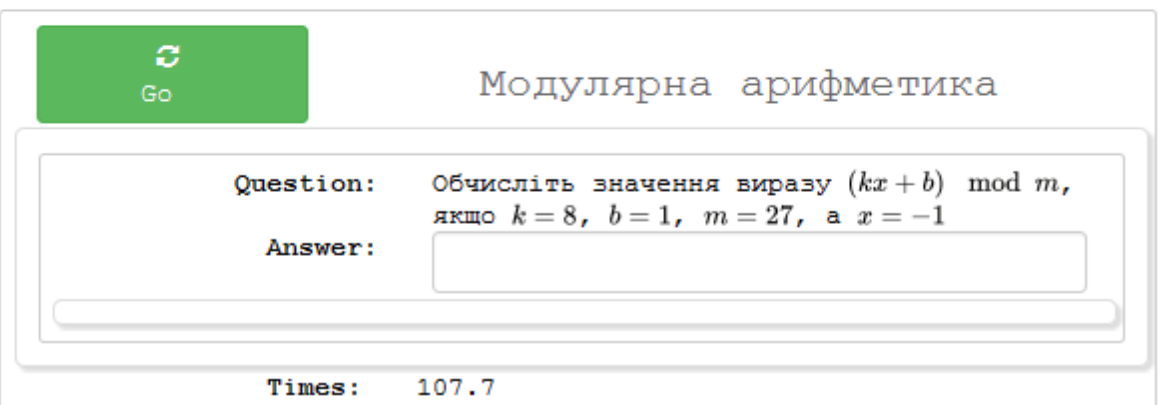
%exercise – вказує, що у комірці подано інтерактивний набір вправ (тестів) у вигляді коду Sage, що містить: а) змінні `question` (HTML-рядок, до якого включені обмежені знаками '\$' математичні вирази) та `answer`

(математичний об'єкт або пара виду (правильна_відповідь, елемент_управління)); б) необов'язкову функцію `check(answer)`, яка повертає логічне значення або пару кортеж (логічне_значення, повідомлення), де перший аргумент є істинним, якщо відповідь правильна, а необов'язковий другий аргумент є повідомленням, яке повинно бути відображене у відповідь на дану відповідь; в) `hints` – необов'язковий список рядків для відображення в заданій послідовності кожного разу, коли користувач вводить неправильну відповідь (останній рядок відображається багаторазово; якщо `hints` не задано, правильна відповідь відображається після трьох спроб). На рис. 2.23 подано приклад використання `%exercise` для підтримки теми 11 «Математичні основи теорії кодування» та результати його виконання.

```

1  %exercise
2  title = "Модулярна арифметика"
3  k = randint(1,10)
4  b = randint(-5,5)
5  x = randint(-3,2)
6  m = randint(17,31)
7  question = "Обчисліть значення виразу  $(kx+b) \pmod m$ , якщо  $k=%s$ ,  $b=%s$ ,  $m=%s$ , а  $x=%s$ "%(k, b, m, x)
8  answer = mod(k*x+b, m)
9  hints = ['mod - залишок', 'Результат - десь біля %s.'%(answer+randint(1,7)),
10 "Правильна відповідь - %s."%answer]
11 def check(attempt):
12     if Integer(attempt) == answer: return True
13     if Integer(attempt) < 0:
14         return False, "Остача від ділення не може бути від'ємною"
15     if Integer(attempt) > m:
16         return False, "Остача від ділення не може бути більше за дільник"
17 %md

```



Times: 107.7

Рис. 2.23. Приклад використання декоратора `%exercise`

`%fork` – виконує код у підпроцесі, що не блокується основним процесом.

Усі глобальні змінні, що змінюються у підпроцесі, змінюються у основному процесі після завершення підпроцесу. Для отримання відомостей про наявні підпроцеси можна використати функцію `fork.children()`, яка повертає словник `{pid:execute_uuid}`. Для завершення підпроцесу необхідно викликати `fork.kill(pid)`.

%fortran – компіляція коду мовою Fortran і налаштування його для подальшого використання. Fortran – одна із найстаріших мов програмування загального призначення: у 2017 році виповнюється 60 років із випуску її першої версії, а на 2018 рік заплановано випуск нової версії Fortran-2018, до якої включено розширену підтримку мови C. Класичний Fortran, за суттю, є також найстарішою мовою математичної інформатики – найбільша кількість якісних математичних бібліотек описані саме цією мовою.

Аналізуючи мови реалізації задач математичної інформатики, студентам доцільно розглянути приклад використання мови Fortran у середовищі CoCalc. Даний приклад надає можливість встановити наступність між історичними та сучасними способами кодування програм: так, 1-5 символи кожного рядка можуть містити мітки, 6 символ застосовується для позначення продовження рядка, 7-72 символи містять програмний код. Така структура була орієнтована на введення програми з перфокарт (рис. 2.24).

%fricas – надає інтерфейс до Fricas (відгалуження системи Axiom).

%gap – надає інтерфейс до GAP (з англ. Groups, Algorithms and Programming): системи комп'ютерної математики для обчислювальної дискретної алгебри з особливим акцентом на обчислювальну теорію груп. Поточна версія GAP – 4.8.6; для використання попередньої версії GAP слід застосувати декоратор **%gap3**.

Дану систему доцільно використовувати у процесі виконання другої лабораторної роботи «Комбінаторні та рекурсивні алгоритми». Для цього необхідно розглянути основні функції системи GAP, пов'язані з комбінаторикою.

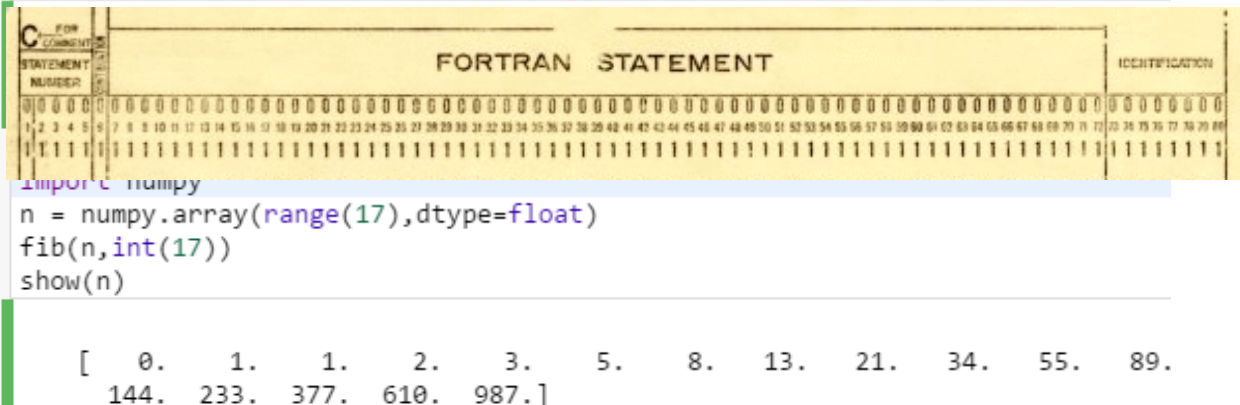
Традиційна функція для обчислення факторіалу натурального числа має

назву Factorial. Наприклад, обчислимо факторіал для цілих чисел від 0 до 20:

```
%gap
List ([0..10], Factorial);

[ 1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, 39916800,
  479001600, 6227020800, 87178291200, 1307674368000, 20922789888000,
  355687428096000, 6402373705728000, 121645100408832000,
  2432902008176640000 ]
```

```
1 %fortran
2 C FILE: FIB1.F
3   SUBROUTINE FIB(A,N)
4 C   CALCULATE FIRST N FIBONACCI NUMBERS
5   INTEGER N
6   REAL*8 A(N)
7   A(1) = 0.0D0
8   A(2) = 1.0D0
9   DO I=3,N
10      A(I) = A(I-1) + A(I-2)
11   ENDDO
12   END
13 C END FILE FIB1.F
```



```
1 import numpy
2 n = numpy.array(range(17),dtype=float)
3 fib(n,int(17))
4 show(n)
```

```
[ 0.  1.  1.  2.  3.  5.  8. 13. 21. 34. 55. 89.
 144. 233. 377. 610. 987.]
```

Рис. 2.24. Генерація ряду чисел Фібоначчі мовою Fortran

$n!$ можна інтерпретувати у такий спосіб. Нехай M – множина з n елементів. Будь-яке розташування елементів множини M в деякому певному порядку називається перестановкою з n елементів. Тоді кількість різних перестановок з n елементів дорівнює $n!$. Зауважимо, що в явному вигляді всі ці перестановки можна отримати за допомогою функції `PermutationsList`. Ця функція повертає множину всіх перестановок з елементів заданої множини (не обов'язково числової):

```
pl:=PermutationsList(['a','d','p','u']);

[ "adpu", "adup", "apdu", "apud", "audp", "aupd", "dapu", "daup", "dpau",
  "dpua", "duap", "dupa", "padu", "paud", "pdau", "pdua", "puad", "puda",
  "uadp", "uapd", "udap", "udpa", "upad", "upda" ]
```

Length (pl) = Factorial (4); # Кількість перестановок = 6 = 3!

true

За допомогою функції `PermutationList` можна також знаходити перестановки з повтореннями, які отримуються, якщо розглядати впорядковані набори з k елементів множини $M=\{s_1, \dots, s_n\}$, в яких елемент s_i міститься v_i разів і $v_1+\dots+v_n=k$ (тоді перестановки множини з n елементів отримуємо при $v_1=\dots=v_n=1$). Число $C_k(v_1, \dots, v_n)$ різних перестановок множини M з повтореннями дорівнює $k!/(v_1!\dots v_n!)$ і може бути отримано за допомогою функції `NrPermutationsList`:

```
NrPermutationsList(['a', 'd', 'p', 'u']);
```

24

Функція `Binomial(n, k)` повертає біноміальний коефіцієнт $C_n^k = n!/(k!(n-k)!)$, тобто коефіцієнт при x^k у многочлені $(x+1)^n$.

```
List ([0..4], k-> Binomial (4, k));
```

[1, 4, 6, 4, 1]

```
x = Indeterminate (Integers, "x");
```

```
(1 + x) ^ 4;
```

x ^ 4 + 4 * x ^ 3 + 6 * x ^ 2 + 4 * x + 1

```
CoefficientsOfUnivariatePolynomial ((1 + x) ^ 4);
```

[1, 4, 6, 4, 1]

```
List ([0..6], n-> List ([0..6], k-> Binomial (n, k))) ;;
```

```
PrintArray (last); # Нижній трикутник називається трикутником Паскаля
```

```
[ [ 1, 0, 0, 0, 0, 0, 0 ],
  [ 1, 1, 0, 0, 0, 0, 0 ],
  [ 1, 2, 1, 0, 0, 0, 0 ],
  [ 1, 3, 3, 1, 0, 0, 0 ],
  [ 1, 4, 6, 4, 1, 0, 0 ],
  [ 1, 5, 10, 10, 5, 1, 0 ],
  [ 1, 6, 15, 20, 15, 6, 1 ] ]
```

Крім того, C_n^k показує кількість сполучень з n елементів по k , тобто кількість k -елементних підмножин множини, що складається з n елементів.

```
Binomial (49, 6);
```

13983816

Самі ж ці сполучення можна отримати за допомогою функції `Combinations(set, k)`. Наприклад, знайдемо всі трійки, які можна скласти з чисел 1, 2, 3, 4, 5:

```
Combinations ([1..5], 3);
  [ [ 1, 2, 3 ], [ 1, 2, 4 ], [ 1, 2, 5 ], [ 1, 3, 4 ], [ 1, 3, 5 ],
    [ 1, 4, 5 ], [ 2, 3, 4 ], [ 2, 3, 5 ], [ 2, 4, 5 ], [ 3, 4, 5 ] ]
```

Якщо другий параметр не вказувати, то повертається список всіх підмножин заданої множини. Кількість сполучень можна також отримати і за допомогою функції `NrCombinations`:

```
Binomial (52,5) = NrCombinations ([1..52], 5);
  true
NrCombinations ([1..52]) = 2 ^ 52;
  true
```

На відміну від функції `Combinations`, що обчислює невпорядковані набори без повторень, функція `Arrangements` обчислює розміщення, тобто всі впорядковані набори з k елементів вихідної множини без повторень. Якщо аргумент k не заданий, то повертаються всі набори для кожного допустимого значення k . Відповідно, кількість наборів із зазначеними властивостями повертає функція `NrArrangements`.

Наприклад, визначимо, скільки різних слів можна скласти з літер a, b, b і c , і, зокрема, скільки різних двобуквених слів можна скласти з них, а потім виведемо всі ці слова:

```
NrArrangements ([ "a", "b", "b", "c"]);
  35
NrArrangements ([ "a", "b", "b", "c"], 2);
  7
List (Arrangements ([ "a", "b", "b", "c"], Concatenation);
  [ [], "A", "ab", "abb", "abbc", "abc", "abcb", "ac", "acb", "acbb", "b",
    "ba", "bab", "babc", "bac", "bacb", "bb", "bba", "bbac", "bbc", "bbca",
    "bc", "bca", "bcab", "bcb", "bcba", "c", "ca", "cab", "cabb", "cb", "cba",
    "cbab", "cbb", "cbba"]
List (Arrangements ([ "a", "b", "b", "c"], 2), Concatenation);
  [ "Ab", "ac", "ba", "bb", "bc", "ca", "cb"]
```

Функція `UnorderedTuples(set, k)` обчислює поєднання з повтореннями, тобто множину невпорядкованих наборів з k елементів вихідної множини з повтореннями. Функція `Tuples(set, k)` обчислює розміщення з повтореннями, тобто множину впорядкованих наборів з k елементів вихідної множини з повтореннями (фактично вона повертає список елементів множини декартових

добутків на себе k разів). Відповідно, кількість наборів із зазначеними властивостями повертають функції `NrUnorderedTuples` та `NrTuples`.

```
UnorderedTuples ([1..3], 2);
    [[1, 1], [1, 2], [1, 3], [2, 2], [2, 3], [3, 3]]
Tuples ([1..3], 2);
    [[1, 1], [1, 2], [1, 3], [2, 1], [2, 2], [2, 3], [3, 1],
    [3, 2], [3, 3]]
NrUnorderedTuples ([1..52], 5);
    3819816
NrTuples ([- 5..5], 5);
    +161051
```

`PartitionsSet(set [,k])` повертає множину різних неупорядкованих розбиттів множини *set* в об'єднання k попарно непересічних непустих множин. Якщо k не задано, то повертаються всі розбиття для всіх допустимих значень k . Кількість таких розбиттів визначається за допомогою функції `NrPartitionsSet(set [,k])`.

```
PartitionsSet ([1,2,3]);
    [[[1], [2], [3]], [[1], [2, 3]], [[1, 2], [3]],
    [[1, 2, 3]], [[1, 3], [2]]]
PartitionsSet ([1,2,3,4], 2);
    [[[1], [2, 3, 4]], [[1, 2], [3, 4]], [[1, 2, 3], [4]],
    [[1, 2, 4], [3]], [[1, 3], [2, 4]], [[1, 3, 4], [2]],
    [[1, 4], [2, 3]]]
NrPartitionsSet ([1..6]);
    203
NrPartitionsSet ([1..10], 3);
    9330
```

`Partitions(n [,k])` повертає множину всіх (неупорядкованих) розбиттів натурального числа n , тобто його представлень у вигляді неупорядкованої суми k доданків. Якщо число k не задано, то повертається множина таких розбиттів для всіх допустимих значень k . Розбиття числа n у вигляді неупорядкованої суми $n = p_1 + p_2 + \dots + p_k$ натуральних чисел описується списком $p = [p_1, p_2, \dots, p_k]$, елементи якого перераховані в незростаючому порядку. Слід врахувати, що `Partitions(40)` повертає 37338 таких розбиттів, а для наступних чисел кількість їх розбиттів буде ще більше. Його можна визначити за допомогою функції

`NrPartitions(n[, k])`, яка не обчислює при цьому самі розбиття.

`Partitions (7);`

```
[[1, 1, 1, 1, 1, 1, 1], [2, 1, 1, 1, 1, 1], [2, 2, 1, 1, 1],
 [2, 2, 2, 1], [3, 1, 1, 1, 1], [3, 2, 1, 1], [3, 2, 2],
 [3, 3, 1], [4, 1, 1, 1], [4, 2, 1], [4, 3], [5, 1, 1], [5, 2],
 [6, 1], [7]]
```

`Partitions (8, 3);`

```
[[3, 3, 2], [4, 2, 2], [4, 3, 1], [5, 2, 1], [6, 1, 1]]
```

`NrPartitions (7);`

15

`NrPartitions (100);`

190569292

Аналогами функцій, що мають справу з упорядкованими розбиттями натурального числа n , є функції `OrderedPartitions(n [, k])` та `NrOrderedPartitions(n [, k])`. Слід урахувати, що вже для $n = 15$ результат функції `OrderedPartitions(15)` буде досить об'ємним.

`OrderedPartitions (5);`

```
[[1, 1, 1, 1, 1], [1, 1, 1, 2], [1, 1, 2, 1], [1, 1, 3],
 [1, 2, 1, 1], [1, 2, 2], [1, 3, 1], [1, 4], [2, 1, 1, 1],
 [2, 1, 2], [2, 2, 1], [2, 3], [3, 1, 1], [3, 2], [4, 1], [5]]
```

`OrderedPartitions (6, 3);`

```
[[1, 1, 4], [1, 2, 3], [1, 3, 2], [1, 4, 1], [2, 1, 3],
 [2, 2, 2], [2, 3, 1], [3, 1, 2], [3, 2, 1], [4, 1, 1]]
```

`NrOrderedPartitions (20);`

524288

Також є функції, що дозволяють розглядати тільки ті розбиття, які задовольняють деяким додатковим умовам. Функція `PartitionsGreatestLE(n, m)` повертає множину всіх неупорядкованих розбиттів натурального числа n , кожний доданок яких менше або дорівнює m . Функція `PartitionsGreatestEQ(n, m)` повертає множину всіх неупорядкованих розбиттів натурального числа n , в яких максимальний доданок дорівнює m .

Функція `RestrictedPartitions(n, set [, k])` повертає множину всіх неупорядкованих розбиттів натурального числа n (на k доданків, якщо заданий аргумент k , або будь-яких, якщо він не заданий), в яких складові належать множині set . Число таких розбиттів можна отримати за допомогою функції `NrRestrictedPartitions(n, set [, k])`.

```
RestrictedPartitions (8, [1,3,5,7]);
  [[1, 1, 1, 1, 1, 1, 1, 1], [3, 1, 1, 1, 1, 1], [3, 3, 1, 1],
  [5, 1, 1, 1], [5, 3], [7, 1]]
```

```
NrRestrictedPartitions (100, [1,2,5,10,25,50]);
```

```
3953
```

Останній приклад показує, що існує 3953 способів представити 1 гривню монетами в 1, 2, 5, 10, 25 та 50 копійок.

Функція `Fibonacci(n)` повертає n -й елемент послідовності Фібоначчі, яка визначена в такий спосіб: $F_1 = F_2 = 1$ і $F_{n+2} = F_{n+1} + F_n$. Отримаємо список перших п'ятдесяти її елементів та продемонструємо таку властивість даної послідовності:

```
fib:=List([1..50],Fibonacci);
  [ 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597,
  2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418,
  317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465,
  14930352, 24157817, 39088169, 63245986, 102334155, 165580141, 267914296,
  433494437, 701408733, 1134903170, 1836311903, 2971215073, 4807526976,
  7778742049, 12586269025 ]
```

```
Gcd (fib [50], fib [25]) = fib [Gcd (50,25)];
```

```
true
```

%giac – надає інтерфейс до інтерпретатора Giac, який є ядром системи динамічної геометрії Geogebra та системи комп'ютерної математики Xcas [29]. У навчанні основ математичної інформатики Web-версія останньої, доступна за посиланням <http://www-fourier.ujf-grenoble.fr/~parisse/giac/xcasen.html>, може використовуватись у якості *хмаро зорієнтованого середовища моделювання* – хмаро зорієнтованої навчальної лабораторії, призначеної для моделювання об'єктів, явищ і процесів, що є предметом вивчення, або надання засобів для побудови і дослідження моделей. Компактність Giac/Xcas надає можливість її використання на широкому спектрі мобільних Інтернет-пристроїв (включно із графічними калькуляторами Texas Instruments: TI89, Voyage 200, TI Nspire CAS та ін.).

У навчанні методів оптимізації (тема 9 спецкурсу) Giac/Xcas можна використати для демонстрації методу дихотомії. На рис. 2.25 наведено приклад функції `dicho` для пошуку кореня деякої функції f між заданими значенням a і b із точністю eps .

При виконанні лабораторної роботи № 2 «Комбінаторні та рекурсивні алгоритми» Giac/Xcas може використовуватись для унаочнення класичних рекурсивних алгоритмів побудови фракталів.

На рис. 2.26 наведено приклад реалізації рекурсивного алгоритму побудови алгебраїчного фракталу – кривої Пеано-Госпера – за допомогою черепахової графіки у хмаро зорієнтованому середовищі моделювання Giac/Xcas.

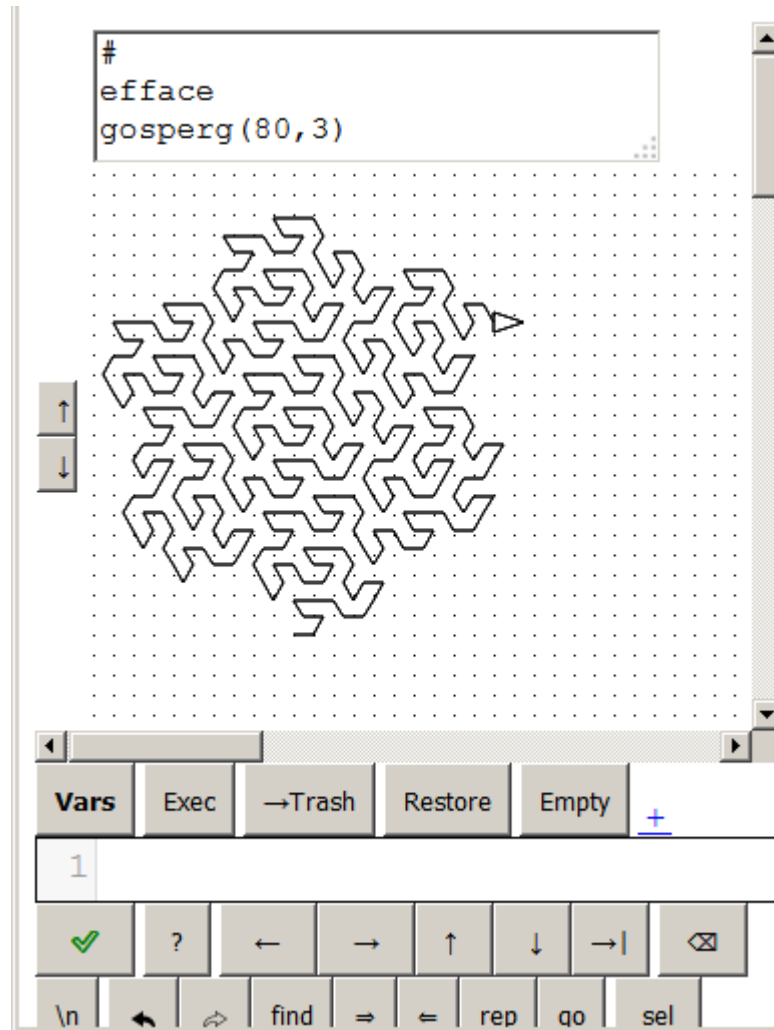


Рис. 2.26. Демонстрація рекурсивної побудови кривої Пеано-Госпера у хмаро зорієнтованому середовищі моделювання Giac/Xcas

Вихідні коди функції `gosperg` та відповідні пояснення пропонуємо студентам отримати із глави 7 «Приклади рекурсії» керівництва Р. Де Граве (Renée De Graeve) [20].

%go – надає можливість виконання програм мовою багатопотокового

програмування Go, що розробляється Google з 2007 року. Приклад:

```
%go
func main() { fmt.Println("Hello World") }
```

Можна запустити на виконання увесь робочий лист, набравши:

```
%default_mode go
```

Зв'язок між сусідніми комітками на робочому аркуші відсутній – кожна програма компілюється та виконується автономно, єдиними побічними ефектами є зміни в файловій системі. Сама програма зберігається у випадковому файлі, який видаляється після того, як запускається.

Команда `%go` автоматично додає `'package main'` та `import "fmt"` у верхній частині програми, оскільки передбачається, що Go використовується в інтерактивному режимі.

`%gp` – надає інтерфейс до PARI/GP (інтерпретатора та бібліотеки для теоретико-числових обчислень). PARI/GP є складовою CoCalc, доступ до якої можливий як через функції SageMath, так й безпосередньо. Найбільш доцільним використанням PARI/GP є у змістовому модулі 4 «Основи криптографії». Так, при виконанні лабораторної роботи 17 «Криптографія з відкритим ключем» необхідно перетворити рядок символів (відкритий текст) на ціле число (шифротекст), реалізувавши RSA – криптосистему з відкритим ключем. Шифрування та дешифрування у RSA виконується над цілими числами за допомогою теоретико-числових операцій.

Нехай задано відкритий текст p , представлений як рядок символів, який потрібно перетворити на ціле число перед шифруванням. Крім того, якщо c – це зашифрований текст, що відповідає p , то йому теж відповідає певне ціле число. Після розшифрування c ми відновлюємо початкове цілочисельне подання p .

Для виконання лабораторної роботи необхідні знання базових відомостей про елементарну теорію чисел, основи програмування та криптографію з відкритим ключем.

PARI/GP надає дві команди для перетворення рядків символів на цілі числа і навпаки: команда `Vecsmall` перетворює рядок символів на відповідний вектор ASCII-кодів, а команда `Strchr` виконує зворотню дію. Наприклад:

```
%gp
M = "Криптографія з відкритим ключем" ; P = Vecsmall(M)
      Vecsmall([208, 154, 209, 128, 208, 184, 208, 191, 209, 130, 208, 190, 208,
179, 209, 128, 208, 176, 209, 132, 209, 150, 209, 143, 32, 208, 183, 32, 208, 178,
209, 150, 208, 180, 208, 186, 209, 128, 208, 184, 209, 130, 208, 184, 208, 188,
32, 208, 186, 208, 187, 209, 142, 209, 135, 208, 181, 208, 188])
Strchr(P)
```

```
"Криптографія з відкритим ключем"
```

Нам потрібно об'єднати всі ASCII-коди P , щоб отримати ціле число, яке відповідає повідомленню M . Це можна виконати за допомогою команди `Str`, яка об'єднує всі її аргументи у єдиний рядок. Результат об'єднання всіх елементів вектору P – це рядкове подання цілого числа. Функція `eval` надає можливість перетворити такий рядок на довге ціле число:

```
a = ""; for (i = 1, length(P), a = Str(a,P[i]))
a
      "20815420912820818420819120913020819020817920912820817620913220915020914332
208183322081782091502081802081862091282081842091302081842081883220818620818720914
2209135208181208188"
```

```
eval(a)
      208154209128208184208191209130208190208179209128208176209132209150209143322
081833220817820915020818020818620912820818420913020818420818832208186208187209142
209135208181208188
```

Алгоритм RSA містить 6 основних кроків:

1. Обрати два простих числа p і q та обчислити $n = pq$.

`PARI/GP` надає список попередньо згенерованих простих чисел, доступних через команду `primes`. Наприклад, `primes(100)` повертає список перших 100 простих чисел. Команда `random(n)` повертає псевдовипадкове ціле число $k \in [0; n-1]$. Таким чином, крок 1 можна виконати у такий спосіб:

```
%gp
p = primes(40000)[random(40000) + 1]
      345047
q = primes(40000)[random(40000) + 1]
      37277
n = p * q
      12862317019
```

2. Дібрати $e \in \mathbb{N}$ таке, що $\text{НСД}(e, \varphi(n)) = 1$.

На цьому кроці необхідно знайти таке натуральне число, яке є

взаємнопростим із співпадає із числом $\varphi(n)$, де φ – функція Ейлера. Команда `eulerphi(n)` підраховує кількість цілих чисел $k \in [1; n]$ таких, що $\text{НСД}(k, n) = 1$. Найбільший спільний дільник двох цілих чисел можна обчислити за допомогою команди `gcd`. Використовуючи цикл, ми можемо обчислити необхідне значення e наступним чином:

```
%gp
e = random(eulerphi(n))
      1903175150
while (gcd(e, eulerphi(n)) != 1, e = random(eulerphi(n)))
e
      6119806961
```

3. Обчислити $d \in \mathbb{Z}$ таке, щоб $de \equiv 1 \pmod{\varphi(n)}$.

Для розрахунку значення d на цьому кроці використаємо розширений алгоритм Евкліда. За визначенням конгруентності, конгруенція $de \equiv 1 \pmod{\varphi(n)}$ є еквівалентною до $de - k\varphi(n) = 1$, де $k \in \mathbb{Z}$. Із попереднього кроку вже відомі числові значення e та $\varphi(n)$. Розширений алгоритм Евкліда надає можливість обчислювати d та $-k$. У PARI/GP це можна виконати за допомогою команди `bezout(x, y)`, яка повертає вектор $[u, v, g]$ такий, що $g = \text{НСД}(x, y)$ і $ux + vy = g$:

```
d = bezout(e, eulerphi(n))[1]
      104588385
```

4. Пара (n, e) є відкритим ключем, трійка (p, q, d) – приватним.

Таким чином, відкритий ключ RSA – це $(n, e) = (12862317019, 6119806961)$, а відповідний приватний ключ $(p, q, d) = (345047, 37277, 104588385)$.

5. Для шифрування будь-якого ненульового числа $m \in \mathbb{Z}$ необхідно обчислити $c \equiv m^e \pmod{n}$.

```
c=Mod(m^e, n)
```

6. Розшифрувати c можна так: $m \equiv c^d \pmod{n}$.

```
m=Mod(c^d, n)
```

Виконання 5 кроку може призвести до переповнення стеку PARI/GP через застосування надвеликих чисел. Існує спосіб ефективного виконання

піднесення до степеню за певним модулем – методом повторюваного піднесення до квадрату [145].

Нехай $n = (\overline{m_k m_{k-1} \dots m_1 m_0})_2$ – двійкове подання степіня n , тобто

$$n = m_k \cdot 2^k + m_{k-1} \cdot 2^{k-1} + \dots + m_1 \cdot 2 + m_0,$$

де $m_k = 1, m_i \in \{0, 1\}$. Тоді

$$\begin{aligned} x^n &= x^{((\dots((m_k \cdot 2 + m_{k-1}) \cdot 2 + m_{k-2}) \cdot 2 + \dots) \cdot 2 + m_1) \cdot 2 + m_0} = \\ &= (((\dots(((x^{m_k})^2 \cdot x^{m_{k-1}})^2 \dots)^2 \cdot x^{m_1})^2 \cdot x^{m_0}. \end{aligned}$$

Таким чином, алгоритм повторюваного піднесення до квадрата зводиться до мультиплікативного аналогу схеми Горнера:

$$\left\{ \begin{array}{l} s_1 = x \\ s_{i+1} = s_i^2 \cdot x^{m_{k-i}} \\ i = 1, 2, \dots, k \end{array} \right\}.$$

Нижче наведено відповідний скрипт PARI/GP:

```
modexp(a, b, n) = { \
  local(d, bin); \
  d = 1; \
  bin = binary(b); \
  for (i = 1, length(bin), \
    d = Mod(d*d, n); \
    if (bin[i] == 1, \
      d = Mod(d*a, n); \
    ); \
  ); \
  return(d); \
}
```

%hide приховує вміст комірки. За замовчуванням приховується код змінної частини комірки, але можна приховати й інші частини шляхом вказання додаткових аргументів 'input', 'output'. Для відображення вмісту використовується функція `cell.show(...)`.

%hideall приховує вміст усіх комірок.

%html є еквівалентом `%pandoc('html')`, надаючи можливість форматування за розміткою HTML тексту, розміщеного одразу після `%html` або у наступних рядках.

%javascript – надає можливість виконання коду JavaScript. Оскільки, як

значалося вище, код мовою CoffeeScript транслюється у JavaScript та виконується безпосередньо у браузері, то, можна стверджувати, що код мовою JavaScript має аналогічне застосування, проте є швидшим.

%julia – надає інтерфейс для Julia – нової функціональної мови програмування для швидких математичних розрахунків великого обсягу. Одним із пріоритетних напрямків у розвитку Julia є підтримка розподілених обчислень: синтаксис мови містить велику кількість стандартних конструкцій для розпаралелювання коду з використанням глобального розподіленого адресного простору. Це означає, що можна керувати посиланням на об'єкт, який знаходиться на іншій машині, що бере участь у розрахунках, легко керувати такими посиланнями та передавати їх між машинами; крім того, під час розрахунків можна підключити нові машини [44].

%lisp – виконання коду мовою штучного інтелекту Лісп.

%kash – надає інтерфейс до комп'ютерної алгебраїчної системи KANT / KASH (Computational Algebraic Number Theory / KAnt SHell), спрямованої на виконання дій над полями різних типів: алгебраїчними числовими, алгебраїчними функціональними та локальними. Остання версія KANT базується на системі Magma, а KASH – на системі GAP [42, с. 10].

%latex – надає можливість використання мови LaTeX для форматування математичних текстів.

У Додатку Г наведено приклад використання LaTeX для оформлення демонстрації методу Гауса для розв'язання систем лінійних алгебраїчних рівнянь. Опис теоретичного матеріалу виконується за допомогою команди `html` (рис. 2.27), параметром якої буде рядок, що містить теги HTML та обмежені знаками «\$» команди LaTeX. Демонстрація містить код SageMath, що надає можливість експериментувати із системами лінійних алгебраїчних рівнянь різної розмірності зі змінною кількістю коефіцієнтів. Головною особливістю даної демонстрації є подання всіх етапів розв'язання в узагальненому вигляді.

Modes ▾ # Data ▾ Control ▾ Program ▾ x Plots ▾ Calculus ▾ Linear ▾ Graphs ▾ Numbers ▾ Rings ▾ %sage

$$a_{\{nn\}}x_{\{n\}} \quad b_{\{n\}}$$

Метод Гауса

Найбільш розповсюдженим точним способом рішення систем лінійних рівнянь є алгоритм послідовного виключення невідомих. Цей метод носить назву *методу Гауса*.

Нехай є система (1.1) n лінійних алгебраїчних рівнянь (СЛАР) з n невідомими:

$$A = \begin{cases} a_{11} & a_{12} \cdots & a_{1n} = b_1, \\ a_{21} & a_{22} \cdots & a_{2n} = b_2, \\ \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} \cdots & a_{nn} = b_n. \end{cases}$$

Позначимо через $A = \begin{pmatrix} a_{11} & a_{12} \cdots & a_{1n} \\ a_{21} & a_{22} \cdots & a_{2n} \\ \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} \cdots & a_{nn} \end{pmatrix}$ матрицю коефіцієнтів системи (1.1), через $b = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{pmatrix}$ стовпець її вільних членів, а вектор невідомих - через $x = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \end{pmatrix}$

Рис. 2.27. Фрагмент опису теоретичного матеріалу з використанням LaTeX

```
@interact
def demosystem(m=slider(1, 9, step_size=1, label="Кількість рівнянь"), \
n=slider(1, 9, step_size=1, label="Кількість невідомих")):
```

Кількість рівнянь

Кількість невідомих

Наступний крок – створення узагальнених змінних a_{ij} , b_i та x_i :

```
L=[]
for i in range(m):
    for j in range(n):
        L.append(var("a%s%s"%(i+1,j+1)))
A=matrix(m,n,L)
L=[]
for i in range(m):
    for j in range(n):
        L.append(var("a%s%s"%(i+1,j+1)))
    L.append(var("b%s"%(i+1)))
Ab=matrix(m,n+1,L)
x=matrix(n,1,var(", ".join("x%s"%(i+1) for i in range(n))))
b=matrix(m,1,var(", ".join("b%s"%(i+1) for i in range(m))))
```

Для проведення експериментів із моделлю спочатку створюється інтерфейс із двома елементами типу «ковзунок»:

Система рівнянь формується відповідними викликами функції `latex`:

```
s="\left \{ \begin{eqnarray}"
for i in range(m):
```

```
s=s+"%s=%s \\\\"%(\latex((A*x)[i][0]),\latex(b[i][0]))
s=s+"\end{eqnarray} \\\right.$"
```

Сформовані система та матриці виводяться за допомогою команди `html`:

```
html("</pre>Система рівнянь:<br>" + s)
      Система рівнянь:
      {
      a11x1 + a12x2 + a13x3 = b1
      a21x1 + a22x2 + a23x3 = b2
html("<p>Матриця коефіцієнтів: <b>A</b>=%s$"%\latex(A))
html("<p>Вектор невідомих: <b><i>x</i></b>=%s$"%\latex(x))
html("<p>Вектор вільних коефіцієнтів: <b><i>b</i></b>=%s$"%\latex(b))
html("<p>Розширена матриця: <b>A|<i>b</i></b>=%s$"%\latex(Ab))
```

$$\text{Матриця коефіцієнтів: } \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

$$\text{Вектор невідомих: } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$\text{Вектор вільних коефіцієнтів: } \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$\text{Розширена матриця: } \mathbf{A|b} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \end{pmatrix}$$

Наступний крок – прямий хід Гауса:

```
for j in range(m):
    i=j+1
    while i<m and j<n:
        M=Ab[i][j]/Ab[j][j]
        Ab[i]=Ab[i]-M*Ab[j]
        i=i+1
html("<p>Розширена матриця після прямого ходу Гауса:<br>
<b>A|<i>b</i></b>=%s$"%\latex(Ab))
```

Розширена матриця після прямого ходу Гауса:

$$\mathbf{A|b} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & -\frac{a_{12}a_{21}}{a_{11}} + a_{22} & -\frac{a_{13}a_{21}}{a_{11}} + a_{23} & -\frac{a_{21}b_1}{a_{11}} + b_2 \end{pmatrix}$$

Далі виконуємо обернений хід Гауса:

```
j=m-1
while j>0:
    i=j-1
    while i>=0:
        if i<m and j<n:
            M=Ab[i][j]/Ab[j][j]
            Ab[i]=Ab[i]-M*Ab[j]
            i=i-1
        j=j-1
html("<p>Розширена матриця після оберненого ходу Гауса:<br>
<b>A|<i>b</i></b>=%s$"%\latex(Ab))
```

Розширена матриця після оберненого ходу Гауса:

$$A|b = \begin{pmatrix} a_{11} & 0 & a_{13} - \frac{a_{12} \left(\frac{a_{13} a_{21}}{a_{11}} - a_{23} \right)}{\frac{a_{12} a_{21}}{a_{11}} - a_{22}} & - \frac{a_{12} \left(\frac{a_{21} b_1}{a_{11}} - b_2 \right)}{\frac{a_{12} a_{21}}{a_{11}} - a_{22}} + b_1 \\ 0 & - \frac{a_{12} a_{21}}{a_{11}} + a_{22} & - \frac{a_{13} a_{21}}{a_{11}} + a_{23} & - \frac{a_{21} b_1}{a_{11}} + b_2 \end{pmatrix}$$

Перетворюємо розширену матрицю на одиничну діленням кожного її рядка на відповідний елемент діагоналі:

```
for j in range(m):
    if j<n:
        Ab[j]=Ab[j]/Ab[j][j]
    html("<p>Розширена матриця після зведення до одиничної:<br>
<b>A|<i>b</i></b>=%s$"%latex(Ab))
```

Розширена матриця після зведення до одиничної:

$$A|b = \begin{pmatrix} 1 & 0 & \frac{a_{13} - \frac{a_{12} \left(\frac{a_{13} a_{21}}{a_{11}} - a_{23} \right)}{\frac{a_{12} a_{21}}{a_{11}} - a_{22}}}{a_{11}} & - \frac{a_{12} \left(\frac{a_{21} b_1}{a_{11}} - b_2 \right)}{\frac{a_{12} a_{21}}{a_{11}} - a_{22}} - b_1 \\ 0 & 1 & \frac{\frac{a_{13} a_{21}}{a_{11}} - a_{23}}{\frac{a_{12} a_{21}}{a_{11}} - a_{22}} & \frac{\frac{a_{21} b_1}{a_{11}} - b_2}{\frac{a_{12} a_{21}}{a_{11}} - a_{22}} \end{pmatrix}$$

Подальші кроки алгоритму виконуються лише у випадку коректного задання системи рівнянь:

```
L=x.list()
L.append(-1)
xr=matrix(n+1,1,L)
if n<m:
    html("<p>Так як кількість невідомих менша за кількість рівнянь,
розв'язання системи методом Гауса неможливо")
```

Якщо система задана коректно, будемо розширений вектор невідомих

```
else:
    html("<p>Розширений вектор невідомих:<br>
<b><i>x'</i></b>=%s$<p>"%latex(xr))
```

Розширений вектор невідомих:

$$x' = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ -1 \end{pmatrix}$$

та знаходимо розв'язок системи:

```
html("Розв'язок системи:<br>")
temp=Ab*xr
L=x.list()
for i in range(m):
    html("%s$<br>"%latex(solve(temp[i][0]==0,x[i][0])[0]))
    L[i]=solve(temp[i][0]==0,x[i][0])[0].rhs()
```

Розв'язок системи:

$$x_1 = -\frac{a_{22}b_1 - a_{12}b_2 - (a_{13}a_{22} - a_{12}a_{23})x_3}{a_{12}a_{21} - a_{11}a_{22}}$$

$$x_2 = \frac{a_{21}b_1 - a_{11}b_2 - (a_{13}a_{21} - a_{11}a_{23})x_3}{a_{12}a_{21} - a_{11}a_{22}}$$

Для перевірки правильності розв'язку виконуємо підстановку обчисленого вектору коренів системи до відповідного матричного рівняння:

```
roots=matrix(n,1,L)
res=A*roots()
html("<p>Вектор коренів: %s%"%latex(roots))
L=res.list()
for j in range(m):
    L[j]=L[j].simplify_rational()
res=matrix(m,1,L)
html("<p>Перевірка:<br> <b>A$\cdot$<i>roots</i></b>=%s%"%latex(res))
```

Вектор коренів:
$$\begin{pmatrix} -\frac{a_{22}b_1 - a_{12}b_2 - (a_{13}a_{22} - a_{12}a_{23})x_3}{a_{12}a_{21} - a_{11}a_{22}} \\ \frac{a_{21}b_1 - a_{11}b_2 - (a_{13}a_{21} - a_{11}a_{23})x_3}{a_{12}a_{21} - a_{11}a_{22}} \\ x_3 \end{pmatrix}$$

Перевірка:

$$A \cdot \mathit{roots} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Після опанування методу та експериментів із моделлю студентам доцільно запропонувати завдання: на основі програмного коду узагальненої моделі розв'язання створити модель, що оперуватиме числовими коефіцієнтами.

%lie надає інтерфейс до LiE – комп'ютерної алгебраїчної системи для виконання дій над групами Лі [73].

%lisp надає інтерфейс до коду мовою Lisp. Зауважимо, що CoCalc не підтримує кириличних символів у кодї мовою Lisp, що змушує використовувати лише окремі, досить прості фрагменти, що не містять коментарів. У спецкурсі «Основи математичної інформатики» ми не пропонуємо виконувати завдання з використанням цієї мови студентам, що не володіють одним із її діалектів. Для тих же, хто володіє, доцільно буде продемонструвати створення простої функції диференціювання символічних виразів:

```
(defun diff (y x) ; y - вираз, що диференцюється, x - змінна диференціювання
  (cond
    (
      (atom y) ; якщо вираз - це змінна або стала
```



```
(/ (- (* (+ (* 0 x) (* 2 1)) 2) (* (* 2 x) 0)) (* 2 2))
```

Згортання результату та його приведення до звичайної (інфіксованої) нотації надає можливість перевірити правильність виконання коду із символьного диференціювання. В якості індивідуального завдання можна запропонувати розширення наведеної функції для інших арифметичних операцій.

Зверніть увагу, що іноді Lisp може знаходитись в очікуванні введення користувача. Його перезапуск здійснюється командою

```
lisp.quit()
```

%load *args* завантажує об'єкт Sage з файлу з ім'ям файлу *args*, який буде мати розширення *.sobj* (якщо воно явно не вказано). Завантажені файли виконуються, але доступ до визначених у них змінних та функцій із CoCalc відсутній. У якості імені файлу можна вказати веб-посилання (для приховування індикатору завантаження необхідно встановити `verbose = False`). *args* – будь-яка кількість файлів одного із наступних типів: *.sobj*, *.sage*, *.py*, *.pyx*, *.html*, *.css*, *.js*, *.coffee*, *.pdf*. `verbose` (за замовчуванням `True`) – показувати завантаження файлів із мережі. При завантаженні веб-типів (*.html*, *.css*, *.js*, *.coffee*) вони завантажуються у DOM (або JavaScript) область веб-браузера, а не у процес Python. При завантаженні pdf-файлу він відображається на робочому аркуші.

%macaulay2 надає інтерфейс до Macaulay2 – системи для обчислень у галузі комутативної алгебри та алгебраїчної геометрії. Дану систему доцільно використовувати для розв'язання систем поліноміальних рівнянь, ілюструючи геометричний зміст відповідних дій [17].

%magma, **%maple**, **%mathematica** та **%matlab** надають інтерфейс до відомих комерційних систем комп'ютерної математики за умови наявності ліцензії на них та встановлення на сервері CoCalc:

```
RuntimeError: unable to start maple because the command 'maple -t -c
"interface(screenwidth=infinity,errorcursor=false)"' failed: The command was not
found or was not executable: maple.
```

In order to use the Maple interface you need to have Maple installed and have a script in your PATH called "maple" that runs the command-line version of Maple. Alternatively, you could use a remote connection to a server running Maple; for hints, type

- ```
print(maple._install_hints_ssh())
```
- (1) You might have to buy Maple (<http://webstore.maplesoft.com/>).
  - (2) \* LINUX: The maple script comes standard with your Maple install.
    - \* APPLE OS X:
      - (a) create a file called maple (in your PATH), with the following contents:
 

```
#!/bin/sh
/Library/Frameworks/Maple.framework/Versions/Current/bin/maple $@
```
      - (b) Save the file.
      - (c) Make the file executable.
 

```
chmod +x maple
```
    - \* WINDOWS:
 You must install Maple-for-Linux into the VMware machine (sorry, that's the only way at present).

Альтернативою налаштування Magma, Maple, Mathematica та MATLAB на сервері CoCalc є використання спрощених хмарних версій, що надають їх розробники. Так, для Mathematica – це Wolfram|Alpha (<http://wolframalpha.com/>), для MATLAB – MATLAB Online (<https://matlab.mathworks.com/>), для Magma – Magma Calculator (<http://magma.maths.usyd.edu.au/calc/>). Наведемо приклад використання останнього для виконання лабораторної роботи № 19 «Алгоритм обміну ключами Діффі-Хеллмана» (рис. 2.28, 2.29).

Алгоритм обміну ключами Діффі-Хеллмана (Diffie-Hellman) – перший практичний метод для отримання загального секретного ключа при спілкуванні через незахищений канал зв'язку [71, с. 51-56]. Сутність алгоритму полягає в наступному: нехай два абоненти знають деякі два числа  $g$  і  $p$  (наприклад, вони можуть бути «захиті» у програмному забезпеченні), які не є секретними і можуть бути відомі також іншим зацікавленим особам. Для того, щоб створити невідомий більш нікому секретний ключ, обидва абонента генерують великі випадкові числа: перший абонент – число  $a$ , другий абонент – число  $b$ . Потім перший абонент обчислює значення  $A = g^a \pmod{p}$  та пересилає його другому, а другий обчислює  $B = g^b \pmod{p}$  і передає першому. Передбачається, що зловмисник може отримати обидва ці значення, але не модифікувати їх (тобто у нього немає можливості втручатися у процес передачі). На другому етапі перший абонент на базі наявного в нього числа  $a$  та отриманого через мережу  $B$  обчислює значення  $B^a \pmod{p} = g^{ab} \pmod{p}$ , а другий абонент на основі наявного у нього  $b$  та отриманого по мережі  $A$  обчислює значення  $A^b \pmod{p} = g^{ab} \pmod{p}$ .



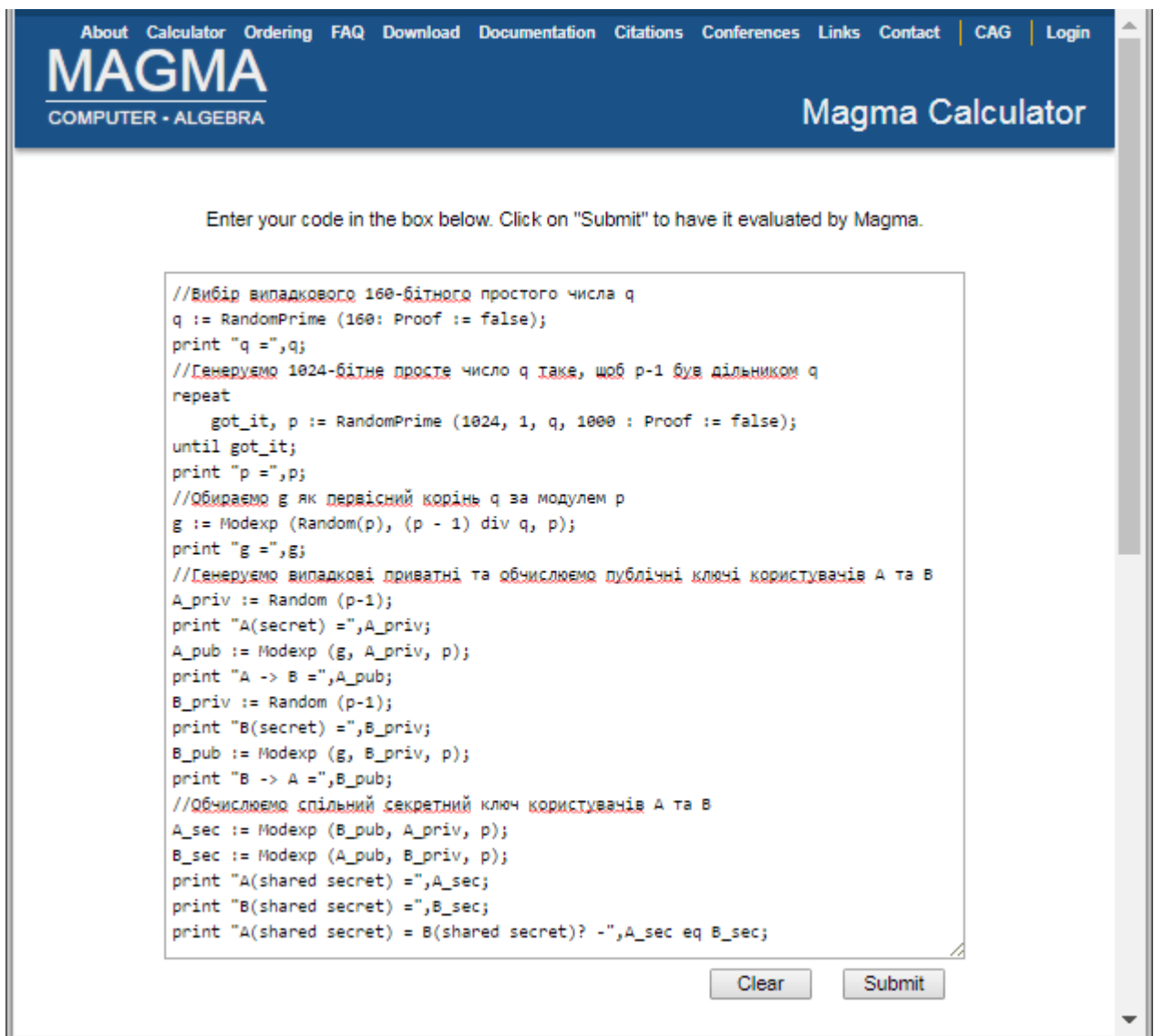


Рис. 2.28. Програмна реалізація алгоритму обміну ключами Діффі-Хеллмана у Magma Calculator

Як результат, у обох абонентів вийшло одне й те саме число  $K = g^{ab} \pmod{p}$ , яке може бути використане в якості секретного ключа, оскільки зловмисник стикається з практично нерозв'язною (за розумний час) проблемою обчислення  $g^{ab} \pmod{p}$  за перехопленими  $g^a \pmod{p}$  та  $g^b \pmod{p}$ , якщо числа  $p$ ,  $a$  та  $b$  вибираються досить великими.

Наведений приклад демонструє, наскільки простим є реалізація криптографічних протоколів на основі модульної арифметики в середовищі Magma [12, с. 185-187]. У прикладі спочатку генерується 160-бітне просте число  $q$ , а потім 1024-бітне – таке просте число  $p$ , що  $p-1$  ділиться на  $q$ . Після

цього створюється основний елемент  $g$ , який має порядок  $q$  по модулю  $p$ . На цьому генерація параметрів завершується.

```

q = 981177992932036143736658900326928865566258839703
p = 158751619376882401754367877119717096992451012630334476363220681648353992052\
9694309255441546169630810266785196399372954120695831403053512106999415368381659\
6338170386212786648228579865186662893542130155875946290925380773840703573957453\
8874639555604399617144881109225043365822459779746538019429561684144948394027
g = 842750653180980413689218252591912388928815981936628321440769437082019410975\
7688455603325774309209651529956551026568029765433577606111216943836765094973399\
7329070673352221441050006225027054452311390446867398720958667275058072920241174\
518946894166057250188916892685082928500183313651967764473561685849514996533
A(secret) = 1518277359540567218120450034079161156820557586694900744536726630633\
6559062663145291780698389583542078175679598158683488561059190231470505698966237\
2747924954293819241295298118584578243669472638900756757413540387664356101081909\
9183319085953525488170944192247965375391912655604348694524667206442477463800733\
15126
A -> B = 4583371897011226176980920733318657668968883544993340542263820703047831\
7145812742109633905901798940268413467064950439088347595724950848065488471632460\
7274698267425463742695430253761093154176488000935483969006083275750782372862686\
8633633814584867392951458786491558290825373103483498810673808119239053289046658\
7
B(secret) = 5339986761164440182193901881271558073295600957939322273491291440158\
8725387413555495753606271695789287446930316998109431834560990437211035722498810\
8923165603009849775015001012836439520668246004003413018318783734872097345566493\
0116192394204443778479402412536594537144163599593815345030619328069345577115042\
3636
B -> A = 1167839360898192686465859937989216803775713857822425755315327983499029\
7762706531582346929050432737091809184346146875905070676532769088910819308403275\
1917281017463827513197530693374656933578375463029705166543413097865110682661663\
3191910678962533160120907619437511873917207084927645635448665430782858754467955\
89
A(shared secret) = 847007313590125358502673258947807482644825191457431179707087\
4433115476386972810532302100402554125021704471165076388773546273455162287592475\
7423040591989527212717336880526715094156593528741515804567856155154613589939506\
3376994707273630233344288834269401018049565640571898972877298568164756124899355\
63376078215
B(shared secret) = 847007313590125358502673258947807482644825191457431179707087\
4433115476386972810532302100402554125021704471165076388773546273455162287592475\
7423040591989527212717336880526715094156593528741515804567856155154613589939506\
3376994707273630233344288834269401018049565640571898972877298568164756124899355\
63376078215
A(shared secret) = B(shared secret)? - true

```

Рис. 2.29. Приклад сеансу роботи студента в процесі виконання лабораторної роботи

```

1337515134874490020173357215531009450590829772871
> Ilog2 (q);
159
> repeat
repeat> got_it, p := RandomPrime (1024, 1, q, 1000 : Proof := false);
repeat> until got_it;
> p;
9580857313574454467613060923425440574687571992321592085613767541323518556424063\
6870999714979396504020106523189126646490023404689751268797973287943112748468858\

```

```

5112188583029439178782333909341452456713751676055272435838466600707466469768637\
02436886400028700931402807964952723748913904077302288212634086165312979
> Ilog2 (p);
1023
> tmp := Random(p);
> exp := (p - 1) div q;
> g := Modexp (tmp, exp, p);
> g;
9099434663087419685135561467286690325533565050389764304406879818429098710086127\
5056323837603364481555882010692591443031542806508304253422643971354617261015205\
9526110036982281836121915695320198181107494957048326450851398389718852451553377\
71985882768398204953603166823927758384246426448586848858743868508080769
> Modexp (g, q, p);
1

```

Функція `RandomPrime` використовується вище двома різними способами. Спочатку для генерації 160-бітної послідовності  $q$ , потім – для генерації 1024-бітного простого числа, яке конгруентне 1 за модулем  $q$  (таке, що  $p-1$  ділиться на  $q$ ). Четвертий аргумент `RandomPrime` – "1000" – це кількість ітерацій, необхідних для знаходження простого числа перед передачею, оскільки є можливість відмови (наприклад, користувач може вибрати параметри, для яких не існує простих чисел заданої форми). `RandomPrime` повертає два значення: перше – це логічне значення, якщо обчислення були успішними, і, друге, якщо перше значення є істинним – саме просте число. Другий виклик `RandomPrime` виконується циклічно для того, щоб переконатися, що знайдене необхідне просте число.

Після налаштування параметрів виконується реалізація протоколу.

Спочатку абоненти обирають приватні ключі і обчислюють відповідні їм відкриті ключі:

```

> A_priv := Random (p-1);
> A_pub := Modexp (g, A_priv, p);
> B_priv := Random (p-1);
> B_pub := Modexp (g, B_priv, p);

```

Далі абоненти використовують публічні ключі один одного для обчислення спільного секретного ключа:

```

> A_sec := Modexp (B_pub, A_priv, p);
> B_sec := Modexp (A_pub, B_priv, p);

```

Наприкінці перевіряється, чи однакові секретні ключі:

```

> A_sec eq B_sec;

```

```

true
> /* секретний ключ */
> A_sec;
8109416952960290247477174360527800613930603221801200765371022080190000393736445\
0307159060302439497132285938077980828986940582351118805711533307729202563454652\
9815387864793079894434226833555249730103795361582588274893664683809421885227721\
95753093058897768884456335283105474598964427905020684463746295562330487

```

Зауважимо, що наведений приклад міг бути реалізований у CoCalc за допомогою команди `magma_free`, що надає доступ до Magma Calculator, проте для користувачів на безоплатних тарифних планах доступ до Інтернет-ресурсів із блокноту SageMath не дозволено, на відміну від SageCell (рис. 2.30).



Type some Sage code below and press Evaluate.

```

1 code='''q := RandomPrime (160: Proof := false); print "q =",q;
2 repeat
3 got_it, p := RandomPrime (1024, 1, q, 1000 : Proof := false);
4 until got_it; print "p =",p;
5 g := Modexp (Random(p), (p - 1) div q, p); print "g =",g;
6 A_priv := Random (p-1); print "A(secret) =",A_priv; A_pub := Modexp (g, A_priv, p); print "A -> B =",A_pub;
7 B_priv := Random (p-1); print "B(secret) =",B_priv; B_pub := Modexp (g, B_priv, p); print "B -> A =",B_pub;
8 A_sec := Modexp (B_pub, A_priv, p); B_sec := Modexp (A_pub, B_priv, p);
9 print "A(shared secret) =",A_sec; print "B(shared secret) =",B_sec;
10 print "A(shared secret) = B(shared secret)? -",A_sec eq B_sec;'''
11 magma_free(code)
12

```

Language: Sage ▾

```

q = 1207003394432193542406194825116808242723996166779
p = 1494770545284801944239482411830158402322606456273800173045546314657833965771239146106691037970953005905398089604
g = 1242656375870307691010831100033825244314797776630574715739625960689176867676899245906780828114461994831202064488
A(secret) = 32917390250253433749641069076440461172274167135932506343203117912636881782578288386619962835109264481301
A -> B = 38755367026857403220659901661067949421467865365714194964217161761595655841048150559794676658410458254863954
B(secret) = 73022099817629627367138388332399291409542344191935010240306119169571611687255274159065112455978135554166
B -> A = 12639809870572939006454358286354125976525538014910610166587142309964345149483030741617751723557572301969769
A(shared secret) = 7648574630124561631485202129059015209623974285875227980102708321498446180500978588689213581132834
B(shared secret) = 7648574630124561631485202129059015209623974285875227980102708321498446180500978588689213581132834
A(shared secret) = B(shared secret)? - true

```

[Help](#) | Powered by SageMath

Рис. 2.30. Доступ до Magma Calculator із SageCell

**%maxima** – надає інтерфейс до системи комп'ютерної математики Maxima, методика використання якої для навчання математичної інформатики студентів педагогічних університетів розкрита у роботі Т. П. Кобильника [139].

**%md** є еквівалентом `%pandoc('markdown_github')`, надаючи можливість

форматування за розміткою Markdown GitHub [30], розміщеного одразу після `%md` або у наступних рядках. Для відображення математичних текстів на сторінках із розміткою Markdown використовується модуль `markdown2Mathjax`, що надає можливість застосування команд LaTeX.

**%mupad** – надає доступ до бібліотеки символічних обчислень MuPAD, яка є складовою комерційних систем комп’ютерної математики MATLAB та Mathcad.

**%mwrnk** – надає можливість виконання коду `mwrnk` (складової бібліотеки `eslib`) для обчислення груп еліптичних кривих Морделла-Вейля (Mordell-Weil) над полем раціональних чисел, їх ізогеній, підгруп кручення тощо [19];

**%octave** – надає інтерфейс до інтерпретатора Octave, чия мова програмування має високий рівень сумісності із мовою MATLAB [31]. Доступ до Octave можливий не лише із CoCalc, а й через Jupyter Notebook.

**%pandoc(fmt)** відображає текст у одному із заданих форматів: `'docbook'`, `'haddock'`, `'html'`, `'json'`, `'latex'`, `'markdown'`, `'markdown_github'`, `'markdown_mmd'`, `'markdown_phpextra'`, `'markdown_strict'`, `'mediawiki'`, `'native'`, `'opml'`, `'rst'`, `'textile'`.

**%perl** – надає доступ до інтерпретатора мови Perl, орієнтованої на опрацювання текстових даних [1].

**%prun** виконує профілювання коду, розміщеного одразу після `%prun` або у наступних рядках (рис. 2.31).

**%python** – надає інтерфейс до «чистого» інтерпретатора мови Python, що не містить розширень мовою SageMath.

**%r** – надає інтерфейс до інтерпретатора мови R, яка містить набір методів для статистичного аналізу (включаючи лінійну і нелінійну регресію, класичні статистичні тести, аналіз часових рядів, кластерний аналіз і багато інших), моделювання, біоінформатики, аналізу даних тощо [62]. Однією із переваг R є розвинені засоби візуалізації даних.

## Salvus Profiler

First sort by   strip\_dirs

```
Tue Feb 7 08:23:34 2017 /projects/bb73fdf2-e5e2-4e2c-a541-9e4c76a76a7d/.sage/temp/computel-us/7872/tmp_GwA
19055 function calls (18550 primitive calls) in 1.211 seconds

Ordered by: internal time

ncalls tottime percall cumtime percall filename:lineno(function)
 26 0.888 0.034 0.888 0.034 {sage.libs.ecl.ecl_eval}
 2 0.135 0.068 0.237 0.118 maxima_lib.py:675(sr_integral)
 1 0.076 0.076 0.965 0.965 maxima_lib.py:71(<module>)
 62 0.008 0.000 0.010 0.000 maxima_lib.py:404(_eval_line)
 14 0.008 0.001 0.011 0.001 operators.py:6(add_vararg)
 36 0.007 0.000 0.024 0.001 operators.py:30(mul_vararg)
 2 0.006 0.003 1.208 0.604 external.py:7(maxima_integrator)
 6 0.006 0.001 0.015 0.002 {method '_maxima_' of 'sage.symbolic.expression.Expression' object}
 30 0.005 0.000 0.043 0.001 {operator.pow}
 1 0.004 0.004 0.008 0.008 expression_conversions.py:8(<module>)
254/2 0.003 0.000 0.087 0.043 maxima_lib.py:1626(max_to_sr)
227/160 0.003 0.000 0.009 0.000 complex_interval_field.py:398(__call__)
 72 0.003 0.000 0.006 0.000 homset.py:543(init)
```

Рис. 2.31. Результат профілювання виклику `integrate(sin(x^2), x)`

У навчанні основ математичної інформатики R доцільно використовувати у другому змістовому модулі «Методи обчислень». Розглянемо приклад використання R у студентському дослідницькому проекті «Моделювання двовимірної моделі Ізінга методом Монте-Карло» [9, с. 194-198].

Модель Ізінга названа на честь німецького фізика Ернста Ізінга (Ernst Ising), була розроблена як статистична механічна модель ферромагнетизму. У своїй найбільш поширеній формі вона складається з дискретних змінних (зазвичай  $\pm 1$ ), що представляють собою спини, розташовані на ґратці. Кожен спін може взаємодіяти з сусідами з енергією взаємодії  $J$  (приймається рівною 1 у подальших розрахунках). Двовимірна модель Ізінга на квадратній решітці є однією з найпростіших статистичних механічних моделей для ілюстрації фазового переходу.

Розроблений Ларсом Онсагером (Lars Onsager) у 1942 році аналітичний розв'язок для двовимірної моделі Ізінга з періодичними граничними умовами на сьогодні є одним з найвищих досягнень сучасної теоретичної фізики. У розробленому нами спецкурсі двовимірна модель Ізінга є корисним прикладом задачі оптимізації: тобто пошуку положення рівноваги (мінімуму вільної енергії) системи взаємодіючих спінів при заданій температурі.

Алгоритм включає три основні частини:

- випадковий вибір спіну та його розгортання;
- обчислення різниці енергії  $\Delta E$ ;

– якщо  $\Delta E < 0$ , новий стан є бажаним та приймається. В іншому випадку генерується випадкове число  $p$  між 0 і 1, і якщо  $\exp(-\beta\Delta E) > p$ , то приймається новий стан. ( $\beta = 1/k_B T$ , де  $k_B$  – константа Больцмана та  $T$  – температура за Кельвіном). Аналогічно, логарифмуючи нерівність з обох боків, рух приймається, якщо  $(-\beta\Delta E) > \ln(p)$ .

Обчислювально інтенсивна частина цього процесу пов'язана з повторенням цих кроків до тих пір, поки не буде досягнута рівновага, а потім повторюється ще багато разів, щоб отримати достатню статистичну вибірку для заданої  $T$ . Для досить малої решітки ( $12 \times 12$ ), що розглядається нижче, рівновага досягається приблизно за  $10^5$  кроків.

Після достатньої кількості кроків можна обчислити середню енергію  $\langle E \rangle$  та намагніченість  $\langle M \rangle$  шляхом безпосереднього підсумовування у вузлах решітки. Для обчислення середньоквадратичних значень енергії  $\langle E^2 \rangle$  та намагніченості  $\langle M^2 \rangle$  будемо використовувати термодинамічні співвідношення:

$$C_V = \frac{\beta}{T} [\langle E^2 \rangle - \langle E \rangle^2]$$

$$\chi = \beta [\langle M^2 \rangle - \langle M \rangle^2]$$

для розрахунку теплоємності  $C_V$  та магнітної чутливості  $\chi$ .

Почнемо із налаштування параметрів моделі. Задамо квадратну решітку (матрицю) спінів  $\mathbf{A}$  з  $nr = 12$  рядів та  $nc = 12$  стовпчиків: однакова кількість рядків та стовпців забезпечує узгодженість з періодичними граничними умовами. Для кожної температури  $n_{pass}$  разів випадково обираємо спін, змінюємо його орієнтацію та перевіряємо енергію отриманої конфігурації за критерієм Метрополіса. Після перших  $n_{equil}$  таких ітерацій приймаємо, що система досягла рівноваги, тому інші ітерації використовуються для отримання середніх значень енергії, намагніченості та їх квадратів.

`nr = 12; nc = 12 # Кількість рядків та стовпців`

```
A = matrix(nrow = nr, ncol = nc)
npass = 2e5 # Кількість ітерації для кожної температури
nequil = 1e5 # Кількість кроків рівноваги для кожної T
```

Встановлюємо верхню та нижню межі температури та крок між ними, таким чином визначаючи кількість температур, для яких виконуватимуться обчислення. Відомо, що температура фазового переходу (критична температура) для цієї моделі дорівнює  $T_c \approx 2,3$ , тому обираємо обмеження, які відповідають цій величині.

```
T_lo = 1.5 # Нижня межа температури
T_hi = 3 # Верхня межа температури
dT = 0.1 # Крок, з яким змінюється температура
nscans = as.integer((T_hi - T_lo)/dT) + 1 # кількість кроків
```

Налаштуємо таблицю (матриця  $M$ ) для запису результатів обчислень на кожному кроці:

```
Створення матриці результатів
M = matrix(nrow = nscans, ncol = 5, byrow=TRUE,
 dimnames=list(rep("", nscans), c("T", "E_av", "Cv", "Mag_av", "Mag_sus")))
```

Створимо функцію  $Ann$  ( $A$ ,  $m$ ,  $n$ ), яка визначає найближчі сусіди для  $(m, n)$ -спіну в  $A$ , з особливим опрацюванням границь матриці для задоволення періодичних граничних умов.

```
Ann = function(A, m, n) {
 if (m == nr) Ann1 = A[1,n] else Ann1 = A[m+1,n] # знизу
 if (n == 1) Ann2 = A[m,nc] else Ann2 = A[m,n-1] # зліва
 if (m == 1) Ann3 = A[nr,n] else Ann3 = A[m-1,n] # зверху
 if (n == nc) Ann4 = A[m,1] else Ann4 = A[m,n+1] # зправа
 return(Ann1 + Ann2 + Ann3 + Ann4)
}
```

Для кожної температури розрахунок починається заново: ініціалізуються змінні в одиницях, обраних так, щоб теплова енергія  $\beta T$  дорівнювала одиниці, а також енергія спінової взаємодії  $J$  (яка у такий спосіб не бере явно участь в обчисленнях). Енергія та намагніченість встановлюються рівними нулю.

```
for (isc in 1:nscans) { # цикл по T
 temp = T_hi - dT*(isc - 1)
 # ініціалізація змінних
 beta = 1/temp
 oc = 0 # output count
 E_av = 0
 E2_av = 0
 mag_av = 0
 mag2_av = 0
```



Матриця заповнюється у шаховому порядку з чергуванням спінів, що вказують догори та донизу.

```
Початкове налаштування спінів у шаховому порядку
A[1,1] = 1
for (i in 1:(nr - 1)) A[i+1,1] = -A[i,1]
for (j in 1:(nc - 1)) A[,j+1] = -A[,j]
```

Далі для кожного значення температури використовується алгоритм Метрополіса-Гастінгса (Metropolis-Hastings) для прийняття або відхилення поточної спроби. Перші *nequil* кроків використовуються для досягнення рівноваги, решта – для накопичення статистичних даних про енергію та намагніченість.

```
for (ipass in 0:npass) { # кроки за методом Монте-Карло для кожної температури
 if (ipass > nequil) {
 oc = oc + 1 # кількість виведень
 mag = sum(A)/(nr*nc)
 mag_av = mag_av + mag
 mag2_av = mag2_av + mag^2
 E = 0
 for (m in 1:nr) {
 for (n in 1:nc) {
 E = E - A[m,n]*Ann(A,m,n)
 }
 }
 E = E/(2*nr*nc)
 E_av = E_av + E
 E2_av = E2_av + E^2
 }
 # Обираємо випадковий спін для зміни
 m = sample(nr,1,replace=TRUE)
 n = sample(nc,1,replace=TRUE)
 ts = -A[m,n] # Розгортаємо спін (змінимо його орієнтацію)
 dU = -2*ts*Ann(A,m,n)
 log_eta = log(runif(1))
 if(-beta*dU > log_eta) A[m,n] = ts
} # кінець циклу, що реалізує метод Монте-Карло
```

Заповнюємо рядок значеннями температури, енергії та намагніченості:

```
M[isc,1] = temp
M[isc,2] = E_av/oc
M[isc,3] = beta^2*(E2_av/oc - (E_av/oc)^2)
M[isc,4] = abs(mag_av/oc)
M[isc,5] = beta*(mag2_av/oc - (mag_av/oc)^2)
cat(c(temp, mag_av, mag2_av, E_av, E2_av), "\n")
} # кінець циклу за T
```

Друкуємо та візуалізуємо результати (рис. 2.32):

```
M # виводимо матрицю M
будуємо графіки
```

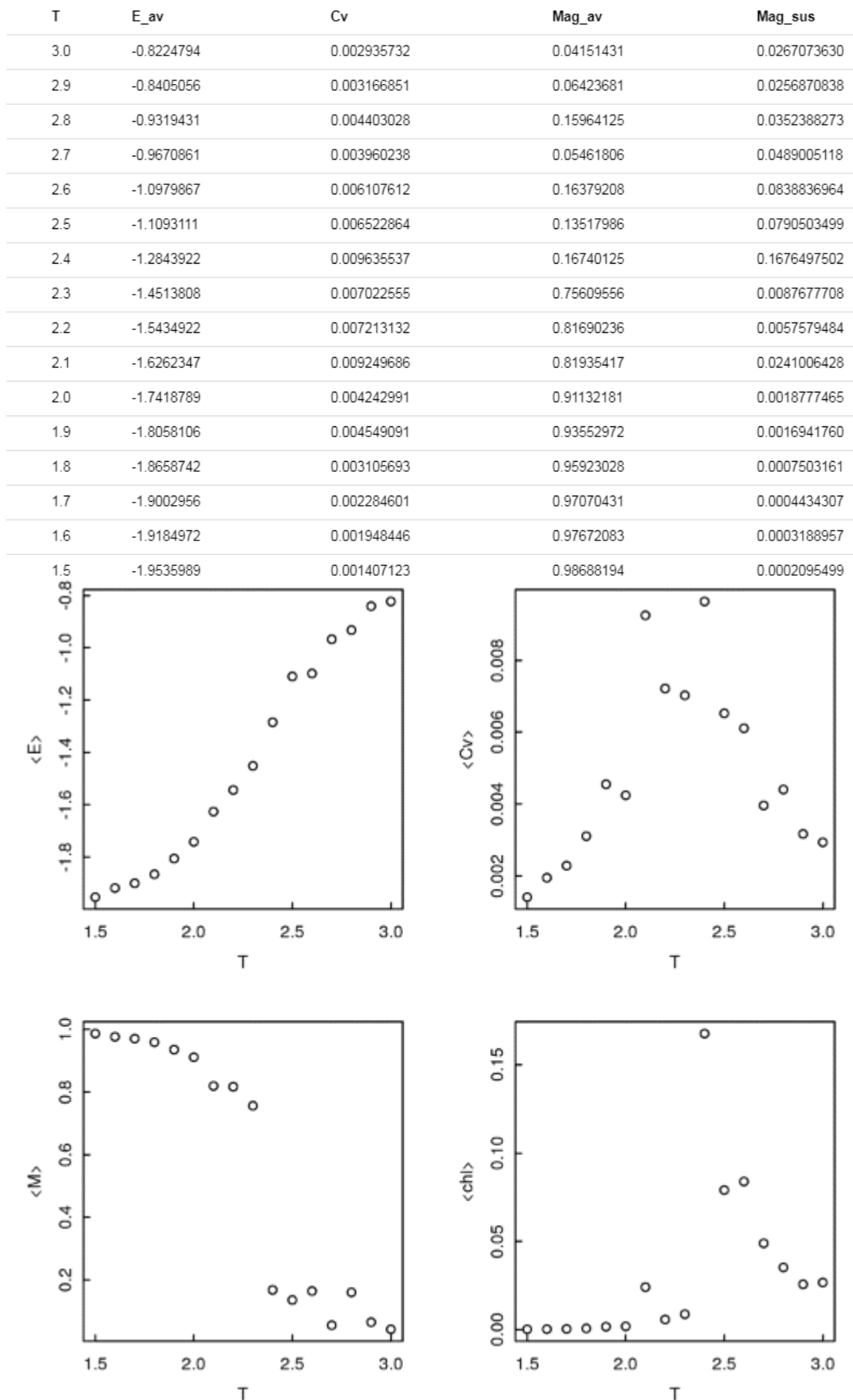


Рис. 2.32. Моделювання двовимірної моделі Ізінга методом Монте-Карло у середовищі CoCalc з використанням R

```

par(mar=c(4,4,1.5,1.5), mex=.8, mgp=c(2,.5,0), tcl=0.3)
par(mfrow=c(2,2))
plot(M[,1], M[,2], xlab="T", ylab="<E>")
plot(M[,1], M[,3], xlab="T", ylab="<Cv>")
plot(M[,1], M[,4], xlab="T", ylab="<M>")
plot(M[,1], M[,5], xlab="T", ylab="<chi>")

```

При розгляді даної моделі доцільно звернути увагу студентів на її зв'язок з іншими моделями – зокрема, нейромережевими. Так, активність нейронів у мозку також може бути статистично змодельована. Так само як і напрямок спіну, активність нейрону може бути позначена «+» (активний) або «-» (неактивний).

Оскільки нейронна активність у довільний момент часу моделюється незалежними бітами, Дж. Дж. Хопфілд (John Joseph Hopfield) запропонував динамічну модель Ізінга як перше наближення нейронної мережі, яка здатна до навчання [38, с. 2556].

Під час моделювання знадобилось 15 хвилин, щоб обчислити значення для 16 температур. Результати моделювання узгоджуються із теорією.

Очікувані результати: енергія зростає із зростанням  $T$ , а намагніченість різко зменшується поблизу критичної температури  $T_c$  і прямує до нуля вище цієї температури. Питома теплоємність та магнітна сприйнятливість, які пов'язані з коливаннями в системі, мають піки навколо  $T_c$ .

**%reset** список\_змінних – якщо список\_змінних не порожній, встановлює усі значення з нього на початкові. Якщо список\_змінних не заданий, то видаляються усі змінні користувача, виконується повернення усім глобальним змінним їх значень за замовчуванням, і очищуються всі інтерфейси до інших систем комп'ютерної математики.

**%ruby** – надає доступ до інтерпретатора Ruby. У навчанні основ математичної інформатики його використання може бути доцільним у процесі навчання першого змістового модулю «Теорія алгоритмів». Так, використовуючи відомості з [39, с. 266-288], студентам можна запропонувати розглянути алгоритми сортування (бульбашковий, швидкий, злиттям), знаходження простих чисел, піднесення до степеню та інші теоретико-числові алгоритми.

**%runfile** – виконує ті самі дії, що й %load (для забезпечення сумісності з Jupyter Notebook);

**%sage0** – завантажує об'єктні файли SageMath (аналогічно до директиви execfile у Python).

**%scilab** – надає інтерфейс до інтерпретатора системи комп'ютерної математики Scilab, переваги використання якої у 2011 році визнано Міністерство національної освіти, вищої освіти і науки Франції, надавши Scilab знак визнання його педагогічної значущості для навчання математики «Reconnu d'Intérêt Pédagogique» [180]. CoCalc надають доступ лише до обчислювального ядра Scilab у текстовому режимі з можливістю побудови елементарних графіків.

**%script** використовується для запуску довільної команди оболонки, уведення до якої виконується із комірки в CoCalc, шляхом розміщення у окремому рядку %script ('команда аргументи') або %script (['команда', 'аргумент1', 'аргумент2', ...]) та послідовності команду наступних рядках. %script можна використовувати в однорядковому режимі. Наприклад:

```
%script('gp -q') factor(2^97 - 1)
```

запустить сеанс gp, подасть на стандартне введення 'factor(2^97-1)' і відобразить отримане розкладання на множники. Результат роботи команди зберігається в атрибуті stdout. Наприклад:

```
s = script('gp -q')
%s factor(2^97-1)
s.stdout
```

**%sh** – надає доступ до інтерпретатора bash для виконання команд операційної системи Linux. Даний спосіб є певною альтернативою використання терміналу:

```
%sh
echo "hi"
pwd
ls -l
```

Також можна просто використати sh не як декоратор, а як функцію:

```
sh('pwd')
```

Для перенаправлення результату виконання команди необхідно

використовувати `%capture`:

```
%capture(stdout='output')
%sh pwd
```

Після виконання змінна `output` міститиме поточний каталог.

**%singular** – надає інтерфейс до Singular – системи комп’ютерної алгебри для поліноміальних обчислень з особливим акцентом на потреби комутативної та некоммутативної алгебри, алгебраїчної геометрії та теорії сингулярності [68].

**%time** – визначає тривалість виконання програмного коду у CoCalc.

**%timeit** – визначає середню тривалість виконання програмного коду у CoCalc.

**%typeset\_mode** – переключення режимів виведення між неформатованим текстом та форматуваним у LaTeX. Наприклад:

```
typeset_mode() # виведення буде виконуватись за допомогою LaTeX
typeset_mode(False) # буде виконуватись неформатоване виведення
typeset_mode(True, display=False) # виведення буде виконуватись за
допомогою LaTeX, але меншим шрифтом та вирівняне не за центром
```

**%var** – створює символні змінні та розміщує їх у глобальному просторі імен.

**%wiki** – є еквівалентом `%pandoc('mediawiki')`, надаючи можливість форматування за розміткою MediaWiki [52], розміщеного одразу після `%wiki` або у наступних рядках. MediaWiki є одним із найпотужніших вікі-рушіїв, який написаний спеціально для Вікіпедії і використовується у багатьох інших проектах фонду Вікімедіа. MediaWiki надає можливість опрацювання тексту як у власному форматі, так і у форматах HTML та TeX (для формул), можливість завантаження зображень або інших файлів, а також інші можливості.

У навчанні основ математичної інформатики можна використовувати різні *хмаро зорієнтовані системи підтримки навчання*. На думку М. А. Кислової, доцільним є використання Google Classroom – хмарного сервісу, що надає можливість викладачам створювати та розміщувати завдання в електронному вигляді, а студентам – обирати необхідне завдання та виконувати його; при цьому надається можливість систематизації завдань та виконаних

студентами робіт у структуру папок та документів на Google Drive; надає можливість організувати процес навчання через Інтернет: створювати класи для навчання та додавати в них учасників, організувати тематичні обговорення, отримувати та виконувати завдання, організувати індивідуальні заняття, зберігати всі документи у структурі папок, оновлювати список виконаних робіт у реальному часі тощо [133, с. 96].

Google Classroom надає можливість поєднання багатьох продуктів Google в одній системі для організації роботи викладача та студента: створення завдань, організація комунікації, зберігання та розповсюдження завдань, створених викладачем та розв'язаних студентом. Робота викладача та студента здійснюється через Google Drive, сервіс файлового хостингу Google; Gmail використовується для забезпечення комунікації.

На сторінці завдань розміщуються завдання для самостійного виконання студентами (студентам достатньо просто натиснути на завдання, щоб приступити до його виконання). Відомості про здані роботи оновлюється в реальному часі, і викладач може оперативно перевірити всі роботи, поставити оцінки і додати свої коментарі.

Перевагами використання Google Classroom є:

1) просте налаштування: викладачі можуть самі додавати студентів для роботи в середовищі або надавати їм код для реєстрації в якості слухачів курсу;

2) економія часу: створення, перевірка та оцінка завдань здійснюються в одному сервісі;

3) ефективна організація навчального процесу: у Google Classroom викладачі можуть розсилати оголошення і починати обговорення, а студенти можуть обмінюватися один з одним навчальними матеріалами і відповідати на задані викладачем запитання;

4) організація роботи з навчальними матеріалами: студенти бачать на сторінці завдань, які роботи ще не здані, а всі матеріали курсу автоматично додаються в їх папки на Google Диску;

5) доступність і безпека: Google Classroom безкоштовний і надійний

ресурс для навчальних закладів [133, с. 100-101].

Інша хмаро зорієнтована система підтримки навчання, що може бути використана у навчанні основ математичної інформатики – MoodleCloud [54]. MoodleCloud надає послугу – повнофункціональний сайт Moodle, на якому можна створювати курси, надавати навчальні ресурси, дозволяти студентам складати завдання та ін. Користувач може обрати один із трьох пакетів послуг, які масштабуються відповідно до його потреб: Moodle для вільного доступу, Starter і Moodle – для корпоративного.

Як було показано вище, провідною хмаро зорієнтованою навчальною лабораторією у навчанні основ математичної інформатики є система CoCalc, у якій також можуть бути розміщені хмаро зорієнтовані додаткові науково-навчальні матеріали – інформаційні ресурси за хмарною моделлю доступу, які сприяють доповненню і розширенню уявлень про об'єкти і процеси, що є предметом вивчення. Інший спосіб їх подання – використання різних сторонніх сервісів для розміщення презентацій, аудіо, відео та ін.

Аналогічно хмаро зорієнтовані комунікаційні засоби, що використовуються у навчанні основ математичної інформатики – це насамперед засоби, убудовані в CoCalc. Провідним засобом обміну текстовими та графічними повідомленням, на думку М. В. Попель [197, с. 51-52], є файл-ресурс типу sage-chat – текстовий чат (рис. 2.33). Текст повідомлення можна форматувати за допомогою тегів HTML та команд вікі-розмітки. Повідомлення математичного змісту може бути подане у звичній математичній нотації за допомогою команд LaTeX.

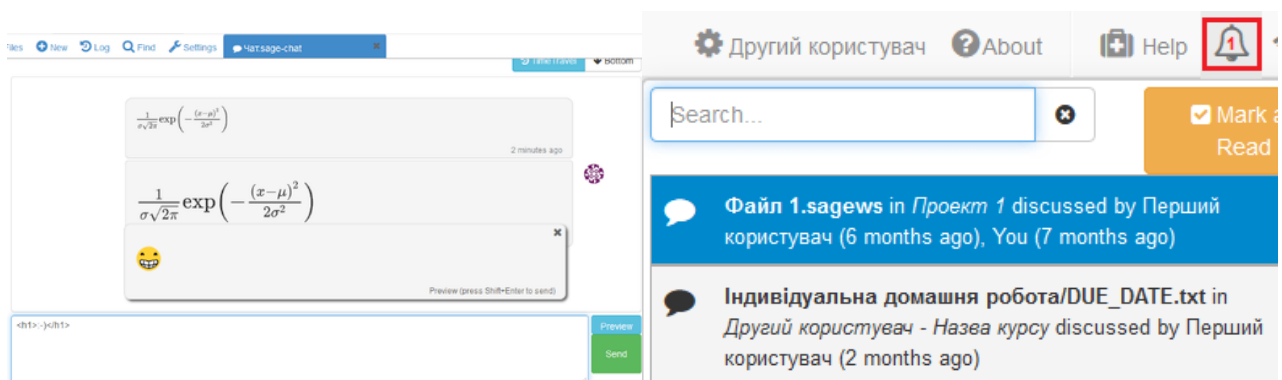


Рис. 2.33. Чат як засіб комунікації у CoCalc

Google Hangouts – це програмне забезпечення для миттєвого обміну повідомленнями та відеоконференцій, розроблене компанією Google. Hangouts надає можливість спілкуватися до 10 користувачам у форматі групових відеоконференцій. За допомогою цієї служби можна проводити вебінари і спілкуватися у відеочатах. Історії чатів зберігаються на серверах Google, що дозволяє синхронізувати їх між пристроями. Фотографії, якими учасники обмінюються під час вебінару, автоматично завантажуються у спеціальний альбом Google+. Також учасники можуть ділитися файлами Google-диска, спільно складати замітки до зустрічі і записувати ідеї на загальній віртуальній дошці. Hangouts інтегрований з календарем Google, так що користувачі можуть планувати відеозустрічі за допомогою календаря [133, с. 96].

Прикладом електронної пошти, що може використовуватись у навчанні основ математичної інформатики, є корпоративна пошта Gmail в Google Apps. Основні переваги – наявність необмеженого простору для зберігання відомостей, можливість корпоративного доступу, розширені можливості електронної пошти (пошук, перегляд ланцюжків листів, вбудований чат) [133, с. 99].

Приклад використання *хмаро зорієнтованої операційної системи* на основі інфраструктури Amazon AWS подано у п. 1.3.

У якості *хмаро зорієнтованих засобів зберігання даних* можна використовувати Dropbox, OneDrive, Google Drive, iCloud та інші хмарні сховища. Так, можливість використання Dropbox як засобу організації спільної роботи студентів над навчально-дослідницькими проектами розглянуто у [178, с. 131-132].

Методику використання *хмаро зорієнтованих офісних пакетів* (зокрема, таких їх складових, як *хмаро зорієнтовані текстові процесори, хмаро зорієнтовані табличні процесори, хмаро зорієнтовані засоби підготовки презентацій*) як засобу організації спільної роботи над навчально-дослідницькими проектами розглянуто у [175]. Так, хмаро зорієнтований офісний пакет Office 365 [56] включає в себе хмаро зорієнтований текстовий



процесор Word Online, хмаро зорієнтований табличний процесор Excel Online, хмаро зорієнтований засіб підготовки презентацій PowerPoint Online, а також *додаткові хмаро зорієнтовані компоненти* – хмаро зорієнтований засіб організації спільної проектної діяльності Microsoft Teams, хмаро зорієнтований «блокнот» OneNote Online та ін. [57].

Приклад використання *хмаро зорієнтованої системи управління базами даних* на основі MySQL подано у п. 1.3.

Заслуговує на увагу також хмаро зорієнтований офісний пакет Google Apps [101], насамперед – через простоту організації спільної роботи та наявність великої кількості розширень. Так, у процесі виконання студентами індивідуального навчально-дослідницького завдання з розробки нейронної мережі в якості доцільного засобу можна запропонувати хмаро зорієнтовані електронні таблиці Google Sheets із додатковим хмаро зорієтованим компонентом (доповненням) Solver (аналогом доповнення «Пошук рішення» для Excel Online).

Розглянемо відповідну методику використання на прикладі класичного набору даних – ірисів Фішера: багатовимірний набір даних для задачі класифікації, за допомогою якого англійський статистик та біолог Р. Е. Фішер (Ronald Aylmer Fisher) у 1936 році продемонстрував розроблений ним метод дискримінантного аналізу. Іриси Фішера складаються з даних про 150 вимірювань ірисів трьох видів (рис. 2.34) – ірису щетинистого (*Iris setosa*), ірису віргінського (*Iris virginica*) та ірису різнокольорового (*Iris versicolor*), по 50 вимірювань для кожного виду, зібраних Е. Ш. Андерсоном (Edgar Shannon Anderson) [5]. Для кожного екземпляра вимірювалися чотири характеристики (рис. 2.35): довжина зовнішньої частки оцвітини (sepal length – SL), ширина зовнішньої частки оцвітини (sepal width – SW), довжина внутрішньої частки оцвітини (petal length – PL), ширина внутрішньої частки оцвітини (petal width – PW).

Для того, щоб зробити обґрунтований висновок про тип ірису, побудуємо тришарову нейронну мережу із такою архітектурою (рис. 2.36):



Рис. 2.34. Зовнішній вигляд ірисів Фішера

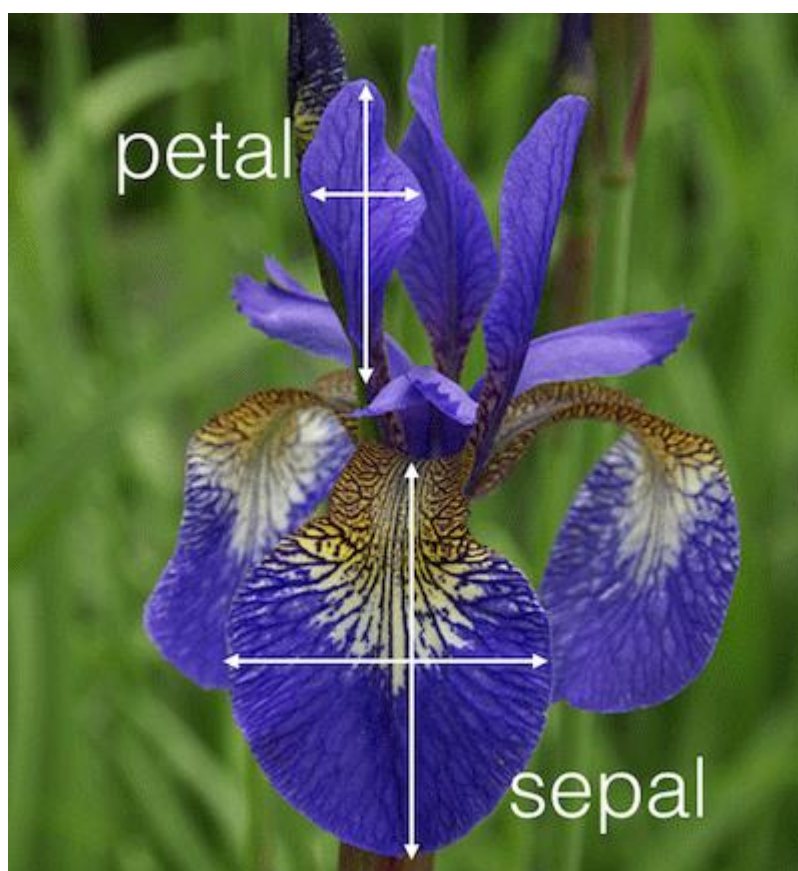


Рис. 2.35. Вимірювані характеристики ірисів Фішера

– вхідний шар – чотиривимірний арифметичний вектор  $(x_1, x_2, x_3, x_4)$ , компонентами якого є чотири відповідні вимірювані характеристики ірисів Фішера (SL, SW, PL, PW), нормовані згідно з обраною функцією активації мережі;

– прихований шар матиме розмірність 9 – мінімальна необхідна кількість згідно теореми Колмогорова – та описуватиметься вектором  $(h_1, h_2, h_3, h_4, h_5, h_6,$

$h_7, h_8, h_9$ );

– вихідний шар – тривимірний арифметичний вектор  $(y_1, y_2, y_3)$ , компонентами якого є ймовірності того, що набір даних відповідає одному із трьох видів ірисів.

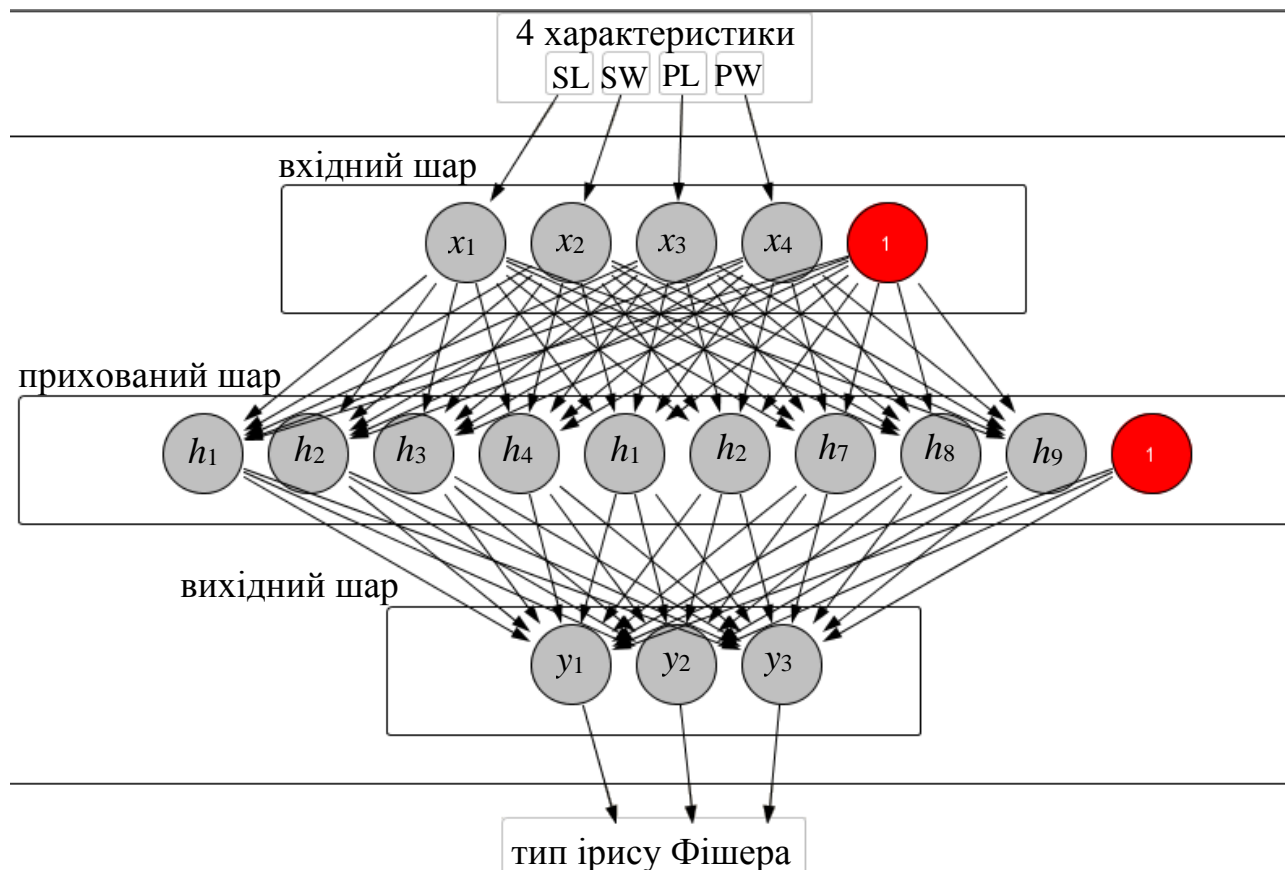


Рис. 2.36. Архітектура нейронної мережі для розв'язання задачі класифікації ірисів Фішера

До нейронів вхідного та прихованого шарів додається нейрон зміщення (bias), значення якого завжди дорівнює одиниці (на рис. 2.36 його позначено червоним кольором). Особливістю нейронів зміщення є те, що вони не мають вхідних синапсів, а тому не можуть бути розташовані на вихідному шарі.

Спочатку внесемо ірисі Фішера до електронних таблиць. Виходячи з того, що цей набір даних найчастіше зустрічається в англійськомовних джерелах, для зручності його опрацювання виконаємо налаштування електронної таблиці, встановивши американську локалізацію (рис. 2.37).

У комірки таблиці занесемо наступні значення:

## Параметри електронної таблиці

**Загальні**    Обчислення

---

Мова

Сполучені Штати ▾      Це вплине на форматування функцій, дат, валюти тощо.

---

Часовий пояс

(GMT+02:00) Kiev ▾      Час занесення записів у журнал електронної таблиці відповідатиме цій часовій зоні. Це впливатиме на всі функції, пов'язані з часом.

---

Мова інтерфейсу: [Українська](#)

Завжди використовувати англійські назви функцій

Рис. 2.37. Налаштування параметрів електронної таблиці

A1    Вихідні дані

A2    SL

B2    SW

C2    PL

D2    PW

E2    Вид ірису

Діапазон комірок A3:E152 міститиме власне ірисис Фішера (рис. 2.38).

|   | A                   | B         | C         | D         | E                |
|---|---------------------|-----------|-----------|-----------|------------------|
| 1 | <i>Вихідні дані</i> |           |           |           |                  |
| 2 | <b>SL</b>           | <b>SW</b> | <b>PL</b> | <b>PW</b> | <b>Вид ірису</b> |
| 3 | 5.10                | 3.50      | 1.40      | 0.20      | <i>setosa</i>    |
| 4 | 4.90                | 3.00      | 1.40      | 0.20      | <i>setosa</i>    |
| 5 | 4.70                | 3.20      | 1.30      | 0.20      | <i>setosa</i>    |
| 6 | 4.60                | 3.10      | 1.50      | 0.20      | <i>setosa</i>    |
| 7 | 5.00                | 3.60      | 1.40      | 0.20      | <i>setosa</i>    |
| 8 | 5.40                | 3.90      | 1.70      | 0.40      | <i>setosa</i>    |

Рис. 2.38. Фрагмент таблиці із ірисисами Фішера

Подати дані цього набору безпосередньо до вхідного шару не можна –

значення 4 характеристик виходить за межі діапазону [0; 1], тому наступним кроком є нормування (приведення даних стовпців А, В, С та D до вказаного діапазону) та кодування видів ірисів із стовпця Е.

Кожен вид ірису закодуємо тривимірним арифметичним вектором: для ірису виду  $i$  (*Iris setosa* – 1, *Iris versicolor* – 2, *Iris virginica* – 3) встановлюємо  $i$ -тий компонент у 1, а інші – у 0. Для цього уводимо наступні значення у комірки:

G1 кодування  
 G2 setosa  
 H2 versicolor  
 I2 virginica  
 G3 =if(\$E3=G\$2,1,0)

Далі копіюємо формулу із комірки G3 у діапазон G3:I152 та отримуємо такі еталонні коди трьох видів ірисів: для ірису щетинистого – (1, 0, 0), для ірису різнокольорового – (0, 1, 0), для ірису віргінського – (0, 0, 1).

Нормування виконуватимемо для кожного стовпця окремо. Для цього знайдемо для них мінімальні та максимальні значення, увівши у комірки такі значення:

E154 min  
 E155 max  
 A154 =min(A3:A152)  
 A155 =max(A3:A152)

Поширимо комірки A154:A155 на діапазон B154:D155 та уведемо в комірки такі значення:

K1 нормування  
 K2  $x_1$   
 L2  $x_2$   
 M2  $x_3$   
 N2  $x_4$   
 K3 =(A3-A\$154)/(A\$155-A\$154)

Останню формулу розширюємо на діапазон K3:N152. Сутність її легко

зрозуміти із виразу:

$$\text{нормоване значення} = \frac{\text{вихідне значення} - \text{мінімальне значення}}{\text{максимальне значення} - \text{мінімальне значення}}.$$

За такого підходу мінімальне значення нормується до 0, а максимальне – до 1.

Відповідно до обраної архітектури, приєднаємо до 4 нейронів вхідного шару нейрон зміщення. Для цього у комірку O2 внесемо його назву ( $x_5$ ), а у діапазон O3:O152 – його значення (1). На цьому кроці вхідний шар сформовано у вигляді ( $x_1, x_2, x_3, x_4, x_5$ ).

Наступний крок – передавання сигналу із вхідного шару нейронної мережі на прихований. Для визначення сили сигналу необхідно мати вагові коефіцієнти нейронної мережі. Позначимо через  $w^{xh}_{ij}$  ваговий коефіцієнт синапсу, що пов'язує нейрон  $x_i$  ( $i = 1, 2, 3, 4, 5$ ) вхідного шару із нейроном  $h_j$  ( $j = 1, 2, \dots, 9$ ) прихованого шару, а через  $w^{hy}_{jk}$  ваговий коефіцієнт синапсу, що пов'язує нейрон  $h_j$  прихованого шару із нейроном  $y_k$  ( $k = 1, 2, 3$ ) вихідного шару. Тоді сила сигналу, що приходить до нейрону  $h_j$  прихованого шару, визначатиметься як скалярний добуток значень сигналу на вхідному шарі та відповідних вагових коефіцієнтів. Для визначення сигналу, який піде далі на вихідний шар, застосуємо логістичну функцію активації  $f(S) = 1/(1+e^{-S})$ , де  $S$  – відповідний скалярний добуток. Формули для визначення сигналів на прихованому та вихідному шарах матимуть вигляд:

$$h_j = f\left(\sum_{i=1}^{4+1} x_i w_{ij}^{xh}\right), y_k = f\left(\sum_{j=1}^{9+1} x_i w_{jk}^{hy}\right).$$

Відповідно необхідно створити дві матриці: матриця  $w^{xh}$  розмірністю  $5 \times 9$  містить вагові коефіцієнти зв'язків 5 нейронів вхідного шару (перші чотири містять нормовані характеристики ірисів Фішера, а п'ятий є нейроном зміщення) з нейронами прихованого шару, а матриця  $w^{hy}$  розмірністю  $10 \times 3$  містить вагові коефіцієнти зв'язків 10 нейронів прихованого шару (з них дев'ять обчислюються, а десятий є нейроном зміщення) з нейронами вихідного

шару. Для «ненавченої» нейронної мережі початкові значення вагових коефіцієнтів можна встановити або у випадковий спосіб, або залишити невизначеними, або рівними нулеві. Для реалізації останнього способу заповнимо комірки такими значеннями:

R1  $w^{xh}$   
 Q2 вхідний/прихований  
 R2 1  
 S2 =R2+1  
 Q3 1  
 Q4 =Q3+1  
 R3 0  
 R9  $w^{hy}$   
 Q10 прихований/вихідний  
 R10 1  
 S10 =R10+1  
 Q11 1  
 Q12 =Q11+1  
 R11 0

Для створення матриць необхідно виконати копіювання комірки R3 у діапазон R3:Z7, R11 – у R11:T20, S2 – у T2:Z2, Q4 – у Q5:Q7, S10 – у T10, Q12 – у Q13:Q20 (рис. 2.39).

Для обчислення скалярного добутку вектор-рядка значень вхідного шару на вектор-стовпець матриці вагових коефіцієнтів  $w^{xh}$  доцільно скористатись функцією множення матриць:

AB1 обчислення прихованого шару  
 AB2  $h_1$   
 AC2  $h_2$   
 AD2  $h_3$   
 AE2  $h_4$   
 AF2  $h_5$

AG2  $h_6$ AH2  $h_7$ AI2  $h_8$ AJ2  $h_9$ AK2  $h_{10}$ AB3 =  $1/(1+\exp(-\text{mmult}(\$K3:\$O3,R\$3:R\$7)))$ 

AK3 1

| fx |           | =(A3-A\$154)/(A\$155-A\$154) |           |            |        |        |        |    |                     |   |   |   |   |   |   |   |   |                     |      |      |      |      |      |      |      |      |      |      |
|----|-----------|------------------------------|-----------|------------|--------|--------|--------|----|---------------------|---|---|---|---|---|---|---|---|---------------------|------|------|------|------|------|------|------|------|------|------|
|    | G         | H                            | I         | J          | K      | L      | M      | N  | O                   | P | Q | R | S | T | U | V | W | X                   | Y    | Z    |      |      |      |      |      |      |      |      |
| 1  | кодування |                              |           | нормування |        |        |        |    |                     |   |   |   |   |   |   |   |   | wxh                 |      |      |      |      |      |      |      |      |      |      |
| 2  | setosa    | versicolor                   | virginica | x1         | x2     | x3     | x4     | x5 | вихідний/прихований |   |   |   |   |   |   |   |   | 1                   | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    |      |      |
| 3  | 1         | 0                            | 0         | 0.2222     | 0.6250 | 0.0678 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 1                   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4  | 1         | 0                            | 0         | 0.1667     | 0.4167 | 0.0678 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 2                   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 5  | 1         | 0                            | 0         | 0.1111     | 0.5000 | 0.0508 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 3                   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6  | 1         | 0                            | 0         | 0.0833     | 0.4583 | 0.0847 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 4                   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 7  | 1         | 0                            | 0         | 0.1944     | 0.6667 | 0.0678 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 5                   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 8  | 1         | 0                            | 0         | 0.3056     | 0.7917 | 0.1186 | 0.1250 | 1  |                     |   |   |   |   |   |   |   |   | why                 |      |      |      |      |      |      |      |      |      |      |
| 9  | 1         | 0                            | 0         | 0.0833     | 0.5833 | 0.0678 | 0.0833 | 1  |                     |   |   |   |   |   |   |   |   | прихований/вихідний |      |      |      |      |      |      |      |      |      |      |
| 10 | 1         | 0                            | 0         | 0.1944     | 0.5833 | 0.0847 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 1                   | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 11 | 1         | 0                            | 0         | 0.0278     | 0.3750 | 0.0678 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 2                   | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 12 | 1         | 0                            | 0         | 0.1667     | 0.4583 | 0.0847 | 0.0000 | 1  |                     |   |   |   |   |   |   |   |   | 3                   | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 13 | 1         | 0                            | 0         | 0.3056     | 0.7083 | 0.0847 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 4                   | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 14 | 1         | 0                            | 0         | 0.1389     | 0.5833 | 0.1017 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 5                   | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 15 | 1         | 0                            | 0         | 0.1389     | 0.4167 | 0.0678 | 0.0000 | 1  |                     |   |   |   |   |   |   |   |   | 6                   | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 16 | 1         | 0                            | 0         | 0.0000     | 0.4167 | 0.0169 | 0.0000 | 1  |                     |   |   |   |   |   |   |   |   | 7                   | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 17 | 1         | 0                            | 0         | 0.4167     | 0.8333 | 0.0339 | 0.0417 | 1  |                     |   |   |   |   |   |   |   |   | 8                   | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 18 | 1         | 0                            | 0         | 0.3889     | 1.0000 | 0.0847 | 0.1250 | 1  |                     |   |   |   |   |   |   |   |   | 9                   | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 19 | 1         | 0                            | 0         | 0.3056     | 0.7917 | 0.0508 | 0.1250 | 1  |                     |   |   |   |   |   |   |   |   | 10                  | 0.00 | 0.00 | 0.00 |      |      |      |      |      |      |      |
| 20 | 1         | 0                            | 0         | 0.2222     | 0.6250 | 0.0678 | 0.0833 | 1  |                     |   |   |   |   |   |   |   |   |                     |      |      |      |      |      |      |      |      |      |      |

Рис. 2.39. Фрагмент електронної таблиці після кодування та нормування вихідних даних та створення матриць вагових коефіцієнтів

Далі копіюємо комірку АК3 у діапазон АК4:АК152, а АВ3 – у АВ3:АJ152.

Ураховуючи, що всі елементи матриці вагових коефіцієнтів  $w^{xh}$  дорівнюють нулю, після копіювання формул всі обчислювані елементи прихованого шару дорівнюватимуть 0,5.

Аналогічно виконаємо обчислення елементів вихідного шару:

AM1 обчислення вихідного шару

AM2  $y_1$

AN2  $y_2$

AO2  $y_3$

AM3 =  $1/(1+\exp(-\text{mmult}(\$AB3:\$AK3,R\$11:R\$20)))$



Далі копіюємо комірку AM3 у діапазон AM3:AO152 (рис. 2.40).

| fx | =1/(1+exp(-mmult(\$AB3:\$AK3,\$S\$11:\$S\$20))) |        |        |        |        |        |        |        |        |     |                           |        |        |        |
|----|-------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|-----|---------------------------|--------|--------|--------|
|    | AB                                              | AC     | AD     | AE     | AF     | AG     | AH     | AI     | AJ     | AK  | AL                        | AM     | AN     | AO     |
| 1  | обчислення прихованого шару                     |        |        |        |        |        |        |        |        |     | обчислення вихідного шару |        |        |        |
| 2  | h1                                              | h2     | h3     | h4     | h5     | h6     | h7     | h8     | h9     | h10 |                           | y1     | y2     | y3     |
| 3  | 0.5000                                          | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 1   |                           | 0.5000 | 0.5000 | 0.5000 |
| 4  | 0.5000                                          | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 1   |                           | 0.5000 | 0.5000 | 0.5000 |
| 5  | 0.5000                                          | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 1   |                           | 0.5000 | 0.5000 | 0.5000 |

Рис. 2.40. Фрагмент електронної таблиці обчислення прихованого та вихідного шару з початковими значеннями вагових коефіцієнтів

Навчання нейронної мережі відбувається шляхом варіювання вагових коефіцієнтів у такий спосіб, щоб із кожним кроком навчання різниця між обчисленими значеннями вихідного шару та бажаними (еталонними) зменшувалась. Для розв'язуваної задачі еталонними є тривимірні вектори, отримані у результаті кодування трьох видів ірисів.

Для визначення різниці між обчисленим та еталонним вихідними векторами застосуємо евклідову відстань:

AQ2 відстань

AR2 сума відстаней

AQ3 =sqrt((AM3-G3)^2+(AN3-H3)^2+(AO3-I3)^2)

AR3 =sum(AQ3:AQ152)

Далі копіюємо комірку AQ3 у діапазон AQ4:AQ152. Комірка AR3 міститиме загальне відхилення обчислених вихідних векторів від еталонних.

За такої постановки навчання нейронної мережі може розглядатися як задача оптимізації, у якій цільова функція – сума відстаней у комірці AR3 – підлягатиме мінімізації шляхом варіювання вагових коефіцієнтів матриць  $w^{xh}$  (діапазон R3:Z7) та  $w^{hy}$  (діапазон R11:T20). Для розв'язання цієї задачі стандартних засобів хмаро зорієнтованих електронних таблиць Google Sheets недостатньо, тому необхідним є встановлення додаткового хмаро зорієнтованого компоненту (доповнення) Solver (рис. 2.41).

На рис. 2.42 показані налаштування доповнення Solver для розв'язання

поставленої задачі: цільова функція (Set Objective) мінімізується (To: Min) шляхом зміни значень (By Changing) матриць вагових коефіцієнтів у діапазоні (Subject To) від  $-10$  до  $+10$  одним із методів оптимізації (Solving Method).

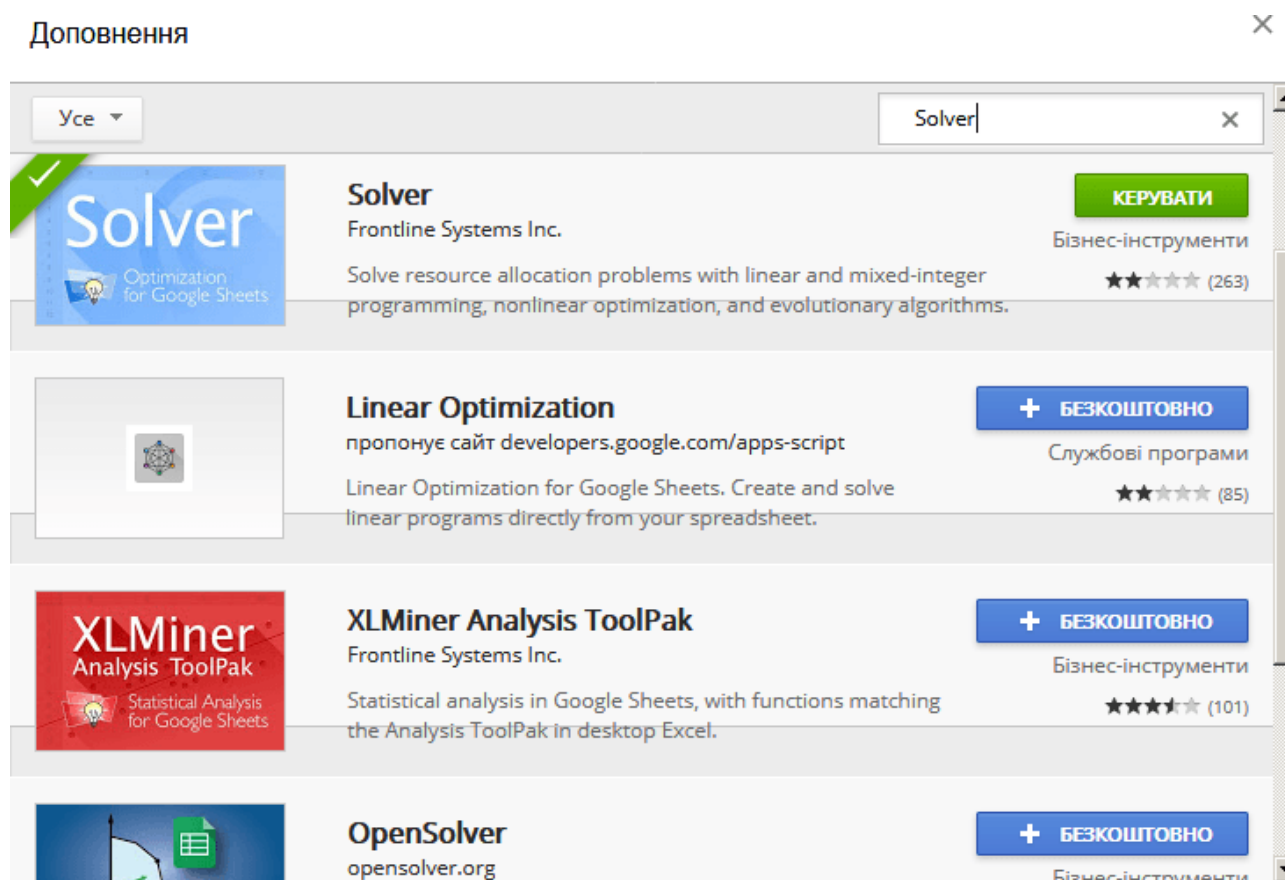


Рис. 2.41. Встановлення доповнення Solver у електронних таблицях Google

Для зменшення суми відстаней дії з Solver можуть бути виконані багаторазово: доцільно поекспериментувати із комбінованим використанням різних методів оптимізації, змінюючи межі варіювання вагових коефіцієнтів. При цьому необов'язково намагатись довести значення суми відстаней до нуля – це може бути й більше (достатньо невелике) значення (рис. 2.43).

Виходячи з обраного способу кодування, вихідний вектор фактично містить три ймовірності:  $y_i$  – ймовірність того, що даний екземпляр є ірисом виду  $i$ , де  $i = 1$  для *Iris setosa*,  $2$  для *Iris versicolor* та  $3$  для *Iris virginica*. Тоді для з'ясування того, який саме вид ірису описує поданий на вхід вектор (SL, SW, PL, PW), необхідно визначити компонент із найбільшою ймовірністю.

Для цього заповнимо комірки у такий спосіб:

**Solver**

Set Objective:  
AR3

To:  Max  Min  Value Of:

By Changing Variable Cells:  
R3:Z7,R11:T20

Subject to:

- R3:Z7 <= 10
- R3:Z7 >= -10
- R11:T20 <= 10
- R11:T20 >= -10

Add Change Delete

Solving Method:

- Standard LSGRG Nonlinear
- Standard LSGRG Nonlinear
- Standard LP/Quadratic
- Standard Evolutionary
- Standard Interval Global
- Standard SOCP Barrier

Рис. 2.42. Налаштування доповнення Solver

AT2 Обчислений вид ірису

AT3 =if(max(AM3:AO3)=AM3,\$G\$2,if(max(AM3:AO3)=AN3,\$H\$2,\$I\$2))

AU3 =if(AT3=E3,"правильно!","невірно")

Далі діапазон AT3:AU3 копіюємо у діапазон AT4:AU152.

Отриманий результат надає можливість унаочнити процес розпізнавання образів, змодельований в електронних таблицях. Побудовану модель будемо вважати адекватною, якщо для всіх 150 випадків стовпець AU міститиме значення "правильно!".

Для перевірки меж застосовності побудованої моделі спробуємо подати на вхід вектор значень, що не співпадає із жодним еталонним вхідним вектором.

Для цього скопіюємо 152 рядок таблиці у 158 та видалимо зміст комірок E158:I158, AQ158, AU158. У комірки A158:D158 уведемо усереднені значення, взяті з опису ірису різнокольорового з статті Е. Андерсона [5, с. 463]: 5.50, 2.75, 3.50 та 1.25. Тоді на вхідний шар подаються нормовані значення  $x_1 = 0.3333$ ,  $x_2 = 0.3125$ ,  $x_3 = 0.4237$ ,  $x_4 = 0.4792$ , на прихованому шарі обчислюються  $h_1 = 0.0206$ ,  $h_2 = 0.4419$ ,  $h_3 = 0.0005$ ,  $h_4 = 0.0001$ ,  $h_5 = 0.9993$ ,  $h_6 = 0.9993$ ,  $h_7 = 0.0001$ ,  $h_8 = 0.0288$ ,  $h_9 = 0.9991$ , а значення вихідного шару  $y_1 = 0.0000$ ,  $y_2 = 1.0000$ ,  $y_3 = 0.0000$ . Так як максимальне значення вихідного шару 1.0000 відповідає другому типу ірису, то можна зробити висновок про те, що ідентифіковано саме різнокольоровий ірис.

| SL |           |           |           |           |           |                           |           |           |           |    |          |                |
|----|-----------|-----------|-----------|-----------|-----------|---------------------------|-----------|-----------|-----------|----|----------|----------------|
| AF | AG        | AH        | AI        | AJ        | AK        | AL                        | AM        | AN        | AO        | AP | AQ       | AR             |
|    |           |           |           |           |           | обчислення вихідного шару |           |           |           |    |          |                |
|    | <b>h5</b> | <b>h6</b> | <b>h7</b> | <b>h8</b> | <b>h9</b> | <b>h10</b>                | <b>y1</b> | <b>y2</b> | <b>y3</b> |    | відстань | сума відстаней |
|    | 0.9984    | 0.1741    | 0.9998    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   | 1.4383         |
|    | 0.9974    | 0.1962    | 0.9947    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9975    | 0.1372    | 0.9992    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9971    | 0.2427    | 0.9980    | 0.0001    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9985    | 0.1677    | 0.9999    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9992    | 0.4072    | 0.9998    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9980    | 0.1907    | 0.9996    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9982    | 0.2323    | 0.9995    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9962    | 0.1928    | 0.9952    | 0.0001    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9972    | 0.2248    | 0.9985    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9989    | 0.2223    | 0.9999    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9979    | 0.2935    | 0.9996    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9968    | 0.1745    | 0.9978    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9959    | 0.0674    | 0.9995    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |
|    | 0.9993    | 0.0890    | 1.0000    | 0.0000    | 1.0000    | 1                         | 1.0000    | 0.0000    | 0.0000    |    | 0.0000   |                |

Set Objective: AR3

To:  Max  Min  Value Of:

By Changing: R3:Z7,R11:T20

Subject To: R3:Z7 <= 20  
R3:Z7 >= -20  
R11:T20 <= 20  
R11:T20 >= -20

Solving Method: Standard Interval Global

Рис. 2.43. Результат оптимізації

За описаною методикою використання хмаро зорієнтованих електронних таблиць як засобу навчання математичної інформатики студентів технічних університетів можна реалізувати всі основні задачі нейромережевого моделювання. Єдине обмеження – не стільки граничний обсяг електронної таблиці, скільки обсяг пам'яті та швидкість пристрою для її опрацювання – на пропонуваннях у спецкурсі проектах якщо й досягається, то лише стає поштовхом для зміни середовища моделювання на більш адекватне [234].

## 2.4 Організація, проведення і результати педагогічного експерименту

Розробка й апробація теоретичних положень дисертаційного дослідження проходили у три етапи:

- 1) аналітико-констатувальний (2003 – 2007 рр.);
- 2) проектувально-пошуковий (2008 – 2014 рр.);
- 3) формувально-узагальнювальний (2015 – 2018 рр.).

Завданням аналітико-констатувального етапу дослідження було вивчення існуючого стану процесу навчання математичної інформатики студентів технічних університетів та виділення вихідних положень дослідження. Для реалізації поставленого завдання було виконано аналіз науково-методичної літератури з інформаційно-комунікаційних технологій, досвід навчання інформатичних дисциплін студентів різних спеціальностей, що надало можливість визначити актуальність дослідження та сформулювати його гіпотезу, освітньо-кваліфікаційних програм та освітньо-кваліфікаційних характеристик бакалавра напряму підготовки 6.050102 «Комп'ютерна інженерія», на підставі яких були визначені виробничі функції, типові задачі діяльності, уміння та компетенції фахівця з інформаційних технологій, що відносяться до математичної інформатики.

Головну увагу на аналітико-констатувальному етапі дослідження було приділено уточненню змісту основ математичної інформатики та спрямуванню його на формування дослідницьких виробничих функцій. У результаті проведеної роботи було виявлено наступне:

1. Сучасна професійна підготовка майбутніх інженерів ІТ-профілю у вищих технічних навчальних закладах України відбувається у галузі знань «Інформатика та обчислювальна техніка» («Інформаційні технології») за декількома напрямами. Це призвело до того, що окремі розділи математичної інформатики – фундаментальної основи для всіх інформатичних напрямів підготовки – відносяться до різних навчальних дисциплін, що викладаються на різних курсах і, як правило, не пов'язані одна з одною.

2. Базова інформатична підготовка полягає в опануванні швидкозмінними технологіями, без урахування умов фундаменталізації інформатичної освіти.

Виявлені невідповідності:

– між цілями та змістом навчання основ математичної інформатики за різними напрямками;

– між науково-методичною літературою, в якій не знайшли відображення міжпредметні зв'язки інформатичних дисциплін та системний підхід до їх опанування;

– між сучасною тенденцією до переорієнтації базової інформатичної підготовки з опанування швидкозмінними технологіями на стабільні наукові основи інформатики зумовили вибір мети дослідження, виділення його об'єкту та предмету і формулювання завдань.

На проектувально-пошуковому етапі дослідження були проаналізовані існуючі програмні засоби навчання математичної інформатики студентів вищих навчальних закладів, розроблені методичні засади навчання основ математичної інформатики студентів технічних університетів засобами хмарних технологій, програмно-методичне забезпечення навчання основ математичної інформатики та методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

На початку етапу значна увага була приділена логічним основам математичної інформатики. Так, було виявлено, що початки булевої алгебри [94] є фундаментальною основою навчання математичної інформатики як у технічних, так й педагогічних ЗВО. Показано, що інтеграція математики, фізики та інформатики сприятиме посиленню прикладної спрямованості як навчання інформатики у ЗЗСО, так й у процесі професійної підготовки ІТ-фахівців. Визначено, що опанування логічних основ інформатики є необхідною умовою успішного навчання такої теми спецкурсу, як завадостійке кодування.

Наступним кроком стала розробка методичних основ навчання дисциплін напряму підготовки «Комп'ютерна інженерія» [153], у процесі якої було

виокремлено основні змістові блоки підготовки майбутніх фахівців на рівнях «бакалавр» та «спеціаліст». Показано, що технологічний уклін у ІТ-підготовці сприяє підвищенню рівня навчальних досягнень студентів, але водночас утруднює їх подальший професійний розвиток.

У статті [243] було визначено, що фундаменталізація навчання студентів технічних університетів – це цілеспрямований процес зміни змістової складової методичних систем навчання на основі: 1) виділення фундаментальної та технологічної складової змісту навчання; 2) математизації фундаментальної складової; 3) стабілізації технологічної складової на основі перспективних напрямів розвитку науки та технології. Провідним напрямом фундаменталізації підготовки майбутніх фахівців у галузі ІКТ є впровадження у загальнопрофесійну підготовку моделей та методів математичної інформатики – фундаменталізованої навчальної дисципліни, у якій стабілізація технологічної складової виконана на основі виділення класів програмних засобів її навчання, насамперед – систем комп'ютерної математики. При цьому об'єднання різних навчальних дисциплін у єдиний блок (метакурс) на основі вихідних положень математичної інформатики (теоретичної основи галузі знань «Інформатика та обчислювальна техніка») та мережних технологій (одного із застосувань інформатичної інженерії) створює умови для реалізації міжпредметних зв'язків та системного підходу в підготовці майбутніх фахівців у галузі ІКТ, а перспективним шляхом інтеграції різних навчальних дисциплін у єдиний метакурс «Математична інформатика» є мережеорієнтований підхід, за якого засоби навчання математичної інформатики переносяться у хмарне (Web) середовище.

У статті [168] було уведено авторське трактування хмарних технологій навчання як комп'ютерно орієнтованої складової педагогічної технології, яка відображає деяку формалізовану модель певного компонента змісту навчання і методики його подання у навчальному процесі, що представлена у цьому процесі педагогічними програмними засобами і передбачає використання хмарних ІКТ для розв'язування дидактичних завдань або їх фрагментів. Місце

хмарні технології навчання у системі інформаційно-комунікаційних технологій навчального призначення показано у [245].

Серед платформ для реалізації хмарних ІКТ особливу увагу на цьому етапі було приділено Amazon EC2 [239]. Показано, що дана платформа може бути використана у ЗВО України на безоплатній основі протягом навчального року; наведено умови та обмеження використання хмаро зорієнтованих ресурсів Amazon EC2. Інші програмні та апаратні засоби хмарних технологій систематизовано у роботі [244].

У роботі [241] встановлені зв'язки математичної інформатики з курсом фізики у технічному університеті. Показано, що основними засобами навчання математичної інформатики є мережеорієнтовані системи комп'ютерної математики, зокрема – мобільне математичне середовище Sage. Визначено доцільність об'єднання систем комп'ютерної математики, систем графічного аналізу та динамічної геометрії у єдиному середовищі для підтримки навчання математичної інформатики у технічних ЗВО.

У [226] показано, що найвища ефективність навчання основ математичної інформатики досягається за умови інтеграції як засобів, так й форм організації її навчання, зокрема, через застосування технологій мобільного та електронного навчання, засобів Web 2.0 тощо, а також через перенесення засобів навчання математичної інформатики у Web-середовище. Наведено складові методики використання, окреслено зміст та задачі педагогічного експерименту [240].

У статті [242] спроектовано зміст навчання основ математичної інформатики студентів технічних університетів.

Починаючи з 2009 року на кафедрі комп'ютерних систем та мереж ДВНЗ «Криворізький національний університет», яка була базовою кафедрою для даного дослідження, було започатковано ІТ-конференцію молодих науковців, на якій під керівництвом автора було представлено ряд доповідей студентів спеціальності «Комп'ютерна інженерія», присвячених розробці та використанню систем управління веб-сайтом, сучасних веб-технологій (зокрема, ASP.NET), засобів хмарних обчислень у комп'ютерній інженерії, навчальних



мереж на основі Wi-Fi. Окремий блок студентських робіт було присвячено використанню SageMath як засобу навчання основ математичної інформатики.

Основні результати, отримані в процесі експериментальної роботи, відображені у статтях [92; 94; 152; 153; 168; 226; 239; 240; 241; 242; 243; 244; 245; 248] та доповідалися на Міжнародній науково-практичній конференції «Науково-методичні засади управління якістю освіти в педагогічних вищих навчальних закладах» (м. Київ, Національний педагогічний університет імені М. П. Драгоманова, 28 травня 2009 р.), VII Міжнародній науково-технічній конференції «Новітні комп'ютерні технології» (м. Севастополь, Державний науково-дослідний інститут автоматизованих систем у будівництві, 15-18 вересня 2009 р.), VII Всеукраїнській науково-практичній конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2010) (м. Черкаси, Черкаський державний технологічний університет, 4-6 травня 2010 р.), VIII Міжнародній науково-технічній конференції «Новітні комп'ютерні технології НОКОТЕ'2010» (м. Севастополь, Державний науково-дослідний інститут автоматизованих систем у будівництві, 14-17 вересня 2010 р.), IV Всеукраїнській конференції аспірантів, студентів та молодих вчених «Комп'ютерні інтелектуальні системи та мережі» (м. Кривий Ріг, Криворізький технічний університет, 23-25 березня 2011 р.), IX Міжнародній науково-технічній конференції «Новітні комп'ютерні технології НОКОТЕ'2011» (м. Севастополь, Державний науково-дослідний інститут автоматизованих систем у будівництві, 13-16 вересня 2011 р.), Міжнародній науково-практичній конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ–2012) (м. Черкаси, Черкаський державний технологічний університет, 25-27 квітня 2012 р.), X Міжнародній науково-технічній конференції «Новітні комп'ютерні технології НОКОТЕ'2012» (м. Севастополь, Державний науково-дослідний інститут автоматизованих систем у будівництві, 11-14 вересня 2012 р.), Міжнародній науково-методичній конференції «Розвиток інтелектуальних умінь і творчих здібностей учнів та студентів у процесі навчання дисциплін природничо-математичного циклу – ІТМ\*плюс-2012» (м. Суми, Сумський

державний педагогічний університет ім. А. С. Макаренка, 6-7 грудня 2012 р.), Всеукраїнському науково-методичному Інтернет-семінарі «Хмарні технології в освіті» (м. Київ – Черкаси – Кривий Ріг, Криворізький національний університет, 21 грудня 2012 р.), VII Всеукраїнській WEB-конференції аспірантів, студентів та молодих вчених «Комп'ютерні інтелектуальні системи та мережі» (м. Кривий Ріг, ДВНЗ «Криворізький національний університет», 25-27 березня 2014 р.), Міжнародному семінарі «Хмарні технології в освіті» (СТЕ'2014) (м. Київ – Кривий Ріг – Черкаси – Харків – Старобільськ – Херсон – Чейні, ДВНЗ «Криворізький національний університет», 26 грудня 2014 р.).

На формуально-узагальнювальному етапі дослідження (2015-2018 рр.) було проведено формульний етап педагогічного експерименту; проаналізовано, опрацьовано та узагальнено одержані результати експериментальної роботи; сформульовані загальні висновки та визначено перспективи подальшого вивчення проблеми; виконано оформлення рукопису дисертації.

У статтях, що відносяться до даного етапу, було:

1) показано, що виокремлення усталених та перспективних хмарних технологій в освіті вимагає наукового прогнозування розвитку хмарних технологій навчання на основі урахування історичних тенденцій розвитку ІКТ [163; 167];

2) завершено процес становлення методики використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів [162; 165; 166];

3) визначені перспективні моделі використання [164] та засоби хмарних технологій [49] у підготовці ІТ-фахівців;

4) подано методику використання CoCalc як засобу навчання у спецкурсі «Основи математичної інформатики» [48; 264];

5) підведено підсумки дослідно-експериментальної роботи [249].

Основні результати, отримані в процесі експериментальної роботи, відображені у статтях [48; 49; 162; 163; 164; 165; 166; 167; 249; 264] та

доповідалися на науково-практичному семінарі «Хмарні технології в сучасному університеті» (ХТСУ-2015) (м. Черкаси, Черкаський державний технологічний університет, 24 березня 2015 р.), ІХ Всеукраїнській науково-практичній WEB конференції аспірантів, студентів та молодих вчених «Комп'ютерні інтелектуальні системи та мережі» (м. Кривий Ріг, ДВНЗ «Криворізький національний університет», 22-24 березня 2016 р.), VII Всеукраїнській науково-методичній конференції «Комп'ютерне моделювання та інформаційні технології в освіті» (КМІТО'2016) (м. Кривий Ріг, ДВНЗ «Криворізький національний університет», 19-20 травня 2016 р.), 5th Workshop on Cloud Technologies in Education (СТЕ 2017) (м. Кривий Ріг, ДВНЗ «Криворізький національний університет», 28 квітня 2017 р.), X Всеукраїнській науково-методичній конференції «Комп'ютерне моделювання та інформаційні технології в освіті» (КМІТО'2017) (м. Кривий Ріг, Криворізький державний педагогічний університет, 27-28 квітня 2017 р.), 14-тій міжнародній конференції «ICT in Education, Research, and Industrial Applications. Integration, Harmonization and Knowledge Transfer» (ICTERI 2018) (м. Київ, Інститут післядипломної освіти Київського національного університету імені Тараса Шевченка, 14-17 травня 2018 р.).

Відповідно до гіпотези дослідження, методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів сприятиме підвищенню рівня їхніх навчальних досягнень за умови:

- теоретичного обґрунтування методичних засад використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів;

- здійснення добору та класифікації засобів хмарних технологій навчання основ математичної інформатики студентів технічних університетів;

- упровадження технологій комбінованого та мультистратегійного навчання;

- розробки хмаро зорієнтованих навчально-методичних матеріалів.

До складу експериментальної групи були включені студенти, які опанували спецкурс «Основи математичної інформатики» з використанням хмарних технологій за розробленою методикою (121 студент), до складу контрольних – студенти, які опанували спецкурс «Основи математичної інформатики» за традиційною методикою (106 студентів), розробленою Т. П. Кобильником та Н. О. Бугаєць.

Рівень навчальних досягнень студентів відповідно до робочої програми спецкурсу визначатимемо за наступною шкалою: дослідницький рівень – 90-100 балів, поглиблений рівень – 80-89 балів, підвищений рівень – 71-79 балів, базовий рівень – 61-70 балів, мінімально-базовий рівень – 50-60 балів, початковий рівень – 30-49 балів, рівень несформованості – 0-29 балів.

У табл. 2.2 подано розподіл студентів контрольної (КГ) та експериментальної груп (ЕГ) за рівнями навчальних досягнень до початку та після завершення формувального етапу педагогічного експерименту.

Таблиця 2.2

| Рівні              | До початку формувального етапу педагогічного експерименту |       |           |       | Після завершення формувального етапу педагогічного експерименту |       |           |       |
|--------------------|-----------------------------------------------------------|-------|-----------|-------|-----------------------------------------------------------------|-------|-----------|-------|
|                    | КГ                                                        |       | ЕГ        |       | КГ                                                              |       | ЕГ        |       |
|                    | кількість                                                 | %     | кількість | %     | кількість                                                       | %     | кількість | %     |
| несформованості    | 5                                                         | 4,72  | 6         | 4,96  | 4                                                               | 3,77  | 2         | 1,65  |
| початковий         | 11                                                        | 10,38 | 14        | 11,57 | 11                                                              | 10,38 | 5         | 4,13  |
| мінімально-базовий | 28                                                        | 26,42 | 26        | 21,49 | 32                                                              | 30,19 | 15        | 12,40 |
| базовий            | 19                                                        | 17,92 | 27        | 22,31 | 21                                                              | 19,81 | 28        | 23,14 |
| підвищений         | 25                                                        | 23,58 | 33        | 27,27 | 20                                                              | 18,87 | 38        | 31,40 |
| поглиблений        | 12                                                        | 11,32 | 8         | 6,61  | 11                                                              | 10,38 | 20        | 16,53 |
| дослідницький      | 6                                                         | 5,66  | 7         | 5,79  | 7                                                               | 6,60  | 13        | 10,74 |

На рис. 2.44, 2.45, 2.46 та 2.47 подано порівняльні розподіли студентів КГ та ЕГ за рівнями навчальних досягнень до початку та після завершення формувального етапу педагогічного експерименту.

З метою з'ясування, чи існують статистично значущі відмінності між отриманими розподілами навчальних досягнень, було використано критерій  $\chi^2$  Пірсона. У відповідності до обмежень застосування даного критерію [218, с. 117], кількість спостережень для кожного із рівнів повинна бути не менше 5,

у той час як у табл. 2.2 наявні й менші значення. Задля дотримання умов застосування критерію виконаємо об'єднання перших двох рівнів (несформованості та початкового) і розрахуємо відповідні емпіричні значення критерію  $\chi^2$ .

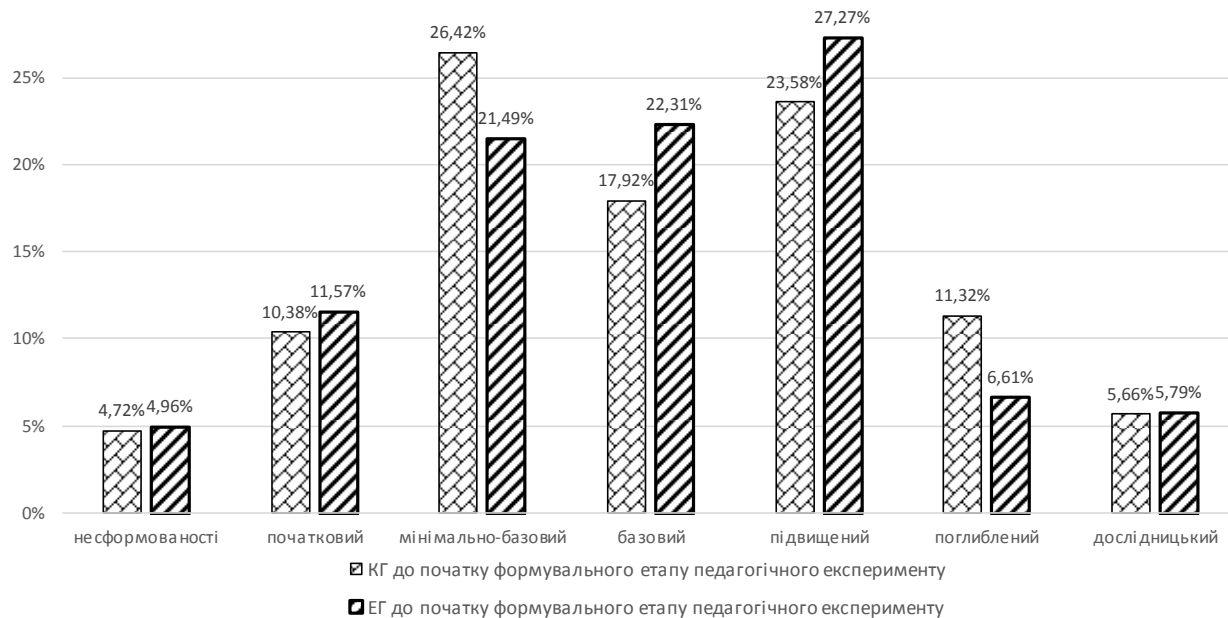


Рис. 2.44. Розподіл студентів КГ та ЕГ за рівнями навчальних досягнень до початку формувального етапу педагогічного експерименту

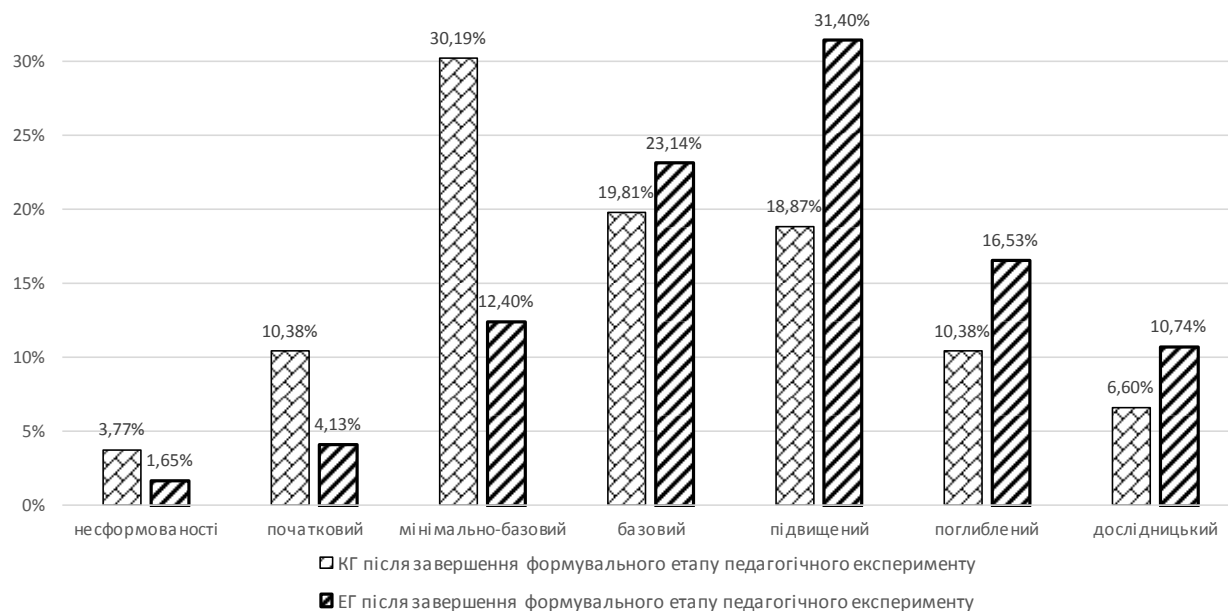


Рис. 2.45. Розподіл студентів КГ та ЕГ груп за рівнями навчальних досягнень після завершення формувального етапу педагогічного експерименту

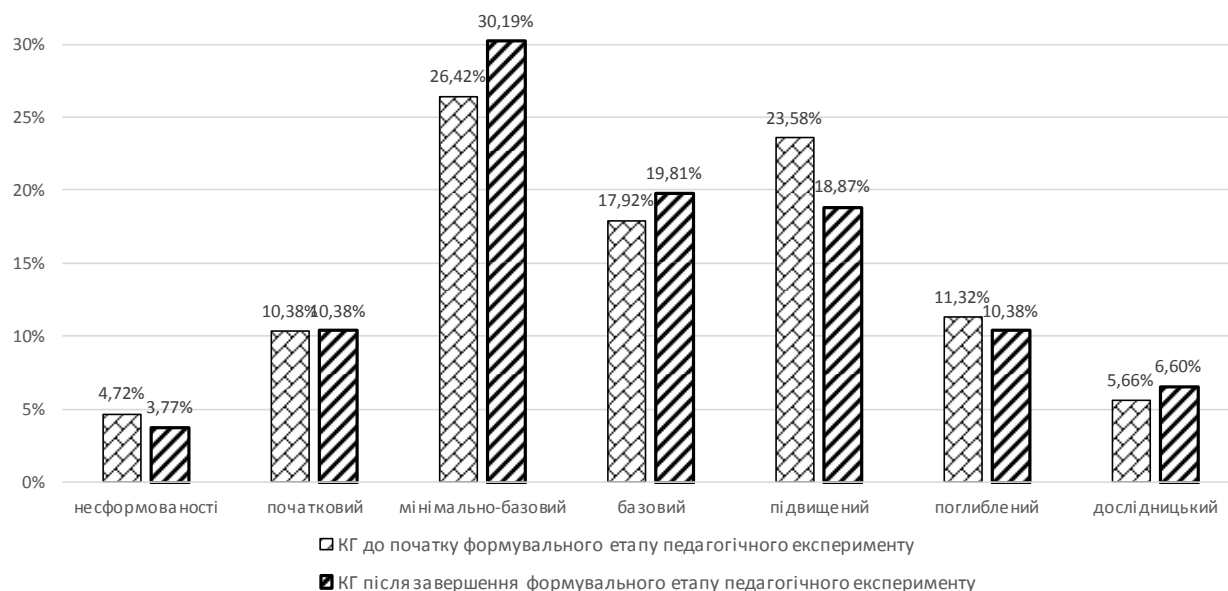


Рис. 2.46. Розподіл студентів КГ за рівнями навчальних досягнень до початку та після завершення формувального етапу педагогічного експерименту

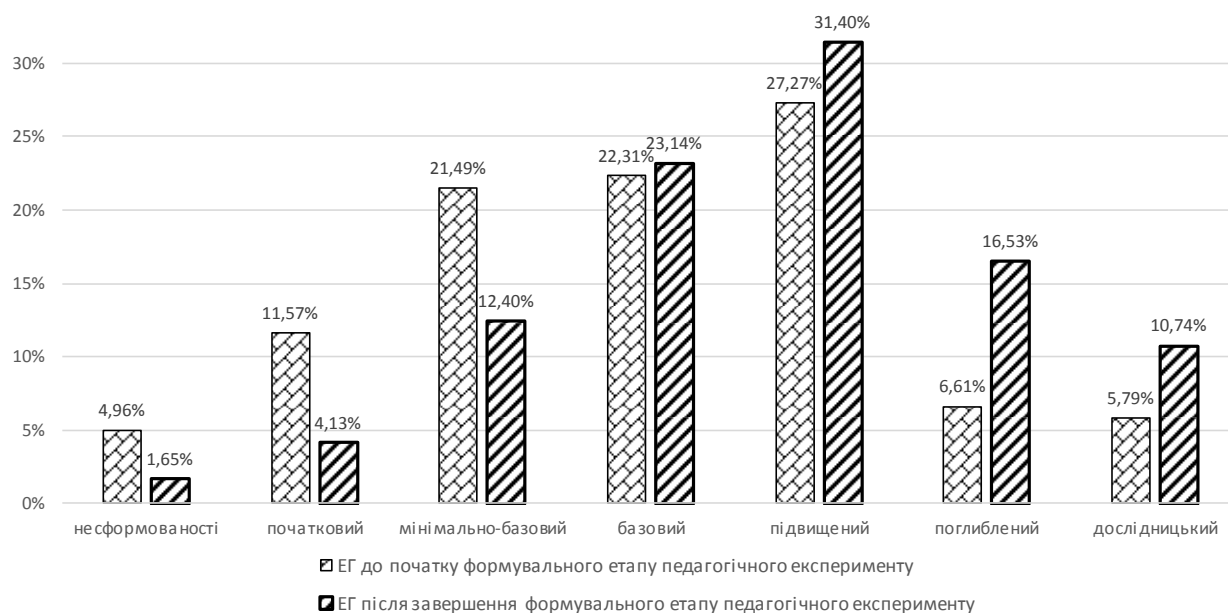


Рис. 2.47. Розподіл студентів ЕГ за рівнями навчальних досягнень до початку та після завершення формувального етапу педагогічного експерименту

Шкалою вимірювань є шкала з  $C=6$  категоріями. Отже, кількість степенів свободи  $\nu = C - 1 = 5$ .

Нульова гіпотеза  $H_0$ : ймовірність попадання студентів контрольної та експериментальної вибірки ( $n_1 = 106$ ,  $n_2 = 121$ ) в кожну з  $i$  ( $i = 1, 2, \dots, 6$ )

категорій однакова, тобто  $H_0: p_{1i} = p_{2i}$  ( $i = 1, 2, \dots, 6$ ).

Альтернативна гіпотеза  $H_1: p_{1i} \neq p_{2i}$  хоча б для однієї із  $k$  категорій.

Значення  $\chi^2$  обчислюється за формулою:

$$T_{\text{емп}} = \frac{1}{n_1 n_2} \sum_{i=1}^c \frac{(n_1 Q_{2i} - n_2 Q_{1i})^2}{Q_{1i} + Q_{2i}}.$$

$p_{1i}$  – ймовірність оцінювання рівня навчальних досягнень учасників контрольних груп за  $i$  рівнем ( $i = 1, 2, \dots, 6$ );

$p_{2i}$  – ймовірність оцінювання рівня навчальних досягнень учасників експериментальних груп за  $i$  рівнем ( $i = 1, 2, \dots, 6$ );

$Q_{1i}$  – кількість учасників контрольних груп, навчальні досягнення яких знаходяться на  $i$  рівні;

$Q_{2i}$  – кількість учасників експериментальних груп, навчальні досягнення яких знаходяться на  $i$  рівні.

Критичні значення критерію Пірсона для кількості степенів свободи  $\nu = 5$  –  $T_{\text{крит}(0,05)} = 11,07$  на рівні статистичної значущості 0,05 та  $T_{\text{крит}(0,01)} = 15,086$  на рівні статистичної значущості 0,01.

Емпіричні значення критерію  $\chi^2$  Пірсона були обчислені для трьох співставлень вибірок табл. 2.2:

1) для розподілів студентів контрольної та експериментальної груп за рівнями навчальних досягнень до початку формувального етапу педагогічного експерименту  $T_{\text{емп}} = 3,89 < T_{\text{крит}(0,05)}$ , що дає підстави для прийняття нульової гіпотези та висновку про те, що до початку формувального етапу педагогічного експерименту статистично значущі відмінності між рівнями навчальних досягнень студентів контрольних та експериментальних груп відсутні;

2) для розподілів студентів контрольної та експериментальної груп за рівнями навчальних досягнень після завершення формувального етапу педагогічного експерименту  $T_{\text{емп}} = 20,06 > T_{\text{крит}(0,01)}$ , що дає підстави для прийняття альтернативної гіпотези та висновку про те, що після завершення формувального етапу педагогічного експерименту між рівнями навчальних

досягнень студентів контрольних та експериментальних груп наявні відмінності на рівні статистичної значущості 0,01;

3) для розподілів за рівнями навчальних досягнень студентів експериментальної групи до початку формувального етапу педагогічного експерименту та після його завершення  $T_{\text{емп}} = 16,52 > T_{\text{крит}(0,01)}$ , що дає підстави для прийняття альтернативної гіпотези та висновку про те, що між рівнями навчальних досягнень студентів експериментальної групи до початку та після завершення формувального етапу педагогічного експерименту наявні відмінності на рівні статистичної значущості 0,01.

Аналіз табл. 2.2 показує, що в експериментальній групі після завершення формувального етапу педагогічного експерименту відсоток студентів, рівень навчальних досягнень яких знаходився на рівні несформованості, початковому, мінімально-базовому рівнях, зменшився (відповідно на 3,31 %, 7,44 % та 9,09 %), а на базовому, підвищеному, поглибленому та дослідницькому рівнях – збільшився (відповідно на 0,83 %, 4,13 %, 9,92 % та 4,95 %). Це надає можливість зробити висновок про те, що збільшення кількості студентів з високими рівнями навчальних досягнень відбулось за рахунок їх переходу із груп з низькими рівнями, тобто наявний ефект підвищення рівня навчальних досягнень студентів експериментальної групи.

Виходячи з того, що в експериментальній групі була застосована розроблена методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів, можна зробити висновок, що саме це було фактором підвищення рівня сформованості рівня їх навчальних досягнень, а, отже, гіпотеза дослідження є доведеною.

## **Висновки до розділу 2**

У процесі розробки методичних засад використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів були отримані такі основні результати та висновки:



1. Теоретичне обґрунтування методичних засад процесу використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів потребувало визначення: змісту навчання спецкурсу «Основи математичної інформатики» (змістові модулі «Теорія алгоритмів», «Методи обчислень», «Теорія кодування», «Основи криптографії») для майбутніх фахівців з інформаційних технологій; цілей навчання (формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій); технології навчання, зокрема форм організації освітнього процесу, методів (лекції, лабораторні заняття, самостійна робота студентів, навчальні конференції, консультації, індивідуальні заняття, навчально-дослідницькі проекти, контрольні заходи) та засобів навчання основ математичної інформатики, провідними з яких було визначено засоби хмарних технологій. Показано, що перспективним засобом розширення можливостей студентів для ініціації, планування, організації, моніторингу та регулювання власної навчальної діяльності та грамотного розв'язання складних навчальних задач є проектування когнітивних навчальних стратегій та їхня подальша інтеграція в мультистратегійне навчання.

2. Проведений аналіз теорії та практики використання засобів хмарних технологій в освітньому процесі надав можливість визначити засоби хмарних технологій навчання основ математичної інформатики як сукупність хмаро зорієнтованих електронних освітніх ресурсів, що застосовуються для інформаційно-процесуального забезпечення виконання дидактичних завдань або їх фрагментів та спрямовані на реалізацію цілей навчання основ математичної інформатики. До основних засобів хмарних технологій навчання основ математичної інформатики відносяться хмаро зорієнтовані: навчально-методичні комплекси (програмно-методичні матеріали, засоби оцінювання навчальних досягнень – тестові системи та тренажери, практикуми, навчально-методичні матеріали – дидактичні демонстраційні матеріали, навчальні посібники та підручники, електронні довідники), системи підтримки навчання, навчальні лабораторії (зокрема, середовища моделювання) та предметні

середовища (системи комп'ютерної математики та середовища програмування). Допоміжними засобами хмарних технологій навчання основ математичної інформатики є хмаро зорієнтовані: додаткові науково-навчальні матеріали, комунікаційні засоби (електронна пошта, засоби аудіо та відеозв'язку), операційні системи, засоби зберігання даних та офісні пакети (текстові та табличні процесори, засоби підготовки презентацій, системи управління базами даних та додаткові хмаро зорієнтовані компоненти). Для підвищення дидактичної ефективності застосовані засоби хмарних технологій навчання основ математичної інформатики використовуються в навчально-виховному процесі спільно з іншими навчально-методичними матеріалами (наприклад, із традиційними підручниками та навчальними посібниками, методичними рекомендаціями для викладачів та студентів тощо), формуючи хмаро зорієнтовані програмно-методичні комплекси.

3. Розроблена методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів складається із трьох основних блоків: цільового (формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій), змістового (навчання основ математичної інформатики) та технологічного (хмаро зорієнтовані засоби ІКТ, методи та форми їх використання в навчанні математичної інформатики). Технологічний блок методики визначає провідний зміст діяльності (індивідуальні та групові навчальні дослідження), форму організації навчання (спецкурс), види діяльності з формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій та відповідні засоби хмарних ІКТ.

4. Експериментальну перевірку ефективності розробленої методики було виконано із залученням 121 студента до експериментальної групи та 106 – до контрольної. З метою з'ясування, чи існують статистично значущі відмінності між отриманими розподілами рівнів навчальних досягнень студентів контрольних та експериментальних груп, використано критерій  $\chi^2$  Пірсона: а) для розподілів студентів контрольної та експериментальної груп за рівнями

навчальних досягнень до початку формувального етапу педагогічного експерименту значення критерію  $\chi^2$  Пірсона  $T_{\text{емп}} = 3,89 < T_{\text{крит}(0,05)}$ , що дає підстави для висновку про відсутність статистично значущих відмінностей між ними; б) для розподілів студентів контрольної та експериментальної груп за рівнями навчальних досягнень після завершення формувального етапу педагогічного експерименту  $T_{\text{емп}} = 20,06 > T_{\text{крит}(0,01)}$ , що дає підстави для висновку про наявність статистично значущих відмінностей між ними на рівні 0,01; в) для розподілів за рівнями навчальних досягнень студентів експериментальної групи до початку формувального етапу педагогічного експерименту та після його завершення  $T_{\text{емп}} = 16,52 > T_{\text{крит}(0,01)}$ , що дає підстави для висновку про наявність статистично значущих відмінностей між ними на рівні 0,01.

Аналіз результатів експериментальної роботи показав, що в експериментальній групі після завершення формувального етапу педагогічного експерименту відсоток студентів, навчальні досягнення яких було діагностовано на рівнях несформованості, початковому та мінімально-базовому, зменшився (відповідно на 3,31%, 7,44% та 9,09%), а на базовому, підвищеному, поглибленому та дослідницькому рівнях – збільшився (відповідно на 0,83%, 4,13%, 9,92% та 4,95%). Це надає можливість зробити висновок про те, що збільшення кількості студентів із високими рівнями навчальних досягнень відбулось за рахунок їх переходу із груп із низькими рівнями, тобто наявний ефект підвищення рівня навчальних досягнень студентів експериментальної групи. Виходячи з того, що в експериментальній групі було застосовано розроблену методику використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів, зроблено висновок про те, що саме це стало чинником підвищення рівня сформованості рівня їхніх навчальних досягнень, а отже, гіпотеза дослідження є доведеною.

Основні результати другого розділу опубліковано в роботах [49; 92; 162; 226; 239; 240; 241; 242; 248; 264].

## ВИСНОВКИ

Отримані результати дослідження дають підстави зробити такі **висновки**:

У дисертації наведено теоретичне обґрунтування й нове вирішення проблеми розробки та впровадження методики використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів.

1. За результатами аналізу наукової літератури, нормативно-законодавчих документів з'ясовано, що забезпечення інноваційного розвитку України можливе на основі комплексного використання в професійній діяльності та професійній підготовці фахівців з інформаційних технологій хмарних технологій і технологій математичної інформатики. Доведено ключову роль математичної інформатики в зменшенні розриву між швидкоплинними змінами в галузі ІКТ та професійною підготовкою фахівців з інформаційних технологій за рахунок застосування технології комп'ютерного моделювання та відповідних програмних засобів навчання, насамперед – систем комп'ютерної математики.

Визначено компетентність з математичної інформатики майбутніх фахівців з інформаційних технологій як особистісно-професійне утворення, яке включає систему знань, умінь, навичок, досвід навчально-дослідницької діяльності з математичної інформатики та позитивне ціннісне ставлення до неї й виявляється в готовності та здатності до модифікації наявних і розробки нових інформаційних технологій на основі моделей і методів математичної інформатики.

2. Ключове поняття дослідження – хмарні технології (хмарні ІКТ) – визначено як сукупність методів, засобів і прийомів, використовуваних для збирання, систематизації, зберігання та опрацювання на віддалених серверах, передавання через мережу та подання через клієнтську програму всеможливих повідомлень і даних. Відповідно під хмарними технологіями навчання в дослідженні розуміються такі ІКТ навчання, що передбачають використання мережних ІКТ із централізованим мережним зберіганням та опрацюванням даних (виконання програм), за якого користувач виступає клієнтом

(користувачем послуг), а «хмара» – сервером (постачальником послуг). Витоки хмарних технологій навчання містяться в застосуванні концепції «комп'ютерної послуги» до освітнього процесу, зокрема наданні місця для зберігання електронних освітніх ресурсів та мобільного доступу до них.

Аналіз вітчизняного та зарубіжного досвіду використання хмарних технологій у підготовці ІТ-фахівців надав можливість обґрунтувати доцільність застосування в навчанні інформатичних дисциплін таких моделей надання хмарних послуг: «програмне забезпечення як послуга», «платформа як послуга» та «інфраструктура як послуга» на основі інформатичної технології віртуальних машин та педагогічної технології дистанційного навчання.

З'ясовано, що розвиток хмарних технологій навчання відзначався неперервністю, взаємозумовленістю та інноваційністю й зумовлений імплементацією положень концепції «комп'ютерної послуги» в організацію освітнього процесу через надання місця для зберігання електронних освітніх ресурсів та мобільного доступу до них.

Доведено, що однією з явних переваг використання хмарних технологій у підготовці майбутніх ІТ-фахівців у технічних університетах є можливість застосування сучасних засобів паралельного програмування як основи хмарних технологій.

3.3 урахуванням теоретично обґрунтованих методичних засад використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів визначено: зміст навчання спецкурсу «Основи математичної інформатики» (змістові модулі «Теорія алгоритмів», «Методи обчислень», «Теорія кодування», «Основи криптографії») для майбутніх фахівців з інформаційних технологій, цілі та технологію навчання, що об'єднує форми організації освітнього процесу й методи навчання (лекції, лабораторні заняття, самостійна робота студентів, навчальні конференції, консультації, індивідуальні заняття, навчально-дослідницькі проекти, контрольні заходи), а також засоби навчання основ математичної інформатики, провідними з яких є засоби хмарних технологій.

4. У дисертації виокремлено засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів як сукупність хмаро зорієнтованих електронних освітніх ресурсів, що застосовуються для інформаційно-процесуального забезпечення виконання дидактичних завдань або їхніх фрагментів та спрямовані на реалізацію цілей навчання основ математичної інформатики і сприяють формуванню компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій.

Класифіковано засоби хмарних технологій навчання основ математичної інформатики студентів технічних університетів на основних (хмаро зорієнтовані програмно-методичні матеріали, тестові системи, тренажери, практикуми, дидактичні демонстраційні матеріали, навчальні посібники та підручники, електронні довідники, системи підтримки навчання, навчальні лабораторії, системи комп'ютерної математики, середовища програмування) та допоміжних (хмаро зорієнтовані додаткові науково-навчальні матеріали, електронна пошта, засоби аудіо- та відеозв'язку, операційні системи, засоби зберігання даних, текстові процесори, табличні процесори, засоби підготовки презентацій, системи управління базами даних, додаткові хмаро зорієнтовані компоненти).

Доведено, що найбільш значущим засобом хмарних технологій навчання основ математичної інформатики є універсальні хмаро зорієнтовані системи комп'ютерної математики, такі як CoCalc, що інтегрують більшість виокремлених засобів.

5. Методика використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів складається із цільового (формування компетентності з математичної інформатики майбутніх фахівців з інформаційних технологій), змістового (навчання основ математичної інформатики) і технологічного (хмаро зорієнтовані засоби ІКТ, методи та форми їх використання в навчанні математичної інформатики) блоків.

Експериментальна перевірка розробленої методики у формі паралельного педагогічного експерименту та результати статистичного опрацювання

отриманих даних підтвердили припущення щодо відсутності в розподілі студентів контрольної та експериментальної груп суттєвих розбіжностей на початку експерименту за рівнями навчальних досягнень ( $\chi^2_{\text{емп}} = 3,89$ ) та засвідчили наявність значущих на рівні 0,01 розбіжностей за рівнями навчальних досягнень ( $\chi^2_{\text{емп}} = 20,06$ ) після завершення педагогічного експерименту, що підтвердило гіпотезу дослідження.

Виконане дослідження не вичерпує всіх аспектів аналізованої проблеми. Подальші наукові пошуки її розв'язання доцільні за такими напрямками: проектування хмаро зорієнтованого середовища навчання майбутніх фахівців з комп'ютерної інженерії; тенденції розвитку хмарних технологій у професійній підготовці та перепідготовці ІТ-фахівців; методика формування дослідницьких компетентностей майбутніх фахівців з інженерії програмного забезпечення засобами хмарних технологій.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. About Perl – [www.perl.org](http://www.perl.org) [Electronic resource] / Perl.org. – 2018. – Access mode : <https://www.perl.org/about.html>.
2. Adams J. E-learning offers myriad opportunities for rapid talent development [Electronic resource] / Jean Adams // T+D. – 2008. – March. – P. 69-73. – Access mode : <http://yellowedge.files.wordpress.com/2008/03/adams.pdf>.
3. Ainsworth S. DeFT: A conceptual framework for considering learning with multiple representations / Shaaron Ainsworth // Learning and Instruction. – 2006. – Vol. 16. – Iss. 3. – P. 183-198. – DOI : 10.1016/j.learninstruc.2006.03.001.
4. Aldakheel E. A. A Cloud Computing framework for computer science education : A Thesis Submitted to the Graduate College of Bowling Green State University in partial fulfillment of the requirements for the degree of Master of Science / Eman A. Aldakheel ; the Graduate College of Bowling Green State University. – [Bowling Green] : December, 2011. – XI, 130 p.
5. Anderson E. The Species Problem in Iris / Edgar Anderson // Annals of the Missouri Botanical Garden. – 1936. – Vol. 23. – No. 3. – P. 457-469+471-483+485-501+503-509. – DOI : 10.2307/2394164.
6. Architecting Low Latency Cloud Networks : Arista Whitepaper [Electronic resource] / Arista. – [2009-05-08]. – 5 p. – Access mode : <https://www.arista.com/assets/data/pdf/CloudNetworkLatency.pdf>.
7. Axiom: The 30 Year Horizon. – Volume 13: Providing Axiom Correct [Electronic resource] / Manuel Bronstein, William Burge, Timothy Daly, James Davenport, Michael Dewar, Martin Dunstan, Albrecht Fortenbacher, Patrizia Gianni, Johannes Grabmeier, Jocelyn Guidry, Richard Jenks, Larry Lambe, Michael Monagan, Scott Morrison, William Sit, Jonathan Steinbach, Robert Sutor, Barry Trager, Stephen Watt, Jim Wen, Clifton Williamson. – [Axiom Project], [2017]. – VII, 73 p. – Access mode : <http://www.axiom-developer.org/axiom-website/bookvol13.pdf>.
8. Bauer F. L. Software Engineering: An Advanced Course / F. L. Bauer, J. B. Dennis, G. Goos, C. C. Gotlieb, R. M. Graham, M. Griffiths, H. J. Helms,



- B. Morton, P. C. Poole, D. Tsihritzis, W. M. Waite ; edited by F. L. Bauer. – Berlin, Heidelberg, New York : Springer-Verlag, 1975. – XII, 548. – (Lecture Notes in Computer Science, Vol. 30) (Formerly published 1973 as Lecture Notes in Economics and Mathematical Systems, Vol. 81). – DOI: 10.1007/3-540-07168-7.
9. Bloomfield V. A. Using R for Numerical Analysis in Science and Engineering / Victor A. Bloomfield. – London : Chapman and Hall/CRC, 2014. – XXIV, 330 p. – (The R Series)
  10. Bonk C. J. The handbook of blended learning: global perspectives, local designs / Curtis J. Bonk, Charles R. Graham. – San Francisco : Pfeiffer, 2006. – 585 p.
  11. Börger E. Computability, Complexity, Logic / Egon Börger. – Amsterdam : , North-Holland, 1989. – XX+592 p. – (Studies in Logic and the Foundations of Mathematics, vol. 128).
  12. Bosma W. Solving Problems with Magma [Electronic resource] / Wieb Bosma, John Cannon, Catherine Playoust, Allan Steel ; School of Mathematics and Statistics, University of Sydney. – 1999. – XII, 233 p. – Access mode : <https://magma.maths.usyd.edu.au/magma/pdf/examples.pdf>.
  13. Brazdil P. B. Multistrategy Learning / Pavel B. Brazdil // Encyclopedia of the Sciences of Learning : With 312 Figures and 68 Tables / Editor : Norbert M. Seel. – New York : Springer, 2012. – P. 2396-2399.
  14. Bull C. N. Studios in software engineering education : A thesis submitted for the degree of Doctor of Philosophy / Christopher Neil Bull ; HighWire CDT, School of Computing and Communications, Lancaster University. – [Bailrigg], 2015. – X, 227 p.
  15. Cloud Service Level Agreement Standardisation Guidelines [Electronic resource]. – Brussels. – 24/06/2014. – 41 p. – Access mode : [http://ec.europa.eu/information\\_society/newsroom/cf/dae/document.cfm?action=display&doc\\_id=6138](http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?action=display&doc_id=6138).
  16. CoCalc - Collaborative Calculation in the Cloud [Electronic resource] / CoCalc by SageMath, Inc. – 2018. – Access mode : <https://cocalc.com/>.
  17. Computations in algebraic geometry with Macaulay 2 / Edited by David

- Eisenbud, Daniel R. Grayson, Michael E. Stillman, and Bernd Sturmfels. – Berlin : Springer-Verlag, 2001. – 345 p. – (Algorithms and Computations in Mathematics. Vol. 8).
18. Computers and the World of the Future / Edited by Martin Greenberger. – New York : M.I.T. Press and Wiley, 1962. – 340 p.
  19. Cremona J. E. eclib (including mwrank and related programs for elliptic curves over Q) [Electronic resource] / J. E. Cremona. – 2017-10-05. – Access mode : <http://homepages.warwick.ac.uk/staff/J.E.Cremona/mwrank/index.html>.
  20. De Graeve, R. La tortue et la géométrie avec Xcas [Electronic resource] / Renée De Graeve. – Access mode : <https://www-fourier.ujf-grenoble.fr/~parisse/giac/doc/fr/castor/>.
  21. Delisle M. Mastering phpMyAdmin 3.4 for Effective MySQL Management : A complete guide to getting started with phpMyAdmin 3.4 and mastering its features / Marc Delisle. – Birmingham – Mumbai : Packt Publishing, 2012. – XIV, 365 p.
  22. Deryhlazov L. V. The Models of Distance Forms of Learning in National Academy of Statistics, Accounting and Audit / L. V. Deryhlazov, V. M. Kukharenko, L. P. Perkhun, N. M. Tovmachenko // *Наук. вісн. Нац. акад. статистики, обліку та аудиту*. – 2017. – № 3. – С. 79-90.
  23. Diaz V. Cloud-based technologies: faculty development, support, and implementation / Veronica Diaz // *Journal of Asynchronous Learning Networks*. – 2011. – Feb. – Volume 15, Issue 1. – P. 95-102.
  24. Dijkstra E. W. On a Methodology of Design / Dijkstra E. W. // *MC-25 Informatica Symposium* / Ed. Jacobus Willem Bakker. – Amsterdam : Mathematisch Centrum, 1971. – P. 4.1-4.10. – (Mathematical Centre tracts, 37)
  25. *Encyclopedia of the Sciences of Learning : With 312 Figures and 68 Tables* / Editor : Norbert M. Seel. – New York : Springer, 2012. – CVI+3536 p.
  26. e-Skills for Cloud Computing, Cyber-security and Green IT [Electronic resource]. – European Union, 2012. – 15 p. – Ref. Ares(2015)2203707 - 27/05/2015. – Access mode : <https://ec.europa.eu/docsroom/documents/10474/attachments/1/translations/en/renditions/pdf>.
  27. Flood M. M. DOUGLAS F. PARKHILL, *The Challenge of the Computer*

- Utility*, Addison-Wesley Publishing Co., Reading, Mass., 1966, 219 pages, \$7.95 / Merrill M. Flood // *Operations Research*. – 1967. – Vol. 15. – Issue 1. – P. 177-178.
28. Garfinkel S. L. *Architects of the Information Society: 35 Years of the Laboratory for Computer Science at MIT* / Simson L. Garfinkel ; edited by Hal Abelson. – Cambridge : The MIT Press, 1999. – 72 p.
  29. Giac/Xcas, a free computer algebra system [Electronic resource] / [Bernard Parisse]. – 2018. – Access mode : <http://www-fourier.ujf-grenoble.fr/~parisse/giac.html>.
  30. GitHub Help [Electronic resource] / GitHub Inc. – 2018. – Access mode : <https://help.github.com/categories/writing-on-github/>.
  31. GNU Octave [Electronic resource] / John W. Eaton. – 2018. – Access mode : <https://www.gnu.org/software/octave/>.
  32. Google and IBM Donate 1,600 Computers to ‘Cloud’ Project [Electronic resource] // Maximum PC. – 2007. – December. – P. 10. – Access mode : <http://dl.maximumpc.com/Archives/MPC1207-web.pdf>.
  33. Google Books Ngram Viewer [Electronic resource] / Google. – 2013. – Access mode : <https://books.google.com/ngrams>.
  34. Google Security Whitepaper [Electronic resource] // Google Cloud Platform. – May 26, 2015. – Access mode : <https://cloud.google.com/security/whitepaper>.
  35. Greenberger M. *The Computers of Tomorrow* [Electronic resource] / Martin Greenberger // *The Atlantic Monthly*. – 1964. – Vol. 213. – No 5, May. – P. 63-67. – Access mode : <http://www.theatlantic.com/past/docs/unbound/flashbks/computer/greenbf.htm>.
  36. Haigh T. *Software in the 1960s as Concept, Service, and Product* / Thomas Haigh // *IEEE Annals of the History of Computing*. – 2002. – Vol. 24. – Issue No. 1, January-March. – P. 5-13.
  37. Hoare C. A. R. *Unifying Theories of Programming* / C. A. R. Hoare, He Jifeng. – London ; New York : Prentice Hall, 1998. – xix, 298 p. : ill. – (International Series in Computer Science).
  38. Hopfield J. J. *Neural networks and physical systems with emergent collective computational abilities* / J. J. Hopfield // *Proceedings of the National Academy*

- of Sciences of the USA. – 1982. – Vol. 79, Iss. 8. – P. 2554-2558.
39. Hudgens J. Comprehensive Ruby Programming: Go from beginner to confident programmer / Jordan Hudgens. – Birmingham : Packt Publishing, 2017. – 330 p.
  40. Irwin D. E. An Operating System Architecture for Networked Server Infrastructure : Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science in the Graduate School of Duke University / David E. Irwin ; Department of Computer Science, Duke University. – [December], 2007. – XVII, 193 p.
  41. Jain A. Role of Cloud Computing in Higher Education [Electronic resource] / Anjali Jain, U.S Pandey // International Journal of Advanced Research in Computer Science and Software Engineering. – 2013. – Vol. 3, Iss. 7. – P. 966-972. – Access mode : [http://ijarcsse.com/Before\\_August\\_2017/docs/papers/Volume\\_3/7\\_July2013/V3I6-0242.pdf](http://ijarcsse.com/Before_August_2017/docs/papers/Volume_3/7_July2013/V3I6-0242.pdf)
  42. KASH: Introduction to KASH3 [Electronic resource] / KANT–Group. – January 31, 2006. – 82 p. – Access mode : <http://page.math.tu-berlin.de/~kant/KASH/pdf/kash3intro.pdf>.
  43. Knuth D. E. Literate Programming / Donald E. Knuth. – Stanford : Center for the Study of Language and Information, 1992. – XVI+368 p. – (CSLI Lecture Notes, no. 27.).
  44. Krill P. New Julia language seeks to be the C for scientists: Developers seek to provide one dynamic language for high productivity, performance [Electronic resource] / Paul Krill // InfoWorld. – Apr 18, 2012. – Access mode : <https://www.infoworld.com/article/2616709/application-development/new-julia-language-seeks-to-be-the-c-for-scientists.html>.
  45. Lardinois F. Google X Announces Project Loon: Balloon-Powered Internet For Rural, Remote And Underserved Areas [Electronic resource] / Frederic Lardinois // TechCrunch / AOL Inc. – Jun 14, 2013. – Access mode : <http://techcrunch.com/2013/06/14/google-x-announces-project-loon-balloon-powered-internet-for-rural-remote-and-underserved-areas/>.
  46. Le Livre Blanc du Cloud Computing: Tout ce que vous devez savoir sur l'informatique dans le nuage / Syntec informatique. – 2<sup>ème</sup> Trimestre 2010. –

- 19 s.
47. Mann A. O. A publicly regulated system of management control services / Alan O. Mann // Management control systems : the proceedings of a symposium held at System Development Corporation, Santa Monica, California, July 29-31, 1959 / Edited by : Donald G. Malcolm, and Alan J. Rowe ; general editor : Lorimer F. McConnell. – Third printing. – New York ; London : John Wiley & Sons, 1962. – P. 245-263.
  48. Markova O. M. CoCalc as a Learning Tool for Neural Network Simulation in the Special Course “Foundations of Mathematic Informatics” [Electronic resource] / Oksana Markova, Serhiy Semerikov, Maiia Popel // ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer 2018 : Proc. of the 13th Int. Conf.. Vol. II: Workshops (ICTERI, 2018). Kyiv, Ukraine, May 14-17, 2018 / Edited by : Vadim Ermolayev, Mari Carmen Suárez-Figueroa, Vitaliy Yakovyna, Vyacheslav Kharchenko, Vitaliy Kobets, Hennadiy Kravtsov, Vladimir Peschanenko, Yaroslav Prytula, Mykola Nikitchenko, Aleksander Spivakovsky. – P. 388-403. – (CEUR Workshop Proceedings, Vol. 2104). – Access mode : [http://ceur-ws.org/Vol-2104/paper\\_204.pdf](http://ceur-ws.org/Vol-2104/paper_204.pdf).
  49. Markova O. M. The tools of cloud technology for learning of fundamentals of mathematical informatics for students of technical universities [Electronic resource] / Oksana M. Markova // Cloud Technologies in Education : Proceedings of the 5th Workshop (CTE 2017). Kryvyi Rih, Ukraine, April 28, 2017 / Edited by : Serhiy O. Semerikov, Mariya P. Shyshkina. – P. 27-33. – (CEUR Workshop Proceedings, Vol. 2168). – Access mode : <http://ceur-ws.org/Vol-2168/paper5.pdf>.
  50. Mayer R. E. Multimedia learning : Second Edition / Richard E. Mayer. – New York : Cambridge University Press, 2009. – 320 p.
  51. McKendrick J. 10 steps to avoid cloud vendor lock-in [Electronic resource] / Joe McKendrick // ZDNet. – July 12, 2013. – Access mode : <http://www.zdnet.com/article/10-steps-to-avoid-cloud-vendor-lock-in/>.
  52. MediaWiki [Electronic resource]. – [2018]. – Access mode : <https://www.mediawiki.org/wiki/MediaWiki>.

53. Mell P. The NIST Definition of Cloud Computing : Recommendation of the National Institute of Standards and Technology [Electronic resource] / Peter Mell, Timothy Grance. – Gaithersburg : National Institute of Standards and Technology, September 2011. – III, 3 p. – (Special Publication 800-415). – Access mode : <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
54. MoodleCloud - Moodle hosting from the people that make Moodle [Electronic resource] / Moodle Pty Ltd. – 2018. – Access mode : <https://moodlecloud.com/>.
55. Oetiker T. Не надто короткий вступ до LATEX2ε або LATEX 2ε за 130 хвилин / Tobias Oetiker, Hubert Partl, Irene Hyna, Elisabeth Schlegl ; пер. Максим Поляков. – Version 4.12. – 5 травня 2003. – 144 с. – Режим доступу : <http://texdoc.net/texmf-dist/doc/latex/lshort-ukr/lshort-ukr.pdf>.
56. Office 365 для бізнесу: запитання й відповіді [Електронний ресурс] / Microsoft. – 2018. – Режим доступу : <https://products.office.com/uk-ua/business/microsoft-office-365-frequently-asked-questions>.
57. Office 365 для освіти [Електронний ресурс] / Microsoft. – 2018. – Режим доступу : <https://www.microsoft.com/uk-ua/education/products/office/default.aspx>.
58. Paul P. K. Cloud Based Educational Systems and Its Challenges and Opportunities and Issues / Prantosh Kr. Paul, Kiran Lata Dangwal // Turkish Online Journal of Distance Education – TOJDE. – 2014. – Vol. 15. – No 1, January. – P. 89-98.
59. Perez S. Facebook Looking Into Buying Drone Maker Titan Aerospace [Electronic resource] / Sarah Perez, Josh Constine // TechCrunch / AOL Inc. – Mar 3, 2014. – Access mode : <http://techcrunch.com/2014/03/03/facebook-in-talks-to-acquire-drone-maker-titan-aerospace/>.
60. Perlis A. J. Identifying and developing curricula in software engineering / Alan J. Perlis / Proceedings of the May 14-16, 1969, spring joint computer conference on XX - AFIPS '69 (Spring). Boston, Massachusetts. – New York : ACM, 1969. – P. 540-541. – DOI: 10.1145/1476793.1476877.
61. Ploetzner R. Cognitive Learning Strategies for Digital Media / Rolf Ploetzner // Encyclopedia of the Sciences of Learning : With 312 Figures and 68 Tables /

- Editor : Norbert M. Seel. – New York : Springer, 2012. – P. 596-599.
62. R: What is R? [Electronic resource] / The R Foundation. – [2018?]. – Access mode : <https://www.r-project.org/about.html>.
  63. Radenković B. IT Education as an Opportunity for Uprising of Serbian Economy [Electronic resource] / Božidar Radenković, Marijana Despotović-Zrakić, Zorica Bogdanović, Dušan Barać, Aleksandra Labus // Management: Journal Of Sustainable Business And Management Solutions In Emerging Economies. – 2017. – Vol. 19, No. 71. – P. 57-71. – DOI : 10.7595/management.fon.2014.0014. – Access mode : <http://management.fon.bg.ac.rs/index.php/mng/article/view/130>.
  64. Rajaei H. Cloud Computing in Computer Science and Engineering Education [Electronic resource] / Hassan Rajaei, Eman A. Aldakheel // Spurring Big Ideas in Education : 119th ASEE Annual Conference & Exposition. – San Antonio, June 10-13, 2012. – 15 p. – Access mode : <http://www.asee.org/public/conferences/8/papers/4956/download>.
  65. Shang J. The Latest Development of Education Informatization in North America and Its Implications / Junjie Shang, Peijie Cao, Huan Nie // Proc. of Int. Conf. of Educational Innovation through Technology (EITT 2014). 27-29 October 2014 – Brisbane, Australia / Edited by Xibin Han, Yuping Wang, Jiangang Cheng, Greer Cavallaro Johnson. – Los Alamitos : IEEE Computer Society, 2014. – P. 170-175. – DOI : 10.1109/EITT.2014.35.
  66. Shor R. M. Cloud Computing for Learning and Performance Professionals / R. M. Shor // INFOLINE. – 2011. – April. – Issue 1104. – 22 p.
  67. Siegle D. Cloud Computing: A Free Technology option to Promote Collaborative learning / Del Siegle // Gifted Child Today. – 2010. – Fall. – Vol. 33, No 4. – P. 41-45.
  68. Singular 4-1-1 – A computer algebra system for polynomial computations [Electronic resource] / Wolfram Decker, Gert-Martin Greuel, Gerhard Pfister, Hans Schönemann. – 2018. – Access mode : <http://www.singular.uni-kl.de/>.
  69. Sommerville I. Teaching cloud computing: A software engineering perspective / Ian Sommerville // Journal of Systems and Software. – 2013. – Vol. 86, Number 9, September. – P. 2330-2332.

70. Spivakovskiy O. V. ICT Advanced Training of University Teachers [Electronic resource] / Oleksandr Spivakovskiy, Nataliya Kushnir, Natalya Valko, Maksym Vinnyk // ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer 2017 : Proc. of the 13th Int. Conf. (ICTERI, 2017). Kyiv, Ukraine, May 15-18, 2017 / Edited by : Vadim Ermolayev, Nick Bassiliades, Hans-Georg Fill, Vitaliy Yakovyna, Heinrich C. Mayr, Vyacheslav Kharchenko, Vladimir Peschanenko, Mariya Shyshkina, Mykola Nikitchenko, Aleksander Spivakovsky. – P. 176-190. – (CEUR Workshop Proceedings, Vol. 1844). – Access mode : <http://ceur-ws.org/Vol-1844/10000176.pdf>.
71. Stein W. Elementary Number Theory: Primes, Congruences, and Secrets: A Computational Approach / William Stein. – New York : Springer Science+Business Media, 2009. – X, 167 p. – (Undergraduate Texts in Mathematics).
72. Stein W. What can SageMathCloud (SMC) do? [Electronic resource] / William Stein // Sage: open source mathematics software This is my blog about things related to sage. – may 1, 2014. – Access mode : <http://sagemath.blogspot.com/2014/05/what-can-sagemathcloud-smc-do.html>.
73. van Leeuwen M. A. A. LiE, A Package for Lie Group Computations / M. A. A. van Leeuwen, A. M. Cohen, B. Lisser. – Amsterdam : Computer Algebra Nederland, 1992. – 113 p.
74. Virtual Education Laboratory (VEL) [Electronic resource] // VMware Solution Exchange / [VMware, Inc.]. – 2014. – Access mode : <https://solutionexchange.vmware.com/store/products/virtual-education-laboratory-vel>.
75. Wang M. Mobile Cloud Learning for Higher Education: A Case Study of Moodle in the Cloud / Minjuan Wang, Yong Chen, Muhammad Jahanzaib Khan // The International Review of Research in Open and Distance Learning. – 2014. – Vol. 15. – April, No 5. – P. 254-267.
76. Watson J. Keeping Pace With K–12 Online Learning: An Annual Review of Policy and Practice [Electronic resource] / John Watson, Amy Murin, Lauren Vashaw, Butch Gemin, Chris Rapp. – 2010. – 148 p. – Access mode : [http://www.kpk12.com/wp-content/uploads/KeepingPaceK12\\_2010.pdf](http://www.kpk12.com/wp-content/uploads/KeepingPaceK12_2010.pdf).



77. Wu S. Education Informatization and University Teaching Reform / Shujuan Wu, Tan Na // Proceedings of the 2013 the International Conference on Education Technology and Information System (ICETIS 2013). – Atlantis Press, 2013. – P. 608-611. – DOI : <https://doi.org/10.2991/icetis-13.2013.137>. – (Advances in Intelligent Systems Research)
78. Алексанян Г. А. Формирование самостоятельной деятельности студентов СПО в обучении математике с использованием облачных технологий : дисс. ... канд. пед. наук : 13.00.02 – теория и методика обучения и воспитания (математика) / Алексанян Георгий Ашотович ; Армавирская гос. пед. академия. – Армавир, 2014. – 150 с.
79. Андрущенко В. П. Освітня політика (огляд порядку денного) / В. П. Андрущенко, В. Л. Савельєв. – К. : Леся, 2010. – 368 с.
80. Атаманчук Ю. М. Особливості управління вищою освітою в контексті адаптації діяльності регіональних ВНЗ до Болонського процесу / Ю. М. Атаманчук, Т. М. Десятов, С. О. Семеріков // Вісн. Черкас. ун-ту. Сер. Пед. науки. – 2010. – Вип. 183, ч. 4. – С. 20-26.
81. Атанов Г. А. Обучение и искусственный интеллект, или Основы современной дидактики высшей школы / Г. А. Атанов, И. Н. Пустынникова ; под общ. ред. Г. А. Атанова. – Донецк : Изд-во ДООУ, 2002. – 504 с.
82. Байдак В. А. Теория и методика обучения математике: наука, учебная дисциплина : моногр. / В. А. Байдак. – 2-е изд., стереотип. – М. : Флинта, 2011. – 264 с.
83. Баляева С. А. Теоретические основы фундаментализации общенаучной подготовки в системе высшего технического образования : дисс. ... д-ра пед. наук : 13.00.01 – общая педагогика / Баляева Светлана Анатольевна ; Мос. пед. гос. ун-т. – М., 1999. – 458 с.
84. Бауэр Ф. Л. Информатика. Вводный курс : в 2 ч. / Ф. Л. Бауэр, Г. Гооз. – М. : Мир, 1990. – Ч. 2. – 423 с. : ил.
85. Беспалько В. П. Киберпедагогика = Cyberpedagogy : введение в теорию и методологию педагогического обеспечения компьютерного обучения / Владимир Беспалько. – М. : Народное образование, 2018. – 238, [1] с.

86. Биков В. Ю. Моделі організаційних систем відкритої освіти : [монографія] / В. Ю. Биков. – К. : Атіка, 2009. – 684 с. : іл.
87. Биков В. Ю. Проблеми та завдання сучасного етапу інформатизації освіти / Биков В. Ю., Спирін О. М., Пінчук О. П. // Наукове забезпечення розвитку освіти в Україні: актуальні проблеми теорії і практики (до 25-річчя НАПН України). : зб. наук. праць / Нац. Акад. пед. наук України. – Київ : Сам, 2017. – С. 191-198.
88. Биков В. Ю. Проект положення про електронні освітні ресурси [Електронний ресурс] / Биков В. Ю., Шишкіна М. П., Лаврентьєва Г. П., Дем'яненко В. М., Лапінський В. В., Запорожченко Ю. Г., Пірко М. В. ; Інститут інформаційних технологій і засобів навчання НАПН України. – [К.], 30.09.2014. – 11 с. – Режим доступу : <http://lib.iitta.gov.ua/1041/>.
89. Болгова Е. В. Автоматизация процесса разработки виртуальных лабораторий на основе облачных вычислений : автореф. дисс. ... канд. техн. наук : 05.13.06 – автоматизация и управление технологическими процессами и производствами (образование) / Болгова Екатерина Владимировна ; [Санкт-Петербургский нац. исслед. ун-т инф. технол., механ.]. – Санкт-Петербург, 2012. – 18 с.
90. Брескіна Л. В. Професійна підготовка майбутніх вчителів інформатики на основі сучасних мережевих інформаційних технологій : автореф. дис... канд. пед. наук : 13.00.02 – теорія та методика навчання інформатики / Брескіна Лада Валентинівна ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2003. – 17 с.
91. Бугаєць Н. О. Розвиток навчально-дослідницьких умінь студентів фізико-математичних спеціальностей у процесі навчання математичної інформатики : автореф. дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання (інформатика) / Бугаєць Наталія Олександрівна ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2016. – 20 с.
92. Бурма С. А. Засоби хмарних обчислень у комп'ютерній інженерії / С. А. Бурма, О. М. Туравініна // Комп'ютерні інтелектуальні системи та мережі : матер. IV Всеукр. конф. асп., студ. та молод. вчен. (23-25 березня 2011 р.) / Кривор. техн. ун-т, Каф. комп. систем та мереж. – Кривий Ріг :

- Видавництво Криворізького технічного університету, 2011. – С. 33-34.
93. Вакалюк Т. А. Вітчизняний досвід проектування хмаро зорієнтованого навчального середовища для закладів освіти / Тетяна Анатоліївна Вакалюк // Новітні комп'ютерні технології. – Кривий Ріг : Видавничий центр ДВНЗ «Криворізький національний університет», 2014. – Том XII : спецвипуск «Хмарні технології в освіті». – С. 20-24.
94. Василенко Н. А. Математика-0 для педагогічних ВНЗ: вступ до булевої алгебри / Наталія Василенко, Володимир Петров, Оксана Туравініна // Математика в школі. – 2008. – № 3. – С. 16-21.
95. Вдовичин Т. Я. Основні завдання впровадження дистанційних технологій у підготовку бакалаврів інформатики заочної форми навчання [Електронний ресурс] / Вдовичин Тетяна Ярославівна // Інформаційні технології і засоби навчання. – 2012. – Том 29, № 3. – 11 с. – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/download/672/509>.
96. Власенко К. В. Застосування хмарних технологій в навчанні вищої математики у закладах вищої технічної освіти / К. Власенко, І. Сітак, О. Чумак // Актуальні питання природничо-математичної освіти. – Суми : СумДПУ імені А. С. Макаренка, 2018. – Вип. 10. – С. 108-113.
97. Власенко К. В. Теоретичні й методичні аспекти навчання вищої математики з використанням інформаційних технологій в інженерній машинобудівній школі : монографія / К. Власенко ; науковий редактор д. пед. н., проф. О. І. Скафа. – Донецьк : Ноулідж, 2011. – 410 с.
98. Волкова Н. П. Інтерактивні технології навчання у вищій школі : навчально-методичний посібник / Н. П. Волкова. – Дніпро : Університет імені Альфреда Нобеля, 2018. – 360 с.
99. Волошина Т. В. Використання гібридного хмаро зорієнтованого навчального середовища для формування самоосвітньої компетентності майбутніх фахівців з інформаційних технологій : дис. ... канд. пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Волошина Тетяна Володимирівна ; Ін-т інформ. технологій і засобів навчання. – К., 2018. – 293 с.
100. Воронкін О. С. «Хмарні» обчислення як основа формування персональних

- навчальних середовищ / О. С. Воронкін // Матеріали другої міжнародної науково-практичної конференції FOSS Lviv 2012 : зб. наук. праць / Львів, 26-28 квітня 2012 р. / Під редакцією : Злобін Г. Г., Апуневич С. Є., Апуневич С. В., Ванькевич Д. Є. – Львів, 2012. – С. 143-146.
101. Все необходимое для учебного заведения [Электронный ресурс] // Google Apps для учебных заведений / Google. – Режим доступа : <https://www.google.com/intx/ru/work/apps/education/benefits.html>.
102. Галкина Л. С. Методика развития ИКТ-компетентности будущих экономистов и менеджеров средствами облачных технологий при обучении дисциплинам информационного цикла : дисс. ... канд. пед. наук : 13.00.02 – теория и методика обучения и воспитания (информатика) / Галкина Людмила Сергеевна ; Сиб. федер. ун-т. – Пермь, 2017. – 177 с.
103. Гершунский Б. С. Философия образования для XXI века / Б. С. Гершунский. – М. : Пед. о-во России, 2002. – 508, [3] с.
104. Глушков В. М. Основы безбумажной информатики. Изд. 2-е, испр. / В. М. Глушков. – М. : Наука. Гл. ред. физ.-мат. лит, 1987. – 552 с.
105. Горошко Ю. В. Система інформаційного моделювання у підготовці майбутніх учителів математики та інформатики : дис. ... д-ра пед. наук : 13.00.02 – теорія та методика навчання (інформатика) / Горошко Юрій Васильович ; Черніг. нац. пед. ун-т імені Т. Г. Шевченка. – Чернігів, 2013. – 470 с.
106. Гуржій А. М. Взаємозв'язок інформатизації суспільства й системи освіти / А. М. Гуржій, В. В. Лапінський // Комп'ютер у школі та сім'ї. – 2015. – № 8. – С. 5-9.
107. Деякі питання визначення середньострокових пріоритетних напрямів інноваційної діяльності загальнодержавного рівня на 2017-2021 роки [Електронний ресурс] : Постанова № 1056 / Кабінет Міністрів України. – К., 28 грудня 2016 р. – Режим доступу : <https://zakon.rada.gov.ua/laws/show/1056-2016-%D0%BF>.
108. Деякі питання визначення середньострокових пріоритетних напрямів інноваційної діяльності загальнодержавного рівня на 2017-2021 роки [Електронний ресурс] : Постанова № 980 / Кабінет Міністрів України. –

- К., 18 жовтня 2017 р. – Режим доступу : <https://zakon.rada.gov.ua/laws/show/980-2017-%D0%BF>.
109. Дичківська І. М. Інноваційні педагогічні технології : підр. / Ілона Дичківська. – 3-тє вид., випр. – К. : Академвидав, 2015. – 304 с. – (Альма-матер+).
  110. Дмитренко Т. А. Дидактические основы управления учебной деятельностью студентов (на материале технических дисциплин) : автореф. дис. ... д-ра пед. наук : 13.00.01 – теория и история педагогики / Тамара Александровна Дмитренко ; Харьков. инж.-пед. ин-т. – Екатеринбург, 1991. – 33 с.
  111. Докучаєва В. В. Проектування інноваційних педагогічних систем у сучасному освітньому просторі : моногр. / Вікторія Вікторівна Докучаєва. – Луганськ : Альма-матер, 2005. – 303 с.
  112. Е Мьинт Найнг. Разработка системы запуска ресурсоемких приложений в облачной гетерогенной среде : автореф. дисс. ... канд. техн. наук : 05.13.15 – вычислительные машины, комплексы и компьютерные сети / Е Мьинт Найнг ; [Санкт-Петербургский гос. электротехн. ун-т «ЛЭТИ» им. В. И. Ульянова (Ленина)]. – Санкт-Петербург, 2013. – 19 с.
  113. Ёлгина Л. С. Фундаментализация образования в контексте устойчивого развития общества: сущность, концептуальные основания : дисс. ... канд. филос. наук : 09.00.11 – социальная философия / Ёлгина Лариса Сергеевна ; Бурятский гос. ун-т. – Улан-Удэ, 2000. – 155 с.
  114. Єфименко В. В. Методика навчання комп'ютерної математики майбутніх учителів інформатики : автореф. дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання (інформатика) / Єфименко Василь Володимирович ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2015. – 22 с.
  115. Жалдак М. И. О некоторых методических аспектах обучения информатике в школе и педагогическом университете / М. И. Жалдак // Методология и технология образования в XXI веке: математика, информатика, физика : матер. межд. науч.-практ. конф. 17-18 ноября 2005 г. / Белорусский гос. пед. ун-т имени Максима Танка. – Минск, 2006. – С. 260-268.
  116. Жалдак М. І. Основи теорії і методів оптимізації : [навч. посіб. для студ.

- мат. спец. вищ. навч. закл.] / М. І. Жалдак, Ю. В. Триус. – Черкаси : Брама-Україна, 2005. – 607 с.
117. Жалдак М. І. Шкільній інформатиці – 25! / Жалдак М. І., Рамський Ю. С. // Наук. часоп. Нац. пед. ун-ту імені М. П. Драгоманова. – Серія №2. Комп'ютерно-орієнтовані системи навчання : зб. наук. пр. / Редкол. – К. : НПУ імені М. П. Драгоманова, 2010. – № 8 (15). – С. 3-17.
118. Жалдак М. І. Проблеми інформатизації навчального процесу в середніх і вищих навчальних закладах / М. І. Жалдак // Комп'ютер в школі та сім'ї. – 2013. – № 3. – С. 8-15.
119. Жугастров О. О. Хмарні обчислення: сутність, недоліки, переваги / О. О. Жугастров // Комп'ютер у школі та сім'ї. – 2011. – № 2. – С. 54-56.
120. Загвязинский В. И. Дидактика высшей школы : текст лекций / В. И. Загвязинский ; Челяб. политехн. ин-т им. Ленинского комсомола, Челяб. регион. центр высш. шк. – Челябинск : ЧПИ, 1990. – 95, [2] с.
121. Зверева А. В. Формирование маркетинга услуг системной интеграции на основе облачных технологий : автореф. дисс. ... канд. эконом. наук : 08.00.05 – экономика и управление народным хозяйством: маркетинг / Зверева Анна Владимировна ; [Финанс. ун-т при Правительстве Российской Федерации]. – М., 2014. – 29 с.
122. Зиновкина М. М. Формирование творческого технического мышления и инженерных умений студентов технических вузов : автореф. дис. ... д-ра пед. наук : 13.00.01 – общая педагогика / Зиновкина Мирослава Михайловна ; НИИ труд. обуч. и проф. ориентации. – М., 1989. – 34 с.
123. Иванников В. П. Облачные вычисления в образовании, науке и госсекторе [Электронный ресурс] / Иванников Виктор Петрович ; Ин-т сист. прогр. – [М.], [2010]. – 23 с. – Режим доступа : <http://grid2010.jinr.ru/files/pdf/cloud.pdf>.
124. Использование SageTeX [Электронный ресурс] // Учебное пособие Sage v 4.3.4 / The Sage Development Team. – 2005-2010. – Режим доступа : <http://freetonik.com/sage/tutorial/sagetex.html>.
125. Иванов С. В. Філософія і фундаменталізація університетської освіти / Сергій Иванов, Микола Кітов // Вища школа. – 2013. – № 1. – С. 20-26.

126. Іваськів І. С. Активізація навчально-пізнавальної діяльності учнів на основі систем штучного інтелекту при навчанні інформатики в старшій школі : автореф. дис. канд. пед. наук : 13.00.02 – теорія та методика навчання інформатики / Іваськів Ігор Степанович ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2000. – 20 с.
127. Інформаційна система "Конкурс" Міністерства освіти і науки України. Вступна кампанія 2011 >> 0501 Інформатика та обчислювальна техніка >> 6.050101 комп'ютерна інженерія [Електронний ресурс]. – 2011. – Режим доступу : <http://www.vstup.info/2011/i2011b0501d6.050102.html>.
128. Кадемія М. Ю. Формування професійних знань учнів профтехучилищ засобами мережних комунікацій : дис. ... канд. пед. наук : 13.00.04 – теорія та методика професійної освіти / Кадемія Майя Юхимівна ; Ін-т педаг. і психол. проф. освіти. – К., 2004. – 260 с.
129. Казанцев С. Я. Дидактические основы и закономерности фундаментализации обучения студентов в современной высшей школе : дисс. ... д-ра пед. наук : 13.00.01 – общая педагогика / Казанцев Сергей Яковлевич ; Казанский гос. ун-т. – Казань, 2000. – 295 с.
130. Калинин С. И. Методическая система обучения студентов педвуза дифференциальному и интегральному исчислению функций в контексте фундаментализации образования : автореф. дисс. ... д-ра пед. наук : 13.00.02 – теория и методика обучения и воспитания (математика) / Калинин Сергей Иванович ; [Ин-т содерж. и метод. обуч.]. – М., 2009. – 44 с.
131. Касьян С. П. Управління документообігом у закладах післядипломної педагогічної освіти на основі хмарних технологій : дис. ... канд. пед. наук : 13.00.06 – теорія і методика управління освітою / Касьян Сергій Петрович ; Ун-т менеджменту освіти. – К., 2016. – 247 с.
132. Кафедра інформаційних систем [Електронний ресурс]. – К. : Київ. нац. ун-т ім. Т. Г. Шевченка, [б. р.] – Режим доступу : <http://scsp.org.ua/uk/faculty/depIS>.
133. Кислова М. А. Розвиток мобільного навчального середовища з вищої математики у підготовці інженерів-електромеханіків : дис. ... канд. пед.

- наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Кислова Марія Алімівна ; Кривор. нац. ун-т. – Кривий Ріг, 2014. – 274 с.
134. Кіяновська Н. М. Теоретико-методичні засади використання інформаційно-комунікаційних технологій у навчанні вищої математики студентів інженерних спеціальностей у Сполучених Штатах Америки : монографія / Н. М. Кіяновська, Н. В. Рашевська, С. О. Семеріков // Теорія та методика електронного навчання. – Кривий Ріг : Видавничий відділ ДВНЗ «Криворізький національний університет», 2014. – Том V. – Випуск 1 (5) : спецвипуск «Монографія в журналі». – 316 с. : іл.
135. Кобилін А. М. Використання сучасних підходів «хмарних» технологій для систем підтримки прийняття рішень в задачах пошуку оптимальних альтернатив в банківській сфері [Електронний ресурс] / А. М. Кобилін, В. Ю. Дубницький // Фінансово-кредитна діяльність: проблеми теорії та практики : зб. наук. пр. – 2011. – №1 (10). – Частина 2. – 8 с. – Режим доступу : [http://www.nbu.gov.ua/old\\_jrn/Soc\\_Gum/Fkd/2011\\_1/part2/51.PDF](http://www.nbu.gov.ua/old_jrn/Soc_Gum/Fkd/2011_1/part2/51.PDF).
136. Кобильник Т. П. Компетентнісний підхід при вивченні «Математичної інформатики» у педагогічному університеті [Електронний ресурс] / Кобильник Тарас Петрович // Інформаційні технології і засоби навчання. – 2007. – Том 2, № 1. – 11 с. – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/download/273/259>.
137. Кобильник Т. П. Методична система навчання математичної інформатики у педагогічному університеті : автореф. дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання (інформатика) / Кобильник Тарас Петрович ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2009. – 19 с.
138. Кобильник Т. П. Методична система навчання математичної інформатики у педагогічному університеті : дис. ... канд. пед. наук : 13.00.02 – теорія і методика навчання (інформатика) / Кобильник Тарас Петрович ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2008. – 279 с.
139. Кобильник Т. П. Системи комп'ютерної математики: Maple, Mathematica, Maxima / Т. П. Кобильник. – Дрогобич : Коло, 2008. – 338 с.
140. Кобильник Т. П. Фундаментальність інформатичної освіти /



- Т. П. Кобильник // Наук. часоп. НПУ імені М. П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання : зб. наукових праць / Редрада. – К. : НПУ імені М.П. Драгоманова, 2007. – № 5 (12). – С. 78-81.
141. Ковтонюк М. М. Теоретичні і методичні засади фундаменталізації загальнопрофесійної підготовки майбутнього вчителя математики : дис. ... д-ра пед. наук : 13.00.04 – теорія і методика професійної освіти / Ковтонюк Мар'яна Михайлівна ; Вінницький держ. пед. ун-т імені Михайла Коцюбинського. – Вінниця, 2014. – 496 с.
142. Колесников А. В. Применение «облачных» вычислений в программах стационарного и дистанционного обучения [Электронный ресурс] / А. В. Колесников, С. А. Деревянко, Е. В. Ромашка // Вісник Східноукр. нац. ун-ту ім. Володимира Даля. – Луганськ, 2011. – № 3 (157). – 5 с. – Режим доступа : [http://www.nbu.gov.ua/old\\_jrn/Soc\\_Gum/VSUNU/2011\\_3/Kolesnikov.pdf](http://www.nbu.gov.ua/old_jrn/Soc_Gum/VSUNU/2011_3/Kolesnikov.pdf).
143. Кондратьев В. В. Фундаментализация профессионального образования специалиста на основе непрерывной математической подготовки в условиях технологического университета : дисс. д-ра пед. наук : 13.00.08 – теория и методика профессионального образования / Кондратьев Владимир Владимирович ; Казанский гос. ун-т. – Казань, 2000. – 421 с.
144. Копаєв О. В. Фундаментальний аспект базового курсу інформатики / Копаєв О. В., Триус Ю. В. // Сучасний стан і перспективи шкільних курсів математики та інформатики у зв'язку з реформуванням у галузі освіти (Дрогобич, 14-16 листопада 2000 р.) : [всеукр. наук.-практ. конф.] : тези доп. – Дрогобич : ДДПУ, 2000. – С. 138-140.
145. Кормен Т. Алгоритмы: построение и анализ / Томас Кормен, Чарльз Лейзерсон, Рональд Ривест, Клиффорд Штайн. – М. : Вильямс, 2013. – 1324 с.
146. Коротун О. В. Використання хмаро зорієнтованого середовища у навчанні баз даних майбутніх учителів інформатики : дис. ... канд. пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Коротун Ольга Володимирівна ; Ін-т інформ. технол. і засоб. навчання. – К., 2018. – 356 с.
147. Коротяев Б. И. Нестандартный взгляд на стандарты высшего

- образования : моногр. / Б. И. Коротяев, В. С. Курило, С. В. Савченко ; Луган. нац. ун-т им. Тараса Шевченко. – Старобельск : Изд-во ГУ «ЛНУ имени Тараса Шевченко», 2016. – 293 с.
148. Корсакова О. К. Принципи навчання / О. К. Корсакова // Енциклопедія освіти / Акад. пед. наук України ; головний ред. В. Г. Кремень. – К. : Юрінком Інтер, 2008. – С. 713-714.
149. Кривонос О. М. Етапи проектування хмаро зорієнтованого середовища навчання баз даних майбутніх учителів інформатики [Електронний ресурс] / О. М. Кривонос, О. В. Коротун // Інформаційні технології і засоби навчання. – 2018. – № 1 (63). – С. 130-145. – Режим доступу : <https://journal.iitta.gov.ua/index.php/itlt/article/view/1866/1299>.
150. Круглик В. С. Система підготовки майбутніх інженерів-програмістів до професійної діяльності у вищих навчальних закладах : дис. ... д-ра пед. наук : 13.00.04 – теорія і методика професійної освіти / Круглик Владислав Сергійович ; Мелітоп. держ. пед. ун-т імені Богдана Хмельницького, Запор. нац. ун-т. – Запоріжжя, 2018. – 682 с.
151. Куписевич Ч. Основы общей дидактики : пер. с польского и предисл. О. В. Долженко / Ч. Куписевич. – М. : Высш. шк., 1986. – 367, [1] с.
152. Купін А. І. Застосування інформаційних технологій для підготовки спеціалістів за напрямом «Комп'ютерна інженерія» / А. І. Купін, В. А. Чубаров, О. М. Туравініна // Новітні комп'ютерні технології : матер. VII Міжнар. наук.-техн. конф. : Київ-Севастополь, 15-18 вересня 2009 р. – К. : Мінрегіон України, 2009. – С. 101-102.
153. Купін А. І. Методології навчання за напрямком «Комп'ютерна інженерія» на базі Криворізького технічного університету / А. І. Купін, В. А. Чубаров, О. М. Туравініна // Наук. часоп. Нац. пед. ун-ту ім. М. П. Драгоманова. Серія № 5. Педагогічні науки: реалії та перспективи. – Вип. 17 : зб. наук. пр. / За ред. В. Д. Сиротюка. – К. : Вид-во НПУ імені М. П. Драгоманова, 2009. – С. 102-108.
154. Лаптев В. В. Методическая теория обучения информатике. Аспекты фундаментальной подготовки / В. В. Лаптев, Н. И. Рыжова, М. В. Швецкий ; Рос. гос. пед. ун-т имени А. И. Герцена. – СПб. : Издание

- С.-Петербурзького університета, 2003. – 352 с.
155. Левченко І. В. Розвиток системи методическої підготовки учителів інформатики в умовах фундаменталізації освіти: дис. ... д-ра пед. наук: 13.00.02 – теорія і методика освіти і виховання (інформатика) / Левченко Ірина Віталіївна. – Москва, 2009. – 527 с.
156. Лингер Р. Теорія і практика структурного програмування / Р. Лингер, Х. Миллс, Б. Уитт. – М. : Мир, 1982. – 406 с.
157. Литвинова С. Г. Методика використання технологій віртуального класу вчителем в організації індивідуального навчання учнів : автореф. дис. ... канд. пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Литвинова Світлана Сергіївна ; Ін-т інф. технол. і зас. навч. – К., 2011. – 25 с.
158. Литвинова С. Г. Проектування хмаро зорієнтованого навчального середовища загальноосвітнього навчального закладу : моногр. / С. Г. Литвинова ; Ін-т інф. технол. і зас. навч. – К. : Компринт, 2016. – 354 с.
159. Литвинова С. Г. Теоретико-методичні основи проектування хмаро зорієнтованого навчального середовища загальноосвітнього навчального закладу : дис. ... д-ра пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Литвинова Світлана Григоріївна ; Ін-т інф. технол. і зас. навч. – К., 2016. – 602 с.
160. Лузік Е. В. Педагог технічного університету 21 століття: пошук інноваційної моделі [Електронний ресурс] / Е. В. Лузік // Вісник Нац. авіац. ун-ту. Серія: Педагогіка, Психологія. – 2016. – № 8. – 6 с. – Режим доступу : <http://jrn1.nau.edu.ua/index.php/VisnikPP/article/download/12351/16695>. – DOI : 10.18372/2411-264X.8.12351.
161. Лук'янченкова В. Є. Конспект лекцій з курсу «Технологія науково-дослідної діяльності» (для студ. 4 курсу ден. форм. навч. спец. 8.050201 – «Менеджмент організацій») / В. Є. Лук'янченкова. – Харків : ХНАМГ, 2007. – 50 с.
162. Маркова О. М. Загальна структура засобів хмарних технологій навчання основ математичної інформатики / Оксана Миколаївна Маркова // Новітні

- комп'ютерні технології. – Кривий Ріг : ДВНЗ «Криворізький національний університет». – 2017. – Том XV. – С. 231-234.
163. Маркова О. М. Історичні аспекти розвитку хмарних технологій / Маркова О. М. // Тези доп. наук.-практ. сем. «Хмарні технології в сучасному університеті» (ХТСУ–2015) : Черкаси, 24 березня 2015 р. – Черкаси : ЧДТУ, 2015. – С. 29-30.
164. Маркова О. М. Моделі використання хмарних технологій у підготовці ІТ-фахівців / Маркова О. М. // Наук. часоп. НПУ імені М. П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання : зб. наук. праць / Редрада. – К. : НПУ імені М. П. Драгоманова, 2016. – № 18 (25). – С. 85-94.
165. Маркова О. М. Модель методичної системи та цілі навчання основ математичної інформатики студентів технічних університетів / Маркова О. М. // Вісник Черкаського університету. Серія Педагогічні науки. – 2016. – № 7. – С. 36-42.
166. Маркова О. М. Теоретичні засади навчання основ математичної інформатики студентів технічних університетів з використанням хмарних технологій / О. М. Маркова // Новітні комп'ютерні технології. – Кривий Ріг : ДВНЗ «Криворізький національний університет». – 2016. – Том XIV. – С. 63-64.
167. Маркова О. М. Хмарні технології навчання: витоки [Електронний ресурс] / Маркова Оксана Миколаївна, Семеріков Сергій Олексійович, Стрюк Андрій Миколайович // Інформаційні технології і засоби навчання. – 2015. – Том 46, № 2. – С. 29-44. – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/download/1234/913>.
168. Маркова О. М. Хмарні технології навчання: спроба визначення / О. М. Маркова // Новітні комп'ютерні технології. – Кривий Ріг : ДВНЗ «Криворізький національний університет». – 2014. – Том XII : спецвипуск «Хмарні технології в освіті». – С. 244-248.
169. Математична інформатика [Електронний ресурс] / Науково-дослідна частина Київського національного університету імені Тараса Шевченка // Наукові школи. – 16.09.2013. – Режим доступу : <http://science.univ.kiev.ua/>

research/scientific\_school/matematychna-informatyka/.

170. Мацко Л. А. Основи психології та педагогіки / Л. А. Мацко, М. Д. Прищак ; Вінниц. нац. техн. ун-т. – Вінниця : ВНТУ, 2009. – 158 с.
171. Меняев А. Ф. Взаимосвязь преподавания и учения в учебном процессе технического вуза : автореф. дис. ... д-ра пед. наук : 13.00.01 – общая педагогика / Меняев Александр Федорович ; Моск. пед. гос. ун-т им. В. И. Ленина. – М., 1995. – 32 с.
172. Меньяйленко О. С. Теоретико-методологічні основи синтезу індивідуалізованих стратегій управління дидактичним процесом в автоматизованих навчальних системах : дис... д-ра техн. наук : 05.13.06 / Меньяйленко Олександр Сергійович ; Луган. нац. пед. ун-т ім. Тараса Шевченка. – Луганськ, 2007. – 404 с.
173. Мерзликін О. В. Перспективні хмарні технології в освіті / Мерзликін О. В., Семеріков С. О. // Тези доп. наук.-практ. сем. «Хмарні технології в сучасному університеті» (ХТСУ–2015) : Черкаси, 24 березня 2015 р. – Черкаси : ЧДТУ, 2015. – С. 31-33.
174. Мерзликін О. В. Хмарні технології як засіб формування дослідницьких компетентностей старшокласників у процесі профільного навчання фізики : дис. ... канд. пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Мерзликін Олександр Володимирович ; Ін-т інформ. технол. і зас. навч. – К., 2016. – 341 с.
175. Мерзликін О. В. Хмаро зорієнтовані електронні освітні ресурси підтримки навчальних фізичних досліджень / Мерзликін Олександр Володимирович // Інформаційні технології і засоби навчання. – 2015. – Том 49, № 5. – С. 106-117. – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/download/1269/956>.
176. Мешко Г. М. Вступ до педагогічної професії : навч. посібник / Г. М. Мешко. – К. : Академвидав, 2010. – 200 с. – (Альма-матер).
177. Михалін Г. О. Професійна підготовка вчителя математики у процесі навчання математичного аналізу : [моногр.] / Г. О. Михалін. – К. : НПУ імені М. П. Драгоманова, 2003. – 320 с.
178. Мінтій І. С. Формування у студентів педагогічних університетів

- компетентностей з програмування на основі функціонального підходу : дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання (інформатика) / Мінтій Ірина Сергіївна ; Нац. пед. ун-т імені М. П. Драгоманова. – 254 с.
179. Моделювання й інтеграція сервісів хмаро зорієнтованого навчального середовища: колект. монографія / [Копняк Н., Корицька Г., Литвинова С., Носенко Ю., Пойда С., Седой В., Сіпачова О., Сокол І., Спирін О., Стромило І., Шишкіна М. ; за заг. ред. С. Г. Литвинової] ; Ін-т інформ. технологій і засобів навчання. – К. : Компринт, 2015. – 160, [1] с.
180. Модло Є. О. Засоби мобільного доступу до Scilab [Електронний ресурс] / Є. О. Модло, С. О. Семеріков, О. В. Сироватський // Міжнар. наук.-метод. Інтернет-конф. «Проблеми вищої математичної освіти: виклики сучасності». 17-18 травня 2018 року. Вінниця, Вінницький нац. техн. ун-т. – 11 с. – Режим доступу : <https://conferences.vntu.edu.ua/index.php/pmouc/pmouc/paper/view/5671>.
181. Мордкович А. Г. Профессионально-педагогическая направленность специальной подготовки учителя математики в педагогическом институте : дисс. ... д-ра пед. наук : 13.00.02 – методика преподавания математики / Александр Григорьевич Мордкович ; Моск. гос. заочн. пед. ин-т. – М., 1986. – 355 с.
182. Морзе Н. В. Основи методичної підготовки вчителя інформатики : [монографія] / Н. В. Морзе ; Нац. пед. ун-т ім. М. П. Драгоманова. – К. : Курс, 2003. – 372 с.
183. Морзе Н. В. Педагогічні аспекти використання хмарних обчислень / Н. В. Морзе, О. Г. Кузьмінська // Інформаційні технології в освіті. – 2011. – Вип. 9. – С. 20-29.
184. Морзе Н. В. Система методичної підготовки майбутніх вчителів інформатики в педагогічних університетах : дис. ... д-ра пед. наук : 13.00.02 – теорія і методика навчання інформатики / Морзе Наталія Вікторівна ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2003. – 605 с.
185. Морзе Н. В. Формування інформатичних компетентностей учнів середньої школи [Електронний ресурс] / Морзе Наталія Вікторівна, Кузьмінська

- Олена Геронтіївна // Інформаційні технології і засоби навчання. – 2011. – Том 23, № 3. – 21 с. – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/download/467/413>.
186. Найповніший тлумачний словник української мови онлайн >> методика [Електронний ресурс] / V&V Project. – 2010-2012. – Режим доступу : <http://eslovnyk.com/методика>.
187. Новиков А. М. Российское образование в новой эпохе: Парадоксы наследия, векторы развития : публицист. моногр. / А. Новиков. – М. : Эгвес, 2000. – 272 с.
188. Облачные технологии и образования / Сейдаметова З. С., Аблялимова Э. И., Меджитова Л. М., Сейтвелиева С. Н., Темненко В. А. : под общ. ред. проф. З. С. Сейдаметовой. – Симферополь : ДИАЙПИ, 2012. – 204 с.
189. Організація та функціонування мережі ресурсних центрів дистанційної освіти загальноосвітніх навчальних закладів : моногр. / [Богачков Ю. М., Биков В. Ю., Пінчук О. П., Олійник В. В., Буров О. Ю., Манако А. Ф., Коневщинська О. Е., Ухань П. С., Іванюк І. В., Рождественська Д. Б., Барладим В. М., Корнієць О. М., Мушка І. В. ; наук. ред. Ю. М. Богачков] ; ІТЗН НАПН України. – К. : Атіка, 2014. – 184 с.
190. Освітньо-кваліфікаційна характеристика бакалавра напрямку підготовки 6.050102 «Комп'ютерна інженерія» : галузевий стандарт вищої освіти України / МОНмолодьспорт України. – К., 2011. – IV+28 с.
191. Освітньо-професійна програма бакалавра напрямку підготовки 6.050102 «Комп'ютерна інженерія» : галузевий стандарт вищої освіти України / МОНмолодьспорт України. – К., 2011. – IV+49 с.
192. Палій С. В. Хмарні механізми формування інформаційно-організаційного середовища довузівської підготовки студентів : дис. ... канд. техн. наук : 05.13.06 – інформаційні технології / Палій Сергій Володимирович ; Київ. нац. ун-т буд-ва і архітектури. – К., 2014. – 201 с.
193. Панченко Л. Ф. Інформаційно-освітнє середовище сучасного університету : моногр. / Л. Ф. Панченко ; Луган. нац. ун-т імені Тараса Шевченка. – Луганськ : ДЗ «ЛНУ імені Тараса Шевченка», 2010. – 279 с.

194. Панченко Л. Ф. Нові тренди аналізу даних [Електронний ресурс] / Л. Ф. Панченко // Науковий вісник Донбасу. – 2013. – № 4. – 17 с. – Режим доступу : <http://nvd.luguniv.edu.ua/archiv/NN24/13plftad.pdf>.
195. Педагогика и психология высшей школы : учеб. пособие для студентов и аспирантов вузов / [М. В. Буланова-Топоркова и др.]. – 2-е изд., доп. и перераб. – Ростов н/Д : Феникс, 2002. – 543 с. – (Высшее образование).
196. Педагогіка : базовий підруч. для студентів вищ. навч. закл. III-IV р. а. / [авт. кол. : І. Ф. Прокопенко, В. Г. Кремень, Є. М. Суліма та ін.] . – Харків : Фоліо, 2015. – 572 с.
197. Попель М. В. Організація навчання математичних дисциплін у SageMathCloud : навч. посібн. / М. В. Попель // Теорія та методика навчання математики, фізики, інформатики. – Кривий Ріг : Видавничий відділ ДВНЗ «Криворізький національний університет», 2015. – Том XIII.– Випуск 1 (35) : спецвипуск «Навчальний посібник у журналі». – 111 с.
198. Попель М. В. Хмарний сервіс SageMathCloud як засіб формування професійних компетентностей вчителя математики : дис. ... канд. пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Попель Майя Володимирівна ; Ін-т інформ. технол. і зас. навч. – К., 2017. – 311 с.
199. Попков В. А. Дидактика высшей школы : учебное пособие / В. А. Попков, А. В. Коржуев. – 4-е изд., испр. и доп. – М. : Юрайт, 2017. – 227 с. – (Образовательный процесс).
200. Преимущество в обучении математике : сб. статей : [пособие для учителей] / Сост. А. М. Пышкало. – М. : Просвещение, 1978. – 239 с.
201. Про вищу освіту [Електронний ресурс] : Закон № 1556-VII / Верховна Рада України. – К., 01.07.2014. – Режим доступу : <http://zakon4.rada.gov.ua/laws/show/1556-18>.
202. Про затвердження Державного стандарту базової і повної загальної середньої освіти : Постанова № 1392, Стандарт, План [Електронний ресурс] / Кабінет Міністрів України. – К., 23.11.2011. – Режим доступу : <http://zakon2.rada.gov.ua/laws/show/1392-2011-%D0%BF>.
203. Про затвердження Положення про дистанційне навчання [Електронний ресурс] / МОН України : Наказ № 466, Положення. – 25.04.2013. – Режим



- доступу : <http://zakon4.rada.gov.ua/laws/show/z0703-13>.
204. Про Рекомендації парламентських слухань на тему: «Створення в Україні сприятливих умов для розвитку індустрії програмного забезпечення» : Постанова, Рекомендації № 4538-VI [Електронний ресурс] / Верховна Рада України. – 15.03.2012. – Режим доступу : <http://zakon2.rada.gov.ua/laws/show/4538-17>.
205. Пышкало А. М. Методическая система обучения геометрии в начальной школе : авторский доклад по монографии «Методика обучения элементам геометрии в начальных классах», представленной на соискание ученой степени докт. пед. наук : 13.00.02 / А. М. Пышкало ; НИИ СиМО АПН СССР. – М., 1975. – 60 с.
206. Рамський Ю. С. Вивчення інформаційно-пошукових систем мережі Інтернет : навч.-метод. посіб. / Ю. С. Рамський, О. В. Резіна ; Нац. пед. ун-т ім. М. П. Драгоманова. – К. : [б. в.], 2004. – 60 с.
207. Решетова З. А. Один из подходов к построению учебной дисциплины / З. А. Решетова, С. А. Беляева // Вестник высшей школы. – 1985. – №1. – С. 35-39.
208. Розов Н. Х. Педагогика высшей школы / Розов Н. Х., Попков В. А., Коржуев А. В. – М. : Юрайт, 2016. – 157 с.
209. Рычкова А. А. Дистанционные образовательные технологии как средство формирования профессиональной самостоятельности будущих инженеров-программистов : дисс. ... канд. пед. наук : 13.00.08 – теория и методика профессионального образования / Рычкова Анастасия Александровна ; [Оренбург. гос. ун-т]. – Оренбург, 2010. – 235 с.
210. Сахаров А. М. Cloud Security [Электронный ресурс] / Сахаров А. М. // Программа «Университетский кластер». Москва, РАН, 31 мая – 3 июня, 2011. – М., 2011. – С. 79. – Режим доступа : <http://www.ispras.ru/unicluster/conf/2011/catalog.pdf>.
211. Сейдаметова З. С. Методична система рівневої підготовки майбутніх інженерів-програмістів за спеціальністю «Інформатика» : автореф. дис... д-ра пед. наук : 13.00.02 – теорія та методика навчання (інформатика) / Сейдаметова Зарема Сейдаліївна ; Нац. пед. ун-т імені М. П. Драгоманова.

- К., 2007. – 40 с.
212. Сейдаметова З. С. Тренды IT-сферы в образовании / З. С. Сейдаметова, Э. И. Абляимова, Л. А. Манжос, В. А. Темненко // Технологии в гуманитарном образовании : матер. V Межд. науч.-прак. конф., 20-24 июня 2012 года. – Пятигорск : ПГЛУ, 2012. – С. 120-125.
213. Селевко Г. К. Педагогические технологии на основе активизации, интенсификации и эффективного управления УВП / Г. К. Селевко. – М. : НИИ шк. технологий, 2005. – 284 с. – (Энциклопедия образовательных технологий).
214. Семеріков С. О. Теоретико-методичні основи фундаменталізації навчання інформатичних дисциплін у вищих навчальних закладах : дис... д-ра пед. наук : 13.00.02 – теорія і методика навчання (інформатика) / Семеріков Сергій Олексійович ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2009. – 536 арк. – Бібліогр. : арк. 470-536.
215. Семеріков С. О. Фундаменталізація навчання інформатичних дисциплін у вищій школі : моногр. / Семеріков С. О. ; наук. ред. акад. АПН України, д. пед. н., проф. М. І. Жалдак. – Кривий Ріг : Мінерал ; К. : НПУ ім. М. П. Драгоманова, 2009. – 340 с.
216. Сергієнко В. П. Створення навчальних ресурсів у середовищі Moodle на основі технології „Cloud computing” [Електронний ресурс] / Сергієнко Володимир Петрович, Войтович Ігор Станіславович // Інформаційні технології і засоби навчання. – 2011. – Том 24, № 4. – 8 с. – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/download/518/434>.
217. Сейтвелієва С. Н. Методика навчання хмарних технологій майбутніх інженерів-програмістів : дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання (інформатика) / Сейтвелієва Сусана Нуріївна ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2017. – 271 с.
218. Сидоренко Е. В. Методы математической обработки в психологии / Е. В. Сидоренко. – СПб. : Речь, 2003. – 350 с.
219. Сисоєва С. О. Інтерактивні технології навчання дорослих : навчально-методичний посібник для викладачів системи формальної, неформальної та інформальної освіти дорослих / Світлана Сисоєва ; Київський ун-т імені

- Бориса Грінченка. – К. : ЕКМО, 2011. – 324 с.
220. Системний підхід у сучасних педагогічних дослідженнях в Україні : монографія / за ред. С. Я. Харченка ; Луган. нац. ун-т імені Тараса Шевченка. – Старобільськ : Вид-во ДЗ «ЛНУ імені Тараса Шевченка», 2016. – 495 с.
221. Скриннік Н. В. Методика навчання української літератури учнів 5-6 класів з використанням хмарних технологій : дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання (українська література) / Скриннік Наталія Володимирівна ; Нац. пед. ун-т ім. М. П. Драгоманова. – К., 2017. – 324 с.
222. Словак К. І. Методика використання мобільних математичних середовищ у процесі навчання вищої математики студентів економічних спеціальностей : автореф. дис ... канд. пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Словак Катерина Іванівна ; ІТЗН. – К., 2011. – 21 с.
223. Соколова Л. Е. Досвід використання технології «хмарних обчислень» в мережевих продуктах для шкільної освіти [Електронний ресурс] / Л. Е. Соколова, В. И. Олевський, Ю. Б. Олевська // Інформаційні технології в освіті. – 2011. – Вип. 9. – С. 82-89. – Режим доступу : [http://ite.ksu.ks.ua/webfm\\_send/207](http://ite.ksu.ks.ua/webfm_send/207).
224. Специальности, специализации, а также присваиваемые квалификации в ГГУ [Электронный ресурс] // Учреждение образования «Гомельский государственный университет имени Франциска Скорины» / TamBY.info : Информационно-справочный портал. – 2012. – Режим доступа : <https://web.archive.org/web/20121004014604/http://www.gsu.by/abiturient/doc/spec12.doc>.
225. Співаковський О. В. Побудова ІКТ інфраструктури ВНЗ: проблеми та шляхи вирішення [Електронний ресурс] / Співаковський Олександр Володимирович, Вінник Максим Олександрович, Тарасіч Юлія Геннадіївна // Інформаційні технології і засоби навчання. – 2014. – Том 39, № 1. – С. 99-116. – Режим доступу : <https://journal.iitta.gov.ua/index.php/itlt/article/download/996/752>.
226. Стрюк А. М. Використання хмарних обчислень у комбінованому навчанні

- системного програмування / А. М. Стрюк, О. М. Туравініна // Тези доп. Міжн. наук.-практ. конф. «Інформаційні технології в освіті, науці і техніці» (ІТОНТ–2012) : Черкаси, 25-27 квітня 2012 р. – У 2 т. – Черкаси : ЧДТУ, 2012. – Т. 2. – С. 96-97.
227. Стрюк А. М. Використання хмарних технологій у комбінованому навчанні інформатики студентів інженерних спеціальностей / А. М. Стрюк, М. В. Рассовицька // Вісн. Дніпропетр. ун-ту імені Альфреда Нобеля. Серія: Педагогіка і психологія. – 2015. – № 1 (9). – С. 221-226.
228. Стрюк А. М. Система «Агапа» як засіб навчання системного програмування бакалаврів програмної інженерії : дис. ... канд. пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Стрюк Андрій Миколайович ; Ін-т інф. технол. і зас. навч. – К., 2012. – 312 с.
229. Ступина М. В. Формирование компетентности студентов в области использования инструментальных средств разработки информационных систем с применением облачных технологий (на примере подготовки будущих бакалавров-разработчиков информационных систем) : дисс. ... канд. пед. наук : 13.00.08 – теория и методика профессионального образования / Ступина Мария Валерьевна ; Ин-т упр. образованием. – М., 2018. – 197 с.
230. Субетто А. И. Проблемы фундаментализации и источников формирования содержания высшего образования: грани государственной политики / Анатолий Иванович Субетто. – Кострома : Изд-во КГПУ, 1995. – 322 с.
231. Сурыгин А. И. Концепция фундаментализации политехнического университетского образования / Сурыгин А. И. // Высокие интеллектуальные технологии образования и науки : матер. X Межд. науч.-метод. конф. / Санкт-Петербургский гос. техн. ун-т. – СПб. : Изд-во СПбГПУ, 2003. – С. 48-52.
232. Суханов А. Д. Концепция фундаментализации высшего образования и ее отражение в ГОСах / А. Д. Суханов // Высшее образование в России. – 1996. – № 3. – С. 36-44.
233. Сучасні технології дистанційного навчання / [Кухаренко Володимир Миколайович] ; Дистанційні курси НТУ «ХП». – Харків. – Режим

- доступу : <http://dl.kharkiv.edu/course/view.php?id=7>.
234. Теплицький І. О. Середовища моделювання: від заміни до інтеграції / І. О. Теплицький, О. І. Теплицький, А. П. Гуменюк // Новітні комп'ютерні технології. – 2008. – Том VI. – С. 67-68.
235. Технології хмарних обчислень – провідні інформаційні технології подальшого розвитку інформатизації системи освіти в Україні (Відповіді доктора технічних наук, професора, академіка НАПН України, лауреата Державної премії, заслуженого діяча науки і техніки України, директора Інституту інформаційних технологій і засобів навчання НАПН України Валерія Юхимовича Бикова на запитання головного редактора науково-методичного журналу «Комп'ютер у школі та сім'ї» В. Д. Руденка) // Комп'ютер у школі та сім'ї. – 2011. – № 6(94). – С. 3-11.
236. Триус Ю. В. Інноваційні технології навчання у вищій школі [Електронний ресурс] / Триус Ю. В. ; Черкаський держ. технол. ун-т // X Міжвузівська школа-семінар «Сучасні педагогічні технології в освіті». – Харків, 31.01-02.02.2012. – 52 с. – Режим доступу : <http://www.slideshare.net/kvntkf/tryus-innovacai-iktvnz>.
237. Триус Ю. В. Комп'ютерно-орієнтовані методичні системи навчання математичних дисциплін у вищих навчальних закладах : дис. ... д-ра пед. наук : 13.00.02 – теорія та методика навчання інформатики / Триус Юрій Васильович ; Черкас. нац. ун-т ім. Богдана Хмельницького. – Черкаси, 2005. – 515 с.
238. Триус Ю. В. Хмарні технології у професійній підготовці студентів комп'ютерних спеціальностей / Ю. В. Триус // Хмарні технології в освіті : матер. Всеукр. наук.-метод. Інтернет-семінару (Кривий Ріг – Київ – Черкаси – Харків, 21 грудня 2012 р.). – Кривий Ріг : Видавничий відділ КМІ, 2012. – С. 147-149.
239. Туравініна О. М. Amazon EC2 як платформа для організації хмарних обчислень / О. М. Туравініна, А. М. Стрюк, Н. В. Рашевська, К. І. Словак // Новітні комп'ютерні технології : матер. IX Міжн. наук.-техн. конф. : Київ-Севастополь, 13-16 вересня 2011 р. – К. : Мінрегіон України, 2011. – С. 187-189.

240. Туравініна О. М. До питання про розробку методики навчання математичної інформатики у технічному ВНЗ / О. М. Туравініна, І. О. Теплицький, І. І. Ліннік // Новітні комп'ютерні технології : матер. VIII Міжн. наук.-техн. конф. : Київ-Севастополь, 14-17 вересня 2010 р. – К. : Мінрегіон України, 2010. – С. 168-169.
241. Туравініна О. М. Застосування мобільних засобів навчання математичної інформатики в курсі фізики / О. М. Туравініна // Тези доп. VII Всеукр. наук.-практ. конф. «Інформаційні технології в освіті, науці і техніці» (ІТОНТ–2010) : Черкаси, 4-6 травня 2010 р. – У 2-х томах. – Черкаси : ЧДТУ, 2010. – Т. 2. – С. 74.
242. Туравініна О. М. Зміст навчання основ математичної інформатики студентів технічних університетів / О. М. Туравініна, С. О. Семеріков // Матер. міжн. наук.-метод. конф. «Розвиток інтелектуальних умінь і творчих здібностей учнів та студентів у процесі навчання дисциплін природничо-математичного циклу» (ІТМ\*плюс – 2012), м. Суми, 6-7 грудня 2012 р. – Суми : Вид-во СумДПУ ім. А. С. Макаренка, 2012. – С. 142-145.
243. Туравініна О. М. Математична інформатика у системі фундаменталізації навчання студентів технічних університетів / О. М. Туравініна // Інновації в навчанні фізики: навчальний та міжнародний досвід : зб. наук. пр. Вип. 18 : серія педагогічна. – Кам'янець-Подільський : Кам'янець-Подільський національний університет ім. І. Огієнка, 2012. – С. 189-191.
244. Туравініна О. М. Хмарні технології навчання студентів / О. М. Туравініна // Новітні комп'ютерні технології : матер. X Міжн. наук.-техн. конф. : Севастополь, 11-14 вересня 2012 р. – К. : Мінрегіон України, 2012. – С. 119-121.
245. Туравініна О. М. Хмарні технології навчання у системі інформаційно-комунікаційних технологій навчального призначення / О. М. Туравініна // Хмарні технології в освіті : матер. Всеукр. наук.-метод. Інтернет-семінару (Кривий Ріг – Київ – Черкаси – Харків, 21 грудня 2012 р.). – Кривий Ріг : Видавничий відділ КМІ, 2012. – С. 9.
246. Уоссермен Ф. Нейрокомпьютерная техника : Теория и практика /

- Ф. Уоссермен ; перевод с англ. Ю. А. Зуева, В. А. Точенова ; под ред. А. И. Галушкина. – М. : Мир, 1992. – 236, [1] с. : ил.
247. Федосин М. С. Виртуализация многокомпонентной системной архитектуры предметно-ориентированной облачной вычислительной среды : автореф. дисс. ... канд. техн. наук : 05.13.15 – вычислительные машины, комплексы и компьютерные сети / Федосин Михаил Евгеньевич ; [Пензенский гос. ун-т]. – Пенза, 2014. – 23 с.
248. Формус Ю. В. Sagemaths як хмарний засіб реалізації основних чисельних методів / Формус Ю. В., Туравініна О. М. // Комп'ютерні інтелектуальні системи та мережі : матер. VII Всеукр. наук.-практ. WEB конф. асп., студ. та молод. вчен. (25-27 березня 2014 р.) / Криворіз. нац. ун-т, Каф. комп. систем та мереж. – Кривий Ріг : Вид-во Криворіз. нац. ун-ту, 2014. – С. 119-120.
249. Халимова Б. А. Хмарні обчислення і технології: тенденції розвитку / Халимова Б. А., Маркова О. М. // Комп'ютерні інтелектуальні системи та мережі : матер. IX Всеукр. наук.-практ. WEB конф. асп., студ. та молод. вчен. (22-24 березня 2016 р.) / Криворіз. нац. ун-т, Каф. комп. систем та мереж. – Кривий Ріг : Видавн. центр ДВНЗ «Криворізький національний університет», 2016. – С. 25-26.
250. Хомутенко М. В. Методика навчання атомної і ядерної фізики старшокласників у хмаро зорієнтованому навчальному середовищі : дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання (фізика) / Хомутенко Максим Володимирович ; Центральноукраїнський держ. пед. ун-т ім. В. Вінниченка. – Кропивницький, 2018. – 397 с.
251. Цапко Е. А. Концепция фундаментализации и ее статус в парадигме образовательного феномена технического университета : автореф. дисс. ... канд. филос. наук : 09.00.11 – социальная философия / Цапко Елена Александровна ; [Томский политех. ун-т]. – Томск, 1998. – 27 с.
252. Цвілик С. Д. Наступність у змісті природничо-математичної та спеціальної підготовки вчителя трудового навчання у вищих педагогічних закладах освіти : автореф. дис. ... канд. пед. наук : 13.00.04 – теорія і методика професійної освіти / Цвілик Світлана Дмитрівна ; Вінницький

- держ. пед. ун-т ім. М. Коцюбинського. – Вінниця, 2005. – 25 с.
253. Чернилевский Д. В. Дидактические технологии в высшей школе : учеб. пособие для студентов вузов, обучающихся по пед. специальностям, магистрантов, аспирантов и слушателей системы доп. проф. образования / Д. В. Чернилевский. – М. : ЮНИТИ, 2002. – 436, [1] с.
254. Чернилевский Д. В. Инновационные технологии и дидактические средства современного профессионального образования : моногр. / Д. В. Чернилевский, В. Б. Моисеев ; Моск. гос. индустр. ун-т. – М., 2002. – 145 с.
255. Черных Л. А. Теоретические основы разработки методической системы обучения / Черных Л. А. // Евристика та дидактика точних наук : збірник наукових робіт. – Вип. 3. – Донецьк : Донецька школа евристики та точних наук, 1995. – С. 15-19.
256. Читалин Н. А. Многоуровневая фундаментализация содержания профессионального образования : дисс. ... д-ра пед. наук : 13.00.01 – общая педагогика / Читалин Николай Александрович ; Ин-т педагогики и психологии проф. образования. – Казань, 2006. – 362 с.
257. Шалкина Т. Н. Информационно-предметная среда как фактор подготовки будущих инженеров-программистов : дисс. ... канд. пед. наук : 13.00.08 – теория и методика профессионального образования / Шалкина Татьяна Николаевна ; Оренбургский гос. ун-т. – Оренбург, 2003. – 190 с.
258. Шваб К. Четвертая промышленная революция / Клаус Шваб. – М. : Эксмо, 2016. – 208 с.
259. Шевченко В. Г. Облачные технологии как средство формирования ИКТ-компетентности будущих учителей информатики : дисс. ... канд. пед. наук : 13.00.02 – теория и методика обучения и воспитания (информатика) / Шевченко Виктория Геннадьевна ; Ин-т стратегии развития образования. – М., 2016. – 263 с.
260. Шиненко М. А. Перспективи розвитку програмного забезпечення як послуги для створення документів електронної бібліотеки на прикладі Microsoft Office 365 [Електронний ресурс] / Шиненко Микола Андрійович, Сороко Наталія Володимирівна // Інформаційні технології і



- засоби навчання. – 2011. – Том 26, №6. – 17 с. – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/download/572/455>.
261. Шишкіна М. П. Проблеми інформатизації освіти України в контексті розвитку досліджень оцінювання якості засобів ІКТ [Електронний ресурс] / Шишкіна Марія Павлівна, Спирін Олег Михайлович, Запорожченко Юлія Григорівна // Інформаційні технології і засоби навчання. – 2012. – Том 27, №1. – 17 с. – Режим доступу : <https://goo.gl/ZkoZP2>.
262. Шишкіна М. П. Теоретико-методичні засади формування і розвитку хмаро зорієнтованого освітньо-наукового середовища вищого навчального закладу : дис. ... д-ра пед. наук : 13.00.10 – інформаційно-комунікаційні технології в освіті / Шишкіна Марія Павлівна ; Ін-т інформ. технологій і засобів навчання. – К., 2016. – 441 с.
263. Шишкіна М. П. Формування і розвиток хмаро зорієнтованого освітньо-наукового середовища вищого навчального закладу : моногр. / М. П. Шишкіна ; Ін-т інформ. технологій і засобів навчання. – Київ : УкрІНТЕІ, 2015. – 256 с. : рис.
264. Шокалюк С. В. SageMathCloud як засіб хмарних технологій комп'ютерно-орієнтованого навчання математичних та інформатичних дисциплін / Шокалюк С. В., Маркова О. М., Семеріков С. О. // Моделювання в освіті: Стан. Проблеми. Перспективи : моногр. / за заг. ред. Соловійова В. М. – Черкаси : Брама, видавець Вовчок О.Ю., 2017. – С. 130-142.
265. Шокалюк С. В. Інформаційні технології математичного призначення в курсі фізики середньої та вищої школи / С. В. Шокалюк, С. О. Семеріков // Зб. наук. пр. Кам'янець-Подільського нац. ун-ту : Серія педагогічна / [редкол. : П. С. Атаманчук (голова, наук. ред.) та ін.]. – Кам'янець-Подільський : Кам'янець-Подільський нац. ун-т, 2008. – Вип. 14 : Інновації в навчанні фізики та дисциплін технологічної освітньої галузі : міжнародний та вітчизняний досвід. – С. 108-113.
266. Шокалюк С. В. Методичні засади комп'ютеризації самостійної роботи старшокласників у процесі вивчення програмного забезпечення математичного призначення : автореф. дис. ... канд. пед. наук : 13.00.02 – теорія і методика навчання (інформатика) / Шокалюк Світлана

- Вікторівна ; Нац. пед. ун-т імені М. П. Драгоманова. – К., 2010. – 21 с.
267. Щодо формування пропозицій обсягу прийому та випуску фахівців з вищою освітою на 2011 рік : Лист МОН №1/9-3 від 06.01.11 р. [Електронний ресурс] / Міністерство освіти і науки України. – Режим доступу : [http://osvita.ua/legislation/Vishya\\_osvita/12928](http://osvita.ua/legislation/Vishya_osvita/12928).
268. Ягупов В. В. Педагогіка : навч. посібник / В. В. Ягупов. – К. : Либідь, 2002. – 560 с.
269. Яцишин А. В. Актуальність підготовки наукових і науково-педагогічних кадрів вищої кваліфікації для інформатизації загальної середньої освіти України / Анна Яцишин // Наукові записки КДПУ ім. В. Винниченка. Серія: Проблеми методики фізико-математичної і технологічної освіти. – Кіровоград : РВВ КДПУ ім. В. Винниченка, 2015. – Випуск 8 (II). – С. 70-78.
270. Яшанов С. М. Системи віртуалізації навчання у фаховій підготовці майбутнього вчителя технологій / С. Яшанов // Наукові записки КДПУ ім. В. Винниченка. Сер. : Педагогічні науки. – 2010. – Вип. 90. – С. 340-344.

## **ДОДАТКИ**

## Додаток А

## Статистичні відомості про підготовку фахівців з інформаційних технологій

Таблиця А.1

Ліцензійний обсяг, державне замовлення, кількість зарахованих на 1 курс абітурієнтів за денною та заочною формами навчання за напрямом підготовки 6.050102 «Комп'ютерна інженерія» (за даними [127])

| №  | ВНЗ                                                                                              | Денна форма |     |     |     | Заочна форма |    |    |    |
|----|--------------------------------------------------------------------------------------------------|-------------|-----|-----|-----|--------------|----|----|----|
|    |                                                                                                  | Л           | Д   | З   | Δ   | Л            | Д  | З  | Δ  |
| 1  | Одеський національний університет імені І. І. Мечникова                                          | 245         | 34  | 13  | -21 | 220          | 0  | 1  | 1  |
| 2  | Відкритий міжнародний університет розвитку людини «Україна»                                      | 100         | 0   | 1   | 1   | 100          | 0  | 0  | 0  |
| 3  | Національний технічний університет «Харківський політехнічний інститут»                          | 330         | 69  | 64  | -5  | 315          | 15 | 16 | 1  |
| 4  | Харківський національний університет радіоелектроніки                                            | 300         | 99  | 162 | 63  | 300          | 8  | 0  | -8 |
| 5  | Ужгородський національний університет                                                            | 30          | 21  | 21  | 0   | 0            | 0  | 0  | 0  |
| 6  | Національний університет кораблебудування імені адмірала Макарова                                | 30          | 12  | 10  | -2  | 30           | 0  | 0  | 0  |
| 7  | Луганський національний університет імені Тараса Шевченка                                        | 30          | 10  | 11  | 1   | 0            | 0  | 0  | 0  |
| 8  | Галицька Академія, ПВНЗ                                                                          | 30          | 0   | 3   | 3   | 30           | 0  | 1  | 1  |
| 9  | Таврійський національний університет імені В. І. Вернадського                                    | 50          | 20  | 20  | 0   | 50           | 0  | 0  | 0  |
| 10 | Херсонський національний технічний університет                                                   | 95          | 20  | 24  | 4   | 140          | 10 | 23 | 13 |
| 11 | Черкаський державний технологічний університет                                                   | 100         | 30  | 30  | 0   | 100          | 7  | 8  | 1  |
| 12 | Чернівецький національний університет імені Юрія Федьковича                                      | 100         | 24  | 27  | 3   | 90           | 4  | 11 | 7  |
| 13 | Національний університет «Львівська політехніка»                                                 | 285         | 125 | 125 | 0   | 225          | 0  | 0  | 0  |
| 14 | Державний університет інформаційно-комунікаційних технологій                                     | 50          | 22  | 22  | 0   | 50           | 0  | 6  | 6  |
| 15 | Луцький інститут розвитку людини університету «Україна»                                          | 25          | 0   | 1   | 1   | 25           | 0  | 0  | 0  |
| 16 | Запорізький інститут економіки та інформаційних технологій                                       | 50          | 0   | 0   | 0   | 50           | 0  | 0  | 0  |
| 17 | Полтавський національний технічний університет імені Юрія Кондратюка                             | 50          | 20  | 32  | 12  | 25           | 0  | 12 | 12 |
| 18 | Криворізький технічний університет                                                               | 50          | 15  | 19  | 4   | 50           | 5  | 7  | 2  |
| 19 | Дніпропетровський національний університет імені Олеся Гончара                                   | 60          | 15  | 15  | 0   | 0            | 0  | 0  | 0  |
| 20 | Східноукраїнський національний університет імені Володимира Даля                                 | 145         | 22  | 22  | 0   | 115          | 3  | 8  | 5  |
| 21 | Горлівський регіональний інститут Відкритого міжнародного університету розвитку людини «Україна» | 25          | 0   | 0   | 0   | 25           | 0  | 0  | 0  |
| 22 | Севастопольський національний університет ядерної енергії та промисловості                       | 60          | 23  | 19  | -4  | 60           | 0  | 5  | 5  |

## Продовження таблиці А.1

| №  | ВНЗ                                                                                                         | Денна форма |     |    |     | Заочна форма |    |    |     |
|----|-------------------------------------------------------------------------------------------------------------|-------------|-----|----|-----|--------------|----|----|-----|
|    |                                                                                                             | Л           | Д   | З  | Δ   | Л            | Д  | З  | Δ   |
| 23 | Прикарпатський національний університет імені Василя Стефаника                                              | 25          | 15  | 15 | 0   | 0            | 0  | 0  | 0   |
| 24 | Донецька академія автомобільного транспорту                                                                 | 60          | 0   | 0  | 0   | 60           | 0  | 0  | 0   |
| 25 | Новокаховський гуманітарний інститут Відкритого міжнародного університету розвитку людини «Україна»         | 30          | 0   | 1  | 1   | 0            | 0  | 0  | 0   |
| 26 | Донецький національний технічний університет                                                                | 225         | 125 | 92 | -33 | 205          | 0  | 4  | 4   |
| 27 | Національний гірничий університет                                                                           | 50          | 15  | 5  | -10 | 50           | 0  | 0  | 0   |
| 28 | Київський національний університет технологій та дизайну                                                    | 0           | 0   | 0  | 0   | 0            | 0  | 0  | 0   |
| 29 | Міжнародний гуманітарний університет                                                                        | 25          | 0   | 0  | 0   | 0            | 0  | 0  | 0   |
| 30 | Запорізький національний технічний університет                                                              | 180         | 35  | 0  | -35 | 180          | 0  | 0  | 0   |
| 31 | Хмельницький національний університет                                                                       | 110         | 50  | 50 | 0   | 65           | 1  | 1  | 0   |
| 32 | Донбаський державний технічний університет                                                                  | 50          | 15  | 18 | 3   | 50           | 2  | 8  | 6   |
| 33 | Кіровоградський національний технічний університет                                                          | 75          | 35  | 26 | -9  | 75           | 3  | 4  | 1   |
| 34 | Тернопільський національний технічний університет імені І. Пулюя                                            | 110         | 49  | 49 | 0   | 110          | 7  | 7  | 0   |
| 35 | Севастопольський національний технічний університет                                                         | 100         | 29  | 34 | 5   | 75           | 0  | 11 | 11  |
| 36 | Київський національний університет імені Тараса Шевченка                                                    | 50          | 25  | 0  | -25 | 0            | 0  | 0  | 0   |
| 37 | Тернопільський національний економічний університет                                                         | 110         | 49  | 49 | 0   | 110          | 7  | 7  | 0   |
| 38 | Луцький національний технічний університет                                                                  | 50          | 20  | 19 | -1  | 50           | 5  | 8  | 3   |
| 39 | Українська інженерно-педагогічна академія                                                                   | 50          | 0   | 0  | 0   | 50           | 0  | 0  | 0   |
| 40 | Одеський національний політехнічний університет                                                             | 185         | 43  | 0  | -43 | 155          | 10 | 0  | -10 |
| 41 | ДВНЗ «Український державний хіміко-технологічний університет»                                               | 30          | 15  | 15 | 0   | 30           | 2  | 3  | 1   |
| 42 | Національний аерокосмічний університет імені М. С. Жуковського «Харківський авіаційний інститут»            | 175         | 30  | 32 | 2   | 180          | 5  | 7  | 2   |
| 43 | Одеська державна академія холоду                                                                            | 75          | 35  | 30 | -5  | 25           | 0  | 9  | 9   |
| 44 | Дубенська філія Відкритого міжнародного університету розвитку людини «Україна»                              | 50          | 0   | 0  | 0   | 0            | 0  | 0  | 0   |
| 45 | Первомайський інститут Одеського національного університету імені І. І. Мечникова                           | 30          | 0   | 0  | 0   | 30           | 0  | 0  | 0   |
| 46 | Чорноморський державний університет імені Петра Могили                                                      | 60          | 25  | 25 | 0   | 0            | 0  | 0  | 0   |
| 47 | Дніпропетровський національний університет залізничного транспорту імені академіка В.Лазаряна               | 50          | 20  | 23 | 3   | 50           | 0  | 0  | 0   |
| 48 | Чернігівський державний технологічний університет                                                           | 150         | 40  | 48 | 8   | 0            | 0  | 0  | 0   |
| 49 | Кременчуцький національний університет імені Михайла Остроградського                                        | 55          | 15  | 11 | -4  | 55           | 0  | 0  | 0   |
| 50 | Технологічний інститут Східноукраїнського національного університету ім. Володимира Даля (м. Северодонецьк) | 155         | 19  | 19 | 0   | 160          | 17 | 17 | 0   |

## Продовження таблиці А.1

| №  | ВНЗ                                                                                                                                | Денна форма |      |      |     | Заочна форма |     |     |     |
|----|------------------------------------------------------------------------------------------------------------------------------------|-------------|------|------|-----|--------------|-----|-----|-----|
|    |                                                                                                                                    | Л           | Д    | З    | Δ   | Л            | Д   | З   | Δ   |
| 51 | Національний технічний університет України «Київський політехнічний інститут»                                                      | 420         | 113  | 241  | 128 | 160          | 9   | 18  | 9   |
| 52 | Вінницький національний технічний університет                                                                                      | 60          | 26   | 19   | -7  | 60           | 3   | 2   | -1  |
| 53 | Миколаївський політехнічний інститут                                                                                               | 25          | 0    | 4    | 4   | 25           | 0   | 0   | 0   |
| 54 | Миколаївський національний університет імені В. О. Сухомлинського                                                                  | 50          | 35   | 15   | -20 | 50           | 0   | 0   | 0   |
| 55 | Івано-Франківський національний технічний університет нафти і газу                                                                 | 30          | 15   | 16   | 1   | 20           | 0   | 0   | 0   |
| 56 | Національний авіаційний університет                                                                                                | 275         | 75   | 54   | -21 | 200          | 10  | 0   | -10 |
| 57 | Іллічівський інститут Одеського національного університету імені І. І. Мечникова                                                   | 0           | 0    | 0    | 0   | 30           | 0   | 3   | 3   |
| 58 | Миколаївський міжрегіональний інститут розвитку людини Відкритого міжнародного університету розвитку людини «Україна»              | 30          | 0    | 3    | 3   | 30           | 0   | 0   | 0   |
| 59 | Черкаський факультет ринкових інформаційних та інноваційних технологій Київського національного університету технологій та дизайну | 60          | 5    | 3    | -2  | 60           | 0   | 0   | 0   |
| 60 | Українська державна академія залізничного транспорту                                                                               | 50          | 0    | 0    | 0   | 50           | 0   | 0   | 0   |
|    | Разом                                                                                                                              | 5555        | 1554 | 1557 | 3   | 4450         | 133 | 207 | 74  |

*Позначення:* Л – ліцензійний обсяг; Д – обсяг державного замовлення; З – кількість зарахованих абітурієнтів; Δ – різниця кількості зарахованих абітурієнтів та обсягу державного замовлення.

## Додаток Б

## Робоча навчальна програма зі спецкурсу «Математична інформатика»

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ДВНЗ «КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»**  
**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  
**КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ**



"ЗАТВЕРДЖУЮ"

/В. І. Вербицький/  
20\_\_ року

**РОБОЧА ПРОГРАМА СПЕЦКУРСУ**  
**«ОСНОВИ МАТЕМАТИЧНОЇ ІНФОРМАТИКИ»**

спеціальність 123 «Комп'ютерна інженерія»

| Форма навчання | Нормативні дані |         |                 |      |              |               |                    |                  |                             |                                |                   |                 |
|----------------|-----------------|---------|-----------------|------|--------------|---------------|--------------------|------------------|-----------------------------|--------------------------------|-------------------|-----------------|
|                | Курс            | Семестр | Загальний обсяг | ESTS | Всього годин | у тому числі  |                    |                  | Самостійна робота<br>(год.) | Кількість змістових<br>модулів | Екзамен (семестр) | Залік (семестр) |
|                |                 |         |                 |      |              | Лекції (год.) | Лабораторні (год.) | Практичні (год.) |                             |                                |                   |                 |
| Денна          | 2               | 4       | 120             | 4    | 72           | 36            | 36                 | –                | 48                          | 4                              | –                 | 4               |
| Заочна         | 2               | 4       | 120             | 4    | 16           | 8             | 8                  | –                | 104                         | 4                              | –                 | 4               |

*Примітка.* Співвідношення кількості годин аудиторних занять до самостійної роботи становить (%): для денної форми навчання – 150 % (72/48), для заочної форми навчання – 15,4 % (16/104).

2015–2016 навчальний рік

Розробники робочої програми навчальної спецкурсу «Основи математичної інформатики»:

старший викладач кафедри комп'ютерних систем та мереж О. М. Маркова  
д. пед. н., проф. С. О. Семеріков

Програму обговорено та рекомендовано до затвердження на засіданні кафедри комп'ютерних систем та мереж «28» серпня 2015 р., протокол № 1.

Завідувач кафедри комп'ютерних систем  
та мереж  
проф., д. т. н.

А. І. Купін

Програму затверджено на засіданні науково-методичної ради інженерно-технічного напрямку ДВНЗ «Криворізький національний університет» «15» вересня 2015 р., протокол № 2.

Проректор з науково-педагогічної  
та навчальної роботи

В. І. Вербицький

Завідувач навчально-методичного відділу

Г. Х. Отверченко

© О. М. Маркова, 2015 р.

© С. О. Семеріков, 2015 р.



## 1. Опис навчальної дисципліни

| Найменування показників                                                                                                                                                                                                                                                 | Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень                                                                          | Характеристика навчальної дисципліни |                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|-----------------------|
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | денна форма навчання                 | заочна форма навчання |
| Кількість кредитів – 4<br>Модулів – 1<br>Змістових модулів – 4<br>Індивідуальне науково-дослідне завдання – розробка нейронної мережі<br>Загальна кількість годин – 120<br>Тижневих годин для денної форми навчання: аудиторних – 4; самостійної роботи студента – 2,66 | Галузь знань<br>12 «Інформаційні технології»<br>Спеціальність<br>123 «Комп'ютерна інженерія»<br>Освітньо-кваліфікаційний рівень: бакалавр | За вибором                           |                       |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | <b>Рік підготовки:</b>               |                       |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | 2-й                                  | 2-й                   |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | <b>Семестр</b>                       |                       |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | 4-й                                  | 4-й                   |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | <b>Лекції</b>                        |                       |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | 36 год.                              | 8 год.                |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | <b>Практичні, семінарські</b>        |                       |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | –                                    | –                     |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | <b>Лабораторні</b>                   |                       |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | 36 год.                              | 8 год.                |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | <b>Самостійна робота</b>             |                       |
|                                                                                                                                                                                                                                                                         |                                                                                                                                           | 48 год.                              | 104 год.              |
| <b>Індивідуальні завдання:</b>                                                                                                                                                                                                                                          |                                                                                                                                           |                                      |                       |
| Вид контролю: залік                                                                                                                                                                                                                                                     |                                                                                                                                           |                                      |                       |

## 2. Мета та завдання навчальної дисципліни

**Мета вивчення дисципліни** – сформувати у студентів компетентність з математичної інформатики як теоретичної основи інформаційних технологій.

### **Завдання навчальної дисципліни:**

- ознайомлення зі структурами даних і алгоритмами, які є фундаментом сучасної методології розробки програм;
- вивчення методів розв'язування інженерних та наукових задач з використанням чисельних методів;
- ознайомлення з основними принципами кодування і модуляції сигналів у процесі передавання даних, опрацювання сигналів, збільшення перешкодозахищеності при передаванні даних по каналах зв'язку;
- формування умінь описувати основні методи реєстрації сигналів, декодування і виявлення помилок за допомогою різних коригуючих кодів;
- вивчення основ алгоритмічних аспектів теорії чисел та їх застосування в сучасній криптографії;
- опанування хмарними технологіями для практичної реалізації основних методів математичної інформатики.

У результаті вивчення дисципліни студент повинен:

*знати:*

- методи аналізу алгоритмів;
- методи оцінки складності алгоритмів;
- методи розробки алгоритмів;
- основи теорії комп'ютерного моделювання;
- методи лінійної алгебри;
- методи розв'язання нелінійних рівнянь та їх систем;
- основи теорії наближення таблично заданих залежностей функціями;
- методи чисельного розв'язання звичайних диференціальних рівнянь;
- методи одновимірної та багатовимірної оптимізації;
- математичні основи нейронних мереж;
- основні відомості з теорії чисел;
- основні поняття теорії завадостійкого кодування;
- методи побудови лінійних кодів;
- циклічні коди та їх властивості;
- коди Боуза-Чоудхурі-Хоквінгема;
- коди Ріда-Соломона та згорткові коди;
- алгоритмічні принципи побудови криптографічних систем;
- алгоритми блочного шифрування за допомогою мережі Фейстеля;
- принципи побудови асиметричних криптосистем;
- алгоритми електронного цифрового підпису;
- методи управління криптографічними ключами;

*уміти:*

- аналізувати, оцінювати та обирати існуючі алгоритми;
- володіти методами та технологіями розробки та оцінювання алгоритмів;
- розробляти нові алгоритми, які пов'язані з проектуванням апаратних та програмних компонентів комп'ютерних систем і мереж;
- використовувати існуючі та розроблювати нові математичні методи для вирішення задач, пов'язаних з проектуванням та використанням комп'ютерних

систем і мереж;

- вибирати та обґрунтовувати методи обчислень, стійкі до похибок;
- розв’язувати лінійні алгебраїчні рівняння та їх системи;
- володіти методами розв’язання нелінійних рівнянь та їх систем;
- застосовувати методи інтерполяції та апроксимації;
- здійснювати вибір методу інтегрування диференціального рівняння;
- ставити та вирішувати оптимізаційні задачі;
- застосувати теорему Колмогорова для апроксимації довільної функції

тришаровою нейронною мережею;

- будувати кільця за заданим модулем;
- застосовувати методи завадостійкого кодування для відновлення даних

при їх пошкодженні;

- будувати лінійні коди;
- будувати генеруючий та перевірючий поліноми для кодування та

декодування циклічних кодів;

- будувати двійкові коди Боуза-Чоудхурі-Хоквінгема;
- застосовувати коди Ріда-Соломона для передавання даних у

комп’ютерних мережах;

– застосовувати методи і засоби забезпечення безпеки програм і даних при проектуванні і експлуатації комп’ютерних систем і мереж;

- враховувати вимоги до систем захисту даних;
- створювати програмні та апаратні підсистеми криптографічного захисту

даних;

- будувати блочні шифри;
- використовувати алгоритм RSA та системи цифрового підпису для

передавання даних у комп’ютерних мережах;

– вміти формувати і управляти ключовою інформацією для підсистем аутентифікації.

**Міждисциплінарні зв’язки.** Необхідною основою для навчання спецкурсу є опанування навчальних дисциплін: «Вища математика», «Теорія

електричних та магнітних кіл», «Комп'ютерна логіка».

Успішне засвоєння спецкурсу сприятиме опануванню навчальних дисциплін: «Алгоритми та методи обчислень», «Дискретна математика», «Комп'ютерна електроніка», «Архітектура комп'ютерів», «Комп'ютерна схемотехніка», «Системне програмування», «Технології проектування комп'ютерних систем», «Захист інформації у комп'ютерних системах», «Інженерія програмного забезпечення».

### **3. Програма навчальної дисципліни**

#### Змістовий модуль 1. Теорія алгоритмів

##### *Тема 1. Аналіз алгоритмів*

Сортування включенням. Машина з довільним доступом до пам'яті. Аналіз алгоритму сортування методом включення. Порядок зростання. Асимптотичні позначення. Порівняння функцій.

##### *Тема 2. Алгоритмічні стратегії*

Асимптотичний аналіз верхньої та середньої оцінок складності алгоритмів; порівняння найкращих, середніх і найгірших оцінок;  $O$ -,  $o$ -,  $\omega$ - та  $\theta$ -нотації; емпіричні вимірювання ефективності алгоритмів; накладні витрати алгоритмів за часом і пам'яттю; рекурентні співвідношення та аналіз рекурсивних алгоритмів. Порівняння алгоритмів, вплив структур даних і особливостей реалізації на ефективність алгоритмів: сортування файлів. Методи розробки алгоритмів: структурне програмування, рекурсія, обходи дерев, «поділяй і пануй», балансування, динамічне програмування, програмування з відходом назад, метод «гілок і меж», евристичні та наближені алгоритми.

##### *Тема 3. Побудова алгоритмів*

Значення сортувань при реалізації алгоритмів. Класифікація сортувань. Характеристики сортувань. Прості сортування як спосіб швидкої реалізації алгоритму. Приклади простих сортувань – метод простого включення, метод простого обміну (бульбашкове сортування), шейкерне сортування, сортування

вставками, сортування підрахунком, цифрове сортування. Переваги і недоліки простих сортувань. Складні сортування як спосіб створення ефективних алгоритмів. Приклади складних сортувань – сортування Шелла, сортування Хоара (швидке сортування), сортування злиттям. Переваги і недоліки складних сортувань. Порівняння простих та складних сортувань.

### Змістовий модуль 2. Методи обчислень

#### *Тема 4. Комп'ютерне моделювання*

Поняття моделі та моделювання. Властивості та класифікація моделей. Алгоритми і програмування. Особливості комп'ютерного моделювання. Особливості імітаційного статистичного моделювання.

#### *Тема 5. Задачі лінійної алгебри*

Розв'язання систем лінійних рівнянь. Визначення власних значень матриць.

#### *Тема 6. Задачі нелінійної алгебри*

Розв'язання нелінійних рівнянь. Розв'язання систем нелінійних рівнянь.

#### *Тема 7. Методи наближення функцій*

Інтерполяція. Інтерполяційні поліноми Лагранжа та Ньютона. Апроксимація. Метод найменших квадратів.

#### *Тема 8. Розв'язання диференціальних рівнянь*

Загальна характеристика методів наближеного розв'язання задачі Коші. Методи Ейлера, Рунге-Кутта, Адамса.

#### *Тема 9. Методи оптимізації*

Класифікація методів оптимізації. Метод випадкового пошуку. Метод дихотомії. Методи «золотого перетину» та метод Фібоначчі. Симплексний пошук.

#### *Тема 10. Нейронні мережі та розпізнавання образів*

Математична модель нейрона. Застосування теореми Колмогорова для апроксимації довільної функції тришаровою нейронною мережею.

### Змістовий модуль 3. Теорія кодування

#### *Тема 11. Математичні основи теорії кодування*

Основні відомості з теорії чисел.

*Тема 12.* Основні поняття теорії завадостійкого кодування

Кодування даних. Завадостійке кодування. Коди із заданою виправною здатністю. Систематичні коди.

*Тема 13.* Лінійні коди

Лінійні коди. Границі об'ємів кодів. Код Хемінга та його властивості. Способи побудови лінійних кодів. Теорема Глаголева.

*Тема 14.* Циклічні коди, їхні властивості

Визначення та властивості. Генеруючий поліном. Кодування циклічних кодів. Перевірочний поліном. Декодування циклічного коду. Мінімальний поліном та його властивості. Число циклічних кодів.

*Тема 15.* Коди Боуза-Чоудхурі-Хоквінгема

Нулі коду. Циклічне подання коду Хемінга. Визначник Вандермонда. Двійкові коди Боуза-Чоудхурі-Хоквінгема.

*Тема 16.* Коди Ріда-Соломона, згорткові коди.

Коди Ріда-Соломона. Згорткові коди. Інші розповсюджені завадостійкі коди. Застосування завадостійких кодів у сучасних телекомунікаційних системах.

Змістовий модуль 4. Основи криптографії

*Тема 17.* Криптографічні системи

Історія криптографії. Основні поняття та визначення. Вимоги до криптографічних систем. Короткі відомості про криптоаналіз.

*Тема 18.* Симетричні криптосистеми

Основні класи симетричних криптосистем. Блочні шифри. Алгоритми блочного шифрування. Режим застосування блочних шифрів. Поточкові шифри.

*Тема 19.* Асиметричні криптосистеми

Односторонні функції та функції-ловушки. Асиметричні системи шифрування.

*Тема 20.* Електронні цифрові підписи

Постановка задачі. Алгоритми електронного цифрового підпису. Функції

хешування.

### Тема 21. Управління криптографічними ключами

Звичайна система управління криптографічними ключами. Управління ключами, засноване на системах з відкритим ключем. Протокол обміну секретними ключами. Використання сертифікатів. Протоколи аутентифікації. Анонімний розподіл ключей.

## 4. Структура навчальної дисципліни

| Назви змістовних модулів                        | Кількість годин |              |           |           |            |              |              |          |          |            |      |      |
|-------------------------------------------------|-----------------|--------------|-----------|-----------|------------|--------------|--------------|----------|----------|------------|------|------|
|                                                 | денна форма     |              |           |           |            | заочна форма |              |          |          |            |      |      |
|                                                 | усього          | у тому числі |           |           |            | усього       | у тому числі |          |          |            |      |      |
|                                                 |                 | л.           | пр.       | лаб.      | інд.       |              | с.р.         | л.       | пр.      | лаб.       | інд. | с.р. |
| <b>Змістовий модуль 1. Теорія алгоритмів</b>    |                 |              |           |           |            |              |              |          |          |            |      |      |
| Аналіз алгоритмів                               |                 | 3,0          |           | 3,0       |            | 4,0          |              | 1,0      |          | 1,0        |      | 8,0  |
| Алгоритмічні стратегії                          |                 | 3,0          |           | 3,0       |            | 4,0          |              |          |          |            |      | 10,0 |
| Побудова алгоритмів                             |                 | 3,0          |           | 3,0       |            | 4,0          |              | 1,0      |          | 1,0        |      | 8,0  |
| <b>Змістовий модуль 2. Методи обчислень</b>     |                 |              |           |           |            |              |              |          |          |            |      |      |
| Комп'ютерне моделювання                         |                 | 1,0          |           |           |            | 2,0          |              |          |          |            |      | 4,0  |
| Задачі лінійної алгебри                         |                 | 1,5          |           | 2,0       |            | 1,0          |              | 1,0      |          |            |      | 3,0  |
| Задачі нелінійної алгебри                       |                 | 1,5          |           | 1,0       |            | 1,0          |              |          |          | 1,0        |      | 3,0  |
| Методи наближення функцій                       |                 | 1,0          |           | 2,0       |            | 1,0          |              |          |          |            |      | 4,0  |
| Розв'язання диференціальних рівнянь             |                 | 1,0          |           | 1,0       |            | 2,0          |              |          |          |            |      | 4,0  |
| Методи оптимізації                              |                 | 1,5          |           | 2,0       |            | 2,0          |              | 1,0      |          |            |      | 4,0  |
| Нейронні мережі та розпізнавання образів        |                 | 1,5          |           | 1,0       |            | 3,0          |              |          |          | 1,0        |      | 4,0  |
| <b>Змістовий модуль 3. Теорія кодування</b>     |                 |              |           |           |            |              |              |          |          |            |      |      |
| Математичні основи теорії кодування             |                 | 1,5          |           | 1,0       |            | 3,0          |              | 1,0      |          |            |      | 4,0  |
| Основні поняття теорії завадостійкого кодування |                 | 1,5          |           |           |            | 4,0          |              | 1,0      |          |            |      | 4,0  |
| Лінійні коди                                    |                 | 1,5          |           | 2,0       |            | 1,0          |              |          |          | 1,0        |      | 4,0  |
| Циклічні коди, їхні властивості                 |                 | 1,5          |           | 2,0       |            | 1,0          |              |          |          | 1,0        |      | 4,0  |
| Коди Боуза-Чоудхурі-Хоквінгема                  |                 | 1,5          |           | 2,0       |            | 1,0          |              |          |          |            |      | 5,0  |
| Коди Ріда-Соломона, згорткові коди              |                 | 1,5          |           | 2,0       |            | 2,0          |              |          |          |            |      | 5,0  |
| <b>Змістовий модуль 4. Основи криптографії</b>  |                 |              |           |           |            |              |              |          |          |            |      |      |
| Криптографічні системи                          |                 | 3,0          |           | 2,0       |            | 3,0          |              |          |          | 2,0        |      | 4,0  |
| Симетричні криптосистеми                        |                 | 2,0          |           | 2,0       |            | 2,0          |              | 1,0      |          |            |      | 5,0  |
| Асиметричні криптосистеми                       |                 | 2,0          |           | 3,0       |            | 2,0          |              | 1,0      |          |            |      | 5,0  |
| Електронні цифрові підписи                      |                 | 1,0          |           | 1,0       |            | 2,0          |              |          |          |            |      | 6,0  |
| Управління криптографічними ключами             |                 | 1,0          |           | 1,0       |            | 3,0          |              |          |          |            |      | 6,0  |
| <b>Усього годин</b>                             | <b>120</b>      | <b>36</b>    | <b>36</b> | <b>48</b> | <b>120</b> | <b>8</b>     | <b>8</b>     | <b>8</b> | <b>8</b> | <b>104</b> |      |      |

### 5. Теми лабораторних занять

| № з/п | Назва теми                                                              | Кількість годин денна/заочна |
|-------|-------------------------------------------------------------------------|------------------------------|
| 1     | Прості цілочисельні алгоритми                                           | 3,0/1,0                      |
| 2     | Комбінаторні та рекурсивні алгоритми                                    | 3,0/–                        |
| 3     | Алгоритми сортування                                                    | 3,0/1,0                      |
| 4     | Розв'язання систем лінійних рівнянь різними методами                    | 2,0/–                        |
| 5     | Методи розв'язання нелінійних рівнянь                                   | 1,0/1,0                      |
| 6     | Інтерполяція та апроксимація табличних функцій                          | 2,0/–                        |
| 7     | Наближене розв'язання звичайних диференціальних рівнянь першого порядку | 1,0/–                        |
| 8     | Чисельне знаходження локальних екстремумів функцій                      | 2,0/–                        |
| 9     | Апроксимація функцій тришаровою нейронною мережею                       | 1,0/1,0                      |
| 10    | Моделювання кільця цілих чисел за модулем $n$                           | 1,0/–                        |
| 11    | Коди Хемінга                                                            | 2,0/1,0                      |
| 12    | Моделювання циклічних кодів                                             | 2,0/1,0                      |
| 13    | Коди Боуза-Чоудхурі-Хоквінгема                                          | 2,0/–                        |
| 14    | Коди Ріда-Соломона                                                      | 2,0/–                        |
| 15    | Криптографічні системи у Sage                                           | 2,0/2,0                      |
| 16    | Моделювання блочних шифрів                                              | 2,0/–                        |
| 17    | Криптографія з відкритим ключем                                         | 3,0/–                        |
| 18    | Хешування ключей                                                        | 1,0/–                        |
| 19    | Алгоритм обміну ключами Діффі-Хеллмана.                                 | 1,0/–                        |
|       | <b>Разом</b>                                                            | <b>36/8</b>                  |

### 6. Самостійна робота

| № з/п | Назва теми                                                                                                                                                                                                                                                                                                                 | Кількість годин денна/заочна |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| 1     | <b>Аналіз алгоритмів.</b> Машина з довільним доступом до пам'яті. Асимптотичні позначення. Порівняння функцій.                                                                                                                                                                                                             | 4/8                          |
| 2     | <b>Алгоритмічні стратегії.</b> $O$ -, $o$ -, $\omega$ - та $\theta$ -нотації; емпіричні вимірювання ефективності алгоритмів; накладні витрати алгоритмів за часом і пам'яттю. Методи розробки алгоритмів: динамічне програмування, програмування з відходом назад, метод «гілок і меж», евристичні та наближені алгоритми. | 4/10                         |
| 3     | <b>Побудова алгоритмів.</b> Приклади простих сортувань – шейкерне сортування, сортування вставками, сортування підрахунком, цифрове сортування. Приклади складних сортувань – сортування Хоара (швидке сортування), сортування злиттям.                                                                                    | 4/8                          |
| 4     | <b>Комп'ютерне моделювання.</b> Алгоритми і програмування. Особливості імітаційного статистичного моделювання.                                                                                                                                                                                                             | 2/4                          |
| 5     | <b>Задачі лінійної алгебри.</b> Визначення власних значень матриць.                                                                                                                                                                                                                                                        | 1/3                          |
| 6     | <b>Задачі нелінійної алгебри.</b> Розв'язання систем нелінійних рівнянь.                                                                                                                                                                                                                                                   | 1/3                          |
| 7     | <b>Методи наближення функцій.</b> Інтерполяція. Інтерполяційні поліноми Лагранжа та Ньютона.                                                                                                                                                                                                                               | 1/4                          |
| 8     | <b>Розв'язання диференціальних рівнянь.</b> Методи Рунге-Кутта вищих порядків, метод Адамса.                                                                                                                                                                                                                               | 2/4                          |
| 9     | <b>Методи оптимізації.</b> Методи «золотого перетину» та метод Фібоначчі.                                                                                                                                                                                                                                                  | 2/4                          |



| № з/п | Назва теми                                                                                                                                                      | Кількість годин денна/заочна |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
|       | Симплексний пошук.                                                                                                                                              |                              |
| 10    | <b>Нейронні мережі та розпізнавання образів.</b> Застосування теореми Колмогорова для апроксимації довільної функції тришаровою нейронною мережею.              | 3/4                          |
| 11    | <b>Математичні основи теорії кодування.</b> Поняття про алгебраїчні структури, групи, поля. Основні теореми теорії чисел.                                       | 3/4                          |
| 12    | <b>Основні поняття теорії завадостійкого кодування.</b> Коди із заданою виправною здатністю. Систематичні коди.                                                 | 4/4                          |
| 13    | <b>Лінійні коди.</b> Теорема Глаголева.                                                                                                                         | 1/4                          |
| 14    | <b>Циклічні коди, їхні властивості.</b> Мінімальний поліном та його властивості. Число циклічних кодів.                                                         | 1/4                          |
| 15    | <b>Коди Боуза-Чоудхурі-Хоквінгема.</b> Двійкові коди Боуза-Чоудхурі-Хоквінгема.                                                                                 | 1/5                          |
| 16    | <b>Коди Ріда-Соломона, згорткові коди.</b> Застосування завадостійких кодів у сучасних телекомунікаційних системах.                                             | 2/5                          |
| 17    | <b>Криптографічні системи.</b> Основні методи криптоаналізу.                                                                                                    | 3/4                          |
| 18    | <b>Симетричні криптосистеми.</b> Режими застосування блочних шифрів. Поточкові шифри.                                                                           | 2/5                          |
| 19    | <b>Асиметричні криптосистеми.</b> Асиметричні системи шифрування. Алгоритм RSA.                                                                                 | 2/5                          |
| 20    | <b>Електронні цифрові підписи.</b> Алгоритми електронного цифрового підпису. Функції хешування.                                                                 | 2/6                          |
| 21    | <b>Управління криптографічними ключами.</b> Протокол обміну секретними ключами. Використання сертифікатів. Протоколи аутентифікації. Анонімний розподіл ключей. | 3/6                          |
|       | <b>Разом</b>                                                                                                                                                    | <b>48/104</b>                |

### 7. Індивідуальні завдання

Для студентів денної форми навчання передбачено індивідуальне навчально-дослідницьке завдання: розробка нейронної мережі.

Для студентів заочної форми навчання передбачено виконання контрольної роботи.

### 8. Методи навчання

Викладення навчальної дисципліни «Основи математичної інформатики» передбачає використання таких провідних методів навчання: репродуктивний метод, пояснювально-ілюстративний метод, метод проблемного подання навчального матеріалу, частково-пошуковий метод, дослідницький метод, методи самостійної роботи та роботи під керівництвом викладача, методи

стимулювання й мотивації навчання, обчислювальний експеримент і програмування, метод демонстраційних прикладів, метод проектів.

## 9. Методи контролю

**Поточний контроль** здійснюється під час проведення лабораторних занять, а також контрольних робіт і має за мету перевірку якості засвоєння матеріалу студентами та зарахування кредитних модулів навчальної дисципліни. Поточний контроль виконання лабораторних робіт здійснюється шляхом моніторингу процесу виконання роботи.

Захист лабораторних робіт здійснюється шляхом усного опитування з одночасною перевіркою наданих звітів. Оцінюються знання та вміння студентів за наступними критеріями:

7 балів одержує студент, який у встановлений термін правильно виконав завдання лабораторної роботи, склав звіт з її виконання, продемонстрував роботу в різних режимах і вичерпно відповів на всі запитання викладача.

6 балів одержує студент, який у встановлений термін правильно виконав завдання лабораторної роботи, склав звіт з її виконання, продемонстрував роботу в різних режимах, але під час захисту не дав правильні відповіді на деякі запитання викладача.

5 балів одержує студент, який у встановлений термін правильно виконав завдання лабораторної роботи, склав звіт з її виконання, продемонстрував роботу в різних режимах, вичерпно відповів на всі запитання викладача, але не вклався у встановлений навчальним графіком термін виконання та захисту.

4 бали одержує студент, який правильно виконав завдання лабораторної роботи, склав звіт з її виконання, продемонстрував роботу в різних режимах, але не вклався у встановлений навчальним графіком термін виконання та захисту і під час захисту не дав правильні відповіді на деякі запитання викладача.

3 бали одержує студент, який виконав завдання лабораторної роботи, продемонстрував роботу в різних режимах, але з помилками склав звіт з її

виконання, не вклався у встановлений навчальним графіком термін виконання та захисту, під час захисту не зміг правильно відповісти на запитання викладача.

2 бали одержує студент, який виконав не всі завдання лабораторної роботи, продемонстрував роботу в деяких режимах, з помилками склав звіт з її виконання, не вклався у встановлений навчальним графіком термін виконання та захисту, під час захисту не зміг правильно відповісти на запитання викладача.

1 бал одержує студент, який приступив до виконання лабораторної роботи, але не може продемонструвати роботу в різних режимах, не склав звіт з її виконання, не вклався у встановлений навчальним графіком термін виконання та захисту, під час захисту не зміг правильно відповісти на запитання викладача.

0 балів отримує студент, який не виконав лабораторні роботи до початку наступного семестру. Він повинен прослухати курс повторно.

**Модульний контроль** здійснюється наприкінці змістових модулів лектором у формі модульної контрольної роботи (тестування). До модульної контрольної роботи входять питання за кожною темою модуля. Відповіді на питання оцінюються за шкалою від 0 (неправильна відповідь) до 1 (правильна відповідь).

При оцінюванні змістових модулів урахується поточний контроль якості навчання.

Підсумкова оцінка з курсу формується наприкінці навчального семестру та складається з суми оцінок за кожний складовий елемент модуля.

Оцінку «зараховано» (А) 90–100 балів одержує студент, який у встановлений термін правильно виконав усі завдання і вичерпно відповів на всі запитання викладача.

Оцінку «зараховано» (В) 80–89 балів одержує студент, який у встановлений термін правильно виконав усі завдання, але під час захисту не дав правильні відповіді на деякі запитання викладача.

Оцінку «зараховано» (С) 71–79 балів одержує студент, який правильно виконав усі завдання, вичерпно відповів на всі запитання викладача, але не

вклався у встановлений навчальним графіком термін виконання та захисту.

Оцінку «зараховано» (D) 61–70 балів одержує студент, який правильно виконав усі завдання, але не вклався у встановлений навчальним графіком термін виконання та захисту і під час захисту не дав правильні відповіді на деякі запитання викладача.

Оцінку «зараховано» (E) 50–60 балів одержує студент, який не вклався у встановлений навчальним графіком термін виконання та захисту, під час захисту не зміг правильно відповісти на запитання викладача та оформив звіт із помилками.

Оцінку «незараховано» (FX) 30–49 балів отримує студент, який не вклався у встановлений навчальним графіком термін виконання та захисту, виконав завдання з грубими помилками і не зміг правильно відповісти на запитання викладача.

Оцінку «незараховано» (F) 0–29 бали отримує студент, який не виконав лабораторні роботи до початку наступного семестру. Він повинен прослухати курс повторно.

Засоби контролю: тестові завдання для виконання модульних контрольних робіт; завдання для виконання лабораторних робіт; орієнтовний перелік тем індивідуальних навчально-дослідницьких завдань.

### **10. Розподіл балів, які отримують студенти**

Підрахунок підсумкової кількості балів, на які заслуговує студент по завершенні вивчення дисципліни «Основи математичної інформатики», здійснюється на підставі визначення сумарної оцінки за поточну успішність при вивченні усіх тем змістових модулів (за результатами всіх оцінок, отриманих протягом вивчення дисципліни) та її поділу на 3 (для студентів денної форми навчання).

Максимальна кількість балів за усіма видами контролю складає:

| Назва теми                                                                                       | Кількість балів |            |
|--------------------------------------------------------------------------------------------------|-----------------|------------|
|                                                                                                  | денна           | заочна     |
| Лабораторна робота № 1 «Прості цілочисельні алгоритми»                                           | 7               | 7          |
| Лабораторна робота № 2 «Комбінаторні та рекурсивні алгоритми»                                    | 7               |            |
| Лабораторна робота № 3 «Алгоритми сортування»                                                    | 7               | 7          |
| Модульна контрольна робота № 1 «Теорія алгоритмів»                                               | 31              |            |
| Лабораторна робота № 4 «Розв'язання систем лінійних рівнянь різними методами»                    | 7               |            |
| Лабораторна робота № 5 «Методи розв'язання нелінійних рівнянь»                                   | 7               | 7          |
| Лабораторна робота № 6 «Інтерполяція та апроксимація табличних функцій»                          | 7               |            |
| Лабораторна робота № 7 «Наближене розв'язання звичайних диференціальних рівнянь першого порядку» | 7               |            |
| Лабораторна робота № 8 «Чисельне знаходження локальних екстремумів функцій»                      | 7               |            |
| Лабораторна робота № 9 «Апроксимація функцій тришаровою нейронною мережею»                       | 7               | 7          |
| Модульна контрольна робота № 2 «Методи обчислень»                                                | 31              |            |
| Лабораторна робота № 10 «Моделювання кільця цілих чисел за модулем $n$ »                         | 7               |            |
| Лабораторна робота № 11 «Коди Хемінга»                                                           | 7               | 7          |
| Лабораторна робота № 12 «Моделювання циклічних кодів»                                            | 7               | 7          |
| Лабораторна робота № 13 «Коди Боуза-Чоудхурі-Хоквінгема»                                         | 7               |            |
| Лабораторна робота № 14 «Коди Ріда-Соломона»                                                     | 7               |            |
| Модульна контрольна робота № 3 «Теорія кодування»                                                | 30              |            |
| Лабораторна робота № 15 «Криптографічні системи у Sage»                                          | 7               | 7          |
| Лабораторна робота № 16 «Моделювання блочних шифрів»                                             | 7               |            |
| Лабораторна робота № 17 «Криптографія з відкритим ключем»                                        | 7               |            |
| Лабораторна робота № 18 «Хешування ключей»                                                       | 7               |            |
| Лабораторна робота № 19 «Алгоритм обміну ключами Діффі-Хеллмана».                                | 7               |            |
| Модульна контрольна робота № 4 «Основи криптографії»                                             | 30              |            |
| Індивідуальне навчально-дослідне завдання «Розробка нейронної мережі»                            | 45              |            |
| Контрольна робота                                                                                |                 | 21         |
| Залік                                                                                            |                 | 30         |
| <b>Разом</b>                                                                                     | <b>300</b>      | <b>100</b> |

## 11. Рекомендована література

### Базова

1. Кормен Т. Алгоритмы: построение и анализ / Томас Кормен, Чарльз Лейзерсон, Рональд Ривест, Клиффорд Штайн. – М. : Вильямс, 2013. – 1324 с.

2. Шаповаленко В. А. Чисельні методи та моделювання на ЕОМ. Модуль 1. Чисельне обчислення функцій, характеристик матриць і розв'язування нелінійних рівнянь та систем рівнянь : навчальний посібник для студентів напрямів бакалаврської підготовки «Автоматизація та комп'ютерно-інтегровані технології», «Телекомунікації», «Мережі та системи поштового

зв'язку» / В. А. Шаповаленко, Л. М. Буката, О. Г. Трофименко ; Міністерство транспорту та зв'язку України, Державна адміністрація зв'язку, Одеська Національна Академія зв'язку ім. О. С. Попова. – Ч. 1. – Одеса : ВЦ ОНАЗ, 2010. – 88 с.

3. Баричев С. Г. Основы современной криптографии : учеб. пособие / С. Г. Баричев, В. В. Гончаров, Р. Е. Серов. – М.: Горячая линия – Телеком, 2011. – 176 с.

4. Сидельников В. М. Теория кодирования / В. М. Сидельников. – М. : ФИЗМАТЛИТ, 2008. – 324 с.

5. Чечкин А. В. Математическая информатика [Электронный ресурс] / А. В. Чечкин. – М. : Наука. Гл. ред. физ.-мат. лит., 1991. – 416 с. – Режим доступа : <http://docslide.ru/documents/-55720e3e497959fc0b8c7349.html>.

#### *Допоміжна*

6. Соловьева Ф. И. Введение в теорию кодирования [Электронный ресурс] : учеб. пособие для студентов высших учебных заведений, обучающихся по специальности «010100 Математика» / Ф. И. Соловьева ; Федеральное агентство по образованию, Новосибирский государственный университет, Механико-математический факультет. – Новосибирск, 2006. – 124 с. – Режим доступа : <http://tc.nsu.ru/uploads/codingtheory.pdf>

7. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 1 : навчальний посібник / [Кветний Р. Н., Богач І. В., Бойко О. Р., Софіна О. Ю., Шушура О. М.; за заг. ред. Р. Н. Кветного] ; Міністерство освіти і науки, молоді та спорту України, Вінницький національний технічний університет. – Вінниця: ВНТУ, 2012. – 193 с.

8. Ахо А. Структуры данных и алгоритмы / Альфред Ахо, Джон В. Хопкрофт, Джеффри Д. Ульман.–М.: Вильямс, 2003. – 384 с.

9. Kohel D. R. Cryptography [Electronic resource] / David R. Kohel. – July 11, 2008. – IV, 138 p. – Access mode : <http://www.sagemath.org/files/kohel-book-2008.pdf>

10. Stallings W. Cryptography and Network Security: Principles and Practice.

Sixth Edition / William Stallings. – Upper Saddle River : Pearson, 2014. – XX, 732 p.

11. Stein W. Elementary Number Theory: Primes, Congruences, and Secrets: A Computational Approach / William Stein. – New York : Springer Science+Business Media, 2009. – X, 167 p. – (Undergraduate Texts in Mathematics)

## **12. Хмаро зорієнтовані засоби навчання**

SageMathCloud [Electronic resource] / SageMath, Inc., 2016. – Access mode : <https://cloud.sagemath.com/>.

## Додаток В

### Доведення правильності алгоритму Евкліда у системі Аxiom

```
Library Coq.ZArith.Znumtheory
```

```
Require Import ZArith_base.
Require Import ZArithRing.
Require Import Zcomplements.
Require Import Zdiv.
Require Import Wf_nat.
```

```
(** For compatibility reasons, this Open Scope isn't local as it should *)
```

```
Open Scope Z_scope.
```

```
(** This file contains some notions of number theory upon Z numbers:
 - a divisibility predicate [Z.divide]
 - a gcd predicate [gcd]
 - Euclid algorithm [euclid]
 - a relatively prime predicate [rel_prime]
 - a prime predicate [prime]
 - properties of the efficient [Z.gcd] function
*)
```

```
Notation Zgcd := Z.gcd (compat "8.3").
Notation Zggcd := Z.ggcd (compat "8.3").
Notation Zggcd_gcd := Z.ggcd_gcd (compat "8.3").
Notation Zggcd_correct_divisors := Z.ggcd_correct_divisors (compat "8.3").
Notation Zgcd_divide_l := Z.gcd_divide_l (compat "8.3").
Notation Zgcd_divide_r := Z.gcd_divide_r (compat "8.3").
Notation Zgcd_greatest := Z.gcd_greatest (compat "8.3").
Notation Zgcd_nonneg := Z.gcd_nonneg (compat "8.3").
Notation Zggcd_opp := Z.ggcd_opp (compat "8.3").
```

```
(** The former specialized inductive predicate [Z.divide] is now
 a generic existential predicate. *)
```

```
Notation Zdivide := Z.divide (compat "8.3").
```

```
(** Its former constructor is now a pseudo-constructor. *)
```

```
Definition Zdivide_intro a b q (H:b=q*a) : Z.divide a b := ex_intro _ q H.
```

```
(** Results concerning divisibility*)
```

```
Notation Zdivide_refl := Z.divide_refl (compat "8.3").
Notation Zdivide_1_1 := Z.divide_1_1 (compat "8.3").
Notation Zdivide_0 := Z.divide_0_r (compat "8.3").
```



```

Notation Zmult_divide_compat_l := Z.mul_divide_mono_l (compat "8.3").
Notation Zmult_divide_compat_r := Z.mul_divide_mono_r (compat "8.3").
Notation Zdivide_plus_r := Z.divide_add_r (compat "8.3").
Notation Zdivide_minus_l := Z.divide_sub_r (compat "8.3").
Notation Zdivide_mult_l := Z.divide_mul_l (compat "8.3").
Notation Zdivide_mult_r := Z.divide_mul_r (compat "8.3").
Notation Zdivide_factor_r := Z.divide_factor_l (compat "8.3").
Notation Zdivide_factor_l := Z.divide_factor_r (compat "8.3").

Lemma Zdivide_opp_r a b : (a | b) -> (a | - b).
Proof. apply Z.divide_opp_r. Qed.

Lemma Zdivide_opp_r_rev a b : (a | - b) -> (a | b).
Proof. apply Z.divide_opp_r. Qed.

Lemma Zdivide_opp_l a b : (a | b) -> (- a | b).
Proof. apply Z.divide_opp_l. Qed.

Lemma Zdivide_opp_l_rev a b : (- a | b) -> (a | b).
Proof. apply Z.divide_opp_l. Qed.

Theorem Zdivide_Zabs_l a b : (Z.abs a | b) -> (a | b).
Proof. apply Z.divide_abs_l. Qed.

Theorem Zdivide_Zabs_inv_l a b : (a | b) -> (Z.abs a | b).
Proof. apply Z.divide_abs_l. Qed.

Hint Resolve Z.divide_refl Z.divide_1_l Z.divide_0_r: zarith.
Hint Resolve Z.mul_divide_mono_l Z.mul_divide_mono_r: zarith.
Hint Resolve Z.divide_add_r Zdivide_opp_r Zdivide_opp_r_rev Zdivide_opp_l
 Zdivide_opp_l_rev Z.divide_sub_r Z.divide_mul_l Z.divide_mul_r
 Z.divide_factor_l Z.divide_factor_r: zarith.

(** Auxiliary result. *)

Lemma Zmult_one x y : x >= 0 -> x * y = 1 -> x = 1.
Proof.
 Z.swap_greater. apply Z.eq_mul_1_nonneg.
Qed.

(** Only [1] and [-1] divide [1]. *)

Notation Zdivide_1 := Z.divide_1_r (compat "8.3").

(** If [a] divides [b] and [b] divides [a] then [a] is [b] or [-b]. *)

Notation Zdivide_antisym := Z.divide_antisym (compat "8.3").
Notation Zdivide_trans := Z.divide_trans (compat "8.3").

```

(\*\* If [a] divides [b] and [b<>0] then [|a| <= |b|]. \*)

Lemma Zdivide\_bounds a b : (a | b) -> b <> 0 -> Z.abs a <= Z.abs b.

Proof.

intros H Hb.

rewrite <- Z.divide\_abs\_l, <- Z.divide\_abs\_r in H.

apply Z.abs\_pos in Hb.

now apply Z.divide\_pos\_le.

Qed.

(\*\* [Z.divide] can be expressed using [Z.modulo]. \*)

Lemma Zmod\_divide : forall a b, b<>0 -> a mod b = 0 -> (b | a).

Proof.

apply Z.mod\_divide.

Qed.

Lemma Zdivide\_mod : forall a b, (b | a) -> a mod b = 0.

Proof.

intros a b (c,->); apply Z\_mod\_mult.

Qed.

(\*\* [Z.divide] is hence decidable \*)

Lemma Zdivide\_dec a b : {(a | b)} + {~ (a | b)}.

Proof.

destruct (Z.eq\_dec a 0) as [Ha|Ha].

destruct (Z.eq\_dec b 0) as [Hb|Hb].

left; subst; apply Z.divide\_0\_r.

right. subst. contradict Hb. now apply Z.divide\_0\_l.

destruct (Z.eq\_dec (b mod a) 0).

left. now apply Z.mod\_divide.

right. now rewrite <- Z.mod\_divide.

Defined.

Theorem Zdivide\_Zdiv\_eq a b : 0 < a -> (a | b) -> b = a \* (b / a).

Proof.

intros Ha H.

rewrite (Z.div\_mod b a) at 1; auto with zarith.

rewrite Zdivide\_mod; auto with zarith.

Qed.

Theorem Zdivide\_Zdiv\_eq\_2 a b c :

0 < a -> (a | b) -> (c \* b) / a = c \* (b / a).

Proof.

intros. apply Z.divide\_div\_mul\_exact; auto with zarith.

Qed.

Theorem Zdivide\_le: forall a b : Z,  
 $0 \leq a \rightarrow 0 < b \rightarrow (a \mid b) \rightarrow a \leq b.$

Proof.

intros. now apply Z.divide\_pos\_le.

Qed.

Theorem Zdivide\_Zdiv\_lt\_pos a b :  
 $1 < a \rightarrow 0 < b \rightarrow (a \mid b) \rightarrow 0 < b / a < b .$

Proof.

intros H1 H2 H3; split.

apply Z.mul\_pos\_cancel\_l with a; auto with zarith.

rewrite <- Zdivide\_Zdiv\_eq; auto with zarith.

now apply Z.div\_lt.

Qed.

Lemma Zmod\_div\_mod n m a:

$0 < n \rightarrow 0 < m \rightarrow (n \mid m) \rightarrow a \bmod n = (a \bmod m) \bmod n.$

Proof.

intros H1 H2 (p, Hp).

rewrite (Z.div\_mod a m) at 1; auto with zarith.

rewrite Hp at 1.

rewrite Z.mul\_shuffle0, Z.add\_comm, Z.mod\_add; auto with zarith.

Qed.

Lemma Zmod\_divide\_minus a b c:

$0 < b \rightarrow a \bmod b = c \rightarrow (b \mid a - c).$

Proof.

intros H H1. apply Z.mod\_divide; auto with zarith.

rewrite Zminus\_mod; auto with zarith.

rewrite H1. rewrite <- (Z.mod\_small c b) at 1.

rewrite Z.sub\_diag, Z.mod\_0\_l; auto with zarith.

subst. now apply Z.mod\_pos\_bound.

Qed.

Lemma Zdivide\_mod\_minus a b c:

$0 \leq c < b \rightarrow (b \mid a - c) \rightarrow a \bmod b = c.$

Proof.

intros (H1, H2) H3.

assert (0 < b) by Z.order.

replace a with ((a - c) + c); auto with zarith.

rewrite Z.add\_mod; auto with zarith.

rewrite (Zdivide\_mod (a-c) b); try rewrite Z.add\_0\_l; auto with zarith.

rewrite Z.mod\_mod; try apply Zmod\_small; auto with zarith.

Qed.

(\*\* \* Greatest common divisor (gcd). \*)

(\*\* There is no unicity of the gcd; hence we define the predicate  
 [Zis\_gcd a b g] expressing that [g] is a gcd of [a] and [b].  
 (We show later that the [gcd] is actually unique if we discard its  
 sign.) \*)

```
Inductive Zis_gcd (a b g:Z) : Prop :=
 Zis_gcd_intro :
 (g | a) ->
 (g | b) ->
 (forall x, (x | a) -> (x | b) -> (x | g)) ->
 Zis_gcd a b g.
```

(\*\* Trivial properties of [gcd] \*)

```
Lemma Zis_gcd_sym : forall a b d, Zis_gcd a b d -> Zis_gcd b a d.
Proof.
 induction 1; constructor; intuition.
Qed.
```

```
Lemma Zis_gcd_0 : forall a, Zis_gcd a 0 a.
Proof.
 constructor; auto with zarith.
Qed.
```

```
Lemma Zis_gcd_1 : forall a, Zis_gcd a 1 1.
Proof.
 constructor; auto with zarith.
Qed.
```

```
Lemma Zis_gcd_refl : forall a, Zis_gcd a a a.
Proof.
 constructor; auto with zarith.
Qed.
```

```
Lemma Zis_gcd_minus : forall a b d, Zis_gcd a (- b) d -> Zis_gcd b a d.
Proof.
 induction 1; constructor; intuition.
Qed.
```

```
Lemma Zis_gcd_opp : forall a b d, Zis_gcd a b d -> Zis_gcd b a (- d).
Proof.
 induction 1; constructor; intuition.
Qed.
```

```
Lemma Zis_gcd_0_abs a : Zis_gcd 0 a (Z.abs a).
Proof.
 apply Zabs_ind.
 intros; apply Zis_gcd_sym; apply Zis_gcd_0; auto.
```

```

 intros; apply Zis_gcd_opp; apply Zis_gcd_0; auto.
Qed.

```

```

Hint Resolve Zis_gcd_sym Zis_gcd_0 Zis_gcd_minus Zis_gcd_opp: zarith.

```

```

Theorem Zis_gcd_unique: forall a b c d : Z,
 Zis_gcd a b c -> Zis_gcd a b d -> c = d \/ c = (- d).

```

```

Proof.

```

```

intros a b c d [Hc1 Hc2 Hc3] [Hd1 Hd2 Hd3].

```

```

assert (c|d) by auto.

```

```

assert (d|c) by auto.

```

```

apply Z.divide_antisym; auto.

```

```

Qed.

```

```

(** * Extended Euclid algorithm. *)

```

```

(** Euclid's algorithm to compute the [gcd] mainly relies on
 the following property. *)

```

```

Lemma Zis_gcd_for_euclid :

```

```

 forall a b d q:Z, Zis_gcd b (a - q * b) d -> Zis_gcd a b d.

```

```

Proof.

```

```

 simple induction 1; constructor; intuition.

```

```

 replace a with (a - q * b + q * b). auto with zarith. ring.

```

```

Qed.

```

```

Lemma Zis_gcd_for_euclid2 :

```

```

 forall b d q r:Z, Zis_gcd r b d -> Zis_gcd b (b * q + r) d.

```

```

Proof.

```

```

 simple induction 1; constructor; intuition.

```

```

 apply H2; auto.

```

```

 replace r with (b * q + r - b * q). auto with zarith. ring.

```

```

Qed.

```

```

(** We implement the extended version of Euclid's algorithm,
 i.e. the one computing Bezout's coefficients as it computes
 the [gcd]. We follow the algorithm given in Knuth's
 "Art of Computer Programming", vol 2, page 325. *)

```

```

Section extended_euclid_algorithm.

```

```

Variables a b : Z.

```

```

(** The specification of Euclid's algorithm is the existence of
 [u], [v] and [d] such that [ua+vb=d] and [(gcd a b d)]. *)

```

```

Inductive Euclid : Set :=

```

```
Euclid_intro :
forall u v d:Z, u * a + v * b = d -> Zis_gcd a b d -> Euclid.
```

```
(** The recursive part of Euclid's algorithm uses well-founded
recursion of non-negative integers. It maintains 6 integers
[u1,u2,u3,v1,v2,v3] such that the following invariant holds:
[u1*a+u2*b=u3] and [v1*a+v2*b=v3] and [gcd(u3,v3)=gcd(a,b)].
*)
```

```
Lemma euclid_rec :
forall v3:Z,
 0 <= v3 ->
 forall u1 u2 u3 v1 v2:Z,
 u1 * a + u2 * b = u3 ->
 v1 * a + v2 * b = v3 ->
 (forall d:Z, Zis_gcd u3 v3 d -> Zis_gcd a b d) -> Euclid.
```

Proof.

```
intros v3 Hv3; generalize Hv3; pattern v3.
apply Zlt_0_rec.
clear v3 Hv3; intros.
destruct (Z_zerop x) as [Heq|Hneq].
apply Euclid_intro with (u := u1) (v := u2) (d := u3).
assumption.
apply H3.
rewrite Heq; auto with zarith.
set (q := u3 / x) in *.
assert (Hq : 0 <= u3 - q * x < x).
replace (u3 - q * x) with (u3 mod x).
apply Z_mod_lt; omega.
assert (xpos : x > 0). omega.
generalize (Z_div_mod_eq u3 x xpos).
unfold q.
intro eq; pattern u3 at 2; rewrite eq; ring.
apply (H (u3 - q * x) Hq (proj1 Hq) v1 v2 x (u1 - q * v1) (u2 - q *
v2))).
tauto.
replace ((u1 - q * v1) * a + (u2 - q * v2) * b) with
 (u1 * a + u2 * b - q * (v1 * a + v2 * b)).
rewrite H1; rewrite H2; trivial.
ring.
intros; apply H3.
apply Zis_gcd_for_euclid with q; assumption.
assumption.
```

Qed.

```
(** We get Euclid's algorithm by applying [euclid_rec] on
[1,0,a,0,1,b] when [b>=0] and [1,0,a,0,-1,-b] when [b<0]. *)
```

Lemma euclid : Euclid.

Proof.

```

case (Z_le_gt_dec 0 b); intro.
intros;
 apply euclid_rec with
 (u1 := 1) (u2 := 0) (u3 := a) (v1 := 0) (v2 := 1) (v3 := b);
 auto with zarith; ring.
intros;
 apply euclid_rec with
 (u1 := 1) (u2 := 0) (u3 := a) (v1 := 0) (v2 := -1) (v3 := - b);
 auto with zarith; try ring.

```

Qed.

End extended\_euclid\_algorithm.

Theorem Zis\_gcd\_uniqueness\_apart\_sign :

```
forall a b d d':Z, Zis_gcd a b d -> Zis_gcd a b d' -> d = d' \/ d = - d'.
```

Proof.

```

simple induction 1.
intros H1 H2 H3; simple induction 1; intros.
generalize (H3 d' H4 H5); intro Hd'd.
generalize (H6 d H1 H2); intro Hdd'.
exact (Z.divide_antisym d d' Hdd' Hd'd).

```

Qed.

(\*\* \* Bezout's coefficients \*)

Inductive Bezout (a b d:Z) : Prop :=

```
Bezout_intro : forall u v:Z, u * a + v * b = d -> Bezout a b d.
```

(\*\* Existence of Bezout's coefficients for the [gcd] of [a] and [b] \*)

Lemma Zis\_gcd\_bezout : forall a b d:Z, Zis\_gcd a b d -> Bezout a b d.

Proof.

```

intros a b d Hgcd.
elim (euclid a b); intros u v d0 e g.
generalize (Zis_gcd_uniqueness_apart_sign a b d d0 Hgcd g).
intro H; elim H; clear H; intros.
apply Bezout_intro with u v.
rewrite H; assumption.
apply Bezout_intro with (- u) (- v).
rewrite H; rewrite <- e; ring.

```

Qed.

(\*\* gcd of [ca] and [cb] is [c gcd(a,b)]. \*)

Lemma Zis\_gcd\_mult :

```
forall a b c d:Z, Zis_gcd a b d -> Zis_gcd (c * a) (c * b) (c * d).
```

Proof.

```

intros a b c d; simple induction 1. constructor; auto with zarith.
intros x Ha Hb.
elim (Zis_gcd_bezout a b d H). intros u v Huv.
elim Ha; intros a' Ha'.
elim Hb; intros b' Hb'.
apply Zdivide_intro with (u * a' + v * b').
rewrite <- Huv.
replace (c * (u * a + v * b)) with (u * (c * a) + v * (c * b)).
rewrite Ha'; rewrite Hb'; ring.
ring.

```

Qed.

(\*\* \* Relative primality \*)

Definition rel\_prime (a b:Z) : Prop := Zis\_gcd a b 1.

(\*\* Bezout's theorem: [a] and [b] are relatively prime if and only if there exist [u] and [v] such that [ua+vb = 1]. \*)

Lemma rel\_prime\_bezout : forall a b:Z, rel\_prime a b -> Bezout a b 1.

Proof.

```

intros a b; exact (Zis_gcd_bezout a b 1).

```

Qed.

Lemma bezout\_rel\_prime : forall a b:Z, Bezout a b 1 -> rel\_prime a b.

Proof.

```

simple induction 1; constructor; auto with zarith.
intros. rewrite <- H0; auto with zarith.

```

Qed.

(\*\* Gauss's theorem: if [a] divides [bc] and if [a] and [b] are relatively prime, then [a] divides [c]. \*)

Theorem Gauss : forall a b c:Z, (a | b \* c) -> rel\_prime a b -> (a | c).

Proof.

```

intros. elim (rel_prime_bezout a b H0); intros.
replace c with (c * 1); [idtac | ring].
rewrite <- H1.
replace (c * (u * a + v * b)) with (c * u * a + v * (b * c));
[eauto with zarith | ring].

```

Qed.

(\*\* If [a] is relatively prime to [b] and [c], then it is to [bc] \*)

Lemma rel\_prime\_mult :

```

forall a b c:Z, rel_prime a b -> rel_prime a c -> rel_prime a (b * c).

```



Proof.

```

intros a b c Hb Hc.
elim (rel_prime_bezout a b Hb); intros.
elim (rel_prime_bezout a c Hc); intros.
apply bezout_rel_prime.
apply Bezout_intro with
 (u := u * u0 * a + v0 * c * u + u0 * v * b) (v := v * v0).
rewrite <- H.
replace (u * a + v * b) with ((u * a + v * b) * 1); [idtac | ring].
rewrite <- H0.
ring.

```

Qed.

Lemma rel\_prime\_cross\_prod :

```

forall a b c d:Z,
 rel_prime a b ->
 rel_prime c d -> b > 0 -> d > 0 -> a * d = b * c -> a = c /\ b = d.

```

Proof.

```

intros a b c d; intros.
elim (Z.divide_antisym b d).
split; auto with zarith.
rewrite H4 in H3.
rewrite Z.mul_comm in H3.
apply Z.mul_reg_l with d; auto with zarith.
intros; omega.
apply Gauss with a.
rewrite H3.
auto with zarith.
red; auto with zarith.
apply Gauss with c.
rewrite Z.mul_comm.
rewrite <- H3.
auto with zarith.
red; auto with zarith.

```

Qed.

(\*\* After factorization by a gcd, the original numbers are relatively prime. \*)

Lemma Zis\_gcd\_rel\_prime :

```

forall a b g:Z,
 b > 0 -> g >= 0 -> Zis_gcd a b g -> rel_prime (a / g) (b / g).

```

Proof.

```

intros a b g; intros.
assert (g <> 0).
intro.
elim H1; intros.
elim H4; intros.

```

```

rewrite H2 in H6; subst b; omega.
unfold rel_prime.
destruct H1.
destruct H1 as (a',H1).
destruct H3 as (b',H3).
replace (a/g) with a';
 [|rewrite H1; rewrite Z_div_mult; auto with zarith].
replace (b/g) with b';
 [|rewrite H3; rewrite Z_div_mult; auto with zarith].
constructor.
exists a'; auto with zarith.
exists b'; auto with zarith.
intros x (xa,H5) (xb,H6).
destruct (H4 (x*g)) as (x',Hx').
exists xa; rewrite Z.mul_assoc; rewrite <- H5; auto.
exists xb; rewrite Z.mul_assoc; rewrite <- H6; auto.
replace g with (1*g) in Hx'; auto with zarith.
do 2 rewrite Z.mul_assoc in Hx'.
apply Z.mul_reg_r in Hx'; trivial.
rewrite Z.mul_1_r in Hx'.
exists x'; auto with zarith.
Qed.

Theorem rel_prime_sym: forall a b, rel_prime a b -> rel_prime b a.
Proof.
 intros a b H; auto with zarith.
 red; apply Zis_gcd_sym; auto with zarith.
Qed.

Theorem rel_prime_div: forall p q r,
 rel_prime p q -> (r | p) -> rel_prime r q.
Proof.
 intros p q r H (u, H1); subst.
 inversion_clear H as [H1 H2 H3].
 red; apply Zis_gcd_intro; try apply Z.divide_1_1.
 intros x H4 H5; apply H3; auto.
 apply Z.divide_mul_r; auto.
Qed.

Theorem rel_prime_1: forall n, rel_prime 1 n.
Proof.
 intros n; red; apply Zis_gcd_intro; auto.
 exists 1; auto with zarith.
 exists n; auto with zarith.
Qed.

Theorem not_rel_prime_0: forall n, 1 < n -> ~ rel_prime 0 n.
Proof.

```

```

intros n H H1; absurd (n = 1 \ / n = -1).
intros [H2 | H2]; subst; contradict H; auto with zarith.
case (Zis_gcd_unique 0 n n 1); auto.
apply Zis_gcd_intro; auto.
exists 0; auto with zarith.
exists 1; auto with zarith.

```

Qed.

```

Theorem rel_prime_mod: forall p q, 0 < q ->
 rel_prime p q -> rel_prime (p mod q) q.

```

Proof.

```

intros p q H H0.
assert (H1: Bezout p q 1).
apply rel_prime_bezout; auto.
inversion_clear H1 as [q1 r1 H2].
apply bezout_rel_prime.
apply Bezout_intro with q1 (r1 + q1 * (p / q)).
rewrite <- H2.
pattern p at 3; rewrite (Z_div_mod_eq p q); try ring; auto with zarith.

```

Qed.

```

Theorem rel_prime_mod_rev: forall p q, 0 < q ->
 rel_prime (p mod q) q -> rel_prime p q.

```

Proof.

```

intros p q H H0.
rewrite (Z_div_mod_eq p q); auto with zarith; red.
apply Zis_gcd_sym; apply Zis_gcd_for_euclid2; auto with zarith.

```

Qed.

```

Theorem Zrel_prime_neq_mod_0: forall a b, 1 < b -> rel_prime a b -> a mod
b <> 0.

```

Proof.

```

intros a b H H1 H2.
case (not_rel_prime_0 _ H).
rewrite <- H2.
apply rel_prime_mod; auto with zarith.

```

Qed.

(\*\* \* Primality \*)

```

Inductive prime (p:Z) : Prop :=

```

```

 prime_intro :

```

```

 1 < p -> (forall n:Z, 1 <= n < p -> rel_prime n p) -> prime p.

```

(\*\* The sole divisors of a prime number [p] are [-1], [1], [p] and [-p]. \*)

```

Lemma prime_divisors :

```

```

 forall p:Z,

```

```

 prime p -> forall a:Z, (a | p) -> a = -1 \/ a = 1 \/ a = p \/ a = - p.
Proof.
 destruct 1; intros.
 assert
 (a = - p \/ - p < a < -1 \/ a = -1 \/ a = 0 \/ a = 1 \/ 1 < a < p \/ a
= p).
 { assert (Z.abs a <= Z.abs p) as H2.
 apply Zdivide_bounds; [assumption | omega].
 revert H2.
 pattern (Z.abs a); apply Zabs_ind; pattern (Z.abs p); apply Zabs_ind;
 intros; omega. }
 intuition idtac.
 (* -p < a < -1 *)
 - absurd (rel_prime (- a) p); intuition.
 inversion H2.
 assert (- a | - a) by auto with zarith.
 assert (- a | p) by auto with zarith.
 apply H7, Z.divide_1_r in H8; intuition.
 (* a = 0 *)
 - inversion H1. subst a; omega.
 (* 1 < a < p *)
 - absurd (rel_prime a p); intuition.
 inversion H2.
 assert (a | a) by auto with zarith.
 assert (a | p) by auto with zarith.
 apply H7, Z.divide_1_r in H8; intuition.

```

Qed.

(\*\* A prime number is relatively prime with any number it does not divide  
\*)

Lemma prime\_rel\_prime :

```
forall p:Z, prime p -> forall a:Z, ~ (p | a) -> rel_prime p a.
```

Proof.

```

 intros; constructor; intros; auto with zarith.
 apply prime_divisors in H1; intuition; subst; auto with zarith.
 - absurd (p | a); auto with zarith.
 - absurd (p | a); intuition.

```

Qed.

Hint Resolve prime\_rel\_prime: zarith.

(\*\* As a consequence, a prime number is relatively prime with smaller  
numbers \*)

Theorem rel\_prime\_le\_prime:

```
forall a p, prime p -> 1 <= a < p -> rel_prime a p.
```

Proof.

```

intros a p Hp [H1 H2].
apply rel_prime_sym; apply prime_rel_prime; auto.
intros [q Hq]; subst a.
case (Z.le_gt_cases q 0); intros H1.
absurd (q * p <= 0 * p); auto with zarith.
absurd (1 * p <= q * p); auto with zarith.
Qed.

```

(\*\* If a prime [p] divides [ab] then it divides either [a] or [b] \*)

```

Lemma prime_mult :
 forall p:Z, prime p -> forall a b:Z, (p | a * b) -> (p | a) \/ (p | b).
Proof.
 intro p; simple induction 1; intros.
 case (Zdivide_dec p a); intuition.
 right; apply Gauss with a; auto with zarith.
Qed.

```

```

Lemma not_prime_0: ~ prime 0.
Proof.
 intros H1; case (prime_divisors _ H1 2); auto with zarith.
Qed.

```

```

Lemma not_prime_1: ~ prime 1.
Proof.
 intros H1; absurd (1 < 1); auto with zarith.
 inversion H1; auto.
Qed.

```

```

Lemma prime_2: prime 2.
Proof.
 apply prime_intro; auto with zarith.
 intros n (H,H'); Z.le_elim H; auto with zarith.
 - contradict H'; auto with zarith.
 - subst n. constructor; auto with zarith.
Qed.

```

```

Theorem prime_3: prime 3.
Proof.
 apply prime_intro; auto with zarith.
 intros n (H,H'); Z.le_elim H; auto with zarith.
 - replace n with 2 by omega.
 constructor; auto with zarith.
 intros x (q,Hq) (q',Hq').
 exists (q' - q). ring_simplify. now rewrite <- Hq, <- Hq'.
 - replace n with 1 by trivial.
 constructor; auto with zarith.

```

Qed.

Theorem prime\_ge\_2 p : prime p -> 2 <= p.

Proof.

intros (Hp, \_); auto with zarith.

Qed.

Definition prime' p := 1 < p /\ (forall n, 1 < n < p -> ~ (n|p)).

Lemma Z\_0\_1\_more x : 0 <= x -> x = 0 \/ x = 1 \/ 1 < x.

Proof.

intros H. Z.le\_elim H; auto.

apply Z.le\_succ\_1 in H. change (1 <= x) in H. Z.le\_elim H; auto.

Qed.

Theorem prime\_alt p : prime' p <-> prime p.

Proof.

split; intros (Hp, H).

- (\* prime -> prime' \*)

constructor; trivial; intros n Hn.

constructor; auto with zarith; intros x Hxn Hxp.

rewrite <- Z.divide\_abs\_1 in Hxn, Hxp |- \*.

assert (Hx := Z.abs\_nonneg x).

set (y := Z.abs x) in \*; clearbody y; clear x; rename y into x.

destruct (Z\_0\_1\_more x Hx) as [->|[->|Hx'|]].

+ exfalso. apply Z.divide\_0\_1 in Hxn. omega.

+ now exists 1.

+ elim (H x); auto.

split; trivial.

apply Z.le\_lt\_trans with n; auto with zarith.

apply Z.divide\_pos\_le; auto with zarith.

- (\* prime' -> prime \*)

constructor; trivial. intros n Hn Hnp.

case (Z.is\_gcd\_unique n p n 1); auto with zarith.

constructor; auto with zarith.

apply H; auto with zarith.

Qed.

Theorem square\_not\_prime: forall a, ~ prime (a \* a).

Proof.

intros a Ha.

rewrite <- (Z.abs\_square a) in Ha.

assert (H := Z.abs\_nonneg a).

set (b := Z.abs a) in \*; clearbody b; clear a; rename b into a.

rewrite <- prime\_alt in Ha; destruct Ha as (Ha, Ha').

assert (H' : 1 < a) by now apply (Z.square\_lt\_simpl\_nonneg 1).

apply (Ha' a).

+ split; trivial.

```

 rewrite <- (Z.mul_1_1 a) at 1. apply Z.mul_lt_mono_pos_r; omega.
 + exists a; auto.
Qed.

```

```

Theorem prime_div_prime: forall p q,
 prime p -> prime q -> (p | q) -> p = q.

```

Proof.

```

 intros p q H H1 H2;
 assert (Hp: 0 < p); try apply Z.lt_le_trans with 2; try apply prime_ge_2;
 auto with zarith.
 assert (Hq: 0 < q); try apply Z.lt_le_trans with 2; try apply prime_ge_2;
 auto with zarith.
 case prime_divisors with (2 := H2); auto.
 intros H4; contradict Hp; subst; auto with zarith.
 intros [H4| [H4 | H4]]; subst; auto.
 contradict H; auto; apply not_prime_1.
 contradict Hp; auto with zarith.

```

Qed.

```

(** we now prove that [Z.gcd] is indeed a gcd in
 the sense of [Zis_gcd]. *)

```

```

Notation Zgcd_is_pos := Z.gcd_nonneg (compat "8.3").

```

```

Lemma Zgcd_is_gcd : forall a b, Zis_gcd a b (Z.gcd a b).

```

Proof.

```

 constructor.
 apply Z.gcd_divide_l.
 apply Z.gcd_divide_r.
 apply Z.gcd_greatest.

```

Qed.

```

Theorem Zgcd_spec : forall x y : Z, {z : Z | Zis_gcd x y z /\ 0 <= z}.

```

Proof.

```

 intros x y; exists (Z.gcd x y).
 split; [apply Zgcd_is_gcd | apply Z.gcd_nonneg].

```

Qed.

```

Theorem Zdivide_Zgcd: forall p q r : Z,
 (p | q) -> (p | r) -> (p | Z.gcd q r).

```

Proof.

```

 intros. now apply Z.gcd_greatest.

```

Qed.

```

Theorem Zis_gcd_gcd: forall a b c : Z,
 0 <= c -> Zis_gcd a b c -> Z.gcd a b = c.

```

Proof.

```

 intros a b c H1 H2.

```

```

case (Zis_gcd_uniqueness_apart_sign a b c (Z.gcd a b)); auto.
apply Zgcd_is_gcd; auto.
Z.le_elim H1.
- generalize (Z.gcd_nonneg a b); auto with zarith.
- subst. now case (Z.gcd a b).
Qed.

```

```

Notation Zgcd_inv_0_l := Z.gcd_eq_0_l (compat "8.3").
Notation Zgcd_inv_0_r := Z.gcd_eq_0_r (compat "8.3").

```

```

Theorem Zgcd_div_swap0 : forall a b : Z,
 0 < Z.gcd a b ->
 0 < b ->
 (a / Z.gcd a b) * b = a * (b/Z.gcd a b).

```

Proof.

```

intros a b Hg Hb.
assert (F := Zgcd_is_gcd a b); inversion F as [F1 F2 F3].
pattern b at 2; rewrite (Zdivide_Zdiv_eq (Z.gcd a b) b); auto.
repeat rewrite Z.mul_assoc; f_equal.
rewrite Z.mul_comm.
rewrite <- Zdivide_Zdiv_eq; auto.

```

Qed.

```

Theorem Zgcd_div_swap : forall a b c : Z,
 0 < Z.gcd a b ->
 0 < b ->
 (c * a) / Z.gcd a b * b = c * a * (b/Z.gcd a b).

```

Proof.

```

intros a b c Hg Hb.
assert (F := Zgcd_is_gcd a b); inversion F as [F1 F2 F3].
pattern b at 2; rewrite (Zdivide_Zdiv_eq (Z.gcd a b) b); auto.
repeat rewrite Z.mul_assoc; f_equal.
rewrite Zdivide_Zdiv_eq_2; auto.
repeat rewrite <- Z.mul_assoc; f_equal.
rewrite Z.mul_comm.
rewrite <- Zdivide_Zdiv_eq; auto.

```

Qed.

```

Notation Zgcd_comm := Z.gcd_comm (compat "8.3").

```

```

Lemma Zgcd_ass a b c : Z.gcd (Z.gcd a b) c = Z.gcd a (Z.gcd b c).

```

Proof.

```

symmetry. apply Z.gcd_assoc.

```

Qed.

```

Notation Zgcd_Zabs := Z.gcd_abs_l (compat "8.3").
Notation Zgcd_0 := Z.gcd_0_r (compat "8.3").
Notation Zgcd_1 := Z.gcd_1_r (compat "8.3").

```



Hint Resolve Z.gcd\_0\_r Z.gcd\_1\_r : zarith.

Theorem Zgcd\_1\_rel\_prime : forall a b,  
Z.gcd a b = 1 <-> rel\_prime a b.

Proof.

```

unfold rel_prime; split; intro H.
rewrite <- H; apply Zgcd_is_gcd.
case (Zis_gcd_unique a b (Z.gcd a b) 1); auto.
apply Zgcd_is_gcd.
intros H2; absurd (0 <= Z.gcd a b); auto with zarith.
generalize (Z.gcd_nonneg a b); auto with zarith.

```

Qed.

Definition rel\_prime\_dec: forall a b,  
{ rel\_prime a b }+{ ~ rel\_prime a b }.

Proof.

```

intros a b; case (Z.eq_dec (Z.gcd a b) 1); intros H1.
left; apply -> Zgcd_1_rel_prime; auto.
right; contradict H1; apply <- Zgcd_1_rel_prime; auto.

```

Defined.

Definition prime\_dec\_aux:

```

forall p m,
{ forall n, 1 < n < m -> rel_prime n p } +
{ exists n, 1 < n < m /\ ~ rel_prime n p }.

```

Proof.

```

intros p m.
case (Z.lt_dec 1 m); intros H1;
[| left; intros; exfalse; omega].
pattern m; apply natlike_rec; auto with zarith.
left; intros; exfalse; omega.
intros x Hx IH; destruct IH as [F|E].
destruct (rel_prime_dec x p) as [Y|N].
left; intros n [HH1 HH2].
rewrite Z.lt_succ_r in HH2.
Z.le_elim HH2; subst; auto with zarith.
- case (Z.lt_dec 1 x); intros HH1.
 * right; exists x; split; auto with zarith.
 * left; intros n [HHH1 HHH2]; contradict HHH1; auto with zarith.
- right; destruct E as (n, ((H0,H2),H3)); exists n; auto with zarith.

```

Defined.

Definition prime\_dec: forall p, { prime p }+{ ~ prime p }.

Proof.

```

intros p; case (Z.lt_dec 1 p); intros H1.
+ case (prime_dec_aux p p); intros H2.
 * left; apply prime_intro; auto.

```

```

 intros n (Hn1,Hn2). Z.le_elim Hn1; auto; subst n.
 constructor; auto with zarith.
 * right; intros H3; inversion_clear H3 as [Hp1 Hp2].
 case H2; intros n [Hn1 Hn2]; case Hn2; auto with zarith.
+ right; intros H3; inversion_clear H3 as [Hp1 Hp2]; case H1; auto.
Defined.

```

Theorem not\_prime\_divide:

```
forall p, 1 < p -> ~ prime p -> exists n, 1 < n < p /\ (n | p).
```

Proof.

```

intros p Hp Hp1.
case (prime_dec_aux p p); intros H1.
- elim Hp1; constructor; auto.
 intros n (Hn1,Hn2).
 Z.le_elim Hn1; auto with zarith.
 subst n; constructor; auto with zarith.
- case H1; intros n (Hn1,Hn2).
 destruct (Z_0_1_more _ (Z.gcd_nonneg n p)) as [H|[H|H]].
+ exfalso. apply Z.gcd_eq_0_1 in H. omega.
+ elim Hn2. red. rewrite <- H. apply Zgcd_is_gcd.
+ exists (Z.gcd n p); split; [split; auto | apply Z.gcd_divide_r].
 apply Z.le_lt_trans with n; auto with zarith.
 apply Z.divide_pos_le; auto with zarith.
 apply Z.gcd_divide_l.

```

Qed.

## Додаток Г

### Демонстрація методу Гауса для розв'язання систем лінійних алгебраїчних рівнянь

```
html(''<h1 style="text-align: center;"><span style="color:
#ff0000;">Метод
Гауса</h1>
```

```
<p style="text-align: justify;">Найбільш
розповсюдженим точним способом рішення систем лінійних рівнянь є алгоритм
послідовного виключення невідомих. Цей метод носить назву методу
Гауса.</p>
```

```
<p style="text-align: justify;">Нехай є
система (1.1) n лінійних алгебраїчних рівнянь (СЛАР)
 n невідомими:</p>
```

```
<center>
$ A = \left\{ \begin{array} {l} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array} \right. = \begin{array} {l} b_1 \\ b_2 \\ \vdots \\ b_n \end{array}
</center>
```

```
<p>Позначимо через
$A = \left(\begin{array} {l} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array} \right)$
```

```
 матрицю коефіцієнтів системи (1.1), через
$b = \left(\begin{array} {l} b_1 \\ b_2 \\ \vdots \\ b_n \end{array} \right)$
стовпець її вільних членів, а вектор невідомих - через
$x = \left(\begin{array} {l} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right)$
```

```
<p style="text-align: justify;">Тоді
систему (1.1) можна записати у матричному вигляді: $A \cdot x = b$ і
створити її розширену матрицю (1.2) (A|b):</p>
```

```
<center>
$A' = \left(\begin{array} {l} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array} \right)$
</center>
```

$$a_{n1} \text{ \& } a_{n2} \text{ \& } \dots \text{ \& } a_{nn} \text{ \& } b_n$$

Перший етап розв'язання системи рівнянь (1.1), що називається прямим ходом методу Гауса, полягає у приведенні розширеної матриці (1.2) до трикутного вигляду. Це означає, що усі елементи матриці (1.2) нижче головної діагоналі повинні дорівнювати 0:

$$A' = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{pmatrix}$$

Для формування першого стовпця матриці (1.3) необхідно із кожного рядка (починаючи із другого) відняти перший, помножений на деяке число  $M$ .

У загальному вигляді процес можна описати так:

2-й рядок = 2-й рядок -  $M$   $x$  1-й рядок,  
 3-й рядок = 3-й рядок -  $M$   $x$  1-й рядок,  
 $\dots$   
 $i$ -й рядок =  $i$ -й рядок -  $M$   $x$  1-й рядок,  
 $\dots$   
 $n$ -й рядок =  $n$ -й рядок -  $M$   $x$  1-й рядок.

Зрозуміло, що перетворення елементів другого рядка буде виконуватись за формулами:

$$a_{21} = a_{21} - M \cdot a_{11}, \quad a_{22} = a_{22} - M \cdot a_{12}, \quad \dots$$

$$a_{2i} = a_{2i} - M \cdot a_{1i}, \quad \dots, \quad a_{2n} = a_{2n} - M \cdot a_{1n},$$

$$b_2 = b_2 - M \cdot b_1.$$

Так як метою даних перетворень є обнулення першого елемента рядка, то  $M$  обирається із умови:

$$a_{21} - M \cdot a_{11} = 0,$$

Отже,

$$M = \frac{a_{21}}{a_{11}}$$

$$a_{i1} = a_{i1} - M \cdot a_{11},$$

$$a_{i2} = a_{i2} - M \cdot a_{12}, \dots$$

$$a_{in} = a_{in} - M \cdot a_{1n}$$

$$a_{1n} = a_{1n} - M \cdot a_{1n}$$

$$b_{i3} = b_{i3} - M \cdot b_{13}$$

$$b_{13} = b_{13} - M \cdot b_{13}$$

$$a_{i1} - M \cdot a_{11} = 0$$

$$\Rightarrow M = \frac{a_{i1}}{a_{11}}$$

Таким чином, перетворення елементів  $i$ -го рядка буде виконуватись наступним чином:

$$a_{i1} = a_{i1} - M \cdot a_{11},$$

$$a_{i2} = a_{i2} - M \cdot a_{12}, \dots$$

$$a_{in} = a_{in} - M \cdot a_{1n}$$

$$b_{i3} = b_{i3} - M \cdot b_{13}$$

$$b_{13} = b_{13} - M \cdot b_{13}$$

Коефіцієнт  $M$  для  $i$ -го рядка обирається із умови:

$$a_{i1} - M \cdot a_{11} = 0$$

$$i$$
  
дорівнює

$$M = \frac{a_{i1}}{a_{11}}$$

Після проведення подібних перетворень для усіх рядків матриця (1.2) прийме вигляд:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ b_{13} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} \\ b_{23} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \\ b_{n3} & 0 & \dots & 0 \end{pmatrix}$$

Очевидно, що якщо повторити описаний вище алгоритм для наступних стовпців матриці (1.2), причому починати перетворення другого стовпця із третього елемента, третій стовпець - з четвертого і т.п., то у результаті буде отримана матриця (1.3).

<p><span style="font-size: large;">Відмітимо, що якщо у матриці (1.2) на головній діагоналі зустрінеться елемент  $a_{kk}$ , рівний нулю, то розрахунок коефіцієнта</span></p>

<p style="text-align: center;"> $M = \frac{a_{ik}}{a_{kk}}$ </p>  
<p><span style="font-size: large;">для  $k$ -го рядка буде неможливий. Минувати ділення на нуль можливо, якщо позбавитись нульових елементів на головній діагоналі. Для цього перед обнуленням елементів у  $k$ -му стовпці необхідно знайти в ньому максимальний по модулю елемент, запам'ятати номер рядка, в якому він знаходиться, і поміняти його місцями з  $k$ -м рядком.</span></p>

<p><span style="font-size: large;">У результаті виконання прямого ходу Гауса, матриця (1.2.) перетвориться у матрицю (1.3), а система рівнянь (1.1) буде мати наступний вигляд:</span></p>

```

$A = \left\{\begin{array}{l} a_{11}x_1+a_{12}x_2+a_{13}x_3+\dots+a_{1n}x_n=b_1, \\ a_{22}x_2+a_{33}x_3+\dots+a_{2n}x_n=b_2, \\ a_{33}x_3+\dots+a_{3n}x_n=b_3, \\ \dots \\ a_{nn}x_n=b_n. \end{array}\right.

```

```

@interact
def demosystem(m=slider(1, 9, step_size=1, label="Кількість рівнянь"), \
n=slider(1, 9, step_size=1, label="Кількість невідомих")):
 L=[]

 for i in range(m):
 for j in range(n):
 L.append(var("a%s%s"%(i+1,j+1)))

 A=matrix(m,n,L)

 L=[]

```

```

for i in range(m):
 for j in range(n):
 L.append(var("a%s%s"%(i+1,j+1)))
 L.append(var("b%s"%(i+1)))

Ab=matrix(m,n+1,L)
x=matrix(n,1,var(", ".join("x%s"%(i+1) for i in range(n))))
b=matrix(m,1,var(", ".join("b%s"%(i+1) for i in range(m))))

s="$\\left \\{ \\begin{eqnarray}"

for i in range(m):
 s=s+"s=%s \\\\"%(latex((A*x)[i][0]),latex(b[i][0]))
s=s+"\\end{eqnarray} \\right.$"

html("</pre>Система рівнянь:
"s)
html("<p>Матриця коефіцієнтів: A=%s$"%latex(A))
html("<p>Вектор невідомих: <i>x</i>=%s$"%latex(x))
html("<p>Вектор вільних коефіцієнтів: <i>b</i>=%s$"%latex(b))
html("<p>Розширена матриця: A|<i>b</i>=%s$"%latex(Ab))

for j in range(m):
 i=j+1
 while i<m and j<n:
 M=Ab[i][j]/Ab[j][j]
 Ab[i]=Ab[i]-M*Ab[j]
 i=i+1

html("<p>Розширена матриця після прямого ходу Гауса:

A|<i>b</i>=%s$"%latex(Ab))

j=m-1
while j>0:
 i=j-1
 while i>=0:
 if i<m and j<n:
 M=Ab[i][j]/Ab[j][j]
 Ab[i]=Ab[i]-M*Ab[j]
 i=i-1
 j=j-1

html("<p>Розширена матриця після оберненого ходу Гауса:

A|<i>b</i>=%s$"%latex(Ab))

for j in range(m):
 if j<n:
 Ab[j]=Ab[j]/Ab[j][j]

```

```

html("<p>Розширена матриця після зведення до одиничної:

A|<i>b</i>=%s$" % latex(Ab))
L=x.list()
L.append(-1)
xr=matrix(n+1,1,L)

if n<m:
 html("<p>Так як кількість невідомих менша за кількість рівнянь,
розв'язання системи методом Гауса неможливо")
else:
 html("<p>Розширений вектор невідомих:

<i>x'</i>=%s$" % latex(xr))
 html("<p>Розв'язок системи:
")
 temp=Ab*xr
 L=x.list()
 for i in range(m):
 html("%s$" % latex(solve(temp[i][0]==0,x[i][0])[0]))
 L[i]=solve(temp[i][0]==0,x[i][0])[0].rhs()
 roots=matrix(n,1,L)
 res=A*roots()
 html("<p>Вектор коренів: %s$" % latex(roots))
 L=res.list()
 for j in range(m):
 L[j]=L[j].simplify_rational()
 res=matrix(m,1,L)
 html("<p>Перевірка:

A$\cdot<i>roots</i>=%s$" % latex(res))

```



## Додаток Д

### Список публікацій за темою дисертації та відомості про апробацію результатів дисертації

#### Список публікацій О. М. Маркової за темою дисертації «Хмарні технології як засіб навчання основ математичної інформатики студентів технічних університетів»

1. Туравініна О. М. Математика-0 для педагогічних ВНЗ: вступ до булевої алгебри / Наталія Василенко, Володимир Петров, Оксана Туравініна // Математика в школі. – 2008. – № 3. – С. 16-21.

2. Туравініна О. М. Методології навчання за напрямком «Комп'ютерна інженерія» на базі Криворізького технічного університету / А. І. Купін, В. А. Чубаров, О. М. Туравініна // Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія № 5. Педагогічні науки: реалії та перспективи. – Випуск 17 : збірник наукових праць / За ред. В. Д. Сиротюка. – К. : Вид-во НПУ імені М. П. Драгоманова, 2009. – С. 102-108.

3. Туравініна О. М. Математична інформатика у системі фундаменталізації навчання студентів технічних університетів / О. М. Туравініна // Збірник наукових праць Кам'янець-Подільського національного університету імені Івана Огієнка. Серія педагогічна / [редкол. : П. С. Атаманчук (голова, наук. ред. та ін.)]. – Кам'янець-Подільський : Кам'янець-Подільський національний університет ім. І. Огієнка, 2012. – Випуск 18 : Інновації в навчанні фізики: навчальний та міжнародний досвід. – С. 189-191.

4. Маркова О. М. Хмарні технології навчання: витоки [Електронний ресурс] / Маркова Оксана Миколаївна, Семеріков Сергій Олексійович, Стрюк Андрій Миколайович // Інформаційні технології і засоби навчання. – 2015. – Том 46, №2. – С. 29-44. – Режим доступу : <http://journal.iitta.gov.ua/index.php/itlt/article/download/1234/913>.

5. Маркова О. М. Модель методичної системи та цілі навчання основ математичної інформатики студентів технічних університетів / Маркова О. М. //

Вісник Черкаського університету. Серія Педагогічні науки. – 2016. – № 7. – С. 36-42.

6. Маркова О. М. Моделі використання хмарних технологій у підготовці ІТ-фахівців / Маркова О. М. // Науковий часопис НПУ імені М. П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання : зб. наук. праць / Редрада. – К. : НПУ імені М. П. Драгоманова, 2016. – № 18 (25). – С. 85-94.

7. Markova O. M. The tools of cloud technology for learning of fundamentals of mathematical informatics for students of technical universities [Electronic resource] / Oksana M. Markova // Cloud Technologies in Education : Proceedings of the 5th Workshop on Cloud Technologies in Education (CTE 2017). Kryvyi Rih, Ukraine, April 28, 2017 / Edited by : Serhiy O. Semerikov, Mariya P. Shyshkina. – P. 27-33. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2168). – Access mode : <http://ceur-ws.org/Vol-2168/paper5.pdf>.

8. Markova O. M. CoCalc as a Learning Tool for Neural Network Simulation in the Special Course “Foundations of Mathematic Informatics” [Electronic resource] / Oksana Markova, Serhiy Semerikov, Maiia Popel // ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer 2018 : Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops (ICTERI, 2018). Kyiv, Ukraine, May 14-17, 2018 / Edited by : Vadim Ermolayev, Mari Carmen Suárez-Figueroa, Vitaliy Yakovyna, Vyacheslav Kharchenko, Vitaliy Kobets, Hennadiy Kravtsov, Vladimir Peschanenko, Yaroslav Prytula, Mykola Nikitchenko, Aleksander Spivakovsky. – P. 388-403. – (CEUR Workshop Proceedings (CEUR-WS.org), Vol. 2104). – Access mode : [http://ceur-ws.org/Vol-2104/paper\\_204.pdf](http://ceur-ws.org/Vol-2104/paper_204.pdf).

9. Туравініна О. М. Застосування інформаційних технологій для підготовки спеціалістів за напрямом «Комп'ютерна інженерія» / А. І. Купін, В. А. Чубаров, О. М. Туравініна // Новітні комп'ютерні технології : матеріали VII Міжнародної науково-технічної конференції : Київ-Севастополь, 15-18 вересня 2009 р. – К. : Міністерство регіонального розвитку та будівництва

України, 2009. – С. 101-102.

10. Туравініна О. М. Застосування мобільних засобів навчання математичної інформатики в курсі фізики / О. М. Туравініна // Тези доповідей VII Всеукраїнської науково-практичної конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2010) : Черкаси, 4-6 травня 2010 р. – У 2-х томах. – Черкаси : ЧДТУ, 2010. – Т. 2. – С. 74.

11. Туравініна О. М. До питання про розробку методики навчання математичної інформатики у технічному ВНЗ / О. М. Туравініна, І. О. Теплицький, І. І. Ліннік // Новітні комп'ютерні технології : матеріали VIII Міжнародної науково-технічної конференції : Київ-Севастополь, 14-17 вересня 2010 р. – К. : Міністерство регіонального розвитку та будівництва України, 2010. – С. 168-169.

12. Туравініна О. М. Засоби хмарних обчислень у комп'ютерній інженерії / С. А. Бурма, О. М. Туравініна // Комп'ютерні інтелектуальні системи та мережі : матеріали IV Всеукраїнської конференції аспірантів, студентів та молодих вчених (23-25 березня 2011 р.) / Міністерство освіти і науки України, Криворізький технічний університет, Кафедра комп'ютерних систем та мереж. – Кривий Ріг : Видавництво Криворізького технічного університету, 2011. – С. 33-34.

13. Туравініна О. М. Amazon EC2 як платформа для організації хмарних обчислень / О. М. Туравініна, А. М. Стрюк, Н. В. Рашевська, К. І. Словак // Новітні комп'ютерні технології : матеріали IX Міжнародної науково-технічної конференції : Київ-Севастополь, 13-16 вересня 2011 р. – К. : Міністерство регіонального розвитку та будівництва України, 2011. – С. 187-188.

14. Туравініна О. М. Використання хмарних обчислень у комбінованому навчанні системного програмування / А. М. Стрюк, О. М. Туравініна // Тези доповідей Міжнародної науково-практичної конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2012) : Черкаси, 25-27 квітня 2012 р. – У 2 т. – Черкаси : ЧДТУ, 2012. – Т. 2. – С. 96-97.

15. Туравініна О. М. Хмарні технології навчання студентів /

О. М. Туравініна // Новітні комп'ютерні технології : матеріали X Міжнародної науково-технічної конференції : Севастополь, 11-14 вересня 2012 р. – К. : Міністерство регіонального розвитку та будівництва України, 2012. – С. 119-121.

16. Туравініна О. М. Зміст навчання основ математичної інформатики студентів технічних університетів / О. М. Туравініна, С. О. Семеріков // Матеріали міжнародної науково-методичної конференції «Розвиток інтелектуальних умінь і творчих здібностей учнів та студентів у процесі навчання дисциплін природничо-математичного циклу» (ІТМ\*плюс - 2012), м. Суми, 6-7 грудня 2012 р. – Суми : Вид-во СумДПУ ім. А. С. Макаренка, 2012. – С. 142-145.

17. Туравініна О. М. Хмарні технології навчання у системі інформаційно-комунікаційних технологій навчального призначення / О. М. Туравініна // Хмарні технології в освіті : матеріали Всеукраїнського науково-методичного Інтернет-семінару (Кривий Ріг – Київ – Черкаси – Харків, 21 грудня 2012 р.). – Кривий Ріг : Видавничий відділ КМІ, 2012. – С. 9.

18. Туравініна О. М. Sagemaths як хмарний засіб реалізації основних чисельних методів / Формус Ю. В., Туравініна О. М. // Комп'ютерні інтелектуальні системи та мережі : матеріали VII Всеукраїнської науково-практичної WEB конференції аспірантів, студентів та молодих вчених (25-27 березня 2014 р.) / Міністерство освіти і науки України, ДВНЗ «Криворізький національний університет», Кафедра комп'ютерних систем та мереж. – Кривий Ріг : Видавництво Криворізького національного університету, 2014. – С. 119-120.

19. Маркова О. М. Історичні аспекти розвитку хмарних технологій / Маркова О. М. // Тези доповідей науково-практичного семінару «Хмарні технології в сучасному університеті» (ХТСУ-2015) : Черкаси, 24 березня 2015 р. / Міністерство освіти і науки України, Черкаський державний технологічний університет, Інститут інформаційних технологій і засобів навчання НАПН України, Криворізький національний університет, Національний педагогічний

університет імені М. П. Драгоманова, Черкаський національний університет імені Богдана Хмельницького. – Черкаси : ЧДТУ, 2015. – С. 29-30.

20. Маркова О. М. Хмарні обчислення і технології: тенденції розвитку / Халимова Б. А., Маркова О. М. // Комп'ютерні інтелектуальні системи та мережі : матеріали ІХ Всеукраїнської науково-практичної WEB конференції аспірантів, студентів та молодих вчених (22-24 березня 2016 р.) / Міністерство освіти і науки України, ДВНЗ «Криворізький національний університет», Кафедра комп'ютерних систем та мереж. – Кривий Ріг : Видавничий центр ДВНЗ «Криворізький національний університет», 2016. – С. 25-26.

21. Маркова О. М. Хмарні технології навчання: спроба визначення / О. М. Маркова // Новітні комп'ютерні технології. – Кривий Ріг : ДВНЗ «Криворізький національний університет». – 2014. – Том XII : спецвипуск «Хмарні технології в освіті». – С. 244-248.

22. Маркова О. М. Теоретичні засади навчання основ математичної інформатики студентів технічних університетів з використанням хмарних технологій / О. М. Маркова // Новітні комп'ютерні технології. – Кривий Ріг : ДВНЗ «Криворізький національний університет». – 2016. – Том XIV. – С. 63-64.

23. Маркова О. М. Загальна структура засобів хмарних технологій навчання основ математичної інформатики / Оксана Миколаївна Маркова // Новітні комп'ютерні технології. – Кривий Ріг : ДВНЗ «Криворізький національний університет». – 2017. – Том XV. – С. 231-234.

24. Маркова О. М. SageMathCloud як засіб хмарних технологій комп'ютерно-орієнтованого навчання математичних та інформатичних дисциплін / Шокалюк С. В., Маркова О. М., Семеріков С. О. // Моделювання в освіті: Стан. Проблеми. Перспективи : монографія / Міністерство освіти і науки України, ДВНЗ «Криворізький державний педагогічний університет», Черкаський національний університет імені Богдана Хмельницького, Південноукраїнський національний педагогічний університет імені К. Д. Ушинського ; за заг. ред. Соловйова В. М. – Черкаси : Брама, видавець Вовчок О.Ю., 2017. – С. 130-142.

**Відомості про апробацію результатів дисертації О. М. Маркової на тему  
«Хмарні технології як засіб навчання основ математичної інформатики  
студентів технічних університетів»**

<b>Назва конференції, конгресу, симпозіуму, семінару, школи</b>	<b>Місце та дата проведення</b>	<b>Форма участі</b>
Міжнародна науково-практична конференція «Науково-методичні засади управління якістю освіти в педагогічних вищих навчальних закладах»	м. Київ, Національний педагогічний університет імені М. П. Драгоманова, 28 травня 2009 р.	очна
VII Міжнародна науково-технічна конференція «Новітні комп'ютерні технології»	м. Севастополь, Державний науково-дослідний інститут автоматизованих систем у будівництві, 15-18 вересня 2009 р.	заочна
VII Всеукраїнська науково-практична конференція «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2010)	м. Черкаси, Черкаський державний технологічний університет, 4-6 травня 2010 р.	заочна
VIII Міжнародна науково-технічна конференція «Новітні комп'ютерні технології НОКОТЕ'2010»	м. Севастополь, Державний науково-дослідний інститут автоматизованих систем у будівництві, 14-17 вересня 2010 р.	заочна
IV Всеукраїнська конференція аспірантів, студентів та молодих вчених «Комп'ютерні інтелектуальні системи та мережі»	м. Кривий Ріг, Криворізький технічний університет, 23-25 березня 2011 р.	очна
IX Міжнародна науково-технічна конференція «Новітні комп'ютерні технології НОКОТЕ'2011»	м. Севастополь, Державний науково-дослідний інститут автоматизованих систем у будівництві, 13-16 вересня 2011 р.	заочна
Міжнародна науково-практична конференція «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2012)	м. Черкаси, Черкаський державний технологічний університет, 25-27 квітня 2012 р.	заочна
X Міжнародна науково-технічна конференція «Новітні комп'ютерні технології НОКОТЕ'2012»	м. Севастополь, Державний науково-дослідний інститут автоматизованих систем у будівництві, 11-14 вересня 2012 р.	заочна
Міжнародна науково-методична конференція «Розвиток інтелектуальних умінь і творчих здібностей учнів та студентів у процесі навчання дисциплін природничо-математичного циклу – ІТМ*плюс-2012»	м. Суми, Сумський державний педагогічний університет ім. А. С. Макаренка, 6-7 грудня 2012 р.	заочна
Всеукраїнський науково-методичний Інтернет-семінар «Хмарні технології в освіті»	м. Київ – Черкаси – Кривий Ріг, Криворізький національний університет, 21 грудня 2012 р.	заочна
VII Всеукраїнська WEB-конференція аспірантів, студентів та молодих вчених «Комп'ютерні інтелектуальні системи та мережі»	м. Кривий Ріг, ДВНЗ «Криворізький національний університет», 25-27 березня 2014 р.	заочна
Міжнародний семінар «Хмарні технології в освіті» (СТЕ'2014)	м. Київ – Кривий Ріг – Черкаси – Харків – Старобільськ – Херсон – Чейні, ДВНЗ «Криворізький національний	очна

Назва конференції, конгресу, симпозіуму, семінару, школи	Місце та дата проведення	Форма участі
	університет», 26 грудня 2014 р.	
Науково-практичний семінар «Хмарні технології в сучасному університеті» (ХТСУ-2015)	м. Черкаси, Черкаський державний технологічний університет, 24 березня 2015 р.	заочна
IX Всеукраїнська науково-практична WEB конференція аспірантів, студентів та молодих вчених «Комп'ютерні інтелектуальні системи та мережі»	м. Кривий Ріг, ДВНЗ «Криворізький національний університет», 22-24 березня 2016 р.	очна
VII Всеукраїнська науково-методична конференція «Комп'ютерне моделювання та інформаційні технології в освіті» (КМІТО'2016)	м. Кривий Ріг, ДВНЗ «Криворізький національний університет», 19-20 травня 2016 р.	очна
X Всеукраїнська науково-методична конференція «Комп'ютерне моделювання та інформаційні технології в освіті» (КМІТО'2017)	м. Кривий Ріг, Криворізький державний педагогічний університет, 27-28 квітня 2017 р.	очна
5th Workshop on Cloud Technologies in Education (CTE 2017)	м. Кривий Ріг, ДВНЗ «Криворізький національний університет», 28 квітня 2017 р.	очна
14th International Conference on ICT in Education, Research, and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI 2018)	м. Київ, Інститут післядипломної освіти Київського національного університету імені Тараса Шевченка, 14-17 травня 2018 р.	очна

## Додаток Е

### Список вищих навчальних закладів та установ, у яких упроваджено результати дослідження

1. Криворізький металургійний факультет Національної металургійної академії України
2. Криворізький технічний університет
3. Запорізький інститут економіки та інформаційних технологій
4. ДВНЗ «Криворізький національний університет»
5. Криворізький коледж Національного авіаційного університет
6. Дніпровський національний університет імені Олеся Гончара
7. Житомирський державний технологічний університет



МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ



НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ

NATIONAL  
METALLURGICAL  
ACADEMY OF UKRAINEКРИВОРІЗЬКИЙ  
МЕТАЛУРГІЙНИЙ  
ФАКУЛЬТЕТ

Україна, 50006 м. Кривий Ріг, вул. Революційна, 5  
Телефон 92-77-39, 92-77-65; Факс 92-79-20  
Код ЗКПО 26236875, МФО 805012,  
р/р 35221001002356 в ГУДКУ у Дніпропетровській області

02. 12. 2011 № 493  
На № \_\_\_\_\_  
Г \_\_\_\_\_ 7

**ДОВІДКА**

про впровадження результатів дисертаційного дослідження  
*Туравініної Оксани Миколаївни*  
на Криворізькому металургійному факультеті  
Національної металургійної академії України

Починаючи з 2009/2010 н.р., результати дисертаційного дослідження О. М. Туравініної впроваджуються у навчальний процес кафедри комп'ютерних систем автоматизованого управління електроприводом Криворізького металургійного факультету Національної металургійної академії України у процесі навчання дисциплін «Інформатика», «Системи автоматизованого управління електроприводом» на спеціальності «Електромеханічні системи автоматизації та електропривод».

Запропонований О. М. Туравініною підхід до навчання математичної інформатики із використанням засобів хмарних обчислень реалізовано у програмно-апаратному комплексі підтримки хмарних обчислень на основі технології Ulteo, доступним у локальній мережі факультету.

Результати впровадження елементів методики навчання математичної інформатики з використанням хмарних обчислень, розробленої О. М. Туравініною, показали її ефективність у процесі навчання студентів комп'ютерних спеціальностей технічних університетів.

Декан Криворізького металургійного факультету  
Національної металургійної академії України

д.т.н., професор



В.Й. Засельський

## ДОВІДКА

про впровадження результатів дисертаційного дослідження  
Туравініної Оксани Миколаївни у Криворізькому технічному університеті

В процесі роботи на посаді старшого викладача кафедри комп'ютерних систем та мереж Туравініна Оксана Миколаївни у 2008–2011 рр. впровадила у навчальний процес за спеціальністю «Комп'ютерні системи та мережі» авторську методику навчання основ математичної інформатики студентів технічних університетів засобами хмарних обчислень. Зміст навчання передбачає вивчення основних моделей, методів і алгоритмів розв'язування задач, що виникають у сфері інтелектуалізації інформаційних систем, а також розгляд проблем використання інформаційних, зокрема математичних, моделей та інформаційних технологій для їх дослідження.

Провідними засобами навчання математичної інформатики обрані системи хмарних обчислень Sage та Open Virtual Desktop, застосування яких гадає можливість інтеграції аудиторної та самостійної роботи студентів у безперервний навчальний процес.

Розроблені О. М. Туравініною методичні засади використання засобу хмарних обчислень Sage використовуються студентами в процесі підготовки курсових та кваліфікаційних робіт за освітньо-кваліфікаційним рівнем «бакалавр».

Матеріали навчального посібника, що розробляється О. М. Туравініною, також використовуються викладачами кафедри комп'ютерних систем та мереж у процесі навчання інформатичних дисциплін.

Вищезазначені матеріали є складовою частиною дисертаційного дослідження Туравініної О.М.

Завідувач кафедри КСМ  
д.т.н., доцент

Проректор з наукової роботи  
д.т.н., професор



А. І. Купін

В. Д. Сидоренко



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

вул. Кіяшка, 16-б м. Запоріжжя  
Код ЄДРПУ 22115979

Тел. / Факс: (061) 239-90-01

№ 28  
від «2» березня 2012 р

**ДОВІДКА**

про впровадження методики навчання основ математичної інформатики з використанням хмарних технологій використання, розробленої  
*Туравініною Оксаною Миколаївною,*  
у Запорізькому інституті економіки та інформаційних технологій

Впровадження розробленої О. М. Туравініною методики навчання основ математичної інформатики з використанням хмарних технологій використання здійснювалось у процесі навчання дисциплін «Теорія інформації та кодування», «Алгоритми та методи обчислень», «Вища математика», «Комп'ютерні системи штучного інтелекту», «Прикладна теорія цифрових автоматів» студентів 1–4 курсів спеціальності «6.0500102. Комп'ютерні системи та мережі».

Визначальною особливістю розробленої О. М. Туравініною методики навчання є застосування засобів хмарних технологій, що створює умови для підвищення технологічної мобільності, та фундаментальний зміст навчання математичної інформатики, що сприяє фундаменталізації підготовки бакалаврів з комп'ютерної інженерії.

Ректор



Г. В. Туровцев



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
 Державний вищий навчальний заклад  
 «КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»

вул. Віталія Магусевича, 11, м. Кривий Ріг, Дніпропетровська обл., 50027, тел. (056) 409-06-06, факс (0564) 74-52-57  
 E-mail: knu@knu.edu.ua Код ЄДРПОУ 37664469

від 04.05.18 № 01/10-01/2018

На № \_\_\_\_\_ від \_\_\_\_\_

ДОВІДКА

про впровадження результатів дисертаційного дослідження  
 Маркової Оксани Миколаївни  
 у навчальний процес кафедри комп'ютерних систем та мереж  
 ДВНЗ «Криворізький національний університет»

У процесі роботи на кафедрі комп'ютерних систем та мереж ДВНЗ «Криворізький національний університет» О. М. Марковою було впроваджено результати дисертаційного дослідження на тему «Хмарні технології як засіб навчання основ математичної інформатики студентів технічних університетів», затверджену на засіданні Вченої ради ДВНЗ «Криворізький національний університет» (протокол № 2 від 31 жовтня 2017 року).

Зокрема, О. М. Марковою були дібрані засоби хмарних технологій навчання математичної інформатики, використання яких сприяє формуванню дослідницьких виробничих функцій майбутніх фахівців з інформаційних технологій – хмаро орієнтовані програмно-методичні матеріали, тестові системи, тренажери, практикуми, дидактичні демонстраційні матеріали, навчальні посібники та підручники, електронні довідники, системи підтримки навчання, навчальні лабораторії, системи комп'ютерної математики, середовища програмування, додаткові науково-навчальні матеріали, електронна пошта, засоби аудіо- та відеозв'язку, операційні системи, засоби зберігання даних, текстові процесори, табличні процесори, засоби підготовки презентацій, системи управління базами даних, додаткові хмаро орієнтовані компоненти. Продемонстровано, що найбільш значущим засобом хмарних технологій навчання основ математичної інформатики є універсальні хмаро орієнтовані системи комп'ютерної математики, такі як CoCalc, що інтегрують у собі більшість виокремлених засобів.

Ефективність розробленої О. М. Марковою методики була підтверджена експериментально. Окремо слід відмітити ефективність використання запропонованих засобів хмарних технологій для підтримки навчально-дослідницької діяльності студентів під час виконання курсових проектів, студентських наукових робіт, випускових робіт бакалавра тощо.

Завідувач кафедри комп'ютерних систем та мереж  
 д. т. н., професор

Проректор з наукової роботи  
 д. т. н., професор



А. І. Купін

В. С. Морзун



## МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Криворізький коледж

Національного авіаційного університету

50045 м. Кривий Ріг, вул. Туполева, 1, тел.27-56-51, факс 90-01-45

### ДОВІДКА

про впровадження результатів дисертаційного дослідження

*Маркової Оксани Миколаївни*

у навчальний процес Криворізького коледжу Національного авіаційного університету

Розроблені в процесі дисертаційного дослідження О. М. Маркової на тему «Хмарні технології як засіб навчання основ математичної інформатики студентів технічних університетів» методичні матеріали використовуються викладачами відділення «Комп'ютерна і програмна інженерія» у процесі навчання студентів спеціальності 123 «Комп'ютерна інженерія» за освітньо-кваліфікаційним рівнем «молодший спеціаліст» дисциплін, що відносяться до блоку математичної інформатики: «Комп'ютерна логіка», «Алгоритми та методи обчислень», «Захист інформації у комп'ютерних системах ЦА».

Крім того, висвітлені в дослідженні моделі використання та засоби хмарних технологій (насамперед – сервіси Google Apps за моделлю SaaS) широко використовуються й у навчанні інших дисциплін (як фундаментальних, так і фахових).

Викладачі відділення відзначають високий науково-методичний рівень розробок з використання хмарних технологій у підготовці майбутніх фахівців з інформаційних технологій.

Завідувач відділення  
«Комп'ютерна і програмна інженерія»

С. С. Терьошина

Начальник Криворізького коледжу  
Національного авіаційного університету  
д. т. н., професор



А. О. Андрусевич

## ДОВІДКА

про впровадження результатів дисертаційного дослідження

*Маркової Оксани Миколаївни*

у навчальний процес кафедри математичного забезпечення електронних  
обчислювальних машин Дніпровського національного університету  
імені Олеся Гончара

Впровадження розробленої О. М. Марковою методики використання хмарних технологій як засобу навчання математичної інформатики здійснювалось у процесі навчання дисциплін «Безпека програм та даних», «Програмування Інтернет», «Організація комп'ютерних мереж», «Інформаційні технології обробки даних», «Аналіз алгоритмів» студентів спеціальності 121 «Інженерія програмного забезпечення».

Запропоновані та методично обґрунтовані дисертантом засоби хмарних технологій (насамперед мережних систем комп'ютерної математики) також можуть бути використані для підтримки навчальних і наукових досліджень в інформаційних технологіях обробки статистичних даних, нейромережових технологіях аналізу даних та ін. на третьому рівні вищої освіти. Для студентів заочної форми навчання розроблена методика надає можливість ефективної організації їх самостійної навчально-дослідницької роботи в середовищі дистанційного навчання.

Декан факультету прикладної математики  
Дніпровського національного університету  
імені О.Гончара, член-кореспондент НАН  
України, д-р фіз.-мат. наук, професор



О.М.Кісельова

Завідувач кафедри математичного забезпечення  
ЕОМ Дніпровського національного університету  
імені О.Гончара, д-р техн. наук, професор



О. Г. Байбуз



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ**  
 Ministry of Education and Science of Ukraine, Zhytomyr State Technological University

вул. Чуднівська, 103, м. Житомир, 10005  
 Chudnivska str., 103, Zhytomyr, 10005, Ukraine  
 Tel./fax: (0412) 24-14-22, 24-14-23, e-mail: rector@ztu.edu.ua, https://ztu.edu.ua, код ЄДРПОУ 05407870

СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ ВІДПОВІДАЄ ДСТУ ISO 9001:2015  
 QUALITY MANAGEMENT SYSTEM ISO 9001:2015

Від 10 Вересня 2018 № 44-22.07/1407  
 На № \_\_\_\_\_ від \_\_\_\_\_

**ДОВІДКА**

про впровадження результатів дисертаційного дослідження  
 Маркової Оксани Миколаївни «Хмарні технології як засіб навчання основ математичної  
 інформатики студентів технічних університетів»  
 у навчальний процес кафедри комп'ютерних наук  
 Житомирського державного технологічного університету

У навчальний процес кафедри впроваджено окремі елементи розробленої О. М. Марковою методики використання хмарних технологій як засобу навчання основ математичної інформатики студентів технічних університетів. Зокрема, під час викладання таких дисциплін, як «BigData та інтелектуальний аналіз даних», «Паралельні та розподілені системи», «Комп'ютерна дискретна математика», «Системи штучного інтелекту», «Хмарні технології» використовуються:

– зміст навчання спецкурсу «Основи математичної інформатики», спрямований на формування дослідницьких виробничих функцій майбутніх фахівців з інформаційних технологій;

– засоби хмарних технологій навчання основ математичної інформатики: хмаро орієнтовані навчально-методичні комплекси (програмно-методичні матеріали, засоби оцінювання навчальних досягнень – тестові системи та тренажери, практикуми, навчально-методичні матеріали – дидактичні демонстраційні матеріали, навчальні посібники та підручники, електронні довідники), хмаро орієнтовані системи підтримки навчання, хмаро орієнтовані навчальні лабораторії (зокрема, середовища моделювання) та хмаро орієнтовані предметні середовища (системи комп'ютерної математики та середовища програмування), хмаро орієнтовані додаткові науково-навчальні матеріали, хмаро орієнтовані комунікаційні засоби (електронна пошта, засоби аудіо та відеозв'язку), хмаро орієнтовані операційні системи, хмаро орієнтовані засоби зберігання даних та хмаро орієнтовані офісні пакети (текстові та табличні процесори, засоби підготовки презентацій, системи управління базами даних та додаткові хмаро орієнтовані компоненти).

Результати дослідження можуть бути використані у процесі підготовки та перепідготовки майбутніх фахівців з інформаційних технологій, зокрема з інженерії програмного забезпечення та комп'ютерної інженерії.

З.о. завідувач кафедри комп'ютерних наук

Т. Н., доцент

Проректор з науково-педагогічної роботи

Т. Н., доцент



І. І. Сугоня

А. В. Мороз