

Ю.В. Прокоп, О.Г. Трофименко, О.В. Задерейко

## АНАЛІЗ ПІДХОДІВ У ВИКЛАДАННІ ПОЧАТКОВОГО КУРСУ ПРОГРАМУВАННЯ В УНІВЕРСИТЕТАХ

*Анотація. Важливість результативного вивчення початкового курсу програмування (CS1), з одного боку, і потреба враховувати реалії вимог, що ставлять перед ІТ-фахівцями роботодавці сьогодні, з іншого боку, спонукають університети гнучко змінювати освітні компоненти, коригувати навчальні плани і вміст дисциплін. Актуальність пошуку підходів, які позитивно впливають на результативність навчання основам програмування, зумовлена складністю цього курсу для студентів і високим рівнем відсіву вже на першій сесії. В роботі проаналізовано підходи передових зарубіжних університетів у викладанні курсу з основ програмування CS1. Проведено співставлення з тенденціями сучасного ринку ІТ-праці на предмет потреб коригування вмісту початкового курсу програмування у закладах вищої освіти.*

*Ключові слова: початковий курс програмування, CS1, мова програмування, ІТ-освіта, алгоритмізація та програмування.*

**Постановка проблеми.** Сьогодні для ІТ-освіти, і як наслідок для ІТ-галузі, характерні проблеми якості освіти і дефіцит спеціалістів водночас. Здобувачі ІТ-освіти мають багато вчитися і під час навчання, і під час професійної діяльності, щоб встигати за змінами в ІТ-індустрії загалом та новими трендами у вузькоспеціалізованих напрямках. Хоча популярність ІТ-освіти зростає, майбутнім студентам варто зважати на потребу певних зусиль і самостійного навчання щодо вдосконалення теоретичних і практичних навичок з розробки програмного забезпечення.

Нині в різних сферах програмної інженерії використовується цілий спектр різних мов програмування, які відрізняються функціоналом, швидкістю, продуктивністю, динамічною типізацією, простотою у навчанні і сприйнятті, наявністю спеціалізованих бібліотек, спрямованістю на оброблення певного роду даних (наприклад, Big Data, машинне навчання) та/або на кодування певного сегмента (наприклад, бекенд, вебфронтенд, мобільна розробка). При чому щорічно кількість робочих місць на ринку ІТ-праці неухильно зростає для

більшості мов. При підготовці IT-фахівців і програмістів зокрема важливим питанням є вибір першої мови програмування для вивчення. Адже від цього багато в чому залежить успішність подальшої IT-кар'єри. Від простоти сприйняття на початку навчання, розуміння базових програмних конструкцій залежить успішність засвоєння концепцій ООП і більш складних конструкцій програмування. З іншого боку, стрімкі реалії сьогодення спонукають шукачів роботи в IT не витрачати зайвого часу на вивчення сильних сторін різних мов програмування, а зосередитись на якійсь одній.

У закладах вищої освіти (ЗВО) нашої країни при підготовці здобувачів галузі 12 «Інформаційні технології», зокрема спеціальностей 121, 122, 126, на першому році навчання базовою дисципліною з вивчення основ програмування переважно є «Алгоритмізація та програмування» (АП). У більшості країн світу (країн ЄС, США, Австралії та ін.) в університетах цей курс називають CS1. Знання і навички, що здобувають студенти в процесі вивчення на цьому етапі, закладають базу для подальшого навчання програмуванню в ЗВО і майбутній професійній діяльності. Успішність вивчення АП впливає на засвоєння інших предметів курсу («Алгоритми та структури даних» (АСД), «Об'єктно-орієнтоване програмування» (ООП) та інших, що складають CS2), пов'язаних із програмуванням, і в значній мірі впливає на становлення програміста.

Важливість результативного вивчення початкового курсу програмування, з одного боку, і потреба враховувати реалії вимог, що ставлять перед IT-фахівцями роботодавці сьогодні, з іншого боку, спонукають ЗВО гнучко змінювати освітні компоненти, коригувати навчальні плани і вміст дисциплін. Серед питань, що викликають у цьому сенсі найбільшу дискусію між викладачами не лише України, а й усього світу, є вибір оптимальної мови програмування для початкового курсу вивчення програмування (CS1) та пошук способів підвищення успішності студентів такого курсу.

**Аналіз останніх досліджень і публікацій.** Дебати про підходи до викладання в університетах курсу CS1 не припиняються упродовж десятиліть. Дослідники відзначають складнощі у сприйнятті зазначеного предмета для студентів і високі показники незадовільних оцінок на іспитах [1]. Автори робіт [2–4] відзначають низькі показники успішності та високий рівень демотивації і навіть відсіву студентів на вступному курсі програмування. Дослідження [5], присвячене аналізу упродовж декількох років успішності з іспиту по курсу CS1, виявило, що третина здобувачів не в змозі його скласти. В роботі [1] відзначається, що викладачі заради зменшення незадовільних

результатів змушені спрощувати курс і знижувати вимоги до студентів. Такий підхід позитивно впливає на статистику невдач, проте завдає шкоди сильним студентам.

Серед негативних факторів з навчання основам програмування автори [1] називають відсутність у першокурсників навіть мінімальних знань і навичок програмування. Серед позитивних факторів в роботі [3] називають сильну мотивацію, особливо внутрішню, пов'язану з реальним інтересом до предмета, а також набір дидактичних процесів і методів, які допомагають засвоїти основні концепції програмування. Автори [6] вважають, що викладачі, які добре знаються на предметі викладання, з більшою ймовірністю будуть випускати успішних учнів.

Викладачі програмування в університетах по всьому світу продовжують шукати способи підвищення результативності курсу CS1, зокрема, змінюючи мову програмування і вводячи в навчальну програму нові класи задач [2, 7–10]. При цьому ефективність запропонованих рішень (зокрема, вибір мови) залежить від регіональної специфіки і традицій ІТ-освіти. Різні підходи і специфічні аспекти у пошуках рішення проблем, пов'язаних з викладанням курсу CS1, свідчать про актуальність цієї тематики і потребу подальшого дослідження.

**Мета дослідження** полягає в аналізі підходів закордонних університетів до викладання курсу з основ програмування, співставлення їх з тенденціями сучасного ринку ІТ-праці та дослідження потреб коригування вмісту дисципліни АП з урахуванням специфіки української вищої ІТ-освіти.

**Викладення основного матеріалу дослідження.** Існує ціла низка факторів, які впливають на процес навчання основам програмування та на ефективність його успішного засвоєння.

Цікавим і продуктивним підходом є залучення студентів, починаючи вже з першого курсу, до розробки реальних програмних проектів, які розробляються на кафедрах в проектних групах разом із фахівцями програмних компаній [11]. Це, з одного боку, наочно показує студентам практичні результати їх навчального процесу, а з іншого мотивує їх до вивчення програмування. Саме такий підхід успішно застосовується на кафедрах інформаційних технологій в ОНАЗ ім. О.С. Попова та НУ «ОЮА» з фахівцями компанії «KeepSolid». Також гарні результати показує підхід, коли в навчанні при розгляді прикладів програм проводяться паралелі щодо того, де реально на практиці можна застосувати такі програмні конструкції. Досягнення результатів вивчення тої чи іншої

теми потребує раціонального застосування способів дій, методів і засобів навчання. Пріоритетний вибір методики навчання та їх поєднання належить викладачеві. Тому багато в чому ефективність засвоєння матеріалу дисципліни залежить від майстерності, досвіду та професіоналізму викладача.

Для подолання складнощів у розумінні основних програмних конструкцій можна застосовувати ігровий підхід. Наприклад, такий підхід ефективно спрацьовує при закріпленні пройденого матеріалу. Для того, щоб зацікавити, варто створювати ігрові ситуації, інтерактивні вправи, активно використовувати мультимедійної засоби. Існують успішні методики [6, 12–14] навчання програмуванню, орієнтованого на розробку студентами ігор або розв'язанні задач, поданих в ігровій формі.

Існує підхід [1, 6, 12], спрямований на підвищення ефективності вивчення курсу CS1, який пропонує використовувати візуальну мову Scratch, принаймні в першій частині курсу з основ програмування. Графічна наочність Scratch полегшує сприйняття програмних конструкцій. Проте ця мова немає зв'язку з реальними мовами програмування, які далі вивчаються в інших дисциплінах і використовуються в професії, що викликає певні труднощі.

Дослідження [5, 15] показують, що вибір мови програмування майже не впливає на результативність і успішність здачі іспиту по CS1. Не зважаючи на те, чи викладається курс з основ програмування синтаксично простою мовою Python, чи то синтаксично складною мовою C++, результати успішності приблизно однакові. Тоді варто вибір початкової мови зосередити на тому, яка саме мова знадобиться надалі у професійній діяльності, що буде додатковим засобом мотивації до вивчення цього курсу. Так, студенти неохоче вчать мови програмування, які не використовуватимуть у своїй подальшій професії через непопулярність їх на ринку праці в конкретній країні.

Щодо реальних мов програмування, затребуваних на ринку IT-праці, найбільш популярними в Україні та світі є мови JavaScript, Java, Python, PHP, C# і C++. Для цього були порівняні рейтинги мов програмування відповідно до індексів PYPL, TIOBE, GitHub та StackOverflow [16]. Відповідно до рейтингу мов програмування DOU 2021 року [17] до десятки найбільш використовуваних у професії нині в Україні потрапили: JavaScript (18%), Java (15%), C# (14%), Python (12%), PHP (10%), TypeScript (7%), C++ (4.6%), Swift (3.1%), Ruby (3%), Kotlin (2.6%).

Суттєво, що кожна з мов програмування має своє специфічне призначення і свою сферу розробки програмного забезпечення. Так, головними мовами

бекенду є Java, C# та PHP. Частково у бекенді застосовують JavaScript, TypeScript, Ruby, Go, Scala та C++. Фактично весь фронтенд пишуть мовами JavaScript і TypeScript. У мобільній розробці основними мовами є Kotlin і Swift, а також зберігає популярність відносно стара мова програмування Java. Для оброблення масивів даних (Big Data та машинне навчання) беззаперечним лідером є Python.

Затребуваність мови на ринку IT-праці є одним із головних факторів для вибору першої мови розробки, оскільки суттєво впливає на мотивацію студентів. Проте від вибору першої мови залежить і культура програмування. Питання вибору першої мови програмування багато років викликає палкі дискусії. Згідно з дослідженнями [7–9] більшість університетів у курсі CS1 використовують одну з мов: Java (40%), Python (30%), C++ (17%), C (8%), причому популярність мови Python зростає за рахунок зниження популярності Java. Решта мов (C#, Matlab і Ruby) значно менш популярні.

У США 42% університетів в курсі CS1 використовують мову Java, мову Python – 26%, C/C++ – 24%, C# – лише 0.65%, ще приблизно 7% складають інші мови [2]. В Австралії найпоширенішими є Java і Python (по 31.25% кожна), C (16.7%), C# (10.4%), Visual Basic (4%), решта – інші мови. В країнах Європи найбільш використовуваною мовою є C (від 30.6% до 45.7% в різних європейських країнах), за нею йде C++ (15–23%), Java (8.3–20.7%), Pascal (7.7%) і Python (5.6%) [10, 18]. Тобто у виборі мови виявляється виражена регіональна специфіка.

Останнім часом великої популярності завдяки простоті синтаксису набула мова Python, проте чи варто починати вивчення програмування саме з цієї мови? Порівняння результатів [14, 15, 19] застосування мов Python та C++ виявило, що студенти, які у вступному курсі програмування вивчали Python, мали більш високі показники складнощів, ніж студенти, які вибрали C++. Перевагу C++ у виборі між мовами Python та C++ для курсу CS1 обґрунтовано у дослідженнях [14, 20–22].

Серед найбільш актуальних аргументів щодо вибору мови програмування для курсу CS1 є: простота синтаксису, популярність на ринку праці, наявність всіх базових конструкцій, підтримка мови різними середовищами програмування, в тому числі наявність онлайн-компіляторів, простота встановлення IDE, педагогічні переваги, наявність онлайн-спільноти для допомоги, максимальна сумісність з іншими мовами програмування.

Важливим питанням є легкість переходу від одного курсу до іншого, тобто

від CS1 до CS2. Дослідження [19, 23] виявили, що у більшості студентів, які вивчали Python або Java у курсі CS1, виникли суттєві складнощі у розумінні динамічних структур даних. Автори [22] пропонують у вступному курсі програмування використовувати C++, а далі в ООП – мову з тієї ж групи (C++, C#, Java), вважаючи, що приналежність мов до одного сімейства сприяє засвоєнню матеріалу.

Грунтуючись на рейтингу затребуваних в Україні мов програмування, аналізі досвіду зарубіжних університетів, необхідності встановлення взаємозв'язку між дисциплінами і специфіці курсів, доречною схемою використання мов програмування є: C/C++ в курсі АП і Java/C# в курсі ООП.

Вибір мови програмування для вивчення у ЗВО може бути зроблений на базі наявності або можливості швидкої підготовки навчально-методичної бази і наявності викладачів, які добре володіють цією мовою. Крім того, аргументом для вибору може бути опитування, проведене серед студентів перед початком курсу.

Простота навчання, потенційна зарплата, довголіття і попит – все це фактори, які майбутні розробники враховують, шукаючи першу для вивчення мову програмування. Аудиторія бажаючих навчитись програмувати зростає, незважаючи на вік. За умов глобальної пандемії COVID, коли багато людей залишилися без роботи, вони прагнуть підготуватися до нової або більш вигідної роботи в галузі розробки програмного забезпечення.

**Висновки.** Універсального рішення проблем, пов'язаних з викладанням початкового курсу програмування в ЗВО не існує. Викладачі програмування в університетах по всьому світу продовжують шукати способи підвищення результативності цього курсу, зокрема, змінюючи мову програмування і вводячи в навчальну програму нові класи задач. При цьому ефективність запропонованих рішень (зокрема, вибір мови) залежить від регіональної специфіки і традицій ІТ-освіти. Застосування різноманітних освітніх технологій: практико-орієнтованої, групової, проблемної, ігрової, тренінгової дозволяє раціональним шляхом досягти поставленої мети.

#### **ЛІТЕРАТУРА / ЛИТЕРАТУРА**

1. Undergraduate Programming Courses, Students' Perception and Success / Krpan D., Mladenovic S., Rosić M. // Procedia - Social and Behavioral Sciences. – 2018. – Vol. 174. – P. 3868–3872. DOI: 10.1016/j.sbspro.2015.01.1126.
2. Sobral S. 30 years of CS1: Programming Languages Evolution / S. Sobral // ICERI-2019 Proceedings. – 2019. – P. 9197–9205. DOI: 10.21125/iceri.2019.2214.

3. Andrzejewska M. Przyczyny niepowodzeń edukacyjnych studentów informatyki na wstępnych kursach programowania / M Andrzejewska. // Edukacja, Technika, Informatyka. – 2018. – Vol. 4. – P. 211–217. DOI: 10.15584/eti.2018.4.29
4. EasyCoding - Methodology to Support Programming Learning. / Almeida M., Alves L., Pereira M., Barbosa G. // First International Computer Programming Education Conference ICPEC-2020. DOI: 10.4230/OASICS.ICPEC.2020.1
5. Watson C. Failure rates in introductory programming revisited. / Watson C., Li F. // ITiCSE '14: Proceedings of the 2014 conference on Innovation & technology in computer science education. – New York, NY, USA, 2014. P. 39–44. DOI: 10.1145/2591708.2591749.
6. Using Games to Help Novices Embrace Programming: From Elementary to Higher Education / Mladenovic S., Krpan D., Mladenović M. // International Journal of Engineering Education. – 2016. – Vol. 32. – P. 521-531.
7. CS2 and the Impact of Programming Language Choice / Siegfried R., Herbert-Berger K., Siegfried J. // 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). – 2020. – P. 1353. DOI: 10.1145/3328778.3372644.
8. A Longitudinal Analysis of the Reid List of First Programming Languages / Siegfried R., Siegfried J., Alexandro G. // Information Systems Education Journal. – 2016. – Vol. 14(6). – P. 47–54.
9. Recent Studies About Teaching Algorithms (CS1) and Data Structures (CS2) for Computer Science Students / Silva D., Aguiar R., Dvconlo D., Silla C. // IEEE Frontiers in Education Conference (FIE). – 2019. – P. 1–8. DOI: 10.1109/FIE43999.2019.9028702.
10. Aleksić V. Introductory Programming Subject in European Higher Education / V. Aleksić, M. Ivanović // Informatics in Education. – 2016. – Vol. 15, No. 2. – P. 163–182. DOI: 10.15388/infedu.2016.09.
11. Using real projects as motivators in programming education / Konecki M., Lovrenčić S., Kaniški M. // 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). – Opatija, Croatia, 2016. – P. 883–886. DOI: 10.1109/MIPRO.2016.7522264.
12. Mediated transfer from visual to high-level programming language / Krpan D., Mladenović S., Zaharija G. // 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). – Opatija, 2017. – P. 800–805. DOI: 10.23919/MIPRO.2017.7973531.
13. Seng W. Y. Computer Game as Learning and Teaching Tool for Object Oriented Programming in Higher Education Institution / Wong Yoke Seng, Maizatul Hayati Mohamad Yatim // Procedia - Social and Behavioral Sciences. – 2014. – Vol. 123. – P.

215–224. DOI: 10.1016/j.sbspro.2014.01.1417.

14. Madeja M. Innovative Approaches in C Introductory Programming Courses / M. Madeja, J. Porubän // ICT in Education, Research, and Industrial Applications, 2019.

15. Python Versus C++: An Analysis of Student Struggle on Small Coding Exercises in Introductory Programming Courses / Alzahrani N., Vahid F., Edgcomb A., Nguyen K., Lysecky R. // 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). – 2018. – P. 86–91. DOI: 10.1145/3159450.3160586.

16. Prokop Y.V. A study of software development tools that are required in the job market in Ukraine and the world. / Prokop Y.V., Trofymenko O.G., Kapustin M.M. // Proceedings of the O.S. Popov ONAT. – 2018. – Vol. 2. – P. 101-108. DOI: 10.33243/2518-7139-2018-1-2-101-108.

17. Рейтинг мов програмування 2021. [Електронний ресурс] – Режим доступу: <https://dou.ua/lenta/articles/language-rating-jan-2021/>

18. An Analysis of Introductory Programming Courses at UK Universities / Murphy E., Crick T., Davenport J. // The Art, Science, and Engineering of Programming. – 2017. – Vol. 1(2). Issue 2. – DOI: 10.22152/programming-journal.org/2017/1/18.

19. Toward Predicting Success and Failure in CS2: A Mixed-Method Analysis / Layman L., Song Y., Guinn C. // ACM Southeast Conference (ACM SE '20). – New York, NY, USA, 2020. – P. 218–225. DOI: 10.1145/3374135.3385277.

20. Analysis of Student Misconceptions using Python as an Introductory Programming Language / Johnson F., McQuistin S., O'Donnell J. // 4th Conference on Computing Education Practice (CEP 2020). – New York, NY, USA, 2020. – Article 4. – P. 1–4. DOI: 10.1145/3372356.3372360.

21. Multivariate analysis when choosing the first programming language studied in universities / Prokop Yu., Trofimenko E., Zadereyko O., Loginova N., Gerganov M. // Advancing Society Through Applied Physics, Electrical and Computer Engineering. 2nd Ukraine Conference on Electrical and Computer Engineering (IEEE UKRCON-2019). – P. 1224-1228.

22. Damyanov I. Programming languages in undergraduate courses and in software industry in Bulgaria / Damyanov I., Borisova N. // International Journal of Pure and Applied Mathematics. – 2017. – Vol. 117, No. 2. – P. 271–278. DOI: 10.12732/ijpam.v117i2.3.

23. A snapshot of current practices in teaching the introductory programming sequence / Davies S., Polack-Wahl J., Anewalt K. // 42nd ACM technical symposium on Computer science education (SIGCSE '11). – New York, NY, USA, 2011. – P. 625–630. DOI: 10.1145/1953163.1953339.

## REFERENCES

1. Undergraduate Programming Courses, Students' Perception and Success / Krpan D., Mladenovic S., Rosić M. // *Procedia - Social and Behavioral Sciences*. – 2018. – Vol. 174. – P. 3868–3872. DOI: 10.1016/j.sbspro.2015.01.1126.
2. Sobral S. 30 years of CS1: Programming Languages Evolution / S. Sobral // *ICERI-2019 Proceedings*. – 2019. – P. 9197–9205. DOI: 10.21125/iceri.2019.2214.
3. Andrzejewska M. Przyczyny niepowodzeń edukacyjnych studentów informatyki na wstępnych kursach programowania / M Andrzejewska. // *Edukacja, Technika, Informatyka*. – 2018. – Vol. 4. – P. 211–217. DOI: 10.15584/eti.2018.4.29
4. EasyCoding - Methodology to Support Programming Learning. / Almeida M., Alves L., Pereira M., Barbosa G. // *First International Computer Programming Education Conference ICPEC-2020*. DOI: 10.4230/OASlcs.ICPEC.2020.1
5. Watson C. Failure rates in introductory programming revisited. / Watson C., Li F. // *ITiCSE '14: Proceedings of the 2014 conference on Innovation & technology in computer science education*. – New York, NY, USA, 2014. P. 39–44. DOI: 10.1145/2591708.2591749.
6. Using Games to Help Novices Embrace Programming: From Elementary to Higher Education / Mladenovic S., Krpan D., Mladenović M. // *International Journal of Engineering Education*. – 2016. – Vol. 32. – P. 521–531.
7. CS2 and the Impact of Programming Language Choice / Siegfried R., Herbert-Berger K., Siegfried J. // *51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. – New York, NY, USA, 2020. – P. 1353. DOI: 10.1145/3328778.3372644.
8. A Longitudinal Analysis of the Reid List of First Programming Languages / Siegfried R., Siegfried J., Alexandro G. // *Information Systems Education Journal*. – 2016. – Vol. 14(6). – P. 47–54.
9. Recent Studies About Teaching Algorithms (CS1) and Data Structures (CS2) for Computer Science Students / Silva D., Aguiar R., Dvconlo D., Silla C. // *IEEE Frontiers in Education Conference (FIE)*. – Covington, KY, USA, 2019. – P. 1–8. DOI: 10.1109/FIE43999.2019.9028702.
10. Aleksić V. Introductory Programming Subject in European Higher Education / V. Aleksić, M. Ivanović // *Informatics in Education*. – 2016. – Vol. 15, No. 2. – P. 163–182. DOI: 10.15388/infedu.2016.09.
11. Using real projects as motivators in programming education / Konecki M., Lovrenčić S., Kaniški M. // *39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. – Opatija, Croa-

tia, 2016. – P. 883–886. DOI: 10.1109/MIPRO.2016.7522264.

12. Mediated transfer from visual to high-level programming language / Krpan D., Mladenović S., Zaharija G. // 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). – Opatija, 2017. – P. 800–805. DOI: 10.23919/MIPRO.2017.7973531.

13. Seng W. Y. Computer Game as Learning and Teaching Tool for Object Oriented Programming in Higher Education Institution / Wong Yoke Seng, Maizatul Hayati Mohamad Yatim // Procedia - Social and Behavioral Sciences. – 2014. – Vol. 123. – P. 215–224. DOI: 10.1016/j.sbspro.2014.01.1417.

14. Madeja M. Innovative Approaches in C Introductory Programming Courses / M. Madeja, J. Porubän // ICT in Education, Research, and Industrial Applications, 2019.

15. Python Versus C++: An Analysis of Student Struggle on Small Coding Exercises in Introductory Programming Courses / Alzahrani N., Vahid F., Edgcomb A., Nguyen K., Lysecky R. // 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). – USA, 2018. – P. 86–91. DOI: 10.1145/3159450.3160586.

16. Prokop Y.V. A study of software development tools that are required in the job market in Ukraine and the world. / Prokop Y.V., Trofymenko O.G., Kapustin M.M. // Proceedings of the O.S. Popov ONAT. – 2018. – Vol. 2. – P. 101-108. DOI: 10.33243/2518-7139-2018-1-2-101-108.

17. Rating of programming languages 2021. URL:  
<https://dou.ua/lenta/articles/language-rating-jan-2021/>

18. An Analysis of Introductory Programming Courses at UK Universities / Murphy E., Crick T., Davenport J. // The Art, Science, and Engineering of Programming. – 2017. – Vol. 1(2). – DOI: 10.22152/programming-journal.org/2017/1/18.

19. Toward Predicting Success and Failure in CS2: A Mixed-Method Analysis / Layman L., Song Y., Guinn C. // ACM Southeast Conference (ACM SE '20). – New York, NY, USA, 2020. – P. 218–225. DOI: 10.1145/3374135.3385277.

20. Analysis of Student Misconceptions using Python as an Introductory Programming Language / Johnson F., McQuistin S., O'Donnell J. // 4th Conference on Computing Education Practice (CEP 2020). – New York, NY, USA, 2020. – Article 4. – P. 1–4. DOI: 10.1145/3372356.3372360.

21. Multivariate analysis when choosing the first programming language studied in universities / Prokop Yu., Trofimenko E., Zadereyko O., Loginova N., Gerganov M. // Advancing Society Through Applied Physics, Electrical and Computer Engineering. 2nd Ukraine Conference on Electrical and Computer Engineering (IEEE UKRCON-2019). – P. 1224-1228.

22. Damyanov I. Programming languages in undergraduate courses and in software industry in Bulgaria / Damyanov I., Borisova N. // International Journal of Pure and Applied Mathematics. – 2017. – Vol. 117. – P. 271–278. DOI: 10.12732/ijpam.v117i2.3.

23. A snapshot of current practices in teaching the introductory programming sequence / Davies S., Polack-Wahl J., Anewalt K. // 42nd ACM technical symposium on Computer science education (SIGCSE '11). – New York, NY, USA, 2011. – P. 625–630. DOI: 10.1145/1953163.1953339.

Received 01.04.2021.

Accepted 05.04.2021.

### **Анализ подходов к преподаванию начального курса программирования в университетах**

*Важность результативного изучения начального курса программирования (CS1), с одной стороны, и потребность учитывать реалии требований, которые ставят перед ИТ-специалистами работодатели сегодня, с другой стороны, побуждают университеты гибко менять образовательные компоненты, корректировать учебные планы и содержание дисциплин. Актуальность поиска подходов, положительно влияющих на результативность обучения основам программирования, обусловлена сложностью этого курса для студентов и высоким уровнем отсева уже на первой сессии. В работе проанализированы подходы передовых зарубежных университетов к преподаванию курса по основам программирования CS1. Проведено сопоставление с тенденциями современного рынка ИТ-труда на предмет потребностей корректировки содержания начального курса программирования в учреждениях высшего образования.*

### **Analysis of approaches in teaching the introductory programming course in universities**

*The importance of the effective study of the initial programming course (CS1), on the one hand, and the need to take into account the requirements for IT professionals today, on the other hand, encourage universities to flexibly change educational components, adjust curricula and disciplines. The relevance of finding approaches that positively affect the effectiveness of learning the basics of programming, due to the complexity of this course for students and the high level of dropout in the first session.*

*The paper analyzes the approaches of foreign universities in teaching the CS1 course. The tendencies of the Ukrainian IT labour market concerning the need for adjustment of the content of the Algorithmization and programming course (Ukrainian analogue CS1) are investigated.*

*It is established that there is no universal solution to the problems associated with teaching the initial programming course in the universities. Important factors that positively affect the success of programming training are the motivation of students, a set of didactic processes and methods, as well as good knowledge of the subject by teachers. To increase the effectiveness of training, researchers propose to change the programming language used in the introductory programming course and to introduce new classes of tasks in the curriculum. The effectiveness of the solutions*

*depends on the regional specifics and traditions of IT education.*

*Considering the ranking of programming languages in demand in Ukraine, the need to establish a relationship between disciplines and specifics of courses, the appropriate scheme of using programming languages are C / C ++ in Algorithmization and programming course (CS1) and then Java / C # in Object-oriented programming course (CS2). Another solution is to collaborate with IT companies and to involve students in writing real programs.*

**Прокоп Юлія Віталіївна** - кандидат історичних наук, старший викладач кафедри інформаційних технологій Одеської національної академії зв'язку ім. О.С. Попова.

**Трофименко Олена Григорівна** - кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія».

**Задерейко Олександр Владиславович** - кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія».

**Прокоп Юлія Витальевна** - кандидат исторических наук, старший преподаватель кафедры информационных технологий Одесской национальной академии связи им. А.С. Попова.

**Трофименко Елена Григорьевна** - кандидат технических наук, доцент, доцент кафедры информационных технологий Национального университета «Одесская юридическая академия».

**Задерейко Александр Владиславович** - кандидат технических наук, доцент, доцент кафедры информационных технологий Национального университета «Одесская юридическая академия».

**Prokop Yuliya** - Candidate of Historical Sciences, Senior lecturer at the Department of Information Technology of the O.S. Popov Odessa National Academy of Telecommunications.

**Trofymenko Olena** - Candidate of Technical Sciences, Associate Professor, Associate Professor at the Department of Information Technologies of the National University "Odessa Law Academy".

**Zadereyko Olexander** - Candidate of Technical Sciences, Associate Professor, Associate Professor at the Department of Information Technologies of the National University "Odessa Law Academy".