

[0000-0002-6608-3668] **Ю. В. Прокоп**¹, канд. іст. наук,
ст. викладач кафедри інформаційних технологій,
e-mail: yulia13.prokop@gmail.com

[0000-0001-7626-0886] **О. Г. Трофименко**², канд. техн. наук, доцент,
доцент кафедри інформаційних технологій,
e-mail: egt@ukr.net

[0000-0002-2614-2109] **А. А. Толокнов**², асистент кафедри інформаційних технологій,
e-mail: arnoldovich1@gmail.com

[0000-0002-3987-9409] **Я. В. Дубовой**¹, магістр з кібербезпеки
e-mail: dubovoy97@gmail.com

¹Державний університет інтелектуальних технологій та зв'язку
вул. Кузнечна, 1, м. Одеса, 65029, Україна

²Національний університет «Одеська юридична академія»
Фонтанська дорога, 23, м. Одеса, 65009, Україна

КОМПЛЕКСНИЙ АНАЛІЗ ПІДХОДІВ ДО ВИКЛАДАННЯ КУРСІВ CS1 І CS2 В УНІВЕРСИТЕТАХ СВІТУ ТА УКРАЇНИ

Стаття присвячена пошуку способів покращення успішності при вивченні базових дисциплін з програмування на 1-му і 2-му курсах навчання бакалаврату при підготовці ІТ фахівців. Досліджено навчальні плани підготовки здобувачів вищої освіти у галузі І2 Інформаційні технології в українських вишах. Розглянуто змістове наповнення дисциплін «Алгоритмізація та програмування», «Алгоритми та структури даних», «Об'єктно-орієнтоване програмування» та аналогічних курсів CS1 і CS2 у західних вишах. З'ясовано, що ці курси є фундаментальними у підготовці якісних фахівців ІТ галузі, а тому важливо залучати передовий світовий досвід та найкращі підходи у викладанні для гнучкого корегування навчальних програм дисциплін, враховуючи сучасні вимоги до фахівців на ринку ІТ праці. Порівняння підходів до викладання зазначених курсів в українських вишах та за кордоном виявило певні розбіжності підходів до вибору мови програмування, що використовується у цих курсах, а також відмінності змістового наповнення, які не повною мірою пояснюються регіональною специфікою ринку ІТ праці і можуть сигналізувати про необхідність вдосконалення робочих програм. Результати дослідження дадуть змогу більш виважено і ґрунтовно підходити до змістового формування зазначених навчальних дисциплін з програмування, вибору підходів до мотивації здобувачів вищої освіти до навчання.

Ключові слова: мова програмування, ІТ освіта, алгоритмізація та програмування, алгоритми та структури даних, об'єктно-орієнтоване програмування.

Вступ. Дисципліни «Алгоритмізація та програмування» (АП), «Алгоритми та структури даних» (АСД), «Об'єктно-орієнтоване програмування» (ООП) є базовими для здобувачів вищої освіти галузі І2 Інформаційні технології. Вони вивчаються в університетах України на першому і другому роках навчання бакалаврату спеціальностей 121, 122, 126. У західних вишах змістові розділи цих дисциплін здебільшого входять у курси CS1 і CS2. Знання та навички, що здобуваються студентами під час вивчення цих предметів, закладають фундаментальну базу для подальшого навчання програмування в університеті та майбутньої професійної діяльності. Успішність вивчення цих

курсів суттєво впливає на гарне засвоєння подальших фахових предметів, де вивчають засоби розробки програмного забезпечення, і значною мірою пов'язана з подальшим становленням майбутнього фахівця ІТ галузі. Важливість результативного опанування курсів АП, АСД та ООП обумовлює потребу гнучкого корегування навчальних програм для приведення їх у відповідність до сучасних вимог в ІТ галузі, а також для пошуків можливостей покращення ефективності засвоєння здобувачами вищої освіти матеріалу цих дисциплін.

Аналіз публікацій та джерел дослідження. Дебати про підходи до викладання в університетах курсів CS1 і CS2 не припиня-

ються упродовж десятиліть. Праць, присвячених окремо різним аспектам вивчення початкових курсів з програмування, доволі багато. Викладачі у вишах по всьому світу продовжують шукати способи підвищення результативності цих курсів [1–30]. Зокрема, вибору мови програмування присвячені роботи [1, 2, 4, 15–30], успішність здобувачів вищої освіти для зазначених предметів аналізується у статтях [3–7], а введенню в навчальну програму нових класів задач присвячені дослідження [8–14]. Однак праць, в яких був би проведений комплексний аналіз підходів до викладання курсів CS1 і CS2, в університетах світу та України немає. Крім того, в наявних дослідженнях не враховується залежність ефективності запропонованих рішень (зокрема вибору мови програмування) від регіональної специфіки і традицій ІТ освіти.

Більшість робіт присвячена дослідженню особливостей викладання лише одного з курсів CS1 або CS2, а взаємозв'язок обох курсів вивчено мало. Робіт із дослідження підходів до викладання трьох дисциплін програмування (АП, АСД і ООП) у взаємозв'язку в вишах Східної Європи та України зокрема ще не було.

Саме тому актуальними є комплексні дослідження, спрямовані, з одного боку, на пошук способів покращення ефективності вивчення базових курсів програмування АП, АСД та ООП в університетах, а з другого – на внесення змін до навчальних програм відповідних дисциплін, враховуючи регіональну специфіку.

Новизна означеної статті полягає в комплексному підході до пошуку способів покращення ефективності вивчення базових курсів програмування, що полягає в аналізі передового світового досвіду у викладанні дисциплін з програмування на 1-му і 2-му курсах навчання бакалаврату при підготовці ІТ фахівців, дослідженні стану відповідних дисциплін у вітчизняних вишах та залученні найкращих підходів задля подальшого гнучкого корегування навчальних програм дисциплін, враховуючи регіональну специфіку ІТ освіти і сучасні вимоги до фахівців на ринку ІТ праці України.

Метою дослідження є аналіз та порівняння підходів до викладання курсів CS1 і CS2 у вищих навчальних закладах світу та України задля вивчення можливості внесення змін до навчальних програм курсів АП, АСД

та ООП відповідно до сучасних тенденцій ІТ-ринку праці в Україні, враховуючи специфіку української вищої ІТ освіти та залучаючи досвід закордонних університетів.

Задачами дослідження є:

- аналіз підходів до викладання CS1 і CS2 в університетах світу задля виявлення і залучення успішного закордонного досвіду;
- аналіз підходів до викладання курсів АП, АСД та ООП в університетах України, порівняння їх із підходами закордонних вишів та регіональними трендами ринку ІТ праці.

Виклад основного матеріалу

1. Аналіз підходів до викладання CS1 і CS2 в університетах світу. Основними питаннями, що викликають найбільшу дискусію між викладачами університетів в усьому світі, є вибір оптимальних мов програмування для курсів CS1 і CS2, а також пошук способів покращення успішності студентів у зазначених курсах. Водночас значна увага у дослідженнях приділяється саме курсу CS1 (61,7 %), CS2 – лише 23,3 %, ще 15 % присвячені їм обом [1]. Викладачі відзначають складність зазначених предметів для студентів і високі показники незадовільних оцінок на іспитах [2]. Автори робіт [3, 4] доходять висновку, що перший і другий курси навчання програмування мають, з одного боку, найнижчий показник успішності, а з другого – високий рівень демотивації і відсіву, особливо на початковому курсі програмування. На невдачі в цьому курсі звертають також увагу роботи [5, 6]. Аналіз успішності на іспитах з курсу CS1 в різні роки показав, що невдачі становлять від 28 % до 33 % [7]. Труднощі фіксуються не лише в початковому курсі CS1, а й у розумінні концепцій ООП [8]. В роботі [2] відзначається, що викладачі заради скорочення незадовільних результатів змушені спрощувати курс і знижувати вимоги до знань здобувачів вищої освіти. Такий підхід позитивно впливає на статистику невдач, проте завдає шкоди сильним студентам.

Дослідники вивчають і чинники, які впливають на процес навчання основ програмування та ООП. З-поміж негативних факторів для навчання основ програмування називають відсутність у першокурсників навіть мінімальних знань про специфіку програмування [2]. Як позитивні фактори відзначають [5] сильну мотивацію, особливо індивідуальну, пов'язану з реальним інтересом до предмета, а також наявність дидактичних методик, які допомагають

засвоїти основні концепції програмування. Зазначається [9], що викладачі, які добре знаються на своєму предметі, з більшою ймовірністю готуватимуть успішних фахівців.

У деяких роботах, присвячених способам підвищення ефективності вивчення курсів CS1 і CS2, пропонується використовувати візуальну мову Scratch, принаймні в першій частині курсу з основ програмування. На думку авторів [2, 9, 10], це позитивно впливає і на другий семестр, коли студенти вивчають ООП. Водночас відзначається відсутність зв'язку між мовами Scratch і C#, на базі якої вивчається ООП, що спричиняє певні труднощі у сприйнятті матеріалу.

Для подолання складнощів у розумінні основних концепцій ООП пропонується [8] використовувати підхід, що ґрунтується на взаємодії з матеріальними інтерфейсами користувача (TUI). Так, у роботі [11] запропоновано проектну методику навчання ООП, орієнтовану на розробку невеликого класу C++, який дає студентам змогу будувати геометричні моделі, маніпулюючи тривимірною матрицею. Іншим поширеним способом заохочення здобувачів вищої освіти є використання в курсах CS1 і CS2 завдань, пов'язаних із розробкою ігор [9, 10, 12, 13]. Обговорюються підходи [14] з використання ще під час навчання у вищій одночасного залучення здобувачів вищої освіти до розробки реальних проектів із метою надання можливості проявити свої практичні знання і мотивувати їх ретельніше вивчати матеріал дисциплін.

Традиційно найбільшу увагу дослідників зосереджено на виборі мови програмування для викладання. Палкі дискусії з цього питання не вщухають багато років. Згідно з дослідженням [15] 88 % університетів у курсі CS1 використовують одну з мов: Java, Python, C++ або C, причому популярність мови Python останнім часом зростає за рахунок зниження популярності Java. Автори [16] називають мови Java, Python і C++ найбільш поширеними. Дослідження [1] виявило таке співвідношення популярності мов: Java – 40 %, Python – 30 %, C++ – 17 %, C – 8 %. Решта мов, як-от: C#, Matlab і Ruby, значно менш популярні для цих курсів. У роботі [4] з'ясовано, що в США в курсі CS1 мову Java використовують в 41,94 % університетів, Python – 26,45 %, C++ – 19,35 %, C – 4,52 %, C# – 0,65 %, ще 7,1 % – інші мови програмування. В Австралії найпоширенішими є Java і Python

(по 31,25 % кожна), C (16,67 %), C# (10,41 %), Visual Basic (4,17 %), решта – інші мови. За даними [17], співвідношення мов програмування на першому курсі європейських університетів таке: C – 45,7 %, C++ – 15 %, Java – 8,3 %, Pascal – 7,7 % і Python – 5,6 %. Показники свідчать [18], що у виборі мови виявляється виражена регіональна специфіка.

Чимало робіт присвячено обговоренню досвіду використання тієї чи іншої мови в курсі CS1, зокрема мови Python [2, 19]. Ця мова останнім часом набула значної популярності і спричинила багато дискусій. Порівняння мов C++ і Python [20] виявило, що Python не полегшує процес навчання, а студенти, які у вступному курсі програмування використовували Python, мали вищі показники складнощів, аніж студенти, які використовували C++. До такого висновку дійшли автори робіт [13, 21, 22]. Автори [13], які спробували змінити мову програмування C у курсі CS1 на Python, після проведеного дослідження повернулися до мови C. Роботи [13, 23, 24–25] також надають перевагу C/C++ для вивчення у курсі CS1.

Дискусії точаться і щодо мови програмування для курсу CS2 (ACD та/або ООП). Загальноприйнято у цьому курсі використовувати об'єктно-орієнтовану мову програмування. Так, 72 % університетів США використовують мову Java, другою за популярністю є Python, яку застосовують 8 % університетів [15]. За даними [1], мова Java є найпопулярнішою для CS2, її частка становить 34,8 %. Крім того, для курсу структур даних застосовують мови C++ [19] і C# [2]. Стосовно використання мов програмування в європейських університетах у другому семестрі першого курсу [17], то розподіл такий: C++ – 32,8 %, C – 28,9 % і Java – 16,1 % відповідно. На другому курсі в третьому і четвертому семестрах: Java – 39 % і 35,1 % відповідно, C – 19,5 % і 13,5 %, C++ – 17 % і 19,6 %, C# – 6 % і 14,2 %. Автори [21] вважають, що концепції типізації даних, керування пам'яттю і посилення на об'єкти, які відсутні в мові Python, є важливими для курсу CS2, а тому вважають мову C++, в якій вони є, більш відповідною для вивчення.

Іншим важливим питанням є безболісність переходу від одного курсу до іншого. Дослідження [21], присвячене достатності вивчення курсу CS1 для підготовки здобувачів вищої освіти до CS2, виявило, що більшість студентів відповіли на це питання ствердно. Проте опитані студенти відзначили структуру

даних «зв'язані списки» складною для розуміння, що може бути наслідком вивчення Python у курсі CS1. Аналогічна складність може виникати після вивчення в цьому курсі мови Java, що підтверджено дослідженням [26]. Автори [25] пропонують у вступному курсі програмування використовувати C++, а в ООП – мову з тієї самої групи (C++, C#, Java), зважаючи на те, що належність мов до одного сімейства сприяє засвоєнню матеріалу.

Серед вагомих аргументів щодо вибору мови називають: простоту синтаксису [2, 20, 22, 27], популярність на ринку праці [2, 19, 27-29], наявність усіх базових конструкцій у мові [27, 29], підтримку мови різними середовищами програмування, в тому числі наявність онлайн-компіляторів [19, 27], простоту встановлення IDE [19, 27, 29], педагогічні переваги [27-29], наявність онлайн-спільноти для допомоги [27, 29], максимальну сумісність з іншими мовами програмування [23]. Є думка [2], що основним джерелом проблем при вивченні вступного курсу програмування є базове планування навчальної програми, а не розуміння синтаксису або навіть концепції.

Підходи і думки дослідників різняться, універсального рішення проблем, пов'язаних із викладанням курсів CS1 і CS2, все ще не знайдено.

2. Аналіз підходів до викладання курсів АП, АСД та ООП у вишах України. Для об'єктивного аналізу різних підходів до ви-

кладання та змістового наповнення відповідних дисциплін на вебпорталі vstup.info було відібрано вищі навчальні заклади, які готують студентів спеціальностей галузі 12 Інформаційні технології. Далі з офіційних вебсайтів цих закладів було зібрано наявні у відкритому доступі навчальні та робочі програми, силбуси та інші навчальні матеріали з дисциплін АП, АСД та ООП. Загалом було проаналізовано матеріали 44 освітніх програм спеціальностей 121–126. Для кожної дисципліни було зібрано такі дані: спеціальність, семестр, в якому вивчається дисципліна, перелік передумов, кількість кредитів, кількість годин лекцій, практичних і лабораторних занять, наявність курсового проекту чи курсової роботи, мова програмування, що використовується під час викладання.

Для курсу АСД додатково визначалася наявність використання стандартних класів для структур даних замість побудови таких структур студентами власноруч.

Крім того, окремо було виписано назви навчальних тем і години, відведені на їх вивчення. Близькі за змістом теми об'єднувалися під однією загальною назвою.

Це дало можливість сформувати рейтинг тем за використанням з метою встановлення переліку найпоширеніших тем, що викладаються в рамках дисципліни АСД (таблиця 1), для подальшого аналізу і порівняння з підходами до викладання відповідного курсу CS2 у західних вишах.

Таблиця 1 – Топ-10 тем, що викладаються в курсі АСД

№	Тема	Частота використання, %
1	Базові алгоритми сортування (бульбашковий, вибором, вставками)	90,5
2	Базові структури даних: списки, черги, стеки	85,7
3	Алгоритми пошуку в лінійних структурах. Бінарний пошук	81,0
4	Бінарні дерева	76,2
5	Розширені алгоритми сортування (Шелла, поділом (Хоара), злиттям, порозрядне, зовнішнє сортування)	76,2
6	Аналіз алгоритмів	66,7
7	Загальні відомості про алгоритми і методи їх опису	47,6
8	Алгоритми на графах	47,6
9	Алгоритми пошуку в тексті	42,9
10	Рекурсія	42,9

Дослідження виявило певні особливості. Так, здебільшого дисципліна АП вивчається першою і переважно в першому семестрі першого року навчання. Далі паралельно або послідовно одна за одною вивчаються дисципліни АСД та ООП.

Логічно, що при викладанні цих предметів варто вибудовувати їх так, щоб новий курс сприймався студентами як продовження попереднього і мав мінімальну складність при переході до нього. Такий підхід дасть змогу знизити поріг складнощів для здобувачів вищої освіти у сприйнятті матеріалу при переході від одного курсу до іншого і покращити результативність та успішність навчання.

Відповідно варто враховувати і вибір мови програмування для зазначених дисциплін, щоби не відволікати студентів на вивчення особливостей нової для них мови, а зосередитися на дослідженні і засвоєнні саме засобів алгоритмізації, на вивченні різноманітних структур даних та основних концепцій ООП. Тому, по-перше, варто розглядати мови, схожі за синтаксисом, а по-друге, ті, які мають відповідний функціонал, зокрема при вивченні структур даних. Позаяк у дисципліні АП вивчаються базові концепції програмування (змінні, засоби введення-виведення даних, умовні і циклічні конструкції, масиви, рядки, структури тощо), то для цього курсу доцільно брати одну з мов програмування Java, Python, C++, C# або C і використовувати імперативну парадигму програмування. Такий підхід підтверджує досвід закордонних університетів, а також результати дослідження [23], проведеного в Державному університеті інтелектуальних технологій і зв'язку, яке встановило, що вибір мови C/C++ є найдоцільнішим.

Дисципліна АСД виявилася такою, що має найбільші відмінності у підходах до викладання у різних вишах як в Україні, так і за кордоном. Вона складається, по суті, з двох частин. Це, по-перше, теорія алгоритмів (різновиди алгоритмів сортування і пошуку, рекурсія тощо та їхній аналіз), а по-друге, структури даних (списки, черги, купа, стеки, дерева, графи, карти і хеш-таблиці тощо). У першій частині студенти переважно працюють із масивами, у другій – із динамічними структурами даних. Для успішного засвоєння матеріалу дисципліни АСД має бути чітко дотримано визначений набір передумов. Деякі теми потребують знання певних розділів ви-

щої та дискретної математики, а з курсу АП студенти повинні володіти базовими навиками програмування, знати специфіку організації вказівників, роботи з пам'яттю і структурами. Все це накладає певні обмеження на вибір мови програмування як для АСД, так і для курсу АП, що йому передуює. Аналіз власного та закордонного досвіду [21] показав, що суттєві переваги при цьому має мова C++, оскільки, на відміну від Python, Java, C#, для вивчення структур даних вона має необхідні мовні конструкції. До того ж, як було зазначено вище, досвід західних колег [21, 26] свідчить про те, що після вивчення Python або Java у початковому курсі програмування студенти можуть мати складнощі із засвоєнням динамічних структур даних.

Через це при виборі мови для курсу АП варто зосередити увагу на C/C++. З огляду на фактори, що впливають на вибір першої мови програмування [30], і вимоги до мови в курсі АСД, викладання мовою C та/або C++ у курсах АП та АСД дасть можливість докладно вивчити усі потрібні у подальшій професії засоби алгоритмізації, а перехід між матеріалом курсів буде логічним і без суттєвих ускладнень його сприйняття.

Проте на практиці як вибір мови у різних університетах для цих дисциплін, так і саме наповнення їх змістовими розділами значною мірою залежать від самого вишу і певного викладача, його досвіду, знань, наявності розроблених навчально-методичних матеріалів тощо.

Розподіл мов програмування, що використовуються в українських вишах у курсі АСД, зображено на рисунку 1. Мова C++ є найпоширенішою, вона вживається у 72 % університетів. Питома вага інших мов є відносно незначною.

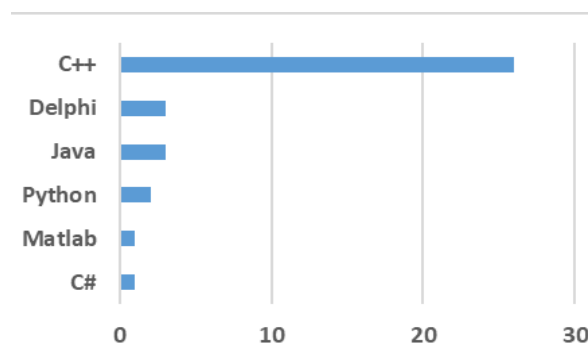


Рисунок 1 – Розподіл мов програмування в курсі АСД у вишах України

Суттєвою відмінністю у підході до вибору мови програмування для курсу АСД (CS2) у західних вишах, на відміну від українських, є переважне використання мов Java, Python або C#. Водночас треба зважати на те, що ці мови не містять вказівників, і через відсутність адрес можливе формування хибного уявлення про динамічну структуру даних або неповне розуміння її природи.

Ще однією відмінністю від західних університетів є різниця у пропорціях алгоритмів та структур даних у цьому курсі, зокрема в українських вишах спостерігається недостатня увага до структур даних. В частині програми виявлено відчутний перекис у бік алгоритмів (зважаючи на кількість годин, відведених на цю тему). У деякі робочі програми вивчення структур даних включено лише номінально або взагалі відсутнє.

Крім того, чимала частина закордонних вишів застосовує такий підхід до викладання другої частини курсу АСД, коли при вивченні динамічних структур даних використовують стандартні класи, зокрема `LinkedList`, `Stack`, `PriorityQueue` (назви подані з мови Java, в інших мовах вони подібні). А такі структури, як двозв'язний список, пропонується створювати студентам власноруч із використанням наявних у мові програмування можливостей. Однак такий підхід як передумову вимагає вивчення курсу ООП або принаймні його основ; припустимим є паралельне вивчення курсів АСД та ООП. Аналогічний підхід застосовують і деякі українські виші. Найчастіше таке простежується у програмах, де існує перекис у бік вивчення теорії алгоритмів.

Курс ООП, в якому вивчають основні концепції об'єктно-орієнтованого програмування, є першим курсом, який може бути безпосередньо пов'язаним із майбутньою професійною діяльністю здобувачів вищої освіти. Здебільшого саме після вивчення цього курсу студенти починають практикувати здобуті знання і навички на стажуваннях у софтверних компаніях. Одним із найважливіших факторів, що впливають на успішність здобувачів вищої освіти, є мотивація до навчання, яка, в свою чергу безпосередньо пов'язана з актуальністю навчальних програм дисциплін та їх сучасним наповненням.

Зокрема, студенти неохоче вчать мови програмування, які не використовуватимуть у своїй професійній діяльності через непопулярність їх на ринку праці в конкретній країні. Тому для вдалого вибору мови програмування для курсу ООП варто регулярно проводити вивчення затребуваності мов на ринку ІТ праці та враховувати тенденції змін їх популярності. Так, за даними дослідження [31], до десятки найбільш використовуваних в Україні мов програмування входять: JavaScript (18,1 %), Java (14,7 %), C# (14,3 %), Python (12,1 %), PHP (10,1 %), TypeScript (6,9 %), C++ (4,6 %), Swift (3,1 %), Ruby (3 %) і Kotlin (2,6 %). Водночас тенденції популярності мови JavaScript, яка впевнено зростала за попередні роки, нині дещо знизилась. Зате зросла популярність мов C#, TypeScript, Swift і Ruby. Частка інших мов скоротилася. Цікаво, що мова Python, яка стрімко нарощувала свою популярність декілька попередніх років, нині вперше дала негативний приріст. Виходячи з досвіду західних колег і зважаючи на останні рейтинги популярності та затребуваності мов програмування, доречно звузити список вибору мови для курсу ООП до Java, Python та C#.

Порівняння та аналіз робочих програм і силабусів дисципліни ООП, розміщених на сайтах українських вишів, виявив, що найпоширенішим є вивчення цього курсу в третьому або третьому і четвертому семестрах на другому році навчання. Однак понад третина вишів починають вивчати ООП ще в другому семестрі першого року навчання. Середня кількість кредитів, відведених на ООП, суттєво різниться у різних університетах та спеціальностях і становить у середньому сім кредитів. Як наслідок, існують певні відмінності у переліку тем, що пропонуються студентам для вивчення: від базових понять ООП до більш глибокого занурення у дисципліну.

На рисунку 2 зображено розподіл мов програмування, що використовуються в курсі ООП у вишах України. В майже половині вишів (47,7 %) віддають перевагу мові C++, 29,5 % університетів використовують мову C# або комбінацію C++ та C#, 20,5 % – Java, 2,3 % – Delphi.

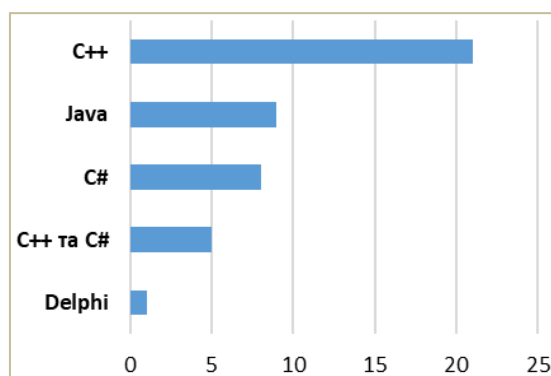


Рисунок 2 – Розподіл мов програмування в курсі ООП у вишах України

Суттєво, що для спеціальності 122 Комп'ютерні науки в українських вишах після курсу ООП нерідко викладається курс «Кросплатформне програмування», в якому переважно використовується мова Java, але водночас у відповідних навчальних програмах здебільшого не відстежується, здавалось би, логічність зв'язку із мовою попереднього курсу ООП. Цікавим підходом деяких університетів є комбінування мов C# у курсі ООП з Java у курсі «Кросплатформне програмування», що забезпечує ознайомлення студентів з обома цими мовами.

З огляду на статистику сфер застосування мов (зокрема, Python переважно використовується для аналізу даних, Java і C# – для Backend) і на сучасний рейтинг цих сфер на ринку праці в Україні (нині запит на фахівців з Backend значно перевищує попит на спеціалістів з Big Data) [31], доцільно зробити вибір у курсі ООП на користь Java або C#. Ще одним аргументом для такого вибору мови є взаємозв'язок курсів. Якщо в АП вивчається C або C++, то перехід до Java або C# для студентів буде практично безболісним, оскільки обидві ці мови є гілками розвитку C/C++ і зберегли в собі базовий синтаксис. Подальший вибір однієї з двох мов може бути зроблений, виходячи з суто педагогічних аргументів: наявності у штаті викладачів, які добре володіють вибраною мовою, наявності розроблених навчально-методичних комплексів або можливості швидкої підготовки методичної бази. Крім того, важливими аргументами для вибору мають бути думки з цього приводу головних стейкхолдерів: поради представників ІТ-компаній та/або опитування, проведене серед здобувачів вищої освіти до початку курсу.

Вельми показово, що у курсі ООП в Україні не використовується мова Python,

навіть якщо студенти вивчали її у курсі АП. Більш специфічне ознайомлення з цією мовою забезпечують інші дисципліни на третьому-четвертому роках навчання.

Відмінності у використанні мов програмування в курсах CS1 і CS2 у закордонних вишах полягають у певній специфіці ринку праці. Зокрема, минулого року в США рейтинг мов програмування за кількістю вакансій був таким: Python – 50 тис., SQL – 50 тис., Java – 45 тис., JavaScript – 38 тис., C++ – 29 тис., C# – 21 тис., PHP – 113 тис., C – 9 тис. [32]. Ці дані відрізняються від статистики українського ІТ ринку праці [31], де більш затребуваними є JavaScript-, Java- та C#-програмісти, що обумовлює відповідний вибір мов програмування в університетах.

Результати досліджень. Аналіз підходів до викладання курсів АП, АСД і ООП в Україні та їх закордонних аналогів CS1 і CS2 виявив значні розбіжності між підходами до викладання цих курсів, деякі з яких мають регіональну специфіку. Значною мірою ці розбіжності пов'язані з мовами програмування, що використовуються в тому чи іншому курсі, а також із тематичним наповненням дисциплін. Виявлено, що підходи до викладання цих курсів в Україні є подібними до інших європейських країн, однак питома вага мови Java в усіх курсах є нижчою, ніж в європейських університетах, і значно нижчою, ніж в університетах США. Водночас питома вага мови C++ є набагато вищою. Особливо вражаючою є різниця між українськими і закордонними університетами у виборі мови програмування для курсу ООП.

Обговорення. Дослідження навчальних програм виявило певні розбіжності між трендами ринку праці та мовами, які використовуються під час викладання програмування, особливо це стосується курсу ООП. Це може свідчити про недостатню гнучкість системи вищої ІТ освіти України і про потребу регулярно досліджувати можливість внесення змін у контент навчальних курсів.

Зважаючи на те, що серед найважливіших чинників, що впливають на ефективність навчання, є мотивація здобувачів вищої освіти, при виборі підходів і мов програмування для зазначених курсів варто спиратися, в першу чергу, на регіональні тенденції ринку праці, для чого треба проводити регулярний моніторинг його змін.

Висновки. Проведене дослідження актуальних проблем викладання курсів CS1 і CS2 у закордонних університетах дало можливість виявити, що ці курси, так само, як їхні аналоги в українських вишах АП, АСД та ООП, є для студентів складними, із високим рівнем невдач на іспитах і відсівом під час сесій, а тому пошуки способів для покращення успішності при вивченні цих курсів є вельми актуальними.

Аналіз літератури і джерел досліджень свідчить про недостатню увагу вчених до взаємозв'язку базових курсів програмування, а також малу увагу вітчизняних дослідників до пошуку шляхів підвищення ефективності викладання цих курсів.

Порівняння підходів до викладання CS1 і CS2 в університетах світу та України дало змогу з'ясувати перелік основних тем їх наповнення і виявити певні розбіжності, зокрема недостатню увагу до вивчення структур даних у вітчизняних вишах. Дослідження статистики використання мов програмування в зазначених дисциплінах показало, що переважно використовують мови Java, Python, C/C++, причому має місце виражена регіональна специфіка.

Встановлено, що важливими факторами успішного оволодіння матеріалом фундаментальних для майбутніх ІТ фахівців дисциплін АСД та ООП є організація гнучкого переходу від курсу АП і наявність в АП розділів, присвячених вивченню структур, вказівників і способів керування пам'яттю.

Грунтуючись на рейтингу затребуваних в Україні мов програмування, необхідності встановлення взаємозв'язку між курсами та їх специфіці, надано перевагу такій схемі використання мов програмування: C/C++ – у курсі АП, C/C++ – у курсі АСД, Java або C# – у курсі ООП.

Отримані результати дадуть можливість більш виважено і ґрунтовно підходити до змістового формування навчальних дисциплін з програмування на першому і другому курсах навчання бакалаврату при підготовці ІТ фахівців.

Новизна дослідження і практична значущість результатів полягають у виявленні розбіжностей підходів до викладання курсів АП, АСД і ООП в Україні та на Заході, які не повною мірою пояснюються регіональною специфікою ринку ІТ праці і можуть сигналізувати про необхідність вдосконалення робочих програм. Крім того, наголошується на доцільності врахування взаємозв'язку між зазна-

ченими курсами для підвищення ефективності опанування студентами базових дисциплін програмування, пошуку підходів до мотивації здобувачів вищої освіти ретельніше вивчати матеріал дисциплін і можливості проявити їм свої практичні знання.

Перспективи подальших досліджень. Складність у сприйнятті та вивченні базових дисциплін з програмування на 1-му і 2-му курсах навчання бакалаврату при підготовці ІТ фахівців для студентів свідчить про потребу подальших пошуків підходів до їх викладання. Водночас варто зважати і враховувати регіональну специфіку ІТ освіти, адже ці курси є фундаментальними у підготовці якісних фахівців ІТ галузі для нашої країни. Важливо залучати передовий світовий досвід та найкращі підходи у викладанні для гнучкого корегування навчальних програм дисциплін, враховуючи сучасні вимоги до фахівців на ринку ІТ праці.

Список використаних джерел

- [1] D. B. Silva, R. L. Aguiar, D. S. Dvconlo, and C. N. Silla, "Recent studies about teaching algorithms (CS1) and data structures (CS2) for computer science students", in *2019 IEEE Frontiers in Education Conf. (FIE)*, Covington, KY, USA, 2019, pp. 1-8. doi: 10.1109/FIE43999.2019.9028702.
- [2] D. Krpan, M. Rosić, and S. Mladenovic, "Teaching basic programming skills to undergraduate students", in *Proc. CIET'14*, 2014, pp. 147-158.
- [3] D. Krpan, S. Mladenović, and M. Rosić, "Undergraduate programming courses, students' perception and success", *Procedia – Social and Behavioral Sciences*, vol. 174, pp. 3868-3872, 2015. doi: 10.1016/j.sbspro.2015.01.1126.
- [4] S. Sobral, "30 years of CS1: Programming languages evolution", in *Proc. ICERI'19*, 2019, pp. 9197-9205. doi: 10.21125/iceri.2019.2214.
- [5] M. Andrzejewska, "Przyczyny niepowodzeń edukacyjnych studentów informatyki na wstępnych kursach programowania", *Edukacja, Technika, Informatyka*, vol. 4, pp. 211-217, 2018. doi: 10.15584/eti.2018.4.29.
- [6] M. Almeida, L. Alves, M. Pereira, and G. Barbosa, "EasyCoding – methodology to support programming learning", in *First Int.*

- Computer Programming Education Conf. ICPEC-2020*, 2020, pp. 1-8.
doi: 10.4230/OASICS.ICPEC.2020.1.
- [7] C. Watson, and F. Li, "Failure rates in introductory programming revisited", in *Proc. ITiCSE '14*, Association for Computing Machinery, New York, USA, 2014, pp. 39-44.
doi: 10.1145/2591708.2591749.
- [8] J. María, R. Corral, A. Balcells, A. Estévez, G. Moreno, and M. Ramos, "A game-based approach to the teaching of object-oriented programming languages", *Computers & Education*, vol. 73, pp. 83-92, 2014.
doi: 10.1016/j.compedu.2013.12.013.
- [9] S. Mladenovic, and D. Krpan, "Using games to help novices embrace programming: From elementary to higher education", *International Journal of Engineering Education*, vol. 32, no. 1 (B), pp. 521-531, 2016.
- [10] D. Krpan, S. Mladenović, and G. Zaharija, "Mediated transfer from visual to high-level programming language", in *40th Int. Convention Information and Communication Technology, Electronics and Microelectronics (MIPRO'2017)*, Opatija, 2017, pp. 800-805.
doi: 10.23919/MIPRO.2017.7973531.
- [11] A. Brito, and A. Medeiros, "A motivating approach to introduce object-oriented programming to engineering students", *The International Journal of Electrical Engineering & Education*, vol. 7, 2019.
doi: 10.1177/0020720919856247.
- [12] W. Yoke, and M. Yatim, "Computer game as learning and teaching tool for object oriented programming in higher education institution", *Procedia – Social and Behavioral Sciences*, vol. 123, pp. 215-224, 2014.
doi: 10.1016/j.sbspro.2014.01.1417.
- [13] M. Madeja, and J. Porubán, "Innovative approaches in C introductory programming courses", in *ICTERI PhD Symposium*, 2019.
- [14] M. Konecki, S. Lovrenčić, and M. Kaniški, "Using real projects as motivators in programming education", in *2016 39th Int. Convention Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2016, pp. 883-886.
doi: 10.1109/MIPRO.2016.7522264.
- [15] R. M. Siegfried, K. G. Herbert-Berger, and J. P. Siegfried, "CS2 and the impact of programming language choice", in *Proc. 51st ACM Technical Symposium Computer Science Education (SIGCSE '20)*, Association for Computing Machinery, New York, 2020, p. 1353.
doi: 10.1145/3328778.3372644.
- [16] R. M. Siegfried, J. P. Siegfried, and G. Alexandro, "A longitudinal analysis of the reid list of first programming languages", *Information Systems Education Journal*, vol. 14 (6), pp. 47-54, 2016.
- [17] V. Aleksić, and M. Ivanović, "Introductory programming subject in european higher education", *Informatics in Education*, vol. 15, no. 2, pp. 163-182, 2016.
doi: 10.15388/infedu.2016.09.
- [18] E. Murphy, T. Crick, and J. Davenport, "An analysis of introductory programming courses at UK universities", *The Art, Science, and Engineering of Programming*, vol. 1 (2), no. 18, 2017.
doi: 10.22152/programming-journal.org/2017/1/18.
- [19] A. Herala, E. Vanhala, and U. Nikula, "Object-oriented programming course revisited", in *Proc. Koli Calling '15*, Association for Computing Machinery, New York, 2015, pp. 23-32.
doi: 10.1145/2828959.2828974.
- [20] N. Alzahrani, F. Vahid, A. Edgcomb, K. Nguyen, and R. Lysecky, "Python versus C++: An analysis of student struggle on small coding exercises in introductory programming courses", in *Proc. 49th ACM Technical Symposium Computer Science Education (SIGCSE '18)*, Association for Computing Machinery, New York, NY, USA, 2018, pp. 86-91.
doi: 10.1145/3159450.3160586.
- [21] L. Layman, Y. Song, and C. Guinn, "Toward predicting success and failure in CS2: A mixed-method analysis", in *Proc. 2020 ACM Southeast Conf. (ACM SE '20)*, Association for Computing Machinery, New York, 2020, pp. 218-225.
doi: 10.1145/3374135.3385277.
- [22] F. Johnson, S. McQuistin, and J. O'Donnell, "Analysis of student misconceptions using Python as an introductory programming language", in *Proc. 4th Conf. Computing Education Practice 2020 (CEP 2020)*, Association for Computing Machinery, New York, NY, USA, 2020, vol. 4, pp. 1-4.
doi: 10.1145/3372356.3372360.
- [23] Y. Prokop, E. Trofimenko, N. Loginova, A. Zadereyko, and M. Gerganov, "Multivariate analysis when choosing the first pro-

- gramming language studied in universities", in *2019 IEEE 2nd Ukraine Conf. Electrical and Computer Engineering (UKRCON)*, 2019, pp. 1224-1228.
doi: 10.1109/UKRCON.2019.8879810.
- [24] S. Naveed, M. Sarim, and A. Nadeem, "C in CS1: Snags and viable solution", *Mehran University Research Journal of Engineering and Technology*, vol. 37, no. 1, pp. 1-14, 2018.
doi: 10.22581/MUET1982.1801.01.
- [25] I. Damyanov, and N. Borisova, "Programming languages in undergraduate courses and in software industry in Bulgaria", *International Journal of Pure and Applied Mathematics*, vol. 117, no. 2, pp. 271-278, 2017.
doi: 10.12732/ijpam.v117i2.3.
- [26] S. Davies, J. Polack-Wahl, and K. Anewalt, "A snapshot of current practices in teaching the introductory programming sequence", in *Proc. 42nd ACM Technical Symposium Computer science education (SIGCSE '11)*, Association for Computing Machinery, New York, NY, USA, 2011, pp. 625-630.
doi: 10.1145/1953163.1953339.
- [27] Y. Prokop, E. Trofimenko, N. Severin, and L. Bukata, "An analysis of criteria for choosing a first programming language in universities", in *15th Int. Conf. ICT in Education, Research, and Industrial Applications (IC-TERI'19)*, 2019, pp. 420-425.
- [28] S. Sobral, "CS1: C, Java or Python? Tips for a conscious choice", in *Proc. ICERI'19*, 2019, pp. 2512-2519.
doi: 10.21125/iceri.2019.0666.
- [29] E. Mehmood, A. Abid, M. S. Farooq, and N. A. Nawaz, "Curriculum, teaching and learning, and assessments for introductory programming course," in *IEEE Access*, 2020, vol. 8, pp. 125961-125981.
doi: 10.1109/ACCESS.2020.3008321.
- [30] Ю. В. Прокоп, О. Г. Трофименко, та О. В. Задерейко, "Аналіз підходів у викладанні початкового курсу програмування в університетах", *Системні технології*, № 4 (135), с. 73-84, 2021.
doi: 10.34185/1562-9945-4-135-2021-08.
- [31] Р. Шевченко, "Рейтинг мов програмування 2021: частка Python зменшується, а TypeScript обійшов C++". [Електронний ресурс]. Режим доступу: <https://dou.ua/lenta/articles/language-rating-jan-2021/?from=lg>. Дата звернення: Feb. 14, 2021.
- [32] The best paying and most in-demand programming languages in 2020. [Online]. Available: <https://www.codeplatoon.org/best-paying-most-in-demand-programming-languages-2020>. Accessed on: Feb. 14, 2021.

References

- [1] D. B. Silva, R. L. Aguiar, D. S. Dvconlo, and C. N. Silla, "Recent studies about teaching algorithms (CS1) and data structures (CS2) for computer science students", in *2019 IEEE Frontiers in Education Conf. (FIE)*, Covington, KY, USA, 2019, pp. 1-8.
doi: 10.1109/FIE43999.2019.9028702.
- [2] D. Krpan, M. Rosić, and S. Mladenovic, "Teaching basic programming skills to undergraduate students", in *Proc. CIET'14*, 2014, pp. 147-158.
- [3] D. Krpan, S. Mladenović, and M. Rosić, "Undergraduate programming courses, students' perception and success", *Procedia – Social and Behavioral Sciences*, vol. 174, pp. 3868-3872, 2015.
doi: 10.1016/j.sbspro.2015.01.1126.
- [4] S. Sobral, "30 years of CS1: Programming languages evolution", in *Proc. ICERI'19*, 2019, pp. 9197-9205.
doi: 10.21125/iceri.2019.2214.
- [5] M. Andrzejewska, "Przyczyny niepowodzeń edukacyjnych studentów informatyki na wstępnych kursach programowania", *Edukacja, Technika, Informatyka*, vol. 4, pp. 211-217, 2018.
doi: 10.15584/eti.2018.4.29.
- [6] M. Almeida, L. Alves, M. Pereira, and G. Barbosa, "EasyCoding – methodology to support programming learning", in *First Int. Computer Programming Education Conf. ICPEC-2020*, 2020, pp. 1-8.
doi: 10.4230/OASICS.ICPEC.2020.1.
- [7] C. Watson, and F. Li, "Failure rates in introductory programming revisited", in *Proc. 2014 Conf. Innovation & Technology in Computer Science Education (ITiCSE '14)*, Association for Computing Machinery, New York, NY, USA, 2014, pp. 39-44.
doi: 10.1145/2591708.2591749.
- [8] J. María, R. Corral, A. Balcells, A. Estévez, G. Moreno, and M. Ramos, "A game-based approach to the teaching of object-oriented programming languages", *Computers & Education*, vol. 73, pp. 83-92, 2014.
doi: 10.1016/j.compedu.2013.12.013.
- [9] S. Mladenovic, and D. Krpan, "Using games to help novices embrace programming: From

- elementary to higher education", *International Journal of Engineering Education*, vol. 32, no. 1 (B), pp. 521-531, 2016.
- [10] D. Krpan, S. Mladenović, and G. Zaharija, "Mediated transfer from visual to high-level programming language", in *40th Int. Convention Information and Communication Technology, Electronics and Microelectronics (MIPRO'2017)*, Opatija, 2017, pp. 800-805. doi: 10.23919/MIPRO.2017.7973531.
- [11] A. Brito, and A. Medeiros, "A motivating approach to introduce object-oriented programming to engineering students", *The International Journal of Electrical Engineering & Education*, vol. 7, 2019. doi: 10.1177/0020720919856247.
- [12] W. Yoke, and M. Yatim, "Computer game as learning and teaching tool for object oriented programming in higher education institution", *Procedia – Social and Behavioral Sciences*, vol. 123, pp. 215-224, 2014. doi: 10.1016/j.sbspro.2014.01.1417.
- [13] M. Madeja, and J. Porubán, "Innovative approaches in C introductory programming courses", in *ICTERI PhD Symposium*, 2019.
- [14] M. Konecki, S. Lovrenčić, and M. Kaniški, "Using real projects as motivators in programming education", in *2016 39th Int. Convention Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2016, pp. 883-886. doi: 10.1109/MIPRO.2016.7522264.
- [15] R. M. Siegfried, K. G. Herbert-Berger, and J. P. Siegfried, "CS2 and the impact of programming language choice", in *Proc. 51st ACM Technical Symposium Computer Science Education (SIGCSE '20)*, Association for Computing Machinery, New York, 2020, p. 1353. doi: 10.1145/3328778.3372644.
- [16] R. M. Siegfried, J. P. Siegfried, and G. Alexandro, "A longitudinal analysis of the reid list of first programming languages", *Information Systems Education Journal*, vol. 14 (6), pp. 47-54, 2016.
- [17] V. Aleksić, and M. Ivanović, "Introductory programming subject in european higher education", *Informatics in Education*, vol. 15, no. 2, pp. 163-182, 2016. doi: 10.15388/infedu.2016.09.
- [18] E. Murphy, T. Crick, and J. Davenport, "An analysis of introductory programming courses at UK universities", *The Art, Science, and Engineering of Programming*, vol. 1 (2), no. 18, 2017. doi: 10.22152/programming-journal.org/2017/1/18.
- [19] A. Herala, E. Vanhala, and U. Nikula, "Object-oriented programming course revisited", in *Proc. 15th Koli Calling Conf. Computing Education Research*, Association for Computing Machinery, New York, 2015, pp. 23-32. doi: 10.1145/2828959.2828974.
- [20] N. Alzahrani, F. Vahid, A. Edgcomb, K. Nguyen, and R. Lysecky, "Python versus C++: An analysis of student struggle on small coding exercises in introductory programming courses", in *Proc. 49th ACM Technical Symposium Computer Science Education (SIGCSE '18)*, Association for Computing Machinery, New York, NY, USA, 2018, pp. 86-91. doi: 10.1145/3159450.3160586.
- [21] L. Layman, Y. Song, and C. Guinn, "Toward predicting success and failure in CS2: A mixed-method analysis", in *Proc. 2020 ACM Southeast Conf. (ACM SE '20)*, Association for Computing Machinery, New York, NY, USA, 2020, pp. 218-225. doi: 10.1145/3374135.3385277.
- [22] F. Johnson, S. McQuistin, and J. O'Donnell, "Analysis of student misconceptions using Python as an introductory programming language", in *Proc. 4th Conf. Computing Education Practice 2020 (CEP 2020)*, Association for Computing Machinery, New York, NY, USA, 2020, vol. 4, pp. 1-4. doi: 10.1145/3372356.3372360.
- [23] Y. Prokop, E. Trofimenko, N. Loginova, A. Zadereyko, and M. Gerganov, "Multivariate analysis when choosing the first programming language studied in universities", in *2019 IEEE 2nd Ukraine Conf. Electrical and Computer Engineering (UKRCON)*, 2019, pp. 1224-1228. doi: 10.1109/UKRCON.2019.8879810.
- [24] S. Naveed, M. Sarim, and A. Nadeem, "C in CS1: Snags and viable solution", *Mehran University Research Journal of Engineering and Technology*, vol. 37, no. 1, pp. 1-14, 2018. doi: 10.22581/MUET1982.1801.01.
- [25] I. Damyanov, and N. Borisova, "Programming languages in undergraduate courses and in software industry in Bulgaria", *International Journal of Pure and Applied Mathematics*, vol. 117, no. 2, pp. 271-278, 2017. doi: 10.12732/ijpam.v117i2.3.

- [26] S. Davies, J. Polack-Wahl, and K. Anewalt, "A snapshot of current practices in teaching the introductory programming sequence", in *Proc. 42nd ACM Technical Symposium Computer science education (SIGCSE '11)*, Association for Computing Machinery, New York, NY, USA, 2011, pp. 625-630. doi: 10.1145/1953163.1953339.
- [27] Y. Prokop, E. Trofimenko, N. Severin, and L. Bukata, "An analysis of criteria for choosing a first programming language in universities", in *15th Int. Conf. ICT in Education, Research, and Industrial Applications (IC-TERI'19)*, 2019, pp. 420-425.
- [28] S. Sobral, "CS1: C, Java or Python? Tips for a conscious choice", in *Proc. ICERI'19*, 2019, pp. 2512-2519. doi: 10.21125/iceri.2019.0666.
- [29] E. Mehmood, A. Abid, M. S. Farooq, and N. A. Nawaz, "Curriculum, teaching and learning, and assessments for introductory programming course," in *IEEE Access*, 2020, vol. 8, pp. 125961-125981. doi: 10.1109/ACCESS.2020.3008321.
- [30] Yu. V. Prokop, O. H. Trofymenko, and O. V. Zadereyko, "Analysis of approaches in teaching the introductory programming course in universities", *Systemni tekhnologii*, vol. 4 (135), pp. 73-84, 2021. [in Ukrainian]. doi: 10.34185/1562-9945-4-135-2021-08.
- [31] R. Shevchenko, "Rating of programming languages 2021: the share of Python is decreasing, and TypeScript has bypassed C++". [Online]. Available: <https://dou.ua/lenta/articles/language-rating-jan-2021/?from=tg>. Accessed on: Feb. 14, 2021 [in Ukrainian].
- [32] The best paying and most in-demand programming languages in 2020. [Online]. Available: <https://www.codeplatoon.org/best-paying-most-in-demand-programming-languages-2020>. Accessed on: Feb. 14, 2021.

Yu. V. Prokop¹, Ph. D., Senior Lecturer at the Department of Information Technologies,

e-mail: yulia13.prokop@gmail.com

O. G. Trofymenko², Ph. D., Docent,

Associate Professor of the Department of Information Technologies,

e-mail: egt@ukr.net

A. A. Toloknov², Assistant of the Department of Information Technologies,

e-mail: arnoldovich1@gmail.com

Ya. V. Dubovoi¹, Master in Cybersecurity

e-mail: dubovoy97@gmail.com

¹State University of Intellectual Technologies and Communications

Kuznechna st., 1, Odesa, 65029, Ukraine

²National University "Odesa Law Academy"

Fontanska doroga st., 23, Odesa, 65009, Ukraine

ANALYSIS OF WAYS TO IMPROVE THE EFFICIENCY IN THE STUDY OF CS1 AND CS2 COURSES IN UNIVERSITIES OF THE WORLD AND UKRAINE

The article is devoted to finding ways to improve the efficiency in the study of basic disciplines of programming in the 1st and 2nd year of bachelor's degree in the training of IT professionals. The curricula for the training of students in the field of 12 (Information Technologies) in Ukrainian universities have been studied. The content of Algorithmization and programming, Algorithms and data structures, Object-oriented programming disciplines and similar CS1 and CS2 courses in Western universities are considered. These courses have been found to be fundamental in training quality IT professionals. A study of current problems of teaching CS1 and CS2 courses in universities of other countries has revealed that these courses, as well as their counterparts in Ukrainian universities, are difficult for students, with a high level of exam failure and dropout during sessions. Therefore, the search for ways to improve the efficiency of these courses is very important. The analysis of different

approaches to teaching CS1 and CS2 in the universities of the world and Ukraine has allowed us to find out the list of the main topics of their content. The study of statistics on the use of programming languages in these disciplines has shown that universities mainly use Java, Python, C/C++ languages, and there is a pronounced regional specificity.

Based on the rating of programming languages in demand in Ukraine, the need to establish a relationship between courses and their specifics, the following scheme of using programming languages is preferred: C/C++ – in Algorithmization and programming course, C/C++ – in Algorithms and data structures course, Java or C# – in Object-oriented programming course. Emphasis is placed on the importance of drawing on world best practices and best approaches in teaching for flexible adjustment of curricula of disciplines taking into account modern requirements for specialists in the IT labor market.

The results of the study will allow a more balanced and thorough approach to the meaningful formation of these programming courses, the choice of approaches to motivate higher education students to study.

Keywords: *programming language, IT education, algorithmization and programming, algorithms and data structures, object-oriented programming.*