

## АНАЛІЗ ЗМІСТУ ДИСЦИПЛІНИ «ВСТУП ДО СПЕЦІАЛЬНОСТІ» В ПРОФЕСІЙНІЙ ПІДГОТОВЦІ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ

Владислав Круглик

*Мелітопольський державний педагогічний університет імені Богдана Хмельницького*

### Анотація:

На основі досвіду викладання дисципліни «Вступ до спеціальності» автор статті дійшов висновку, що її зміст вимагає досконалого аналізу, розробки нової парадигми навчання й перевірки її ефективності. Саме тому в дослідженні наведені оновлена мета й завдання цієї дисципліни, визначені знання, уміння й компетенції, якими мають оволодіти майбутні інженери-програмісти в процесі її засвоєння, зміст двох модулів, перший з яких присвячено питанням програмування й підготовки до освоєння майбутньої професійної діяльності, а другий – організації наукового дослідження в галузі інформаційних технологій і програмування, а також запропоновано використання інтерактивного онлайн-середовища Codecademy як засобу навчання. Апробація модифікованого змісту дисципліни дала змогу дійти висновку про доцільність нововведень і корекцію процесу викладання дисципліни «Вступ до спеціальності» для вивчення професійної лексики англійською мовою.

### Ключові слова:

інженер-програміст; зміст дисципліни; вступ до спеціальності.

### Аннотация:

**Круглик Владислав. Анализ содержания дисциплины «Введение в специальность» в профессиональной подготовке инженеров-программистов.**

На основе опыта преподавания дисциплины «Введение в специальность» автор статьи делает вывод о том, что ее содержание требует досконального анализа, разработки новой парадигмы обучения и проверки ее эффективности. Именно поэтому в исследовании приведены обновленные цель и задачи этой дисциплины, определены знания, умения и компетенции, которыми должны овладеть будущие инженеры-программисты в процессе ее освоения; содержание двух модулей, первый из которых посвящен вопросам программирования и подготовке к освоению будущей профессиональной деятельности, а второй – организации научного исследования в области информационных технологий и программирования, а также предложено использовать в качестве средства обучения интерактивную онлайн-среду Codecademy. В результате апробации модифицированного содержания дисциплины сделан вывод о целесообразности нововведений и коррекции процесса преподавания дисциплины «Введение в специальность» с целью изучения профессиональной лексики на английском языке.

### Ключевые слова:

инженер-программист; содержание дисциплины; введение в специальность.

### Resume:

**Kruhlyk Vladyslav. Analysis of the discipline "Introduction to the specialty" in professional training of software engineers.**

Based on teaching experience of "Introduction to specialty" the author of the article concludes that it requires a thorough analysis of the content, the development of a new learning paradigm and verification of its effectiveness. It is proposed to update the purpose and objectives of this discipline, knowledge, skills and competences, which future software engineers should master in the course of its development; use as a learning tool Codecademy interactive online environment; the contents of two modules, the first of which is devoted to the issues of programming and preparation for the development of future professional activity, and the second – the organization of scientific research in the field of information technology and programming. As a result of testing of the modified content of the discipline, the conclusion has been made of the feasibility of innovations and correction process of teaching the discipline to study the professional vocabulary in English.

### Key words:

software engineer; content of discipline; introduction to specialty.

Постановка проблеми. Сфера професійної діяльності інженерів-програмістів досить широка й різноманітна. Як домінантні можна назвати такі професійні завдання: встановлення й налаштування програмного й апаратного забезпечення, розробка програм та інформаційних систем різної тематики й масштабів, написання документації для розробки й супроводу програмних проєктів, проєктування й налаштування комп'ютерної мережі, створення цифрової графічної продукції, надання послуг з системного аналізу та інших послуг у сфері інформатизації, програмування, мережних технологій. Студенти, що навчаються на спеціальностях «Інженерія програмного забезпечення», «Комп'ютерні науки та інформаційні технології», «Комп'ютерна інженерія», «Системний аналіз», «Кібербезпека», опановують сферу своєї майбутньої професійної діяльності поетапно. Першим етапом стає вивчення дисципліни «Вступ до спеціальності», яка має ознайомити майбутніх інженерів-програмістів з академічними основами їхньої

професійної підготовки й допомогти адаптуватися до навчання у ВНЗ.

Про важливість цієї дисципліни свідчить той факт, результати проведеного нами опитування до вивчення дисципліни «Вступ до спеціальності» показали, що більшість студентів першого курсу не уявляє сутності професії та змісту майбутньої професійної діяльності, не мотивовані до її освоєння й мають сумніви щодо обраного напряму навчання. Картина змінюється після завершення вивчення дисципліни: студенти визначаються зі спеціалізацією у сфері інформаційних технологій, у них підвищується мотивація до освоєння професійної діяльності, починає вибудовуватися індивідуальна траєкторія професійного розвитку. Отже, через недостатню профорієнтаційну діяльність щодо висвітлення особливостей професії інженера-програміста зміст дисципліни «Вступ до спеціальності» вимагає досконалого аналізу, розробки нової парадигми навчання і перевірки її ефективності. Це має сприяти розв'язанню проблем, що

виникають у процесі професійного становлення майбутніх інженерів-програмістів.

Аналіз останніх досліджень і публікацій. Теоретичні та практичні аспекти викладання дисципліни «Вступ до спеціальності» розкриті в працях В. Гвоздєєвої, В. Осадчого, К. Осадчої, І. Сердюк, З. Сейдаметової та ін. Так, зокрема, В. Гвоздєєва [1] мету дисципліни «Вступ до спеціальності програміста» вбачає у формуванні уявлення про професію програміста й забезпечення підстав для набуття навичок у цій сфері діяльності. Для її досягнення науковець пропонує розглядати основні поняття програмування, принципи структурного програмування й початку програмування в середовищі Turbo Pascal і системі Delphi, особливості програмного забезпечення для обчислювальної техніки й автоматизованих систем, питання правового та програмного захисту комп'ютерної інформації, стандартизації та ліцензування програмних продуктів.

Автори навчального посібника «Вступ до спеціальності програміста» В. Осадчий, К. Осадча, І. Сердюк [5] вважають, що теоретичний матеріал дисципліни має ознайомити студентів з предметом діяльності програміста, особливостями його професійної діяльності, класифікацією програмного забезпечення, основами теорії програмування, видами й типами сучасних мов програмування, основними етапами життєвого циклу створення й супроводу програмних продуктів, організацією колективів програмістів, процесом тестування програмного забезпечення, питаннями правового захисту комп'ютерної інформації, стандартизації й ліцензування програмних продуктів. У практичному блоці дисципліни автори пропонують виконання таких завдань: пошук і відбір необхідної інформації за допомогою різних пошукових систем; оформлення технічного завдання в установленому порядку й відповідно до вимог ДСТУ / ГОСТ; визначення цілей створення програми, вимог до неї й основних вихідних даних, які необхідні для її розробки; підготовка документів для отримання авторського права на програмне забезпечення; створення власного портфоліо, написання резюме, складання сертифікаційних іспитів з інформаційних технологій [4, с. 369–371].

З. Сейдаметова розглядає дисципліну «Вступ до спеціальності» в рамках адаптації студентів першого курсу комп'ютерних спеціальностей до майбутньої професійної діяльності. Вона виділяє такі педагогічні цілі викладання цієї дисципліни: формування чіткого знання й розуміння, що необхідно робити в тій чи іншій ситуації, що виникатиме під час навчання в університеті; формування професійного кругозору студента й розуміння ним майбутньої

професії; орієнтація студентів у нових ідеях і технологіях комп'ютерингу; формування розуміння історії комп'ютерингу й логіки розвитку основних ідей; створення мотивацій і стимулів для інтенсивного вивчення фундаментальних концепцій комп'ютерингу; формування мотивації для самостійного пошуку в спеціалізованій літературі, спеціалізованих джерелах, а також у мережі Інтернет інформації щодо сучасних проблем комп'ютерингу. У змісті дисципліни авторка виділяє дванадцять тем, у межах яких студенти ознайомлюються з організацією навчального процесу у вищому навчальному закладі загалом і з особливостями навчання на спеціальності «Інформатика» зокрема; вивчають найважливіші історичні факти та події комп'ютерингу; отримують дані про відомі фірми у сфері комп'ютерингу й персоналії; обговорюють питання, пов'язані з мовами програмування (Паскаль, Сі, Сі++) і компіляторами; засвоюють поняття про розмаїття операційних систем, завдання та функції, що покладаються на такі програми, а також поняття про віруси й антивірусне програмне забезпечення, про вільно поширюване програмне забезпечення й програмне забезпеченням з відкритим кодом. Також до змісту внесено теми, присвячені проектам розподілених обчислень у мережі Інтернет, проектам майбутнього (квантові комп'ютери, квантові обчислення, нанотехнології, нанокomp'ютери, біоавтомати, біокomp'ютери), професіям у комп'ютерингу, Інтернет-сервісам і вебометрії та періодичним виданням, інформаційним і освітнім ресурсам комп'ютерингу паперового та електронного форматів [7, с. 188–196].

Аналіз наведених вище джерел дав змогу дійти таких висновків: науковці часто розглядають теоретичні аспекти вступу до професії програміста (З. Сейдаметова та В. Гвоздєєва) і рідше – практичні (В. Осадчий). Замало уваги приділяється питанням, що стосуються власне особливостей професійної діяльності програміста, його кар'єрного розвитку, правового захисту продуктів діяльності й перспективам розвитку інформаційних технологій. На нашу думку, найбільш наближеною до сучасних реалій розвитку сфери професійної діяльності майбутніх інженерів-програмістів є тематика теоретичного матеріалу, запропонована колективом авторів – В. Осадчий, К. Осадча, І. Сердюк. Проте вона потребує корекції в зв'язку з появою нових технологій і тенденцій у галузі інформаційних технологій і програмування. Крім того, ми не погоджуємося з практичним блоком, запропонованим цими авторами. На наше переконання, слід зробити акцент на основах програмування

й дослідницькій діяльності майбутніх інженерів-програмістів.

Формулювання цілей статті. Зважаючи на викладене вище, метою статті було обрано пошук доцільного й ефективного змісту дисципліни «Вступ до спеціальності» та з'ясування її значення для професійної підготовки майбутніх інженерів-програмістів.

Виклад основного матеріалу дослідження. Зауважимо, що в рекомендаціях Асоціації з обчислювальної техніки (The Association for Computing Machinery – ACM) та Інституту інженерів з електротехніки та електроніки (Institute of Electrical and Electronic Engineers – IEEE) пропонуються вступні дисципліни з програмування («Вступ до програмування» («Introduction to Programming»), «Вступ до інформатики» («Introduction to Computer Science»)), метою яких є надання початкових знань для вивчення інших дисциплін [1]. Аналіз змісту цих дисциплін, що викладаються в Університеті Брауна, Харві-Мад-коледжі, Університеті Крейтона, свідчить про внесення до нього питань з фундаментальної інформатики, розробки, упровадження й тестування комп'ютерних програм, реалізації алгоритмів на певній мові програмування, об'єктно-орієнтованого програмування, а також керівні структури, масиви, рекурсія та подієве програмування. Проте не ставиться завдання висвітлення проблем адаптації до навчання у ВНЗ, освоєння професії, планування кар'єрного зростання й ознайомлення з організацією наукових досліджень у галузі інформаційних технологій і програмування.

Ми проаналізували навчальні плани спеціальностей «Інформатика», «Комп'ютерні науки», «Системна інженерія», «Програмна інженерія» освітнього рівня «бакалавр» таких українських ВНЗ, як Херсонський державний університет, Запорізький національний університет, Сумський державний університет, Уманський державний педагогічний університет імені Павла Тичини, Національний технічний університет «ХПІ», Національний університет «Києво-Могилянська академія», Національний педагогічний університет імені М. П. Драгоманова, Кіровоградський державний педагогічний університет імені Володимира Винниченка, Національний технічний університет України «КПІ», Національний університет біоресурсів і природокористування України. Проведений аналіз дав змогу дійти висновку, що в більшості вищих навчальних закладів для названих спеціальностей не викладається дисципліна «Вступ до спеціальності» («Вступ до фаху») або вона входить до варіативної частини (Сумський державний університет). Вивчення змісту цих

дисциплін показало, що в окремих випадках до нього не вносять питання з програмування, натомість висвітлюють питання, які стосуються інформаційно-комунікаційних технологій. Крім того, як і в проаналізованих закордонних анотаціях дисципліни, не розглядають питання професійної кар'єри й особливості проведення наукових досліджень у цій галузі.

Згідно з навчальним планом, за яким навчаються студенти напряму підготовки «Інформатика» освітнього рівня «бакалавр» Мелітопольського державного педагогічного університету імені Богдана Хмельницького, дисципліна «Вступ до спеціальності» є нормативною. Мета її вивчення полягає в наданні інформації про структуру навчальної програми напряму підготовки «Інформатика» й місце кожної з дисциплін, що вивчаються, у загальній схемі професійної освіти, а також у формуванні в майбутніх спеціалістів у галузі програмування уявлення про їхню майбутню професію й уміння застосовувати наукові методи пізнання в майбутній професійній діяльності. Мета вивчення дисципліни конкретизується в таких завданнях: формування професійного світогляду майбутнього фахівця з інформаційних технологій, цілісного уявлення про сутність і значення професії інженера-програміста для суспільства та економіки, спрямованість на оволодіння професією в галузі інформаційних технологій; ознайомлення зі структурою, видами, методами й послідовністю виконання наукових досліджень, розвиток умінь здійснювати наукові дослідження в цій галузі.

Унаслідок вивчення цієї навчальної дисципліни студенти повинні знати: класифікацію програмного забезпечення й сучасних мов програмування; етапи розвитку, засоби опису, основні поняття мов програмування; характеристики якості й види тестування програмного забезпечення, програмні помилки й засоби їх усунення; вимоги до сертифікації ІТ-фахівців; поняття й парадигми програмування; науковий апарат, методи, організацію, упровадження результатів та експертизи якості наукового дослідження в галузі інформаційних технологій. У процесі навчання студенти мають набути таких умінь: розробляти алгоритми й записувати їх в різних формах, виділяти й здійснювати опис об'єктів завдання з розробки програми та їх взаємозв'язки, читати й розуміти готові алгоритми, записувати програмний код, розуміти семантику основних структур управління в програмному коді, ефективно використовувати засоби програмування під час розв'язання задач, володіти термінологією програмування; грамотно складати резюме,

оформляти власне портфоліо; працювати з бібліографічним каталогом, опрацьовувати теоретичні джерела й аналізувати досвід, розробляти план наукового дослідження роботи й поетапно втілювати його, виконувати дослідну частину наукової роботи, оформлювати виконану роботу відповідно до вимог, здійснювати презентацію результатів наукового дослідження.

Дисципліна спрямована на набуття таких загальних і професійних компетенцій: аналіз і синтез – здатність до аналізу та синтезу на основі логічних аргументів і перевірених фактів; гнучкість мислення – набуття гнучкого мислення, відкритість до застосування фізичних знань і компетентностей у широкому діапазоні можливих місць роботи й у повсякденному житті; глибокі знання й розуміння – здатність аналізувати методи проектування й реалізації програмних продуктів систем на основі певної мови програмування; розв’язання проблем – здатність розв’язувати нескладні проблеми й задачі структурного, модульного, функціонального й об’єктно-орієнтованого програмування, здатність використовувати для цього відповідне програмне забезпечення; здатність до навчання – здатність шляхом самостійного навчання освоїти нові галузі, використовуючи здобуті знання з інформатики та програмування.

На вивчення дисципліни відведено 180 годин, з них 64 – на аудиторні заняття (28 – лекційні, 38 – лабораторно-практичні), 114 – на самостійну роботу. Дисципліна складається з двох модулів: «Вступ до спеціальності програміста», «Основи наукових досліджень». На перший модуль відведено 100 годин, з них 38 – на аудиторні заняття (14 – лекційні, 24 – лабораторні) і 62 – на самостійну роботу. На другий модуль відведено 80 годин, з них 28 – на аудиторні заняття (14 – лекційні, 14 – практичні), 52 – на самостійну роботу.

Аналіз і узагальнення праць, у яких йдеться про зміст дисципліни «Вступ до спеціальності» для майбутніх інженерів-програмістів, спонукав нас до вдосконалення цього змісту з опорою на фундаменталізацію професійної підготовки в галузі програмування, що має характеризуватися цілісним уявленням про всі аспекти майбутньої професійної діяльності. Саме тому ми модифікували запропонований у навчальних посібниках [1], [2], [5], [6], [8] теоретичний матеріал. Унаслідок цього лекційний матеріал першого модуля охоплює питання з основ програмування, програмного забезпечення комп’ютера, управління програмними проектами, а другого – питання організації та проведення науково-дослідної роботи (табл. 1).

Таблиця 1

Орієнтовна тематика лекційних занять і розподіл годин з дисципліни «Вступ до спеціальності»

№ п/п	Теми	Питання	К-ть годин
Модуль 1			
1.	Місце програмування й програмного забезпечення в суспільстві	Місце програмування в суспільстві, програмне забезпечення, розвиток мов і підходів до програмування, технологія програмування	2
2.	Кросплатформність. Види й типи сучасних мов програмування	Кросплатформність, універсальні й спеціалізовані мови програмування, класифікація мов програмування за типами задач, рейтинг мов програмування	2
3.	Основи баз даних	Класифікація баз даних, призначення і сфери застосування баз даних, характеристика СУБД	2
4.	Архітектура системи (проекту)	Поняття архітектури системи, уніфікована мова моделювання, шаблони проектування	2
5.	Життєвий цикл програмного засобу	Загальносистемні принципи та етапи створення програм, життєвий цикл програмного засобу, моделі MSF, екстремальне програмування, культура програмування	2
6.	Технічне завдання. Тестування програмного продукту	Поняття про сутність і структуру технічного завдання, мета технічного завдання, основні вимоги до складання технічного завдання, програмні помилки, загальні питання організації тестування програмного продукту, види тестування програмного забезпечення	2
7.	Стандартизація, сертифікація та ліцензування програмного продукту	Документування й стандартизація програм, порядок сертифікації, ліцензування програмного продукту, авторське право на програмний продукт	2

№ п/п	Теми	Питання	К-ть годин
Модуль 2			
8.	Науково-дослідна робота в процесі підготовки спеціаліста	Поняття, історія розвитку, функції та класифікації науки, перетворення науки на безпосередню продуктивну силу, основні закономірності розвитку й принципи науки, організація науки у ВНЗ	2
9.	Структура та логіка наукового дослідження	Наукова проблема й вибір теми, формулювання мети, завдань і питань дослідження, етапи проведення наукового дослідження	2
10.	Методи наукового дослідження	Класифікація методів наукового дослідження, теоретичні та емпіричні методи дослідження	2
11.	Метод вимірювання в дослідженнях	Методи вимірювання, методи опитування	2
12.	Науковий експеримент	Сутність і види експерименту, планування експерименту, програма й методика експерименту, ефективність наукових досліджень та її критерії	2
13.	Обробка та інтерпретація результатів дослідження	Групування й графічна обробка даних, статистичні методи обробки даних, методи перевірки статистичних гіпотез	2
14.	Підвищення якості наукового дослідження	Поняття якості наукового дослідження, основні напрями вдосконалення наукових досліджень, правові основи регуляції наукових досліджень, плагіат і його ознаки, етика в науковій спільноті	2

Відповідно до цього були розроблені завдання до лабораторних і практичних занять. Метою лабораторних занять першого модуля було поступове засвоєння найбільш уживаних парадигм програмування (структурне, модульне, ітеративне, об'єктно-орієнтоване) на прикладі мультипарадигмальної мови програмування Python. Під час виконання практичних робіт другого модуля студенти поступово проходили всі етапи проведення наукового дослідження за індивідуальними темами (табл. 2).

Практичні роботи мають типову структуру: назва, мета, список додаткової літератури, обладнання, питання до контролю попередніх занять, обговорення, самостійне вивчення й осмислення навчального матеріалу, хід роботи (короткі теоретичні відомості, покрокова проілюстрована інструкція, завдання до виконання), вимоги до складання звіту.

Лабораторні завдання першого модуля орієнтовані на роботу в інтерактивному онлайн-середовищі Codecademy, у якому на основі вправ відбувається поступове навчання мов програмування [10]. У середовищі вивчення будь-якої мови програмування, зокрема й Python, розділено на кілька основних тем, які мають певну кількість вправ для досягнення високого рівня розуміння всіх питань теми, а також завдання у вигляді кінцевого проекту програми. Середовище надає тому, хто навчається, редактор коду й інтерактивний інтерпретатор для перевірки правильності виконання кожної вправи й завдання. Якщо

завдання виконано правильно, здійснюється перехід до наступної вправи або теми. Ресурси для навчання надаються безкоштовно, а можливість пройти тестування або створити власний проект є платними послугами.

Наголосимо, що використання під час виконання лабораторних робіт можливостей середовища Codecademy переслідує ще одну навчальну мету – вивчення професійної лексики англійською мовою. Адже середовище є англійським і вправи також подаються англійською мовою. У процесі їх виконання студенти можуть не тільки вивчати мову програмування, а й удосконалювати свої знання з англійської мови шляхом розширення словникового запасу й занурення в професійну лексику англійською мовою. Це, своєю чергою, дасть змогу їм у майбутньому бути більш конкурентоспроможними на ринку праці.

Завдання для самостійної роботи передбачають вивчення теоретичного матеріалу й виконання практичних завдань з таких тем: «Наслідки тривалої роботи за комп'ютером», «Сертифікація IT-фахівців», «Оформлення резюме», «Оформлення портфоліо», «Презентування доповіді за результатами наукового дослідження». Для забезпечення самостійної роботи студентів із огляду на досить великий фактичний обсяг інформації для розуміння й засвоєння ми розробили електронний навчально-методичний комплекс у системі дистанційного навчання МДПУ ([dfn.mdpu.org.ua](http://dfn.mdpu.org.ua)).

**Орієнтовна тематика лабораторно-практичних занять  
і розподіл годин з дисципліни «Вступ до спеціальності»**

№ п/п	Теми	Питання	К-ть годин
Модуль 1			
1.	Working It blockly games (Робота з blockly games)	Ознайомлення з Blockly. Виконання завдань на Blockly games	2
2.	Python syntax (Синтаксис Python)	Алфавіт мови Python, концепція даних, поняття про лексику, прагматику, синтаксис і семантику мови	2
3.	Strings and console output (Рядки й виведення в констоль)	Типи даних у Python, функції введення й виведення	2
4.	Conditionals and control flow (Умовні оператори й управління потоком)	Поняття умови, логічні оператори, умовний оператор	2
5.	Functions (Функції)	Створення власних функцій і їх виклик	2
6.	Lists & dictionaries (Списки та словники)	Вбудовані складники типів даних, кортежі, словники, списки	2
7.	Student becomes the teacher («Студент стає вчителем»)	Розробка програми з використанням різних типів даних, умов, функцій, списків	2
8.	Lists and functions (Списки та їх функції)	Використання списків і функцій для розробки програми «Морський бій»	2
9.	Loops (Цикли)	Оператор циклу з параметром, циклічні програми, ітераційні цикли	2
10.	Advanced topics in Python (Додаткові питання Python)	Перебирання й сортування списків, лімба-вирази, бітові операції	2
11.	Introduction to classes (Уведення в класи)	Основи об'єктно-орієнтованого програмування у Python, оголошення класів, властивості об'єктів	2
12.	File input and output (Уведення й виведення файлів)	Основні завдання обробки файлів, запис даних у файл, виведення даних з файлу	2
Модуль 2			
13.	Вивчення організаційних основ наукових досліджень	Структура наукового дослідження, види науково-дослідної роботи студентів, аналіз результатів наукових досліджень	2
14.	Аналіз стану досліджуваної проблеми	Послідовність і особливості пошуку й обробки інформації, пошук і обробка літературних джерел за проблемою дослідження	2
15.	Методологія пошуку інформації в бібліотеці	Методи пошуку й збирання наукової інформації, особливості використання каталогів бібліотек, електронні бібліотеки	2
16.	Пошук інформації в мережі Інтернет	Методи й способи пошуку інформації в мережі Інтернет, способи узагальнення інформації	2
17.	Формулювання наукового апарату дослідження	Науковий апарат дослідження, засоби обґрунтування актуальності, об'єкта, предмета, мети, завдань і гіпотези дослідження	2
18.	Підготовка та проведення опитування	Особливості використання опитувальних методів, правила складання анкет і опитувальників	2
19.	Оформлення наукової роботи й літературних джерел	Правила оформлення наукової роботи, вимоги до оформлення літературних та Інтернет-джерел	2

Відповідно до плану роботи МДПУ імені Богдана Хмельницького на 2015-2016 н. р., дисципліна викладалася студентам I курсу факультету інформатики, математики та економіки (напрямок підготовки – «Інформатика») у першому семестрі (1.09-15.12)

й охоплювала два модулі: «Вступ до спеціальності програміста», «Основи наукових досліджень».

Після апробації розробленого нами змісту дисципліни «Вступ до спеціальності» ми провели опитування для отримання зворотного

відгуку щодо реалізованих нововведень у процесі її викладання. Студентам були поставлені запитання, що стосувалися досвіду у сфері програмування, складності теоретичного матеріалу й практичних завдань, труднощів у процесі засвоєння дисципліни, результату вдосконалення знань англійської мови, а також від них очікувалися зауваження й побажання щодо подальшого викладання дисципліни.

Аналізуючи результати опитування, ми дійшли таких висновків: 75% студентів до навчання в університеті не мали досвіду з програмування, 62,5% відзначають великий обсяг теоретичного матеріалу; незначна кількість опитуваних (25%) наголошують на його складності. Цікавим є той факт, що для більшості майбутніх інженерів-програмістів (62,5%) найбільш важкими стали матеріали другого модуля. Проте, незважаючи на це, 50% студентів вважають, що вони засвоїли матеріал і першого й другого модулів на однаковому рівні. 25% опитуваних відповіли, що завдання вправ з програмування англійською мовою не були для них складними, 37,5% студентів вказали на незначні труднощі, а решті було досить важко зрозуміти зміст таких завдань. Але попри деякі труднощі, спричинені недостатнім розумінням завдань англійською мовою, більшість студентів (62,5%) визначила їх як завдання середньої складності. Що ж до

завдань другого модулю, то переважна більшість студентів (87,5%) вважає їх завданнями середнього рівня складності.

Аналіз питань, пов'язаних з використанням вправ англійською мовою, засвідчив, що для більшості студентів було складно зрозуміти завдання, до того ж особливих успіхів у поповненні словникового запасу професійними термінами англійською мовою у них виявлено не було. Незначна кількість опитуваних (25%) змогла назвати кілька нових слів, які були вивчені в процесі виконання лабораторних робіт. Такі результати змушують замислитися над подальшим удосконаленням процесу вивчення професійної лексики англійською мовою шляхом співпраці з викладачами англійської мови.

Наприкінці вивчення дисципліни студенти ознайомилися з традиційною схемою викладання дисципліни, відображеною в працях В. Гвоздєєвої та З. Сейдаметової. Після чого було поставлене запитання: яку схему викладання вони вважають найкращою для свого професійного розвитку? За результатами опитування 75% студентів вважають найкращою нову схему вивчення дисципліни «Вступ до спеціальності», що передбачає оновлений зміст і використання нових засобів навчання (таких, як Codecademy) (рис. 1).

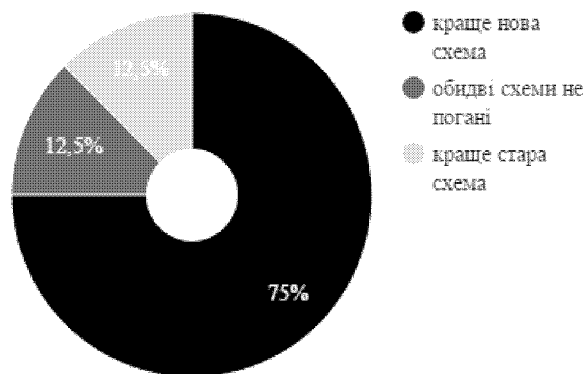


Рис. 1. Результати опитування стосовно оновленої схеми вивчення дисципліни

Також ми проаналізували результати навчання студентів з дисципліни «Вступ до спеціальності» на основі вхідного й фінального контролю. Вхідний контроль проводився перед початком вивчення дисципліни й охоплював перелік питань у тестовому вигляді з усіх тем. За результатами вхідного контролю 30% студентів отримали оцінку «добре», 50% – «задовільно», 20% – «незадовільно». За результатами фінального контролю, який було проведено у формі іспиту, завдання до якого містили теоретичні питання й практичні завдання з двох модулів, 30% студентів отримали оцінку «відмінно», а

70% – «добре». Порівнявши дані вхідного контролю й результати підсумкового іспиту можемо дійти висновку про те, що рівень знань студентів підвищився за рахунок удосконалення змісту дисципліни, добору засобів і прийомів навчання, спрямованих на формування професійної компетентності майбутніх інженерів-програмістів, орієнтованих на завдання, які студентам потрібно буде виконувати в процесі професійної діяльності.

Висновки. Аналіз результатів опитування студентів І курсу напряму підготовки «Інформатика» стосовно змісту та якості викладання дисципліни «Вступ до спеціальності»

свідчить, що дібрана тематика лекційного матеріалу є вдалою, а практичні завдання – доцільними, хоча використання матеріалів англійською мовою не дало значних результатів у вивченні професійної лексики, що потребує подальшого вдосконалення прийомів навчання спільно з викладачами англійської мови. Аналіз результатів навчання показав, що якість знань з дисципліни становить 100%, а ступінь навченості – 74,8%, що також підтверджує

доцільність дібраного нами змісту цієї дисципліни. Перспективу подальших досліджень вбачаємо в удосконаленні процесу вивчення професійної лексики англійською мовою шляхом співпраці з викладачами дисципліни «Іноземна мова за професійним спрямуванням»: перегляд обсягу теоретичного матеріалу, розробка індивідуальних завдань для організації диференційованого навчання.

**Список використаних джерел**

1. Гвоздева В. А. Введение в специальность программиста: учебник. – [2-е изд., испр. и доп.]. / В. А. Гвоздева. – М.: ИД «ФОРУМ»: ИНФРА-М, 2007. – 208 с.
2. Колесников О. В. Основи наукових досліджень: навч. посібник / О. В. Колесников. – К.: Центр учбової літератури, 2011. – 144 с.
3. Круглик В. С. Основы алгоритмизации и программирования на языке Python: учеб. пособие / Круглик В. С., Львов М. С., Спиваковский А. В. – Херсон: Айлант, 2015. – 252 с.
4. Осадчий В. В. Анализ проблемы профессиональной подготовки программиста и пути ее решения / В. В. Осадчий, Е. П. Осадчая // Образовательные технологии и общество. – Казань, 2014. – № 3 (17). – С. 362–378.
5. Осадчий В. В. Вступ до спеціальності програміста. Навчальний посібник / Осадчий В. В., Осадча К. П., Сердюк І. М. – Мелітополь: ТОВ «Видавничий будинок ММД», 2011. – 296 с.
6. Основи наукових досліджень: конспект лекцій / укл. Е. В. Колісниченко. – Суми: Сумський державний університет, 2012. – 83 с.
7. Сейдаметова З. С. Методическая система уровневой подготовки будущих инженеров-программистов по специальности «Информатика»: дис... д-ра пед. наук: 13.00.02 / Сейдаметова Зарема Сейдалиевна. – К., 2007. – 546 с.
8. Сурмін Ю. П. Майстерня вченого: підручник для науковця / Ю. П. Сурмін. – К.: Навчально-методичний центр «Консорціум з удосконалення менеджмент-освіти в Україні», 2006. – 302 с.
9. Computer Science 2013: Curriculum Guidelines for Undergraduate Programs in Computer Science [Електронний ресурс]. – Режим доступу: URL: <http://www.acm.org/education/CS2013-final-report.pdf> (20. 03. 2016).
10. Wortham J. Codecademy Offers Free Coding Classes for Aspiring Entrepreneur [Електронний ресурс] / J. Wortham // The New York Times. – Режим доступу: URL: <http://bits.blogs.nytimes.com/2011/09/14/codecademy-offers-free-coding-classes-for-aspiring-entrepreneurs> (20. 03. 2016).

**Рецензент:** Сегеда Н.А.– д.пед.н., професор

**Відомості про автора:**

**Круглик Владислав Сергійович**  
okp@mdpu.org.ua

Мелітопольський державний педагогічний університет імені Богдана Хмельницького  
вул. Гетьманська, 20, м. Мелітополь,  
Запорізька обл., 72312, Україна  
doi: <http://dx.doi.org/10.7905/nvmdpu.v0i16.1385>

Матеріал надійшов до редакції 25. 03. 2016 р.  
Прийнято до друку 26.04.2016 р.

**References**

1. Hvozdeeva, V. A. (2007). *Introduction to the specialty of a programmer: textbook. 2nd ed., corrected and improved.* Moscow: ID "FORUM": INFRA-M. [in Russian]
2. Kolesnykov, O. V. (2011). *Basics of scientific research: teaching guide.* Kyiv: Centre of educational literature. [in Ukrainian]
3. Kruhlyk, V. S., Lvov, M. S., Spivakovskii, A. V. (2015). *Fundamentals of algorithmic and programming language to Python: textbook.* Kherson: Ailant. [in Russian]
4. Osadchiy, V. V., Osadcha, Ye. P. (2014). Analysis of the problem of a programmer's professional training and the ways of its solution. *Obrazovatelnyie tekhnologii i obshchestvo.* Kazan, №3 (17), 362-378. [in Russian]
5. Osadchyi, V. V., Osadcha, K. P., Serdiuk, I.M. (2011). *Introduction to the specialty of a programmer. Textbook.* Melitopol: LLC "Publishing house MMD". [in Ukrainian]
6. Kolisnichenko, Ye. V. (2012). *Basics of scientific research: synopsis of lectures.* Sumy: Sumy State University. [in Ukrainian]
7. Seydametova, Z. S. (2007). *Methodical system of the level training of urovnevoy future engineers-programmers to specialty "Informatics": thesis for the degree of Doctor of Pedagogical Sciences: 13.00.02.* [in Russian]
8. Surmin, Yu. P. (2006). *Workshop of a scientist: handbook for a scientist.* Kyiv: Training Centre "Consortium to improve management-education in Ukraine". [in Ukrainian]
9. *Computer Science 2013: Curriculum Guidelines for Undergraduate Programs in Computer Science.* Retrieved from: URL: <http://www.acm.org/education/CS2013-final-report.pdf>. [in English]
10. Wortham, Jenna. *Codecademy Offers Free Coding Classes for Aspiring Entrepreneur.* *The New York Times.* Retrieved from: URL: <http://bits.blogs.nytimes.com/2011/09/14/codecademy-offers-free-coding-classes-for-aspiring-entrepreneurs>. [in English]