



DOI 10.31110/2413-1571-2022-035-3-007

УДК 378.147 : 004

МЕТОДИЧНІ АСПЕКТИ ВИКЛАДАННЯ ДИСЦИПЛІНИ "ТЕХНОЛОГІЇ СТВОРЕННЯ ПРОГРАМНИХ ПРОДУКТІВ"

Вікторія ЛОГВІНЕНКО ✉

Сумський національний аграрний університет, Суми, Україна
 2014lv@gmail.com
<https://orcid.org/0000-0003-0993-0821>

METHODICAL ASPECTS OF TEACHING THE DISCIPLINE "TECHNOLOGIES OF CREATING SOFTWARE PRODUCTS"

Viktoriia LOGVINENKO ✉

Sumy National Agrarian University, Sumy, Ukraine
 2014lv@gmail.com
<https://orcid.org/0000-0003-0993-0821>

АНОТАЦІЯ

Формулювання проблеми. Технології створення програмних продуктів є дуже важливою дисципліною для вивчення майбутніми фахівцями з інформаційних систем та технологій. Про високий рівень їх підготовки, можна говорити в тому випадку, коли ця підготовка спирається на методичну систему вивчення технологій створення програмних продуктів та є усвідомлення викладачем методичних аспектів викладання цієї дисципліни.

Матеріали і методи. Аналіз та систематизація навчальної та науково-методичної літератури; узагальнення досвіду викладання комп'ютерних дисциплін.

Результати. Розроблено методичну систему вивчення "Технології створення програмних продуктів". Описано змістовно-тематичні лінії дисципліни. Окреслено методичні аспекти викладання «Технології створення програмних продуктів» для майбутніх фахівців з інформаційних систем та технологій: дотримання розглянутої методичної системи; організація теоретичного матеріалу для вивчення послідовними змістовно-тематичними лініями; надання можливості студентам зосередити увагу на головних аспектах з розробки програмних продуктів; врахування й озброєння студентів системою знань про основні аспекти майбутньої діяльності; використання завдань практичного характеру, наближених до фахової діяльності; застосування під час практичних та самостійних робіт інструментарію технологій створення програмних продуктів.

Висновки. Оскільки "Технології створення програмних продуктів" є базовою дисципліною при підготовці комп'ютерних фахівців у вищій школі, тому її викладання повинно ґрунтуватися відповідно до розглянутої методичної системи. Розглянуті змістовно-тематичні лінії викладання можна доповнювати, модифікувати та оновлювати з урахуванням галузевих тенденцій. Подальше дослідження буде спрямовано на методичних аспектах викладання різних технологій створення програмних продуктів.

КЛЮЧОВІ СЛОВА: підготовка фахівців з інформаційних систем та технологій; технології створення програмних продуктів; методична система; зміст навчання; аспекти викладання.

ABSTRACT

Formulation of the problem. Technologies for creating software products are a very important educational component to study by future information systems and technology professionals. The high level of their training can be said in the case when this training is based on the methodical system of studying the technology of creating software products and the teacher is aware of the methodical aspects of teaching this discipline.

Materials and methods. Analysis and systematization of educational and scientific-methodical literature; generalization of experience in teaching computer science.

Results. A methodical system for studying "Technologies of creating software products" has been developed. Content-thematic lines of the discipline are described. The methodical aspects of teaching "Technologies of creating software products" for future specialists in information systems and technologies are outlined: compliance with the considered methodical system; organization of theoretical material for study by consistent content-thematic lines; enabling students to focus on the main aspects of software development; taking into account and equipping students with a system of knowledge about the main aspects of future activities; using of tasks of a practical nature that are close to professional activities; application during practical and independent work of tools of technologies of creating software products.

Conclusions. Since "Technologies of creating software products" is a basic discipline in the training of computer specialists in higher education, its teaching should be based on a methodical system. The considered content-thematic lines of teaching can be supplemented, modified, and updated based on industry trends. Further research will focus on the methodical aspects of teaching different technologies for creating software products.

KEYWORDS: training of specialists in information systems and technologies; technology of creating software products; methodical system; content of training; aspects of teaching.

ВСТУП

Постановка проблеми. Інформаційні системи та технології є вагомим галуззю сучасної економіки України. Підготовка майбутніх фахівців з інформаційних систем та технологій, здатних розробляти та впроваджувати нові програмні продукти та враховувати певні вимоги щодо якості програмного забезпечення, що створюється, є одним із завдань вищої ІТ-освіти України. Принципового значення в такій підготовці надається тому, щоб заклад вищої освіти давав своїм випускникам здобути необхідні фахові компетенції.

Чинна освітня програма підготовки комп'ютерних фахівців у вищій школі створює основу для вивчення відповідних дисциплін, забезпечує досягнення певного рівня фахової підготовки, формує загальні й професійні компетентності для розробки, впровадження й дослідження інформаційних систем та технологій.

Вивчення технологій створення програмних продуктів є дуже важливою дисципліною для вивчення майбутніми фахівцями з інформаційних систем та технологій. Самі технології створення програмних продуктів включають широке коло комп'ютерних програм та систем, які можна розділити на технології планування розробки програмних продуктів,

Для цитування:

Логвіненко В. Методичні аспекти викладання дисципліни "Технології створення програмних продуктів". *Фізико-математична освіта*, 2022. Том 35. № 3. С. 50-58. DOI: 10.31110/2413-1571-2022-035-3-007

Логвіненко, В. (2022). Методичні аспекти викладання дисципліни "Технології створення програмних продуктів". *Фізико-математична освіта*, 35(3), 50-58. <https://doi.org/10.31110/2413-1571-2022-035-3-007>

For citation:

Logvinenko, V. (2022). Methodical aspects of teaching the discipline "Technologies of creating software products". *Physical and Mathematical Education*, 35(3), 50-58. <https://doi.org/10.31110/2413-1571-2022-035-3-007>

Logvinenko, V. (2022). Metodichni aspekty vykladannia dystsyplyny "Tekhnologii stvorennia prohramnykh produktiv" [Methodical aspects of teaching the discipline "Technologies of creating software products"]. *Fiziko-matematichna osvita – Physical and Mathematical Education*, 35(3), 50-58. <https://doi.org/10.31110/2413-1571-2022-035-3-007>

технології проектування програмних продуктів, технології реалізації програмних продуктів, технології тестування. Вважаю, що говорити про високий рівень підготовки майбутніх фахівців спеціальності 126 “Інформаційні системи та технології”, можна в тому випадку, коли ця підготовка спирається на методичну систему вивчення технологій створення програмних продуктів.

Аналіз актуальних досліджень. Проблеми підготовки ІТ-фахівців присвячено широке коло досліджень: актуальні стратегії та підходи до підготовки ІТ-спеціалістів (Малихін & Ярмольчук, 2020; Власюк & Грицюк, 2013); моделі підготовки майбутнього ІТ-фахівця в університетах аграрного профілю (Глазунова, 2014); аналіз ринку ІТ-спеціальностей щодо професій та спеціалізацій в цій сфері (Острога, 2019); підготовка майбутніх фахівців з інформаційних технологій до творчої самореалізації (Роценюк, 2019).

Перегляд й аналіз наявних стандартів щодо підготовки відповідних фахівців та робочих програм із дисципліни “Технології створення програмних продуктів” за спеціальностями галузі знань 12 Інформаційні технології показує, що навчальна дисципліна є обов’язковою при підготовці майбутніх фахівців комп’ютерного напрямку. Так у професійному стандарті фахівця з розробки програмного забезпечення вказується, що “трудові функції фахівця з розробки програмного забезпечення вводяться на базі процесів життєвого циклу програмних засобів” і перелічуються основні задачі діяльності фахівця з розробки програмного забезпечення (“Професійний стандарт”, 2014). Але в деякі дослідники вказують на “проблему швидкого застарівання технологічного змісту навчання, розв’язання якої полягає у його фундаменталізації через виокремлення базових основ галузі” (Стрюк, 2018). Базові основи технологій створення програмних продуктів розглядаються багатьма науковцями (Алексенко, 2018). Втім неоднозначною є і сама проблема вибору технологій створення програмного забезпечення інформаційних систем (Асеева & Кулаковська, 2019). Все це спонукає викладача комп’ютерних наук до пошуку шляхів застосування та вивчення різних комп’ютерних технологій при їх викладенні майбутнім фахівцям.

Отже, на основі вищевикладеного, **метою** даної статті є висвітлення деяких методичних аспектів до викладання технологій створення програмних продуктів для бакалаврів спеціальності 126 – Інформаційні системи та технології.

МЕТОДИ ДОСЛІДЖЕННЯ

У статті використано наступні методи досліджень: аналіз та систематизація навчальної та науково-методичної літератури; узагальнення досвіду викладання комп’ютерних дисциплін.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Методологічною основою методики вивчення дисципліни «Технології створення програмних продуктів» є системний підхід. Методична система ґрунтується на авторському дослідженні (Логвіненко, 2005) та включає наступну структуру (рис. 1):

- виконання аналізу освітніх програм спеціальності 126;
- уточнення цілей вивчення “Технології створення програмних продуктів” за технологією Блума та Кратволя (Кларин, 1988), що подано у таблиці 1 та таблиці 2;
- виконання аналізу змісту дисципліни “Технології створення програмних продуктів”, конструювання навчальних матеріалів за дисципліною;
- виконання діагностики та аналізу базового рівня знань студентів перед вивченням даної дисципліни;
- вибір технології (або способів) та методів викладання даної дисципліни: мотиваційні технології, технології формування нових знань, технології засвоєння діяльності, технології контролю знань студентів;
- організація процесу навчання на підґрунті цілей вивчення дисципліни “Технології створення програмних продуктів”;
- виконання контролю сформованих знань та умінь, і в залежності від результатів виконання коригування як навчального матеріалу, так і технологій викладання за дисципліною.

Зосередимо увагу на змісті навчання технологій створення програмних продуктів. Він визначається освітньо-професійною програмою “Інформаційні системи та технології” для першого (бакалаврського) рівня вищої освіти галузі знань 12 Інформаційні технології за спеціальністю 126 – Інформаційні системи та технології (“ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА “Інформаційні системи та технології”; 2021). Ця програма забезпечує міцне і свідоме оволодіння студентами “загальних та професійних компетентностей, необхідних для вирішення актуальних інженерних завдань та практичної реалізації отриманих знань в галузі інформаційних систем та технологій, передбачає здобуття спеціальних професійних знань, оволодіння методологією розробки, впровадження й дослідження прикладних інформаційних систем та технологій в економіці, в т.ч. в аграрному секторі, на базі використання методів економіко-математичного моделювання, технологій програмування, новітніх інформаційних технологій та сучасного прикладного програмного забезпечення”. Поряд з цим вивчення технологій створення програмних продуктів має свою специфічну мету, свої конкретні завдання.

Навчальна дисципліна “Технології створення програмних продуктів” синтезує та узагальнює вивчення сучасних методів і засобів створення програмних продуктів для подальшої професійної діяльності, набуття навичок практичної роботи з програмними засобами для створення програмних продуктів. Він передбачає вивчення майбутніми фахівцями теоретичної та методологічної бази даного напрямку діяльності. У процесі навчання студенту надаються знання деяких основних технологій зі створення програмних продуктів та практичні навички їх реалізації. Мета дисципліни “Технології створення програмних продуктів” полягає у формуванні й засвоєнні знань про спеціальні технології організації та проведення процесу розробки програмного забезпечення; у засвоєнні знань з основ створення великих програмних систем з використанням процесів життєвого циклу, починаючи з виявлення вимог і закінчуючи його супроводом програмного забезпечення. Можна вказати на те, що створення програмного продукту або системи – це процес, що включає різні трудові завдання, і майбутній фахівець повинен мати уяву про методи аналізу, проектування, реалізації та тестування програмних систем, орієнтуватися в тенденціях та наявних підходах до технологій розробки програмних продуктів.

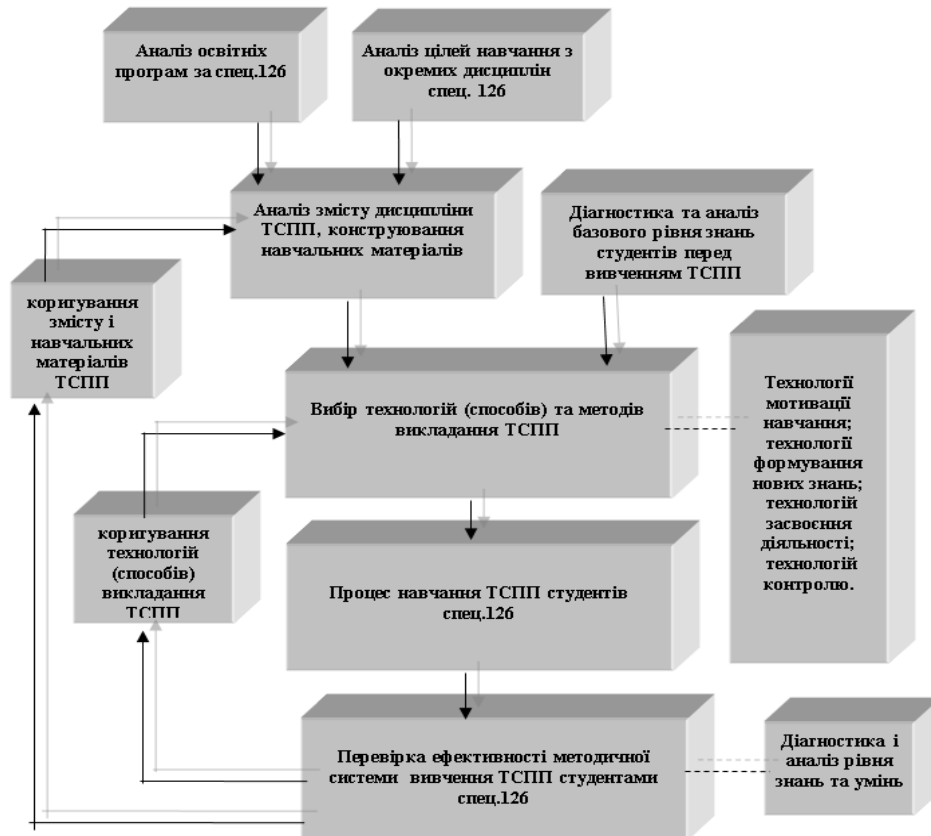


Рис. 1. Структура методичної системи вивчення ТСПП студентами спеціальності 126 “Інформаційні системи та технології”

Таблиця 1

Опис цілей вивчення “Технології створення програмних продуктів” (ТСПП) у когнітивній області через діяльність студента

Категорія	Дії студента
1. Знання	студент знає і відтворює основні поняття, терміни, методи, правила і принципи основних технологій за дисципліною ТСПП; студент знає етапи розробки програмних продуктів (і їх суть) технологічних процедур; студент знає алгоритми роботи, що відтворюють порядок розробки нового програмного продукту.
2. Розуміння	студент розуміє основні поняття, терміни, методи, правила і принципи основних технологій за дисципліною ТСПП; студент інтерпретує схеми, алгоритми роботи із відповідними програмними засобами; студент перетворює словесний матеріал на мову схем і на мову програмування, і навпаки; студент розуміє суть застосування технологій з курсу ТСПП; студент розуміє суть методів, що використовуються для створення нового програмного продукту; студент розуміє суть технологічних процедур.
3. Застосування	студент пояснює використання основ з курсу ТСПП; студент використовує поняття, терміни, правила, методи і принципи при виконанні практичних та самостійних робіт; студент демонструє правильне застосування методів і процедур, відомих алгоритмів дій; студент застосовує певний інструментарій технології з курсу ТСПП.
4. Аналіз	студент здійснює аналіз початкових вимог щодо створення нового програмного продукту; студент визначає причинно-наслідкові зв'язки в завданнях з курсу; студент виконує висновки після виконання завдань; студент зіставляє відмінності застосування різних технологій розробки; знаходить (бачить) помилки в логіці вирішення наданих завдань; студент зіставляє застосування різних технологій при вирішенні певного класу завдань.
5. Синтез	студент складає план або алгоритм розробки нового програмного продукту; студент виконує постановку завдання; студент має навички грамотної постановки задач для їх розв'язання за допомогою комп'ютера; студент має вміння формалізованого опису поставлених задач; студент має вміння побудови моделей вирішення поставлених завдань; студент пропонує використання знань зі споріднених областей знань.

Категорія	Дії студента
6. Оцінка	студент оцінює логіку розроблення нового програмного продукту; студент оцінює одержані результати виходячи з початкових даних, студент оцінює відповідні висновки; студент оцінює значущість свого продукту навчальної діяльності; студент оцінює застосування різних технологій при вирішенні певного класу завдань.

Таблиця 2

**Опис цілей вивчення “Технології створення програмних продуктів” (ТСПП)
в афективній області через діяльність студента**

Категорія	Дії студента
1. Сприйняття	студент усвідомлює важливість навчання, важливість вивчення курсу ТСПП, усвідомлює свою реалізацію в навчанні як чинника професійного зростання; студент уважно слухає матеріал в аудиторії; студент уважно вивчає методичні матеріали, за якими працює.
2. Реагування	студент виконує завдання за методичними матеріалами курсу ТСПП; студент підкоряється внутрішньому розпорядку і правилам поведінки; студент добровільно викликається виконувати завдання; студент виявляє цікавість до предмету та до завдання, що виконує; студент задоволений процесом виконання завдання.
3. Засвоєння ціннісної орієнтації	студент проявляє стійке бажання оволодіти вміннями з курсу ТСПП; вирішувати завдання із застосуванням відповідних технологій; студент проявляє переконаність, відстоюючи застосування того або іншого способу дій, методу або застосування певного програмного засобу при вирішенні навчальних завдань.
4. Організація ціннісної орієнтації	студент бере на себе відповідальність за свою поведінку, за виконання отриманих завдань з курсу ТСПП; студент розуміє свої можливості при вивченні певної теми з курсу ТСПП; студент розуміє значення своєї реалізації в навчанні, в майбутній професійній діяльності; студент проявляє готовність до виконання завдань з курсу ТСПП; студент проявляє прагнення до співпраці в групі; студент планує свою роботу над завданням відповідно до своїх здібностей, потреб.

Таким чином, завданням “Технології створення програмних продуктів”, є: забезпечення належного теоретичного рівня навчання і формування в студентів застосовувати набуті знання; формування умінь, що включають як знання теоретичних основ дій, знання про способи виконання дій, їх змісту і послідовності, призначення необхідних приладів, інструментів та програмного забезпечення, навичок роботи з ними, практичний досвід виконання аналогічних дій.

Розглянемо детально й послідовно змістовно-тематичні лінії викладання дисципліни “Технології створення програмних продуктів”.

Перша лінія даного курсу – “Вступ у технології програмування”. Це початок вивчення даної дисципліни, де студенти опановують базові поняття та види програмного забезпечення, історичні аспекти розроблення програмного забезпечення. Викладачем виконується акцент на тлумаченні поняття “технології програмування” як процесу, узагальнюються відомості, отримані з інших споріднених дисциплін. Також формуються основні поняття, твердження і методи та зв’язки між ними, наголошується практичне значення курсу, що вивчається.

Друга лінія даного курсу – “Моделі життєвого циклу програмного забезпечення”. В ній йде ознайомлення й аналіз важливого поняття “життєвий цикл програмного забезпечення”, обов’язково враховуючи галузевий стандарт (“IEEE Standard Glossary of Software Engineering Terminology”, 1990). Розглядаються також питання щодо аналізу вимог замовником до програмного продукту, питання розробки технічного завдання та стандарту (“Информационная технология. Комплекс стандартов на автоматизованные системы”, 1990), який нещодавно використовувався, і, на наш погляд, також є корисним для ознайомлення студентів різним підходам до розробки програмних продуктів. Також студенти ознайомлюються із характеристиками каскадної, ітеративної й спіральної моделей життєвого циклу програмного забезпечення, розглядаються їх недоліки та переваги.

Третя лінія даного курсу - «Планування та управління процесом розроблення та супроводу програмного забезпечення», де ретельно вивчаються особливості процесом керування та організації робіт з розроблення програмного забезпечення, пояснюються питання ролей процесу, умінь, координації роботи, питання комунікації між розробниками, забезпечення якості програмного забезпечення. В цьому матеріалі, який безперечно пов’язаний із попередньою предметною лінією, увагу студентів акцентують на послідовність етапів розробки програмних продуктів: аналіз вимог замовника та дослідження предметної області, функціональна специфікація нового програмного продукту, технічне завдання на нього, проектування та моделювання нової програмної розробки, реалізація або кодування, тестування та експлуатація. На нашу думку, ця тема потребує найбільшої уваги, тому що розглядає цілісність процесу створення програмних продуктів і важливість кожного етапу розробки програмного забезпечення.

Четверта лінія даного курсу – «Стандарти на розроблення та супровід програмного забезпечення». В ній розглядаються питання стандартизації розроблення програмного забезпечення, міжнародні стандарти ISO (International Organization for Standardization) та стандарти організації IEEE (Institute of Electrical and Electronics Engineers). Основними документами, що регламентують процес розроблення програмного забезпечення є державний стандарт щодо

розроблення систем та програмного забезпечення ДСТУ ISO/IEC/IEEE 16326:2015 ("Розроблення систем та програмного забезпечення", 2015), міжнародний стандарт на процеси розроблення та організації життєвого циклу програмного забезпечення ISO/IEC/IEEE 12207:2017 ("Systems and software engineering", 2017) та галузевий стандарт Інституту інженерів з радіоелектроніки та електротехніки (IEEE), що систематизує основні види діяльності з програмної інженерії, Guide to the Software Engineering Body of Knowledge або SWEBOK (Bourque & Fairley, 2014). І безперечно, знання основних стандартів сфери інформаційних технологій та управління IT-послугами є фундаментом вивчення даної дисципліни, є ключем до розуміння підходів до управління IT-послугами, а також ключем розуміння до організації, контролю й вдосконалення цієї діяльності. Змістовне розкриття основних стандартів, пов'язаних з ними понять, тверджень і наголошення на практичному значенні використання вказаних стандартів забезпечує обов'язковий мінімум знань зі стандартизації програмних продуктів майбутніми фахівцями.

П'ятою лінією даного курсу є "Сучасні методології розроблення програмних систем. Візуальне моделювання мовою UML (Unified Modeling Language) предметної області". В ній студенти знайомляться та опановують матеріал щодо інструментарію проектування на мові UML (Буч та ін., 2008; Torre et al., 2018; Bernhard, 2016; Filho & Braga, 2017). Сюди відносяться знання таких основних понять як "спеціальна уніфікована мова моделювання", "сутності", "структурні сутності", "поведінкові сутності", "групування", "відносини", "залежності", "асоціації", "узагальнення", "реалізації", "діаграми", "структурні діаграми", "поведінкові діаграми", "діаграми варіантів використання", "діаграми взаємодії" та інші. Всі види сутностей й відносин UML у діаграмах подаються у свій спосіб графічного зображення. Зауважимо, що базові поняття мови UML дозволяють формувати вміння з проектування певних систем, програмних продуктів, бо для розуміння системи, що розробляється, потрібно застосовувати різні діаграми, які уточнюють конкретні аспекти цієї системи. Так діаграма використання – це концептуальна модель системи. Діаграма класів – логічна модель, в якій зображуються статичні аспекти всієї структури. Діаграма діяльності є теж логічною моделлю. Вона відтворює динамічні аспекти функціонування системи, яка розробляється. І отримання умінь з моделювання на мові UML охоплює знання теоретичних основ, знання правил та способів використання нотацій сутностей UML, призначення необхідного інструментарію візуалізації, знання з виконання основних алгоритмів дій при створенні моделей, вказаних вище. А формування умінь із проектування на мові UML дозволить майбутнім фахівцям бути готовими діяти в змінюваних, умовах, в нових практичних ситуаціях.

Шостою лінією даного курсу є "Методологія Rational Unified Process", за матеріалами якої студенти ознайомлюються з уніфікованим процесом розроблення програмного забезпечення, з вимогами до нього (Kruchten, 2004). Оскільки весь процес розроблення розподіляється за життєвим циклом програмного забезпечення, то велика увага спочатку приділяється "моделі варіантів використання", що описує повні функціональні вимоги до системи, що розробляється. Далі будується "модель аналізу", що є концептуальною моделлю системи. Вона детально описує усі вимоги та роботи, необхідні для реалізації. Наступним етапом є створення "моделі проектування", що відтворює функціональні та нефункціональні вимоги щодо системи, яка проектується, та "моделі розгортання", що дозволяє встановити реалізованість варіантів використання. Після цього настає черга до "моделі реалізації", що описує елементи "моделі проектування" як файли із кодом програм. І нарешті розглядається "модель тестування", в якій тестуються компоненти "моделі реалізації".

Сьомаю лінією даного курсу є "Методологія Microsoft Solution Framework (MSF). Методологія eXtreme Programming (XP)". Студенти опановують базові принципи MSF (Turner, 2006) та XP (Beck, 2005), ознайомлюються з моделями та основними методами. До базових принципів MSF, що відтворюють досвід груп-розробників програмних продуктів, відносяться: "сприяння відкритій комунікації", "робота у напрямку спільного бачення проєкту", "надання прав і можливостей членам команди", "визнання індивідуальної та спільної відповідальності", "зосередження на бізнес-цілях", "бути гнучкими, очікувати на зміни", "інвестування у якість", "навчання за досвідом". Розглядаючи XP, акцентується увага на те, що ця методологія є спрощеною методологією організації створення програмних продуктів для невеликих та середніх за розміром команд. Ця методологія включає методи щодо виконання чотирьох видів діяльності створення програмного продукту: кодування, тестування, слухання, проектування.

Восьмою лінією даного курсу є "Гнучке розроблення ПЗ на основі Agile. Патерни проектування при розробленні програмних систем", де розглядаються зі студентами принципи гнучкого розроблення програмного забезпечення, Agile-методи (Stelman & Greene, 2014) та деякі патерни проектування програмних систем (Gamma et al., 1995; Wedyan & Abufakher, 2020; Horstmann, 2006). Серед Agile-методів має велике застосування Scrum-методів (Sharma & Wadhwa, 2015). При ознайомленні з патернами проектування програмних систем студентам потрібно усвідомити, що на цей час вже розроблено досить велику кількість патернів в залежності від сфери використання, тому їх застосування при вирішенні кожного завдання – є визначення конкретного рішення, що пов'язане з певною проблемою.

Весь матеріал для викладання студентам може бути модифікований або доповнений, виходячи з останніх наукових розробок. Вище в загальних рисах схарактеризовано навчальний матеріал, який запропоновано до вивчення в даній дисципліні. Завдання викладача донести матеріал до студентів, сформувавши їх цілісне й системне уявлення про технології створення програмних продуктів. Проте теоретичні відомості за даним курсом не є самоціллю вивчення даної дисципліни.

Вважаємо, що важливим методичним аспектом дисципліни "Технології створення програмних продуктів" є залучення завдань практичного характеру, які пронизують теоретичний матеріал. Від студентів вимагається виконати практичні роботи, виконати надані в них приклади, застосувати отримані знання та вміння у завданнях самостійної роботи. Це дозволяє посилити теоретичну підготовку, активізувати мислення студентів, посилити розуміння ними предметної області та майбутньої професійної діяльності, сформувавши вміння аналізувати об'єкт проектування або його функціонування, сформувавши вміння оцінювати сукупність факторів, що мають вплив на життєвий цикл програмного забезпечення. І справа тут не тільки у так званому "перенесенні" знань і умінь у нові ситуації, які зазвичай

супроводжуються певними психологічними труднощами та відсутністю досвіду студентів із розробки програмних продуктів.

Вивчаючи конкретні теми та виконуючи конкретні практичні роботи, необхідно створювати основу для систематизації отриманих знань, розкривати всі зв'язки навчального матеріалу за даною дисципліною та іншими фаховими курсами, враховувати їх специфіку.

Розглянемо зміст деяких практичних робіт, що надаються студентам. Оскільки створення програмних продуктів підпорядковано стадіям життєвого циклу розробки програм, то студентами послідовно виконуються практичні роботи спочатку щодо аналізу вимог з розробки програмних продуктів, далі надається робота щодо функціональної розробки специфікації програмного продукту, після передбачено практичну роботу з розробки технічного завдання на програмне забезпечення, далі передбачається практична робота із ознайомленням технології для планування робіт по створенню програмних продуктів, після пропонуються декілька практичних робіт, що передбачають ознайомлення із технологією про проектування на мові UML, і т.д. Результатом виконання кожної практичної роботи є оформлення звіту, в якому студент показує результати своїх дій з опанування основних навичок щодо технологій створення програмних продуктів. Зауважуємо на те, що в процесі виконання практичних робіт акцент припадає на специфіку розробки програмних продуктів, а в цілому формується досвід використання відповідних технологій з дотриманням відповідних етапів розробки програмних продуктів. Вважаємо, що такий підхід, що спирається на приклади реальної дійсності, є базисом для опанування студентами навичок з розробки нових інформаційних систем та технологій.

Наведемо приклад практичної роботи "Планування та управління процесом розроблення програмного продукту в MS Project (Project Libre)". Після ознайомлення студентів з алгоритмами роботи у відповідних програмах їм пропонується одне із завдань: розробити проєкт зі створення сайту-візитки. Обговорюючи зі студентами це завдання, доходимо до висновку, що проєкт передбачає наступні етапи: а) створення контенту сайту: створення слайдера, наповнення сайту контентом, робота з віджетами, тест; б) розміщення сайту на хостингу: реєстрація домену, оплата хостингу, заливка сайту на хостинг. Далі студенти, використовуючи відповідне програмне забезпечення, реалізують планування процесу розроблення сайту у своїх практичних діях та отримують план роботи над власним проєктом з розробки сайту, що представлено на рис. 2.

		Название	Продолжи...	Начало	Окончание
1		☐ Створення контенту сайту	6 днів?	27.09.21 8:00	04.10.21 17:00
2		Створення слайдера	2 днів?	27.09.21 8:00	28.09.21 17:00
3		Наповнення сайту контентом	2 днів?	29.09.21 8:00	30.09.21 17:00
4		Робота з віджетами	2 днів?	01.10.21 8:00	04.10.21 17:00
5		Тест	0 днів?	04.10.21 8:00	04.10.21 8:00
6		☐ Розміщення сайту на хостингу	0,625 днів	04.10.21 8:00	04.10.21 14:00
7		Регістрація домену	0,25 днів	04.10.21 8:00	04.10.21 10:00
8		Оплата хостингу	0,25 днів	04.10.21 8:00	04.10.21 10:00
9		Заливка сайту на хостинг	0,625 днів	04.10.21 8:00	04.10.21 14:00

Рис. 2. Приклад результату планування проєкту із створення сайту

При виконанні практичних робіт за тематикою проектування основ UML студенти створюють різні діаграми, в т.ч. діаграми використання. Так, на прикладі проектування системи інтернет-магазину, вони створюють різні діаграми варіантів використання, в т.ч. діаграму для замовлення товарів в системі інтернет (рис. 3). На різних завданнях-прикладях зі студентами обговорюються "актори" інформаційної системи, їх функції, варіанти використання і т.д. Таким чином студенти виконують усвідомлено проектування системи, яка пропонується. А для закріплення даного матеріалу, студентам доцільно пропонувати різнопланові інші завдання вже для самостійного проектування.

Вважаємо, що принциповою вимогою до викладання "Технологій створення програмних продуктів" є озброєння студентів системою знань про основні аспекти майбутньої діяльності. Важливим є ґрунтовне засвоєння студентами матеріалу основних змістовно-тематичних ліній. Весь матеріал є доброю основою для розвитку в студентів чіткості та логічності мислення, та є знаряддям для розв'язування завдань практичного характеру, наближених до фахової діяльності. Дисципліну "Технології створення програмних продуктів" можна назвати опорним курсом для таких дисциплін як "Проектування інформаційних систем", "Управління ІТ-проєктами", "Інформаційні системи в бізнесі" та інші, що передбачені освітньою програмою.

Дуже важливо максимально використовувати можливості "Технологій створення програмних продуктів" для розвитку самостійного мислення, творчої активності й під час виконання самостійних робіт, що мають творчий характер. В результаті виконання таких завдань, долаючи певні труднощі, студенти отримують упевненість у своїх пізнавальних можливостях, привчаються до правильності, лаконічності й точності висловлювання своїх думок при роботі над "новим програмним продуктом".

Викладання основних технологій створення програмних продуктів та вивчення їх студентами спеціальності 126 – "Інформаційні системи та технології" забезпечує формування професійної компетентності студентів. Навчання й вивчення матеріалу кожної тематичної лінії курсу "Технології створення програмних продуктів" ґрунтується на нашій методологічній основі. Наше викладання втілює поєднання освітніх цілей навчання й фахового розвитку, узагальнення нових досягнень та підходів до розробки програмних продуктів. Викладання нами даного курсу передбачає вивчення алгоритмів роботи зі спеціалізованим програмним забезпеченням, яке охоплює життєвий цикл розробки програм.

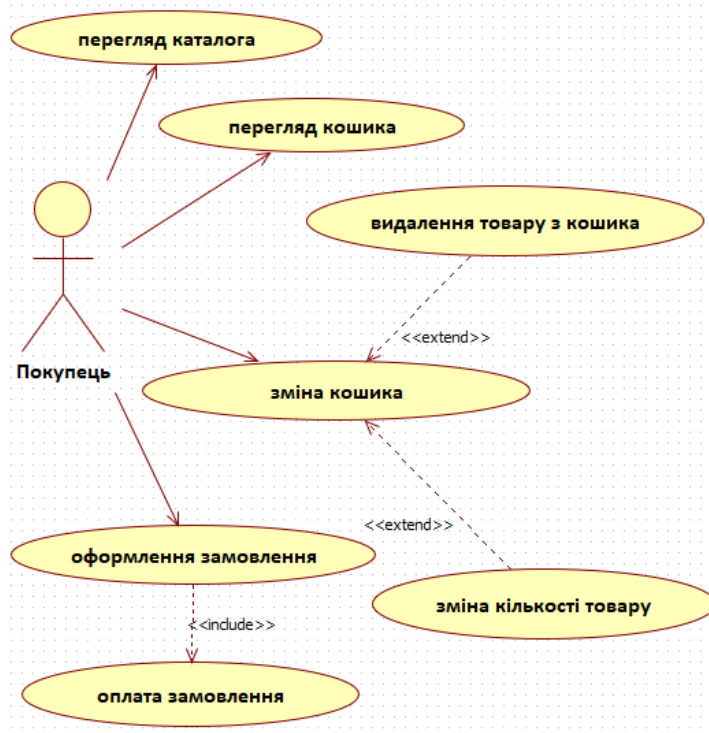


Рис. 3. Приклад результату проектування на мові UML: діаграма варіантів використання, що пояснює основний прецедент «замовлення товарів»

Таким чином, важливими методичними аспектами викладання “Технологій створення програмних продуктів” студентами спеціальності 126 “інформаційні системи та технології” є:

- дотримання методичної системи вивчення студентами даної дисципліни;
- організація теоретичного матеріалу даної дисципліни за послідовними змістовно-тематичними лініями, які дозволяють чітко узагальнювати та систематизувати зміст, що вивчається;
- надання можливості студентам зосередити увагу на головних аспектах з розробки програмних продуктів: аналіз предметної області, аналіз вимог замовник програмного продукту, проектування розробки, реалізація і т.д.;
- врахування й озброєння студентів системою знань про основні аспекти майбутньої діяльності;
- використання завдань практичного характеру, завдань наближених до фахової діяльності;
- застосування під час практичних та самостійних робіт інструментарію технологій створення програмних продуктів.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО ДОСЛІДЖЕННЯ

Оскільки “Технології створення програмних продуктів” є базовою дисципліною при підготовці комп’ютерних фахівців у вищій школі, тому її викладання повинно ґрунтуватися відповідно до розглянутої методичної системи. В даному дослідженні окреслено методичні аспекти викладання дисципліни, але детально розглянуто змістовно-тематичні лінії викладання, які можна доповнювати, модифікувати та оновлювати з урахуванням галузевих тенденцій.

Подальше дослідження буде спрямовано на методичних аспектах викладання різних технологій створення програмних продуктів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Beck, K. (2005). *Extreme programming explained: embrace change* (2nd ed). Pearson Education, Inc.
2. Bernhard, R.(2016). *Modeling with UML*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-33933-7>
3. Bourque, P., & Fairley, R. E. (Eds.). (2014). *SWEBOOK.V3. Guide to the Software Engineering Body of Knowledge*. IEEE Computer society. <https://cs.fit.edu/~kgallagher/Schtick/Serious/SWEBOOKv3.pdf>
4. Filho, J.L., Braga, J. (2017). UML: Unified Modeling Language. *Encyclopedia of GIS*. Springer, Cham. https://doi.org/10.1007/978-3-319-17885-1_1419.
5. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison-Wesley, Inc.
6. Horstmann, C. (2006). *Object-Oriented Design and Patterns*. Books by John Wiley Sons Inc. <https://manualzz.com/doc/47794372/object-oriented-design-and-patterns--2nd-edition>.
7. Kruchten, P. (2004). *The rational unified process : an introduction*. Boston : Addison-Wesley.
8. Malykhin, O. B., & Yarmolchuk, T. M. (2020). Topical strategies in the professional training for information technologies specialists. *Information Technologies and Learning Tools*, 76(2), 43–57. <https://doi.org/10.33407/itlt.v76i2.2682>.
9. Rasnacs, A., & Berzisa, S. (2017). Method for Adaptation and Implementation of Agile Project Management Methodology. *Procedia Computer Science*, 104, 43-50. <https://doi.org/10.1016/j.procs.2017.01.055>

10. Sharma, N., Wadhwa, M. (2015). eXSRUP: Hybrid Software Development Model Integrating Extreme Programming, Scrum & Rational Unified Process. *International Journal of Computer Sciences and Engineering*, 3(5), 383-392. https://www.ijcseonline.org/pdf_paper_view.php?paper_id=537&69-IJCSE-01035-3.pdf
11. Software Engineering. (1990). *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12-1990, 1-84. <https://doi.org/10.1109/IEEESTD.1990.101064>.
12. Stelman, A., & Greene, J. (2014). *Learning Agile*. O'Reilly Media, Inc.
13. Systems and software engineering. (2017). *Software Life Cycle Processes* (ISO/IEC/IEEE 12207:2017). International standard. <https://www.iso.org/standard/63712.html>.
14. Torre, D., Labiche, Y., Genero, M., Baldassarre, M. T., & Elaasar, M. (2018). UML Diagram Synthesis Techniques: A Systematic Mapping Study. *2018 IEEE/ACM 10th International Workshop on Modelling in Software Engineering (MiSE)*, 33-40. <https://ieeexplore.ieee.org/document/8445456/references#references>.
15. Turner, M. S. V. (2006). *Microsoft Solutions Framework Essentials: Building Successful Technology Solutions*. Microsoft Press.
16. Wedyan, F., & Abufakher, S. (2020). Impact of Design Patterns on Software Quality: A Systematic Literature Review. *IET Software*, 14(1), 1-17. <https://doi.org/10.1049/iet-sen.2018.5446>.
17. Алексенко, О. В. (2018). *Технології програмування та створення програмних продуктів: конспект лекцій*. Сумський державний університет.
18. Асєєва, А.В., & Кулаковська, І. В. (2019). Аналіз проблем вибору технології для розробки програмного забезпечення. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*, 37, 10-18. <https://doi.org/10.36910/6775-2524-0560-2019-37-2>.
19. Буч, Г., Рамбо, Д., & Якобсон, І. (2008). *Язык UML. Руководство пользователя*. (2-е изд.: Пер. с англ. Мухин Н.). М.: ДМК Пресс.
20. Власюк, А., & Грицюк, П. (2013). Підготовка фахівців з інформаційних технологій у контексті сучасних вимог. *Нова педагогічна думка*, (1.1), 109. http://nbuv.gov.ua/UJRN/Npd_2013_1_28.
21. Глазунова, О.Г. (2014). Модель підготовки майбутнього іт фахівця в університетах аграрного профілю в умовах глобалізації та євроінтеграції. *Вісник Національного університету оборони України*, 5 (42), 36-42.
22. Информационная технология. Комплекс стандартов на автоматизированные системы. (1990). *Техническое задание на создание автоматизированной системы* (ГОСТ 34.602-89). Межгосударственный стандарт. http://online.budstandart.com/ru/catalog/doc-page?id_doc=57803.
23. Кларин, М.В. (1988). Педагогическая технология в учебном процессе. Анализ зарубежного опыта. *Новое в жизни, науке и технике*. Сер. «Педагогика и психология», 6. М.: Знание.
24. Логвіненко, В.Г. (2005). *Методика формування пізнавальної самостійності студентів технічних спеціальностей в процесі вивчення інформаційно-комунікативних технологій* [Дис. канд. пед. наук]. Українська інженерно-педагогічна академія.
25. Освітньо-професійна програма «Інформаційні системи та технології». (2021). <https://eim.snau.edu.ua/kafedri/kibernetiki-ta-informatiki/osvitno-profesijni-programi/>.
26. Острога, М.М. (2019). Професійна підготовка в галузі ІТ: аналіз ринку ІТ-спеціальностей. *Фізико-математична освіта*, 4(22), частина 2, 52-57. <https://doi.org/10.31110/2413-1571-2019-022-4-034>.
27. Професійний стандарт. (2014). *Фахівець з розробки програмного забезпечення*. <https://mon.gov.ua/storage/app/media/vyshcha/IT-prof-standarty/6-ps-rozrobnik-pz-13.12.2014.pdf>.
28. Розроблення систем та програмного забезпечення. (2015). *Процеси життєвого циклу. Керування проектами* (ДСТУ ISO/IEC/IEEE 16326:2015). Державний Стандарт України. http://online.budstandart.com/ua/catalog/doc-page?id_doc=67052.
29. Рощенюк, А.М. (2019). *Підготовка майбутніх фахівців з інформаційних технологій до творчої самореалізації в адаптаційний період*. Рукопис. [Автореф. Дис. к-та пед. наук, Рівненський державний гуманітарний університет]. Репозитарій РГДУ. https://rshu.edu.ua/images/afto/avtoref_roshcheniuk.pdf.
30. Стрюк, А.М. (2018). Становлення та розвиток інженерії програмного забезпечення як галузі знань. *Інформаційні технології в освіті*, 4 (37), 103-136.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Beck, K. (2005). *Extreme programming explained: embrace change* (2nd ed). Pearson Education, Inc.
2. Bernhard, R. (2016). *Modeling with UML*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-33933-7>
3. Bourque, P., & Fairley, R. E. (Eds.). (2014). *SWEBOOK V3. Guide to the Software Engineering Body of Knowledge*. IEEE Computer society. <https://cs.fit.edu/~kagallagher/Schtick/Serious/SWEBOOKv3.pdf>
4. Filho, J.L., Braga, J. (2017). UML: Unified Modeling Language. *Encyclopedia of GIS*. Springer, Cham. https://doi.org/10.1007/978-3-319-17885-1_1419.
5. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison-Wesley, Inc.
6. Horstmann, C. (2006). *Object-Oriented Design and Patterns*. Books by John Wiley Sons Inc. <https://manualzz.com/doc/47794372/object-oriented-design-and-patterns--2nd-edition>.
7. Kruchten, P. (2004). *The rational unified process: an introduction*. Boston: Addison-Wesley.
8. Malykhin, O. B., & Yarmolchuk, T. M. (2020). Topical strategies in the professional training for information technologies specialists. *Information Technologies and Learning Tools*, 76(2), 43-57. <https://doi.org/10.33407/itlt.v76i2.2682>.
9. Rasnacs, A., & Berzisa, S. (2017). Method for Adaptation and Implementation of Agile Project Management Methodology. *Procedia Computer Science*, 104, 43-50. <https://doi.org/10.1016/j.procs.2017.01.055>
10. Sharma, N., Wadhwa, M. (2015). eXSRUP: Hybrid Software Development Model Integrating Extreme Programming, Scrum & Rational Unified Process. *International Journal of Computer Sciences and Engineering*, 3(5), 383-392. https://www.ijcseonline.org/pdf_paper_view.php?paper_id=537&69-IJCSE-01035-3.pdf
11. Software Engineering. (1990). *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12-1990, 1-84. <https://doi.org/10.1109/IEEESTD.1990.101064>.
12. Stelman, A., & Greene, J. (2014). *Learning Agile*. O'Reilly Media, Inc.
13. Systems and software engineering. (2017). *Software Life Cycle Processes* (ISO/IEC/IEEE 12207:2017). International standard. <https://www.iso.org/standard/63712.html>.
14. Torre, D., Labiche, Y., Genero, M., Baldassarre, M. T., & Elaasar, M. (2018). UML Diagram Synthesis Techniques: A Systematic Mapping Study. *2018 IEEE/ACM 10th International Workshop on Modelling in Software Engineering (MiSE)*, 33-40. <https://ieeexplore.ieee.org/document/8445456/references#references>.
15. Turner, M. S. V. (2006). *Microsoft Solutions Framework Essentials: Building Successful Technology Solutions*. Microsoft Press.

16. Wedyan, F., & Abufakher, S. (2020). Impact of Design Patterns on Software Quality: A Systematic Literature Review. *IET Software*, 14(1), 1-17. <https://doi.org/10.1049/iet-sen.2018.5446>.
17. Aleksenko, O. V. (2018). *Tekhnologii prohramuvannia ta stvorennia prohramnykh produktiv: konspekt lektsii [Technologies of programming and creation of software products: lecture notes]*. Sumskyi derzhavnyi universytet. (in Ukrainian).
18. Asieieva, A., & Kulakovska, I. (2019). Analiz problem vyboru tekhnologii dlia rozrobky prohramnoho zabezpechennia [Analysis of technology selection problems for software development]. *Kompiuterno-intehrovani tekhnologii: osvita, nauka, vyrobnytstvo – Computer-integrated technologies: education, science, production*, (37), 10-18. <https://doi.org/10.36910/6775-2524-0560-2019-37-2>. (in Ukrainian).
19. Buch, G., Rambo, D., & Jakobson, I. (2008). *Jazyk UML. Rukovodstvo pol'zovatelja [UML language. User's manual]*. (2-e izd.: Per. s angl. Muhin N.). M.: DMK Press. (in Russian).
20. Vlasiuk, A., & Hrytsiuk, P. (2013). Pidhotovka fakhivtsiv z informatsiinykh tekhnologii u konteksti suchasnykh vymoh [Training of information technology specialists in the context of modern requirements]. *Nova pedahohichna dumka – A new pedagogical idea*, (1.1), 109. http://nbuv.gov.ua/UJRN/Npd_2013_1_28. (in Ukrainian).
21. Hlazonova, O.H. (2014). Model pidhotovky maibutnoho it fakhivtsia v universytetakh ahrarynoho profilu v umovakh hlobalizatsii ta yevrointehratsii [The model of training future IT specialists in agricultural universities in the context of globalization and European integration]. *Visnyk Natsionalnoho universytetu oborony Ukrainy – Bulletin of the National University of Defense of Ukraine*, 5 (42), 36-42. (in Ukrainian).
22. Informacionnaja tehnologija. Kompleks standartov na avtomatizovannye sistemy [Information technology. A set of standards for automated systems]. (1990). *Tekhnicheskoe zadanie na sozdanie avtomatizirovannoj sistemy – Terms of reference for the creation of an automated system* (GOST 34.602-89). Mezhgosudarstvennyj standart. http://online.budstandart.com/ru/catalog/doc-page?id_doc=57803. (in Russian).
23. Klarin, M.V. (1988). Pedagogicheskaja tehnologija v uchebnom processe. Analiz zarubezhnogo opyta [Pedagogical technology in the educational process. Analysis of foreign experience]. *Novoe v zhizni, nauke i tehnike – New in life, science and technology*. Ser. «Pedagogika i psihologija», 6. M.: Znanie. (in Russian).
24. Lohvinenko, V. H. (2005). *Metodyka formuvannia piznavalnoi samostiinosti studentiv tekhnichnykh spetsialnostei v protsesi vyvchennia informatsiino-komunikatyvnykh tekhnologii [Methods of forming cognitive independence of students of technical specialties in the process of studying information and communication technologies]*. [Dys. kand. ped. nauk]. Ukrainska inzhenerno-pedahohichna akademiia. (in Ukrainian).
25. Osvitno-profesiina prohrama «Informatsiini systemy ta tekhnologii» [Educational and professional program "Information systems and technologies"]. (2021). <https://eim.snau.edu.ua/kafedri/kibernetiki-ta-informatiki/osvitno-profesijni-programi/>. (in Ukrainian).
26. Ostroha, M. (2019). Profesiina pidhotovka v haluzi IT: analiz rynku IT-spetsialnostei [IT professional training: analysis of the IT specialties market]. *Fizyko-matematychna osvita – Physical and Mathematical Education*, 4(22), 2, 52-57. <https://doi.org/10.31110/2413-1571-2019-022-4-034>. (in Ukrainian).
27. Profesiinyi standart [Professional standard]. (2014). *Fakhivets z rozrobky prohramnoho zabezpechennia – Software development specialist*. <https://mon.gov.ua/storage/app/media/vyshcha/IT-prof-standarty/6-ps-rozrobnik-pz-13.12.2014.pdf>. (in Ukrainian).
28. Rozroblennia system ta prohramnoho zabezpechennia [Systems and software development]. (2015). *Protsesy zhyttievoho tsykladu. Keruvannia proektamy – Life cycle processes. Project management* (DSTU ISO/IEC/IEEE 16326:2015). Derzhavnyi Standart Ukrainy. http://online.budstandart.com/ua/catalog/doc-page?id_doc=67052. (in Ukrainian).
29. Roshcheniuk, A.M.(2019). Pidhotovka maibutnykh fakhivtsiv z informatsiinykh tekhnologii do tvorchoi samorealizatsii v adaptatsiinyi period [Training of future specialists in information technology for creative self-realization in the adaptation period]. Rukopys.[Avtoref. Dys. k-ta ped. nauk, Rivnenskyi derzhavnyi humanitarnyi universytet]. Repozytarii RHDU. https://rshu.edu.ua/images/afto/avtoref_roshcheniuk.pdf. (in Ukrainian).
30. Striuk, A.M. (2018). Stanovlennia ta rozvytok inzhenerii prohramnoho zabezpechennia yak haluzi znan [Formation and development of software engineering as a field of knowledge]. *Informatsiini tekhnologii v osviti – Information technology in education*, 4 (37), 103-136. (in Ukrainian).

