

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ  
Декан факультету комп'ютерних  
інформаційних технологій  
Ігор ЯКИМЕНКО  
"29" 2025 р.

ЗАТВЕРДЖУЮ  
Проректор  
з науково-педагогічної роботи  
Віктор ОСТРОВЕРХОВ  
"29" 2025 р.

ЗАТВЕРДЖУЮ  
Директор навчально-  
наукового інституту новітніх  
освітніх технологій  
Святослав ПИТЕЛЬ  
"29" 2025 р.



## РОБОЧА ПРОГРАМА

з дисципліни «Проектний практикум»

Ступінь вищої освіти: бакалавр  
Галузь знань – 12 «Інформаційні технології»  
Спеціальність – 121 «Інженерія програмного забезпечення»  
Освітньо-професійна програма – «Інженерія програмного забезпечення»

Кафедра комп'ютерних наук

Форма навчання	Курс	Семестр	Лекції (год.)	Лаб. (год.)	ІРС (год.)	Тренінг (год.)	СРС (год.)	Разом (год.)	Екзамен (сем.)
денна	4	7	-	104	6	6	124	240	7
заочна	4	7	-	4	-	-	236	240	8

*Handwritten signature*

Тернопіль – ЗУНУ  
2025

Робоча програма складена на основі освітньо-професійної програми підготовки бакалавра галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення», затвердженої вченою радою ЗУНУ (протокол №9 від 15 червня 2022 р.).

Робоча програма розроблена доцентом кафедри комп'ютерних наук, к.т.н., Наталією ПОРПЛИЦЕЮ.

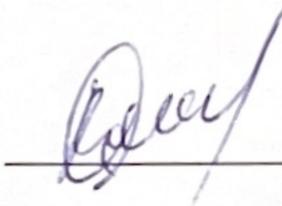
Робоча програма затверджена на засіданні кафедри комп'ютерних наук, протокол № 1 від 26.08 2025 р.

Завідувач кафедри д.т.н., професор



Андрій ПУКАС

Гарант ОП,  
к.т.н., доцент



Світлана КРЕПИЧ

# СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «Проектний практикум»

## 1. Опис дисципліни «Проектний практикум»

Дисципліна – Проектний практикум	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів – 8	Галузь знань: 12 «Інформаційні технології»	Статус дисципліни: обов'язкова Мова навчання: українська
Кількість залікових модулів – 5	Спеціальність: 121 «Інженерія програмного забезпечення»	Рік підготовки: денна – 4; заочна – 4. Семестр: денна – 7; заочна – 7,8.
Кількість змістовних модулів – 3	Ступінь вищої освіти: бакалавр	Лабораторні заняття: денна – 104 год.; заочна – 4 год.
Загальна кількість годин – 240		Самостійна робота: денна – 124 год.; заочна – 236 год. Тренінг: денна – 6 год. Індивідуальна робота: денна – 6 год.
Тижневих годин: 16 год., з них аудиторних – 7 год.		Вид підсумкового контролю – екзамен.

## 2. Мета й завдання вивчення дисципліни «Проектний практикум»

### 2.1. Мета вивчення дисципліни.

Метою курсу є засвоєння практичних навичок у галузі створення програмних систем. Практично засвоїти життєвий цикл програмного продукту. Реалізувати один з методів організації розробки програмної системи. Одержати навички створення програм у складі групи програмістів (розподіл обов'язків, робочі та етичні відношення між членами групи, особливості виконання кожної зі складових процесу). Поглибити свої знання та навички створення моделі, алгоритму та програми для конкретної задачі. Навчитися створювати програми, що є складовими систем вищого рівня.

### 2.2. Завдання вивчення дисципліни.

Завданням дисципліни є отримання та закріплення практичних навиків у створенні прикладних програмних додатків, а також програмних компонент прикладних бізнес систем та опанування практичних методів і засобів розробки та експлуатації даних систем.

### **2.3. Найменування та опис компетентностей, формування яких забезпечує вивчення дисципліни «Проектний практикум»:**

- ✓ ФК15. Здатність формувати нові конкурентоспроможні ідеї й реалізовувати їх у проєктах (стартапах).
- ✓ ФК16. Здатність ефективно працювати в національному та міжнародному контексті, як особистість і як член команди.

### **2.4. Передумови для вивчення дисципліни.**

Вивчення курсу «Проектний практикум» передбачає наявність знань з курсу «Якість програмного забезпечення та тестування», «Іноземна мова», «Системний аналіз», «Елементи дуальної освіти», курсовий проєкт з дисципліни «Засоби програмування баз даних і знань».

### **2.5. Результати навчання.**

У результаті вивчення курсу «Проектний практикум» студенти повинні:

- ✓ РН25. Вміти ефективно працювати в національному та міжнародному контексті, як особистість і як член команди.
- ✓ РН26. Знати та вміти застосовувати методи та засоби інженерії програмного забезпечення у проєктній діяльності та реалізації конкурентоспроможних ідей.

## **3. Програма навчальної дисципліни «Проектний практикум»**

### ***Змістовий модуль 1. Формування вимог до програмного забезпечення***

Тема 1. Вибір технологічного стеку.

*Питання для обговорення:* Вступ. Критерії вибору технології та мови програмування. Популярні мови програмування: переваги та недоліки. Середовища розробки (IDE) та редактори коду. Ініціалізація Git-репозиторію. Практичний приклад: вибір технологій для проєкту.

Література: 1,2, 5, 10.

Тема 2. Аналіз і уточнення вимог до програмного продукту.

*Питання для обговорення:* Життєвий цикл вимог. Процес збору вимог. Інструменти для збору вимог. Аналіз вимог. Валідація вимог. Специфікація вимог.

Література: 1, 2, 4-6, 10.

### ***Змістовий модуль 2. Проектування програмного забезпечення***

Тема 3. Проектування архітектури програмного забезпечення.

*Питання для обговорення:* Вступ: Роль архітектури ПЗ та основи об'єктно-орієнтованого підходу (ООП). Архітектурні шаблони: MVC, багатошарова архітектура, мікросервіси. Компонентне та модульне проектування: поділ на класи та компоненти. UML-діаграми для архітектури: діаграми класів, компонентів, розгортання.

Література: 6-10.

Тема 4. Проектування структури баз даних.

*Питання для обговорення:* Вступ: Значення правильного проектування баз даних у розробці ПЗ. Основні моделі даних: реляційна, документна, графова. Нормалізація даних: концепція нормальних форм, запобігання дублюванню. Зв'язки між таблицями: типи зв'язків (один до одного, один до багатьох, багато до багатьох). Індеси та оптимізація: підвищення продуктивності запитів.

Література: 6-10.

### **Змістовий модуль 3. Програмна реалізація та тестування програмного продукту**

Тема 5. Рефакторинг, оптимізація та безпека програмного коду.

*Питання для обговорення:* Значення оптимізації коду для продуктивності програм. Профілювання та аналіз: інструменти для виявлення вузьких місць (Profiler, Debugger). Алгоритми та структури даних: вибір ефективних рішень для оптимізації. Мікрооптимізація: робота з циклами, рекурсією, кешуванням. Проведення рефакторингу за принципами SOLID та Clean Code. Застосування засобів ШІ для рефакторингу та оптимізації програмного коду.

Література: 1-5, 9, 10.

Тема 6. Тестування програмного продукту та створення тестових артефактів.

*Питання для обговорення:* Роль тестування у забезпеченні якості ПЗ. Тестові артефакти: тест-план, тест-кейси, тестові сценарії, звіти про дефекти. Забезпечення високого покриття коду. Процес тестування: визначення вимог, створення тестів, проведення, аналіз.

Література: 3, 4, 8.

Тема 7. Розгортання та презентація проєкту.

*Питання для обговорення:* Контейнеризація проєкту (Docker). Підготовка та проведення презентації, демонстрація роботи проєкту та обґрунтування всіх технічних рішень.

Література: 2-4.

## **4. Структура залікового кредиту дисципліни «Проєктний практикум»**

Денна форма навчання	Кількість годин				
	Лекції	Лабораторні заняття	СРС	Тренінг	Контрольні заходи
<b>Змістовий модуль 1. Формування вимог до програмного забезпечення</b>					
Тема 1. Вибір технологічного стеку.	-	10	10	2	Усне опитування/тестування
Тема 2. Аналіз і уточнення вимог до програмного продукту.	-	10	20		Усне опитування/тестування
<b>Змістовий модуль 2. Проєктування програмного забезпечення з використанням ООП</b>					
Тема 3. Проєктування архітектури програмного забезпечення.	-	14	20	2	Усне опитування/тестування
Тема 4. Проєктування структури баз даних.	-	15	20		Усне опитування/тестування
<b>Змістовий модуль 3. Програмна реалізація та тестування програмного продукту</b>					
Тема 5. Рефакторинг, оптимізація та безпека програмного коду.	-	20	20	2	Усне опитування/тестування

Тема 6. Тестування програмного продукту та створення тестових артефактів.	-	25	20		Усне опитування/тестування
Тема 7. Розгортання та презентація проєкту.	-	10	14		Усне опитування/тестування
<b>Разом:</b>	-	<b>104</b>	<b>124</b>	<b>6</b>	

Заочна форма навчання	Кількість годин		
	Лекції	Лабораторні заняття	Самостійна робота
Тема 1. Вибір технологічного стеку.	-	1	30
Тема 2. Аналіз і уточнення вимог до програмного продукту.	-		30
Тема 3. Проектування архітектури програмного забезпечення.	-	1	36
Тема 4. Проектування структури баз даних.	-		30
Тема 5. Рефакторинг, оптимізація та безпека програмного коду.	-	1	30
Тема 6. Тестування програмного продукту та створення тестових артефактів.	-		40
Тема 7. Розгортання та презентація проєкту.	-		40
<b>Разом:</b>	-	<b>4</b>	<b>236</b>

## 5. Тематика лабораторних робіт

### Лабораторна робота №1

Тема: Вибір технологічного стеку.

Мета: Навчитись вибирати технології та середовища програмування залежно від типу та складності задачі.

### Лабораторна робота №2

Тема: Аналіз і уточнення вимог до програмного продукту.

Мета: Навчитись визначати та уточнювати вимоги до програмного продукту.

### Лабораторна робота №3

Тема: Проектування архітектури програмного забезпечення.

Мета: Навчитись розробляти архітектуру програмного забезпечення, обґрунтувати вибір архітектурного патерну (наприклад, MVC, багатошарова) та створити відповідні UML-діаграми (класів, компонентів).

### Лабораторна робота №4

Тема: Проектування структури баз даних.

Мета: Здобути навички проєктування реляційних баз даних. Розробити логічну схему БД (ER-діаграму), застосування принципів нормалізації даних та визначити RESTful API для взаємодії з даними.

### Лабораторна робота №5

Тема: Рефакторинг, оптимізація та безпека програмного коду.

Мета: Набути практичного досвіду підвищення якості наявного коду шляхом рефакторингу (застосування принципів SOLID, Clean Code).

### **Лабораторна робота №6**

Тема: Тестування програмного продукту та створення тестових артефактів.

Мета: Освоїти інженерні практики забезпечення якості. Розробити тестові артефакти (тест-план, тест-кейси) та провести функціональне тестування (чорна скринька) для виявлення та фіксації дефектів.

### **Лабораторна робота №7**

Тема: Розгортання та презентація проєкту.

Мета: Набути практичних навичок підготовки проєкту до кінцевого розгортання (наприклад, за допомогою Docker або інструментів CI/CD), презентувати та демонструвати функціональність готового програмного продукту, обґрунтовуючи всі етапи та технічні рішення, прийняті в ході практикуму.

## **6. Самостійна робота**

Для успішного вивчення і засвоєння дисципліни «Проектний практикум» студенти повинні володіти значним обсягом інформації, частину якої вони отримують і опрацьовують шляхом самостійної роботи. Самостійна робота полягає у опрацюванні однієї із запропонованих (або узгоджених з керівником) тем та підготовці звіту. Орієнтовна тематика:

*Теми для самостійної роботи:*

1. Дослідження етичних аспектів збору та аналізу даних користувачів.
2. Техніки пріоритезації вимог у високоризикових проєктах.
3. Вплив неявних вимог (Implicit Requirements) на успіх проєкту.
4. Застосування швидкого прототипування (Rapid Prototyping) для валідації вимог.
5. Трасування вимог (Requirements Traceability) та інструменти для його підтримки.
6. Дослідження впливу User Experience (UX) на формулювання вимог.
7. Функціональна декомпозиція як метод структуризації вимог.
8. Дослідження антипатернів у об'єктно-орієнтованому проєктуванні.
9. Архітектурний патерн Layered Architecture (Багатошарова архітектура) і його реалізація у Enterprise-застосунках.
10. Патерн Repository для ізоляції бізнес-логіки від даних.
11. Роль та значення документації архітектури (C4 Model).
12. Clean Code (Чистий код): принципи та практики, які забезпечують підтримку коду.
13. Впровадження парного програмування (Pair Programming): переваги та недоліки.
14. Застосування статичного аналізу коду (Static Code Analysis) для підвищення якості.
15. Роль Code Review (Перевірки коду) у процесі командної розробки.
16. Оптимізація продуктивності шляхом асинхронного програмування та багатопотоковості.
17. Порівняльний аналіз інструментів для автоматизованого тестування інтерфейсу (UI Automation).
18. Практика розробки через тестування (TDD): переваги та виклики.

19. Методи аналізу складності алгоритмів і їхнє застосування в оптимізації.
20. Застосування метрик коду (Code Metrics) для оцінки якості та складності.
21. Управління випусками (Release Management): планування та розгортання продукту.
22. Гейміфікація у командній розробці та її вплив на мотивацію.
23. Логування та моніторинг (Logging and Monitoring): як інструменти підтримки якості після розгортання.
24. Методи управління ходом виконання робіт ІТ-проєкту.
25. Оцінка економічної ефективності ІТ-проєкту.

## 7. Тренінг з дисципліни

Тематика: Визначення вимог до програмного забезпечення.

Порядок проведення тренінгу:

1. Здійснити аналіз предметної області з метою встановлення вимог до програмного забезпечення.
2. Визначити бізнес процеси, які необхідно автоматизувати.
3. Провести формалізацію вимог до програмного забезпечення.
4. Побудувати UML моделі розроблюваного програмного забезпечення.

Результати виконання завдань тренінгу з дисципліни «Проєктний практикум» оцінюються за 100-бальною шкалою у вигляді презентації виконаних завдань відповідно до обраної з переліку теми.

*Індивідуальні варіанти:*

1. Додаток для проведення соціологічних опитувань
2. Додаток для продажу та бронювання квитків в кінотеатрі
3. Додаток управління ресурсами компанії, що займається виробництвом меблів
4. Додаток управління відносинами з постачальниками
5. Додаток «Готель»
6. Додаток обліку оплати послуг інтернет-провайдера
7. Додаток обліку канцтоварів на підприємстві
8. Додаток управління процесом обробки кореспонденції для компанії експрес доставки поштових відправлень
9. Додаток взаємодії з клієнтами для туристичної компанії
10. Додаток для обліку основних засобів виробництва на підприємстві
11. Додаток «Салон краси»
12. Додаток для обліку пацієнтів в медичному закладі
13. Додаток «Салон продажу автомобілів»
14. Додаток «Бібліотека»
15. Додаток для обліку розрахунків з квартиронаймачами ЖЕКу
16. Додаток «Агенція нерухомості»
17. Додаток для продажу і бронювання авіаційних квитків
18. Додаток для обліку продаж в магазині комп'ютерної техніки
19. Додаток для обліку резюме для відділу кадрів компанії

20. Додаток для пошуку ліків у мережі аптек міста

### 8. Методи оцінювання

У процесі вивчення дисципліни «Проектний практикум» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- стандартизовані тести;
- поточне опитування;
- залікове модульне тестування та опитування;
- презентації результатів виконаних завдань та досліджень;
- студентські презентації та виступи на наукових заходах;
- тренінги;
- екзамен.

### 9. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Проектний практикум» визначається як середньозважена величина, в залежності від питомої ваги кожної складової залікового кредиту:

Модуль 1		Модуль 2		Модуль 3	Модуль 4	Модуль 5
10%	10 %	10 %	10 %	5%	15%	40%
Поточне оцінювання	Модульний контроль 1	Поточне оцінювання	Модульний контроль 2	Тренінги	Самостійна робота	Екзамен
Середнє арифметичне із оцінок, отриманих за виконання лабораторних робіт: № 1 – 4.	Підсумкове модульне тестування за темами №1-4– 25 питань по 4 бали.	Середнє арифметичне із оцінок, отриманих за виконання лабораторних робіт: № 5 – 7.	Підсумкове модульне тестування за темами №5-8 – 25 питань по 4 бали.	Оцінювання презентації виконаних завдань по 100-ій шкалі.	Оцінювання аналітичного огляду наукових джерел або літератури відповідно до обраної з переліку теми по 100-ій шкалі.	1. Теоретичні питання: 2 питання по 30 балів - max 60 балів. 2. Практичне завдання - max 40 балів.

#### Виконання лабораторних робіт:

90-100 балів (Відмінно) - здобувач самостійно, без помилок, виконав усі кроки лабораторної роботи, дотримуючись принципів програмної інженерії (best practices). Вільно оперує ключовими поняттями, принципами та патернами, що стосуються проектування та розробки якісних програмних систем.

75-89 балів (Добре) - здобувач виконав завдання лабораторної роботи, але з кількома дрібними помилками, які не вплинули на кінцевий результат (наприклад, неточна послідовність дій), виникали питання під час роботи.

60-74 балів (Задовільно) - здобувач виконав завдання, але з суттєвими помилками або неточностями (наприклад, функціональність реалізована не повністю, або робота вимагала кількох ітерацій виправлень). Розуміння поставлених у лабораторній роботі завдань є поверхневим або фрагментарним.

1-59 балів (Незадовільно) – здобувач не зміг виконати основне завдання або кінцевий результат був повністю невірним чи нефункціональним. Здобувач не продемонстрував базових навичок роботи з ключовими інструментами та програмним забезпеченням.

**Підсумкове модульне тестування** - вид контролю, при якому засвоєний здобувачем освіти теоретичний та практичний матеріал оцінюється у форматі тестування. Тестування містить 25 запитань кожна правильна відповідь оцінюється у 4 бали, максимум 100 балів.

#### **Тренінг:**

90-100 балів (Відмінно) - здобувач самостійно, без помилок, усі етапи завдання, правильно задокументував усі етапи роботи, дотримуючись принципів програмної інженерії (best practices). Вільно оперує ключовими поняттями, принципами та патернами, що стосуються проектування та розробки якісних програмних систем.

75-89 балів (Добре) - здобувач виконав завдання, але з кількома дрібними помилками, які не вплинули на кінцевий результат (наприклад, неточна послідовність дій), виникали питання під час роботи.

60-74 балів (Задовільно) - здобувач виконав завдання, але з суттєвими помилками або неточностями (наприклад, функціональність реалізована не повністю, або робота вимагала кількох ітерацій виправлень). Розуміння поставлених у завдань є поверхневим або фрагментарним.

1-59 балів (Незадовільно) – здобувач не зміг виконати основне завдання або кінцевий результат був повністю невірним чи нефункціональним. Здобувач не продемонстрував базових навичок роботи з ключовими інструментами та програмним забезпеченням.

#### **Самостійна робота:**

90–100 балів (Відмінно) - звіт охоплює всі ключові аспекти обраної теми, демонструючи глибоке розуміння предмета. Дослідження містить не лише опис, але й глибокий аналіз, порівняння різних підходів та методів, а також обґрунтовані висновки. Використані сучасні та актуальні джерела інформації, що свідчить про обізнаність здобувача з останніми тенденціями у сферах розробки ІТ-проектів. Здобувач вільно володіє матеріалом, виступає впевнено та відповідає на всі запитання.

75–89 балів (Добре) - звіт розкриває основні питання теми, але може бути менш деталізованою. У роботі є аналітичні елементи, але вони можуть бути недостатньо глибокими. Використані джерела інформації є релевантними, але можуть бути не найновішими. Здобувач демонструє знання матеріалу, але може мати незначні труднощі з відповідями на додаткові питання.

60–74 балів (Задовільно) - звіт охоплює лише основні аспекти теми. Можуть бути пропущені важливі деталі. Робота має описовий характер, аналіз та висновки є поверхневими. Здобувач не завжди впевнено відповідає на запитання.

1–59 балів (Незадовільно) - зміст звіту не відповідає темі або є компіляцією застарілих даних. Робота є прямим копіюванням без самостійного опрацювання.

**Екзамен** - вид підсумкового контролю, який проводиться з метою оцінювання засвоєння здобувачем вищої освіти вивченого матеріалу. Екзаменаційний білет складається з двох блоків.

**Перший блок** містить два теоретичних запитань, за кожне з яких можна отримати від 0 до 30 балів, що в підсумку дає максимально 60 балів. За відповідь на питання здобувач отримує 16–30 балів, якщо у повному обсязі володіє навчальним матеріалом, всебічно, самостійно та аргументовано відповідає на питання білету і 1–15 балів – якщо володіє навчальним матеріалом не в повному обсязі, викладає його фрагментарно, допускаючи при цьому суттєві неточності.

**Другий блок** містить практичне завдання, яке оцінюється:

31 – 40 балів - доповідь охоплює всі основні аспекти обраної теми, демонструючи глибоке розуміння предмету дисципліни. Здобувач вільно володіє матеріалом, відповідає впевнено на всі запитання.

21– 30 балів - доповідь розкриває основні питання теми, але може бути недостатньо деталізованою; є аналітичні елементи, але вони можуть бути не достатньо глибокими. Здобувач демонструє знання матеріалу, але має незначні труднощі з відповідями на додаткові питання.

11 – 20 балів - відповідь охоплює лише основні аспекти теми, бути пропущені важливі деталі. Здобувач не завжди впевнено відповідає на запитання.

1 – 10 балів - відповідь містить значні помилки або не зовсім відповідає завданню.

**Шкала оцінювання:**

За шкалою ЗУНУ	За національною шкалою	За шкалою ECTS
90-100	відмінно	A (відмінно)
85-89	добре	B (дуже добре)
75-84		C (добре)
65-74	задовільно	D (задовільно)
60-64		E (достатньо)
35-59	незадовільно	FX (незадовільно з можливістю повторного складання)
1-34		F (незадовільно з обов'язковим повторним курсом)

## 10. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1	Мультимедійний проектор	1-7
2	Проекційний екран	1-7
3	Комунікаційне програмне забезпечення (Internet Explorer, Google Chrome, Firefox)	1-7
4	Операційна система Windows, наявність доступу до мережі Internet	1-7
5	Персональні комп'ютери, ноутбук.	1-7
6	Комунікаційне програмне забезпечення (Zoom) для проведення занять у режимі онлайн (за необхідності)	1-7
7	Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності)	1-7
8	Базове програмне забезпечення Microsoft Office	1-7
9	Системи управління базами даних: MySQL, MongoDB. <a href="https://dbdiagram.io">https://dbdiagram.io</a> – онлайн-додаток для побудови діаграм	1-7

	<p>зв'язків для баз даних (як додаток до основного).          Спеціалізоване програмне забезпечення для моделювання:          Microsoft Visio, Rational Rose, Ramus.  <a href="https://staruml.io/">https://staruml.io/</a> - онлайн-додаток для побудови UML-діаграм          Інтегроване середовище розробки програмного забезпечення:          Microsoft Visual Studio, Microsoft Visual Studio Code</p>	
--	---	--

## РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Вступ до інженерії програмного забезпечення /Є. В. Левус, Т. А. Марусенкова. – Навчальний посібник. Львів: Видавництво Львівської політехніки, 2021. – 248.
2. Gergely Orosz, The Software Engineer's Guidebook: Navigating Senior, Tech Lead, and Staff Engineer Positions at Tech Companies and Startups. - Pragmatic Engineer, 2023. – 414 p.
3. Essential Software Development Career + Technical Guide. - AppJungle.NET LLC, 2023. – 686 p.
4. Fernando Doglio, Skills of a Successful Software Engineer. - Manning, 2022. – 192 p.
5. Баран С. В. Розробка програмного забезпечення з використанням патернів проектування. Навчальний посібник. – Кривий Ріг, Державний університет економіки і технологій, 2023, 203 с.
6. Ізмайлова, О. В. Проектування інформаційних систем : навч. посібник : для студ. галузі знань 12 "Інформаційні технології" / О. В. Ізмайлова ; Київ. нац. ун-т буд-ва і архіт. - Київ : КНУБА, 2022. - 87 с.
7. Конструювання програмного забезпечення. Об'єктно-орієнтований підхід/ О. Б. Кунгурців, Н. О. Новікова. - Видавництво «Кондор», 2024. - 228 с.
8. Winteringham Mark, Software Testing with Generative AI. - Simon and Schuster, 2024. – 304 p.
9. Software Architecture in Practice (SEI Series in Software Engineering). /Len Bass, Paul Clements, Rick Kazman. – 4th Edition, Print2print, 2022. – 442.
10. Програміст-прагматик: друге ювілейне видання. / Девід Томас, Ендрю Хант. – 2-ге видання, Науковий світ, 2024. – 392.