

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

ЗАТВЕРДЖУЮ

В.о. декана факультету комп'ютерних
інформаційних технологій
Ігор ДЖИМЕНКО

" 31 " 2023 р.



ЗАТВЕРДЖУЮ

В.о. проректора з
науково-педагогічної роботи
Віктор ОСТРОВЕРХОВ

" 31 " 2023 р.



ЗАТВЕРДЖУЮ

Директор навчально-наукового
інституту новітніх освітніх
технологій

Святослав ПИТЕЛЬ

2023 р.



РОБОЧА ПРОГРАМА

з дисципліни «Архітектура та проектування програмного забезпечення»

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 КПЗ (год.)	СРС (год.)	Разом (год.)	Іспит. (сем.)
денна	3	5	42	42	5	12	49	150	5
заочна	3	5,6	8	4	-	-	138	150	6

Тернопіль – ЗУНУ
2023

В.О. Островерхов

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 26.05 2021 р.).

Робоча програма розроблена, викладачем кафедри комп'ютерних наук Андрієм ЮШКО.

Робоча програма затверджена на засіданні кафедри комп'ютерних наук, протокол №1 від 28 серпня 2023р.

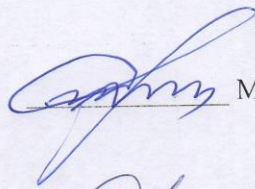
Завідувач кафедри д.т.н., доцент



Андрій ПУКАС

Розглянуто та схвалено групою забезпечення спеціальності 121 Інженерія програмного забезпечення, протокол № 1 від 30.08 2023 р.

Голова групи
забезпечення спеціальності,
д.т.н., професор



Микола ДИВАК

Гарант ОП,
к.т.н., доцент



Світлана КРЕПИЧ

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Архітектура та проектування програмного забезпечення»

1. Опис дисципліни «Архітектура та проектування програмного забезпечення»

Дисципліна «Архітектура та проектування програмного забезпечення»	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів – 5	Галузь знань: 12 «Інформаційні технології»	Статус дисципліни: нормативна Мова навчання: українська
Кількість залікових модулів - 4	Спеціальність: 121 «Інженерія програмного забезпечення»	Рік підготовки: <i>Денна – 3</i> <i>Заочна – 3</i> Семестр: <i>Денна – 5</i> <i>Заочна – 5,6</i>
Кількість змістових модулів – 4	Ступінь вищої освіти: бакалавр	Лекції: <i>Денна – 42 год,</i> <i>Заочна – 8 год.</i> Лабораторні роботи: <i>Денна – 42 год,</i> <i>Заочна – 4 год.</i>
Загальна кількість годин – 150		Самостійна робота: <i>Денна – 49 год,</i> тренінг – 12 год. <i>Заочна – 138 год.</i> Індивідуальна робота: <i>Денна – 5 год.</i>
Тижневих годин – 10 год., з них аудиторних – 4 год.		Вид підсумкового контролю – іспит

2. Мета й завдання вивчення дисципліни «Архітектура та проектування програмного забезпечення»

2.1. Мета вивчення дисципліни

Метою дисципліни «Архітектура та проектування програмного забезпечення» є вивчення методів проектування програмного забезпечення, вивчення шаблонів проектування, середовищ розробки і архітектур, дослідження існуючих архітектур проміжного програмного забезпечення, проектування розподілених систем з використанням проміжного програмного забезпечення.

Мета проведення лекцій полягає в:

- ознайомленні студентів з теоретичними основами проектування ПЗ та архітектур, ознайомлення з шаблонами проектування, середовищами розробки і архітектур;
- дослідженні існуючих архітектур проміжного програмного забезпечення;
- викладенні студентам у відповідності з програмою та робочим планом ключових понять, методів і засобів проектування програмного забезпечення;
- сформувані у студентів цілісну систему теоретичних знань з курсу «Архітектура та проектування програмного забезпечення».

Програма та тематичний план дисципліни орієнтовані на глибоке вивчення проектування програмного забезпечення. Вивчення шаблонів проектування, середовищ розробки і архітектур. Дослідження існуючих архітектур проміжного програмного забезпечення. Проектування розподілених систем з використанням проміжного програмного забезпечення. Компонентне проектування. Теорія вимірювання і використання метрик в проектуванні. Проектування з урахуванням таких якостей як продуктивність, безпека, захищеність, можливість повторного використання, надійність і т.д. Основи еволюції програмного забезпечення та реінжинірингу. Основи проектування програмного забезпечення. Архітектурне проектування програмного забезпечення. Проектування об'єктно-орієнтованого програмного забезпечення.

Аналіз та оцінка проектів програмного забезпечення.

2.2. Завдання вивчення дисципліни

У результаті вивчення курсу «Архітектура та проектування програмного забезпечення» студенти повинні:

- проектувати і реалізовувати програмне забезпечення, використовуючи різні технології проміжного програмного забезпечення;
- розуміти поняття архітектури програмних засобів, підходи до її аналізу та побудови;
- знати основні існуючі архітектури, програмні засоби їх реалізації, їх характеристики та особливості, а також вміти застосовувати ці засоби на практиці;
- використовувати адекватні метрики якості як засіб оцінки якості проектування, оцінювати відповідність результатів проектування поставленим цілям;
- оволодіти основними методами об'єктно-орієнтованого проектування ПЗ, вміти застосовувати ці методи на практиці;

- модифікувати проекти, використовуючи продумані підходи до управління змінами;
- використовувати методи зворотної інженерії (reverse engineering).

2.3. Завдання проведення лабораторних занять

Мета проведення практичних та лабораторних занять полягає у тому, щоб виробити у студентів практичні навички застосування різних шаблонів проектування, середовищ розробки архітектури програмного забезпечення.

Завдання проведення практичних та лабораторних занять:

- сформування у студента систему практичних навичок розробки архітектури та проектування програмного забезпечення;
- сформувати практичні навички проектування програмного забезпечення на основі шаблонів та проміжного програмного забезпечення;
- глибше засвоїти та закріпити теоретичні знання, одержані на лекціях. У курсі розглянуті теми, які актуальні для фахівців на шляху до створення працездатного і якісного програмного продукту.

2.4. Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни «Архітектура та проектування програмного забезпечення»:

Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

Здатність розробляти архітектури, модулі та компоненти програмних систем.

2.5. Передумови для вивчення дисципліни.

Дисципліни, які повинні бути вивчені попередньо:

- «Алгоритми та структури даних»;
- «Операційні системи»;
- «Аналіз вимог до програмного забезпечення».

2.6. Результати навчання

У результаті вивчення курсу «Архітектура та проектування програмного забезпечення» студенти повинні:

Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.

Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

Застосовувати на практиці інструментальні програмні засоби доменного

аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

3. Програма навчальної дисципліни

«Архітектура та проектування програмного забезпечення»

Змістовий модуль 1. Структура та архітектура ПЗ.

Тема 1. Вступ до архітектури та проектування програмного забезпечення.

Значення архітектури для розробки програмного забезпечення. Основні принципи та поняття проектування програмного забезпечення.

Література: 1,2,8.

Тема 2. Архітектурні стилі

Архітектурні стилі.

Література: 1,2,8.

Тема 3. Архітектурне проектування

Структуровані системи. Моделі управління. Модульна декомпозиція Проблемно-залежні архітектури.

Література: 4,7,9.

Тема 4. Архітектура розподілених систем

Багатопроцесорна архітектура. Архітектура клієнт/сервер. Архітектура розподілених об'єктів. CORBA.

Література: 3,5.

Змістовий модуль 2. Стратегії і методи проектування ПЗ.

Тема 5. Об'єктно-орієнтоване проектування

Об'єкти і класи об'єктів. Процес об'єктно-орієнтованого проектування. Модифікація системної архітектури.

Література: 1,6,9.

Тема 6. Проектування систем реального часу

Проектування систем. Керуючі програми. Системи спостереження і управління. Системи збору даних.

Література: 2,7.

Тема 7. Проектування з повторним використанням компонентів

Покомпонентна розробка. Множини додатків. Проектні патерни.

Література: 2,3.

Змістовий модуль 3. Аналіз якості та оцінка програмного дизайну

Тема 8. Проектування інтерфейсу користувача

Принципи проектування інтерфейсів. Взаємодія з користувачем. Оцінка інтерфейсу.

Література: 3,4,6.

Змістовий модуль 4. Нотації та засоби підтримки проектування

Тема 9. Інструментальні засоби підтримки проектування. Представлення нотацій.

Продукти призначені для моделювання та проектування. Інструменти моделювання архітектури. Техніка Моделювання Архітектури. Ідентифікація цілей архітектури. Представлення додатка. Ідентифікація ключових проблем. Література: 3,4,6.

4. Структура залікового кредиту з дисципліни «Архітектура та проектування програмного забезпечення»

Денна форма навчання	Кількість годин					
	Лекції	Лабораторна робота	СРС	ІРС	Тренінг КПІЗ	Контрольні заходи
Змістовий модуль 1. Структура та архітектура ПЗ.						
Тема 1. Вступ до архітектури та проектування програмного забезпечення.	4	2	4		4	Усне опитування/тестування
Тема 2. Архітектурні стилі	4	4	4			Усне опитування/тестування
Тема 3. Архітектурне проектування	4	6	4			Усне опитування/тестування
Тема 4. Архітектура розподілених систем	4	4	4			Усне опитування/тестування
Змістовий модуль 2. Стратегії і методи проектування ПЗ						
Тема 5. Об'єктно-орієнтоване проектування	4	4	6		4	Усне опитування/тестування
Тема 6. Проектування систем реального часу	4	4	6			Усне опитування/тестування
Тема 7. Проектування з повторним використанням компонентів	4	4	6			Усне опитування/тестування
Змістовий модуль 3. Аналіз якості та оцінка програмного диз						
Тема 8. Проектування інтерфейсу користувача	6	6	6		2	Усне опитування/тестування
Змістовий модуль 4. Нотації та засоби підтримки проектуван						
Тема 9. Інструментальні засоби підтримки проектування. Представлення нотацій.	8	8	9	1	2	Усне опитування/тестування
Разом	42	42	49	5	12	

Заочна форма навчання	Кількість годин		
	Лекції	Лабораторна робота	Самостійна робота
Тема 1. Вступ до архітектури та проектування програмного забезпечення.	2	2	16
Тема 2. Архітектурні стилі			14
Тема 3. Архітектурне проектування	14		
Тема 4. Архітектура розподілених систем	14		
Тема 5. Об'єктно-орієнтоване проектування	2	2	16
Тема 6. Проектування систем реального часу			16
Тема 7. Проектування з повторним використанням компонентів			16
Тема 8. Проектування інтерфейсу користувача			16
Тема 9. Інструментальні засоби підтримки проектування. Представлення нотаций.	2		16
Разом	8	4	138

5. Тематика лабораторних робіт

Лабораторне заняття № 1 (5 год.)

Тема: Концепція системи. Представлення Use Case і діаграма кооперації.

Мета: Навчитись проектувати Use Case діаграми та діаграми кооперації.

Питання для обговорення:

1. Який результат структурної класифікації системи стосовно індивідуального завдання?
2. Яким чином здійснюється аналіз складності системи і в якому вигляді представляється результат?
3. Що таке представлення Use Case?
4. Які основні компоненти використовуються в представленні Use Case?
5. Для чого використовується представлення Use Case ?
6. Що таке діаграма кооперації?
7. Характеристика акторів, асоціацій і розширень.

Лабораторне заняття № 2 (5 год.)

Тема: Розробка архітектури інформаційної системи. Розробка діаграми компонентів і діаграми послідовності

Мета: Навчитись проектувати архітектуру інформаційної системи, діаграму компонентів і діаграму послідовності.

Питання для обговорення:

1. Дайте характеристику діаграми компонентів
2. Що таке компонент?
3. Для чого призначений забезпечений інтерфейс?
4. Що таке необхідний інтерфейс?
5. Дайте характеристику діаграми послідовності.
6. Що таке специфікація входження?
7. Що таке специфікація виконання ?
8. Що таке фрагмент взаємодії ?

Лабораторне заняття № 3 (8 год.)

Тема: Розробка діаграми внутрішньої структури, діаграма основних класів системи і діаграми пакетів

Мета: Навчитись проектувати діаграму внутрішньої структури, діаграму основних класів системи і діаграму пакетів.

Питання для обговорення:

1. Дайте характеристику діаграми внутрішньої структури
2. Що таке з'єднувач і інтерфейс?
3. У чому відмінності забезпеченого і необхідного інтерфейсів?
4. Що таке діаграма основних класів системи?
5. Поясніть поняття класу і асоціації
6. Поясніть поняття узагальнення і залежності
7. Дайте характеристику діаграми пакетів
8. Що таке обмеження. Приведіть приклад обмеження.

Лабораторне заняття № 4 (5 год.)

Тема: Обґрунтування вибору засобів розробки, операційної системи і СУБД. Вимоги до програмного і апаратного забезпечення. Діаграма розгортання системи.

Мета: Навчитись обирати засобів розробки, операційну систему і СУБД, формулювати вимоги до програмного і апаратного забезпечення, проектувати діаграму розгортання системи.

Питання для обговорення:

1. Для чого використовується діаграма розгортання.
2. Що таке артефакт?
3. Поясніть поняття «залежність» стосовно діаграми розгортання і приведіть приклад
4. Вкажіть критерії, які Ви використовували при виборі операційної системи. Поясніть, чому саме цими критеріями Ви керувалися.

5. У чому полягають відмінності тонкого і товстого клієнтів?
6. У чому полягають відмінності інтранет веб-сервера додатку і веб-сервера сайту?
7. Обґрунтуйте, чому Ваша системи зможе забезпечити обслуговування необхідного числа клієнтів.

Лабораторне заняття № 5 (5 год.)

Тема: Розробка прототипу інтерфейсу.

Мета: Навчитись розробляти прототип інтерфейсу.

Питання для обговорення:

1. Які вимоги пред'являються до інтерфейсу користувача.
2. Назвіть основні теоретичні передумови, пов'язані з проектуванням інтерфейсу
3. Що є важливішим критерієм: зручність використання інтерфейсу або простота його розробки. Чому.
4. Поясніть, в яких випадках слід використовувати елементи інтерфейсу сторонніх розробників. Які труднощі можуть при цьому виникати.
5. На які чинники слід звертати увагу при виборі компонентів сторонніх розробників.

Комплексне практичне індивідуальне завдання

Індивідуальні завдання з дисципліни «Архітектура та проектування програмного забезпечення» виконується самостійно кожним студентом. КППЗ охоплює усі основні теми дисципліни «Архітектура та проектування програмного забезпечення». Метою виконання КППЗ є оволодіння навичками застосування теоретичних. КППЗ оформлюється у відповідності з встановленими вимогами. Виконання КППЗ є одним із обов'язкових складових модулів залікового кредиту з дисципліни «Архітектура та проектування програмного забезпечення».

Варіанти КППЗ з дисципліни «Архітектура та проектування програмного забезпечення»:

1. Система управління ресурсами підприємства, що займається розробкою програмного забезпечення.
2. Система управління ресурсами підприємства, що займається виробництвом меблів.
3. Система управління ресурсами підприємства, що займається збіркою і продажем комп'ютерів.
4. Система управління ресурсами підприємства, що займається ремонтом і обслуговуванням автомобілів.
5. Система управління ресурсами підприємства, що надає різні поліграфічні і видавничі послуги.

6. Розробити систему управління складами підприємства розташованими в 3-х містах.
7. Розробити систему інтернет магазину з продажу книг
8. Розробити систему автоматизації процесу забезпечення запасними частинами центрів технічного обслуговування автомобілів
9. Розробити систему управління проектами для компанії, що займається розробкою програмного забезпечення.
10. Розробити систему відстежування помилок для компанії, що займається розробкою програмного забезпечення.
11. Розробити систему обліку оплати послуг інтернет провайдера
12. Розробити систему автоматизації формування заявок на виконання ремонтних робіт і постачання запасних частин для електроенергетичної компанії.
13. Розробити систему управління і моніторингу стану проектів в інвестиційній компанії.
14. Розробити систему продажу і бронювання авіаційних квитків.
15. Розробити систему управління процесом транспортування контейнерів.
16. Розробити систему автоматизації документообігу компанії.
17. Розробити систему електронного «аукціону» для постачальників будівельних матеріалів.
18. Розробити автоматизовану систему історії хвороби.
19. Розробити систему управління процесом обробки кореспонденції для компанії експрес доставки поштових відправлень.
20. Розробити систему управління вантажними перевезеннями для автотранспортної компанії.
21. Розробити систему обліку основних засобів виробництва на підприємстві.
22. Розробити систему обліку резюме для відділу кадрів компанії.
23. Розробити систему взаємодії з клієнтами для туристичної компанії.
24. Розробити систему автоматизації обслуговування для готельного комплексу.
25. Система управління кадрами на підприємстві.
26. Система управління відносинами з постачальниками.
27. Система управління відносинами з клієнтами.
28. Система управління нерухомістю.

7 Самостійна робота

- 1 Місце проектування в інженерії ПЗ.

- 2 Характеристики проектної діяльності.
- 3 Суттєві елементи проектування.
- 4 Характеристики слабоструктурованих задач.
- 5 Фактори, які повинні бути відображені в проектах.
- 6 Труднощі створення система на базі програмного забезпечення.

- 7 Що таке модель? Учасники процесу моделювання.
- 8 Когнітивні та прагматичні, абстрактні та матеріальні моделі.

- 9 Функції моделювання.
- 10 Вимоги до моделей.
- 11 Контекст проектування у життєвому циклі ПЗ.
- 12 Процес проектування ПЗ.
- 13 Основні етапи проектування програмного забезпечення.
- 14 Класифікація атрибутів якості ПЗ.
- 15 Ієрархічна модель якості ПЗ.
- 16 Реляційна модель якості ПЗ.
- 17 Надійність та готовність.
- 18 Здатність до супроводу та портативність.
- 19 Принцип розділу задач.
- 20 Принцип та наслідки абстракції.
- 21 Принцип та наслідки модуляризації.
- 22 Модуль та його категорії.
- 23 Відношення між модулями, бажані властивості відношень.
- 24 Ієрархія відношень USES.
- 25 Зчеплення та зв'язаність.
- 26 Принцип приховування інформації.
- 27 Задачі проектування ПЗ.
- 28 Повторне використання ПЗ, його вигоди та проблеми.
- 29 Ландшафт повторного використання ПЗ.
- 30 Фактори планування повторного використання ПЗ.
- 31 Шаблони проектування.
- 32 Генератори ПЗ.
- 33 Аспектно-орієнтована розробка ПЗ.
- 34 Каркаси застосувань.
- 35 Повторне використання продуктів COTS.
- 36 Лінійки продуктів програмного забезпечення.
- 37 ERP системи.
- 38 Метод проектування ПЗ та його склад.
- 39 Покрокова деталізація в проектуванні ПЗ.
- 40 Метод функціонально-орієнтованого проектування.

- 41 Структурне програмування Джексона.
- 42 Розробка систем за Джексонем.
- 43 Архітектура ПЗ визначення.
- 44 Роль архітектури в проектуванні програмного забезпечення.
- 45 Архітектурні структури та представлення.
- 46 Підхід опису архітектури “4+1 views”.
- 47 “Layers”.
- 48 “Pipes-and-Filters”.
- 49 “Broker”.

8. Тренінг з дисципліни

Тема: Проектування різних видів архітектур програмних систем.

Порядок проведення:

1. Здійснити аналіз методів проектування моделей предметних областей.
2. На основі вищезгаданих методів навчитись визначати вид моделі та підхід до побудови архітектури програмної системи.
3. Розглянути методи проектування систем з використанням об’єктних і стандартних структур програмних систем.
4. Представити результати архітектурного проектування у вигляді короткого звіту – нотації у вигляді діаграм (сутність–зв’язок, переходи станів, потоки даних і дій і т. п.) та їх опис. Література: 9.

9. Методи оцінювання

У процесі вивчення дисципліни «Архітектура та проектування програмного забезпечення» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточне опитування;
- стандартизовані тести;
- залікове модульне тестування та опитування;
- реферати;
- презентації результатів виконання завдань;
- оцінювання результатів КППЗ;
- завдання на лабораторному обладнанні;
- ректорська контрольна робота;
- тренінги;
- іспит.

10. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальної шкалою) з дисципліни «Архітектура та проектування програмного забезпечення» визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

Заліковий модуль 1	Заліковий модуль 2 (РКР)	Заліковий модуль 3 (КПІЗ)	Заліковий модуль 4 (іспит)	Разом
20%	20%	20%	40%	100%
Виконання лабораторних робіт (3 роботи по 20 балів – 60 балів) Написання модульної роботи – 40 балів	Виконання лабораторних робіт (2 роботи по 15 балів – 30 балів) Написання ректорської контрольної роботи – 70 балів	Виконання завдань під час тренінгу (20 балів) Написання та захист КПІЗ (80 балів)	Тестові завдання (10 питань по 5 балів – 50 балів) Завдання по теорії (2 завдання по 15 балів – 30 балів) Практичне завдання (20 балів)	100

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)

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

№	Найменування	Номер теми
1.	Мультимедійний проектор	1-9
2.	Проекційний екран	1-9

3.	Комунікаційне програмне забезпечення (Internet Explorer, Google Chrome, Firefox)	1-9
4.	Операційна система Windows, наявність доступу до мережі Internet	1-9
5.	Персональні комп'ютери	1-9
6.	Комунікаційне програмне забезпечення (Zoom) для проведення занять у режимі онлайн (за необхідності)	1-9
7.	Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності)	1-9
8.	Базове програмне забезпечення Microsoft Office	1-9
9.	Спеціалізоване програмне забезпечення: - векторний графічний редактор MS Visio, - Visual Paradigm - редактор коду MS Visual Studio	3-9

Рекомендовані джерела інформації

1. Мартін Р. Чистий код / Роберт Мартін., 2019. – 416 с.
2. Head First. Патерни проектування. Легкий для сприйняття довідник / Е.Фрімен, Е. Робсон, К. Сьєра, Б. Бейтс., 2019. – 672 с.
3. Мартін Р. Чиста архітектура / Роберт Мартін., 2019. – 416 с.
4. The Software Engineering Institute (SEI) Architecture Website -- architecture definitions, offers a good example. Режим доступу: <http://www.sei.cmu.edu/architecture/definitions.html>
5. IEEE Computer Society, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems: IEEE Std 1472000. 2019.
6. Швець О. Занурення в патерни проектування / Олександр Швець., 2021. – 395 с.
7. Len Bass, Paul Clements, and Rick Kazman, Software Architecture in Practice, Second Edition. Addison Wesley 2018.
8. Швець О. Рефакторинг [Електронний ресурс] / Олександр Швець. – 2022. – Режим доступу до ресурсу: <https://refactoring.guru/uk/refactoring>
9. Murray Cantor, «Rational Unified Process for Systems Engineering» The Rational Edge, August 2018. Режим доступу: http://download.boulder.ibm.com/ibmdl/pub/software/dw/rationaledge/aug03/f_rupse_mc.pdf