

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.

РОБОЧА ПРОГРАМА

з дисципліни «Технологія розробки програмного забезпечення»

ступінь вищої освіти – бакалавр

галузь знань – 12 “Інформаційні технології”

спеціальність – 122 „Комп'ютерні науки”

освітньо-професійна програма – „ Штучний інтелект ”

Кафедра інформаційно-обчислювальних систем і управління

Форма навчання	Курс	Семестр	Лекції (год.)	Лабораторні заняття (год.)	ІРС (год.)	Тренінг (год.)	Самост. робота студ. (год.)	Разом (год.)	Екз. (сем.)
Денна	3	6	46	44	5	12	73	180	6

Тернопіль – ЗУНУ
2025

29.08.2025
[Signature]

Робоча програма складена на основі освітньо-професійної програми «Штучний інтелект» підготовки бакалавра галузі знань 12 «Інформаційні технології» спеціальності 122 – «Комп'ютерні науки», затвердженої Вченою Радою ЗУНУ (протокол № 11 від 26.06.2024 р.).

Робочу програму склали викладачі кафедри інформаційно-обчислювальних систем і управління ЗУНУ Назар МЕЛЬНИК та Віталій Дорош

Робоча програма затверджена на засіданні кафедри інформаційно-обчислювальних систем і управління, протокол № 1 від 26.08.2025 року.

В.о. завідувача кафедри
інформаційно-обчислювальних
систем і управління



Надія ВАСИЛЬКІВ

Гарант освітньо-професійної
програми «Штучний інтелект»,
к.т.н., доцент



Василь КОВАЛЬ

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ " ТЕХНОЛОГІЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ "

1. Опис дисципліни "Технологія розробки програмного забезпечення "

Дисципліна «Технологія розробки програмного забезпечення»	Галузь знань, спеціальність, СВО	Характеристика навчальної дисципліни
Кількість кредитів – 6	Галузь знань – 12 “Інформаційні технології”	Статус дисципліни: обов’язкова дисципліна циклу професійної підготовки Мова навчання: Українська
Кількість залікових модулів – 5	Спеціальність – 122 «Комп’ютерні науки»	Рік підготовки: 3 Семестр: Денна – 6
Кількість змістових модулів – 2	Ступінь вищої освіти – бакалавр	Лекції: Денна – 46 год. Лабораторні заняття: Денна – 44 год.
Загальна кількість годин – 180		Самостійна робота: Денна – 73 год., Тренінг: Денна – 12 год. Індивідуальна робота: Денна – 5 год.
Тижневих годин – 12, з них аудиторних – 6 год.		Вид підсумкового контролю – екзамен

2. Мета і завдання дисципліни

"ТЕХНОЛОГІЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ"

2.1. Мета вивчення дисципліни

Мета дисципліни – є формування системи теоретичних знань і набуття практичних умінь і навичок з питань теоретичних основ та використання сучасних методології та технологій розроблення, тестування та експлуатації програмних продуктів, та для моделювання і проектування інформаційних систем, документування та маркетингу програмного забезпечення.

2.2. Завдання вивчення дисципліни

Завдання навчальної дисципліни «Технологія розробки програмного забезпечення» – вивчення понять, принципів, методології та технологій створення програмних продуктів як сукупності процесів розроблення програмних систем на засадах життєвого циклу (ЖЦ) програмного забезпечення інформаційних систем; вивчення призначення засобів об'єктного підходу до проектування програмного забезпечення інформаційних систем; вивчення стандартів програмної інженерії; вивчення методів створення вимог під час розроблення програмних продуктів; вивчення методів та засобів тестування програмне забезпечення сучасними засобами тестування програмних продуктів; вивчення стандартів якості під час створення програмного забезпечення інформаційних систем; вивчення інструментальних засобів програмної інженерії та їх практичне застосування під час проектування програмне забезпечення інформаційних систем.

2.3. Найменування та опис компетентностей, формування яких забезпечує вивчення дисципліни:

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

СК14. Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.

2.4. Передумови для вивчення дисципліни

Вивчення курсу „Технологія розробки програмного забезпечення” передбачає наявність систематичних та ґрунтовних знань із дисциплін: «Основи комп'ютерних наук», «Основи програмування», «Алгоритми і структури даних», «Сучасні парадигми програмування», «Бази і сховища даних», «Розробка клієнт-серверних застосувань».

2.5. Результати навчання

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПР11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог

і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

ПР15. Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних.

3. Програма навчальної дисципліни «ТЕХНОЛОГІЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Змістовий модуль 1 - Загальні особливості розробки програмного забезпечення

Тема 1. Базові поняття створення програмних продуктів

Поняття моделі життєвого циклу програмного забезпечення та інформаційної системи. Основні етапи життєвого циклу інформаційної системи – аналіз, розроблення вимог, проектування, тестування, ввід у експлуатацію, супровід. Причини виникнення технології програмної інженерії. Структура життєвого циклу за міжнародним стандартом ISO/IEC 12207. Концепція інформаційної безпеки та принципи безпечного проектування програмного забезпечення.

Тема 2. Класифікація та сутність процесів розроблення програмного забезпечення

Принципи і методи проектування програмного забезпечення. Методи програмної інженерії. Рівні управління вимогами. Особливості застосування та формулювання вимог.

Тема 3. Моделі життєвого циклу ПЗ.

Технології розроблення програмного забезпечення. Програмний продукт. Життєвий цикл програмного забезпечення. Моделі життєвого циклу. Стратегії розробки ПЗ. Водоспадна модель (Waterfall model) ЖЦ. Інкрементна (incremental) модель. Макетування. Спіральна модель. Компонентно-орієнтована модель. RAD-модель. V- модель (V-Model)

Тема 4. Гнучкі методології.

Методологія Scrum. Важкі та полегшені процеси. Agile-маніфест. Методологія Kanban. Методологія XP (Extreme Programming). Методологія Scrum. Порівняння Agile-методологій

Тема 5. Інженерія вимог до ПЗ.

Методики формування вимог. Вимоги до програмного забезпечення. Категорії вимог. Етапи інженерії вимог. Типи вимог. Формування вимог. Діаграма прецедентів. Аналіз вимог (класичний підхід). Характеристики якісних вимог. Опис вимог у вигляді текстових сценаріїв. Формування та аналіз вимог в процесі Scrum. Історія користувача (user story). Критерії якісної історії користувача. Ієрархія історій. Критерії якісної задачі. Карта історій.

Тема 6. Аналіз вимог до ПЗ.

Опис детальних вимог за допомогою діаграм діяльності та станів UML. Мова моделювання UML. Використання UML для аналізу вимог. Режими використання. Режим ескізу. Режим проектування. Режим мови програмування. Типи діаграм UML 2.0. Діаграма діяльності. Діаграма станів.

Тема 7. Проектування програмного забезпечення.

Діаграма класів та об'єктів UML. Проектування програмного забезпечення. Використання нотації UML для проектування ПЗ. Діаграма класів: клас, ім'я, видимість полів та атрибутів, атрибути, операції, відношення між класами, генерування коду. Діаграма об'єктів.

Тема 8. Проектування ПЗ з допомогою структурних та поведінкових діаграм UML.

Діаграма послідовності (Sequence diagram). Діаграма пакетів (Package Diagram). Діаграма компонентів (Component Diagram). Діаграма розгортання

(Deployment Diagram).

Тема 9. Якість ПЗ.

Якість ПЗ. Види якості. Модель якості програмних систем. Модель Мак-Кола. Модель якості ISO/IEC 9126. Стандартні показники якості. Метрики якості.

Тема 10. Проектування UI.

Місце прототипу в життєвому циклі ПЗ. Стандарти проектування людино-машинного інтерфейсу. Засоби створення прототипів.

Тема 11. Об'єктні моделі та реляційні бази даних.

Архітектурні рішення. Функціональні проблеми. Зчитування даних. Взаємне відображення об'єктів та реляційних структур. Реалізація відображення. Наслідування з одною таблицею (Single Table Inheritance). Наслідування з таблицями для кожного класу (Class Table Inheritance). Наслідування з таблицями для кожного конкретного класу (Concrete Table Inheritance).

Тема 12. Чистий код.

Поняття чистого коду. Стандарти створення чистого коду: змістовні імена, стандарти написання функцій, коментарів, класів. Оброблення помилок.

Тема 13. Модульне тестування.

Призначення. Переваги. Характеристики якісного теста. Тестові двійники: об'єкт-заглушка (Dummy Object), тестова заглушка (Test Stub), тестовий агент (Test Spy), імітація (Fake), підставний об'єкт (Mock). TDD. Фреймворки тестування. Підтримка модульного тестування на рівні мови.

Змістовий модуль 2 – Керування конфігурацією, архітектура тестування та супровід ПЗ

Тема 14. Системи контролю версій.

Основи Git. Про контроль версій. Особливості GIT: принципи збереження даних, стани файлу в робочому каталозі, процеси роботи з гілками, робочий процес одного розробника, робочий процес взаємодії декількох розробників, хостинг репозиторіїв, GitHub.

Тема 15. Безперервна інтеграція та розгортання.

Концепція безперервної інтеграції. Автоматизація складання. Гаки (hooks) в системах контролю версій. Інструменти складання. Засоби безперервної інтеграції. Концепція безперервного розгортання. Засоби безперервного розгортання.

Тема 16. Принципи проектування.

Шаблони DI. Зв'язність та зв'язаність. Інверсія залежності (Dependency Inversion). Шаблони DI: впровадження в конструктор, впровадження у властивість, впровадження в метод, навколишній контекст (Ambient Context). Антишаблони Dependency Injection: Диктатор (Control Freak), «Гібридне впровадження» (Bastard Injection), Обмежене конструювання» (Constrained Construction), Локатор сервісів (Service Locator). Принципи SOLID: принцип єдиної відповідальності (SRP), принцип відкритості / закритості (Open / Closed Principle), принцип підстановки Лісков (Liskov Substitution Principle), принцип поділу інтерфейсів (Interface Segregation Principle), принцип інверсії залежностей (Dependency Inversion Principle, DIP). Принцип KISS. Принцип YAGNI. Принцип DRY. Принцип «Говори, а не питай» (TellDon't-Ask).

Тема 17. Архітектурні стилі.

Огляд. Поєднання архітектурних стилів. Архітектура клієнт/сервер. Компонентна архітектура. Проектування на основі предметної області. Багатошарова архітектура (N-layer architecture). Гексагональна архітектура (Hexagonal Architecture). N-рівнева / 3-рівнева архітектура (N-tier architecture). Об'єктно-орієнтована архітектура. Сервісно-орієнтована архітектура (Service-Oriented Architecture). Мікро-сервісна архітектура.

Тема 18. Багатошарова архітектура.

Концепція розшарування. Архітектура базових типів застосунків. Мобільні застосунки. Веб-застосунок. Особливості типових шарів. Рівень представлення.

Шаблони реалізації бізнес логіки. Шаблони реалізації шару доступу до даних. Шар служб. Наскрізна функціональність. Чиста архітектура.

Тема 19. Породжуючі шаблони проектування.

Factory Method. Абстрактна фабрика (Abstract Factory). Одинак (Singleton). Прототип (Prototype). Будівельник (Builder).

Тема 20. Структурні шаблони проектування.

Декоратор (Decorator). Адаптер (Adapter). Фасад (Facade). Компоновщик (Composite). Заступник (Proxy). Міст (Bridge). Пристосуванець (Flyweight).

Тема 21. Поведінкові шаблони проектування.

Стратегія (Strategy). Спостерігач (Observer). Команда (Command). Шаблонний метод (Template Method). Ітератор (Iterator). Стан (State). Ланцюжок Обов'язків (Chain of responsibility). Інтерпретатор (Interpreter). Посередник (Mediator). Хранитель (Memento). Відвідувач (Visitor).

Тема 22. Супровід програмного забезпечення

Сутність та методи супроводу програмних систем. Завдання супроводу програмних систем. Якість супроводу програмних систем. Управління супроводом програмних систем.

4. Структура залікового кредиту з дисципліни «Технологія розробки програмного забезпечення»

Денна форма навчання

Тема	Кількість годин					
	Лекції	Лабора- торні заняття	Індивідуа- льна робота	Тренінг	Самостійна робота	Контрольні заходи
Змістовий модуль 1 - Загальні особливості розробки програмного забезпечення						
Тема 1. Базові поняття створення програмних продуктів	2	-	-	6	3	Опитування під час заняття
Тема 2. Класифікація та сутність процесів розроблення програмного забезпечення	2	6	-		3	Опитування під час заняття
Тема 3. Моделі життєвого циклу ПЗ.	4	-	1		4	Опитування під час заняття
Тема 4. Гнучкі методології.	2	7	-		4	Опитування під час заняття
Тема 5. Інженерія вимог до ПЗ.	2	-	-		3	Опитування під час заняття
Тема 6. Аналіз вимог до ПЗ.	2	6	-		4	Опитування під час заняття
Тема 7. Проектування програмного забезпечення.	2	7	1		3	Опитування під час заняття
Тема 8. Проектування ПЗ з допомогою структурних та поведінкових діаграм UML.	2	6	1		4	Опитування під час заняття

Тема 9. Якість ПЗ.	2	-	-		4	Опитування під час заняття
Тема 10. Проектування UI.	2	4	1		3	Опитування під час заняття
Тема 11. Об’єктні моделі та реляційні бази даних.	2	-	-		3	Опитування під час заняття
Тема 12. Чистий код.	2	-	-		4	Опитування під час заняття
Тема 13. Модульне тестування.	2	4	-		3	Опитування під час заняття
Змістовий модуль 2 – Керування конфігурацією, архітектура тестування та супровід ПЗ						
Тема 14. Системи контролю версій	2	4	-	6	3	Опитування під час заняття
Тема 15. Безперервна інтеграція та розгортання	2	-	-		3	Опитування під час заняття
Тема 16. Принципи проектування	2	-	-		3	Опитування під час заняття
Тема 17. Архітектурні стилі	2	-	1		3	Опитування під час заняття
Тема 18. Багатошарова архітектура	2	-	-		4	Опитування під час заняття
Тема 19. Породжуючі шаблони проектування	2	-	-		3	Опитування під час заняття
Тема 20. Структурні шаблони проектування	2	-	-		3	Опитування під час заняття
Тема 21. Поведінкові шаблони проектування	2	-	-		3	Опитування під час заняття
Тема 22. Супровід програмного забезпечення	2	-	-		3	Опитування під час заняття
Разом	46	44	5	12	73	

5. Тематика лабораторних занять

Лабораторна робота № 1.

Тема: Вибір програмного проекту, формування команди, підготовка статуту.

Лабораторна робота № 2

Тема: План проекту.

Лабораторна робота № 3

Тема: Специфікація вимог до програмної системи.

Лабораторна робота № 4.

Тема: Стандарт UML.

Лабораторна робота № 5.

Тема: UML.

Лабораторна робота № 6.

Тема: UI/UX дизайн.

Лабораторна робота №7

Тема: Тестування програмної системи

Лабораторна робота №8

Тема: Система контролю версій Git. GitHub.

6. Самостійна робота

Студенти проводять дослідження на обрану з поданого нижче переліку тему та готують результати дослідження у вигляді реферату. Студенти можуть самостійно запропонувати тему дослідження, відповідно до власних вподобань та в рамках даного предмету і погодити її з викладачем.

Перелік тем:

1. Базові поняття та види програмного забезпечення.
2. Розроблення програмного забезпечення.
3. Поняття технології програмування як процесу.
4. Поняття життєвого циклу програмного забезпечення.
5. Каскадна модель (waterflow model).
6. Ітеративна модель (Iterative and incremental development).
7. Спіральна модель.
8. Розроблення ПЗ як проектна діяльність.
9. Основні форми планів робіт.

10. Забезпечення якості ПЗ.
11. Стандартизація розроблення ПЗ.
12. Міжнародні стандарти ISO.
13. Базовий стандарт ISO 12207.
14. Зміст основних процесів ЖЦ ПЗ.
15. Стандарт на процеси життєвого циклу систем ISO/IEC 15288.
16. Стандарти якості серії ISO 9000.
17. CASE–засоби та нотації моделювання програмних систем.
18. Візуальне моделювання мовою UML.
19. Базові терміни та нотація UML.
20. Методології розроблення ПЗ.
21. Вимоги щодо розроблення програм у RUP.
22. Методологія Microsoft Solution Framework (MSF).
23. Методологія eXtreme Programming (XP).
24. Маніфест гнучкого розроблення (Agile manifesto).
25. Принципи гнучкого розроблення ПЗ.
26. Agile-методи.
27. Патерни проектування програмних систем.
28. Забезпечення надійності розробки ПЗ.
29. Складність програмної системи.
30. Моделі якості процесів конструювання.
31. Методи тестування.
32. Види тестування програмного забезпечення.
33. Рівні тестування.
34. Історія патернів.
35. Класифікація патернів.
36. Приклади патернів.

7. Тренінг з дисципліни

Тренінг з дисципліни «Технологія розробки програмного забезпечення» виконується самостійно кожним студентом. Під час тренінгу прикладну область обирає студент самостійно. Метою виконання тренінгу є оволодіння навичками застосування методів технології розробки програмного забезпечення при розв’язуванні прикладних проблем. Тренінг оформлюється згідно з встановленими вимогами.

Тренінг формується на основі лабораторних робіт за наступною структурою:

1. Процент робіт, виконаних розробником (по кожному виду звітів).
2. Процент програмного коду, написаного розробником.
3. Опис особистого внеску в розробку системи.
4. Опис реалізованих розробником алгоритмів й опанованих в ході розробки проекту технологій/систем.

№ з/п	Вид роботи	Порядок проведення тренінгу
1	Вступна частина	ознайомлення студентів з темою тренінгового заняття і видача завдання
2	Практична частина	виконання завдань студентами згідно з індивідуальним завданням; оформлення короткого звіту
3	Підведення підсумків	Презентація та оцінювання результатів виконаних завдань

8. Засоби оцінювання та методи демонстрування результатів навчання

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

- поточне тестування та опитування;
- оцінювання результатів лабораторних робіт;
- оцінювання виконання завдань самостійної роботи;
- оцінювання виконання завдань тренінгу;
- екзамен.

9. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Технологія розробки програмного забезпечення» визначається як середньозважена величина, залежно 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	Тренінг	Самостійна робота	Екзамен
Оцінка визначається як середнє арифметичне з оцінок, отриманих за виконання і захист 4 лабораторних	Модульна контрольна робота (15 тестових завдань та 1 практичне)	Оцінка визначається як середнє арифметичне з оцінок, отриманих за виконання і захист 4 лабораторних	Модульна контрольна робота (15 тестових завдань та 1 практичне)	Виконання завдань під час тренінгу (4 завдання)	Виконання завдання для самостійної роботи	25 тестових завдань та 2 практичних завдання

Модуль 1 та Модуль 2 включають по дві складові за якими оцінюють знання студента:

Оцінка за “**Поточне оцінювання**” визначається як середнє арифметичне з оцінок, отриманих під час занять (тестові завдання під час лекцій та оцінки

лабораторних робіт).

ECTS, бали	Критерії оцінювання
A 90 – 100	Виконані всі завдання лабораторної роботи. Реалізований код програми є працездатним, оптимізованим та структурованим. Правильно виконані всі етапи роботи (налаштування середовища, програмування, відлагодження, перевірка результатів). Звіт оформлено акуратно з поясненням принципів роботи програми. Студент впевнено відповідає на запитання викладача, надає повні пояснення, вміє узагальнити результати та відповідає на додаткові запитання. Звіт оформлено акуратно з поясненням принципів роботи програм та без помилок.
B 85 – 89	Виконані всі основні завдання та частина додаткових. Результати працездатні, хоча присутні дрібні неточності у коді чи описі. Студент демонструє добре розуміння теми та пояснює ключові рішення. Звіт містить незначні недоліки в оформленні.
C 75 – 84	Виконані всі основні завдання, але без додаткових. Програма працює, але з обмеженнями (відсутні додаткові перевірки, код програми не оптимізований, відсутня обробка виняткових ситуацій). Частково зроблений аналіз отриманих результатів. У звіті є недоліки в оформленні або опис.
D 65 – 74	Виконана лише частина основних завдань. Представлені рішення містять суттєві недоліки в коді або логіці роботи, але студент демонструє базове розуміння теми. Робота виконана за зразком, без власних пояснень. Аналіз та висновки неповні або неточні. У звіті є помилки чи пропуски.
E 60 – 64	Виконано мінімальну частину базових завдань з допомогою викладача. Отримані результати обмежено працездатні. Аналіз та висновки відсутні або поверхневі. Звіт має значні недоліки.
FX 35 – 59	Більшість завдань не виконано. Код програми або відсутній, або непрацездатний. Студент не може пояснити свої дії та не розуміє суті лабораторної роботи. Звіт відсутній або не відповідає вимогам.

“Модульний контроль” - оцінка визначається за результатами складання тесту. Тест включає 20 питань теоретичного характеру, які охоплюють основні теми дисципліни та одне завдання практичного характеру. Загальне число балів, що отримав студент, показує наскільки він має глибокі, міцні і системні знання із дисципліни.

Виконання тестових завдань оцінюється за 60-бальною шкалою, тому одна правильна відповідь на тестове завдання оцінюється кількістю балів, яка розраховується наступною формулою:

$$K = 100 / N,$$

де: K - кількість балів за 1 питання; N - кількість питань у тесті.

Загальна кількість балів за весь тест оцінюється наступною формулою:

$$Z = K * P,$$

де: Z - загальна кількість балів; K - кількість балів за 1 запитання; P - кількість правильних відповідей на тестові завдання.

Практичне завдання оцінюється у 40 балів.

Модуль 3 - “Тренінг” – оцінка визначається, як середнє арифметичне оцінок, отриманих за виконання окремих завдань тренінгу, відповідно п. 7 даної робочої програми.

ECTS, бали	Критерії оцінювання
A 90 – 100	Завдання тренінгу виконані в повному обсязі та реалізовані на високому рівні з урахуванням всіх вимог. Показана креативність або оптимізація, що покращує результат. Студент упевнено та аргументовано пояснює прийняті рішення, відповідає на запитання. Демонструє впевнене володіння матеріалом. Продемонстровано якісний кінцевий програмний продукт/прототип.
B 85 – 89	Виконано всі основні завдання. Присутні незначні недоліки у деталях або оформленні. Результат відповідає вимогам та є цілком працездатним.
C 75 – 84	Виконано більшість завдань. Деякі вимоги реалізовані поверхово. Представлений результат працездатний, але має обмежену функціональність.
D 65 – 74	Завдання виконано частково. Присутні суттєві недоліки в реалізації або якості результату. Досягнуто мінімально необхідного рівня. Реалізована основна ідея, але без глибокої деталізації.
E 60 – 64	Виконано лише базову частину завдань. Робота відповідає мінімальним вимогам та демонструє обмежений рівень знань і практичних умінь.
FX 35 – 59	Завдання виконано частково, значна частина ключових етапів відсутня. Результат неповний або не відповідає встановленим вимогам. Демонстрація роботи програми/прототипу обмежена або неможлива.

Модуль 4 - “Самостійна робота” – оцінка визначається, як середнє арифметичне оцінок, отриманих під час виконання завдання самостійної роботи, а саме: 1) представлення проведеного дослідження з обраної теми; 2) аналіз отриманих результатів.

ECTS, бали	Критерії оцінювання
A 90 – 100	Робота виконана повністю, тема розкрита всебічно. Використано достатню кількість джерел, є чітка структура (вступ, основна частина, висновки). Аналіз результатів глибокий та аргументований, зроблено власні висновки. Оформлення відповідає вимогам.
B 85 – 89	Тема досліджена на високому рівні, використано якісні джерела. Присутня логічна структура та аналіз, проте глибина висновків або критичний огляд обмежені. Невеликі недоліки в оформленні.
C 75 – 84	Виконано основний обсяг роботи, тема розкрита частково. Аналіз поверховий, висновки загальні. Використано обмежену кількість джерел, але робота залишається цілісною та зрозумілою.
D 65 – 74	Тема досліджена неповністю, значна частина матеріалу подана описово без належного аналізу. Висновки слабкі або нечіткі. Є помітні недоліки в структурі та оформленні.
E 60 – 64	Представлено лише базову інформацію по темі, відсутній достатній аналіз. Висновки формальні. Робота відповідає мінімальним вимогам.
FX 35 – 59	Виконано частину завдання. Тема розкрита поверхово, аналіз практично відсутній. Структура нечітка, джерела використані обмежено. Робота не відповідає більшості вимог.

Модуль 5 - “Екзамен” – оцінка визначається за результатами екзаменаційного білету, який складає 3 теоретичні питання з розгорнутими відповідями по 20 балів та 2 практичні завдання по 20 балів.

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.	Мультимедійне обладнання	1-22

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Volker Gruhn Rüdiger Striemer The Essence of Software Engineering / Volker Gruhn Rüdiger Striemer. – Saint Philip Street Press, 2020. – 248 p.
2. Chhavi Raj Dosaj The Self-Taught Software Tester A Step By Step Guide to Learn Software Testing Using Real-Life Project / Chhavi Raj Dosaj. – Amazon Digital Services LLC - KDP Print US, 2020. – 217 p.
3. Liping Liu Requirements Modeling and Coding: An Object-Oriented Approach / L. Liping. – World Scientific Publishing Europe Ltd, 2020. – 451 p.
4. Постіл С. Д. UML. уніфікована мова моделювання інформаційних систем / С. Д. Постіл : Ун-т держ. фіск. служби України. - Ірпінь : Ун-т держ. фіск. служби України, 2019. - 321 с.
5. Мартін Р. Чистий код / Р. Мартін – М.: Фабула, 2019. – 416 с.
6. Роберт М. Чиста архітектура: мистецтво розробки програмного забезпечення» / Роберт Мартін, Фабула, 2019. – 416 с.
7. UML Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/uml/index.htm>.
8. Java Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/java/index.htm>.
9. C++ Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/cplusplus/index.htm>.