

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

ЗАТВЕРДЖУЮ:

Декан факультету комп'ютерних
інформаційних технологій
Ігор ЯКИМЕНКО



2025 р.

ЗАТВЕРДЖУЮ:

Проректор з науково-педагогічної
роботи
Віктор ОСТРОВЕРХОВ



2025 р.

РОБОЧА ПРОГРАМА

з дисципліни

«Тестування та верифікація інтелектуальних систем»

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.)	Лабор. (год.)	ІРС (год.)	Тренінг (год.)	СРС (год.)	Разом (год.)	Залік (сем.)
Денна	4	7	30	14	3	6	97	150	7

Тернопіль – ЗУНУ
2025

29.08.2025
[Signature]

Робочу програму склав доцент кафедри інформаційно-обчислювальних систем і управління, к.т.н., доцент Павло Биковий

Робоча програма затверджена на засіданні кафедри інформаційно-обчислювальних систем і управління, протокол № 1 від 26 серпня 2025 р.

В.о. завідувача кафедри



Надія ВАСИЛЬКІВ

Гарант освітньо-професійної
програми "Штучний інтелект",
канд. техн. наук, доцент



Василь КОВАЛЬ

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «ТЕСТУВАННЯ ТА ВЕРИФІКАЦІЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ»

1. ОПИС ДИСЦИПЛІНИ «ТЕСТУВАННЯ ТА ВЕРИФІКАЦІЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ»»

Дисципліна – «Тестування та верифікація інтелектуальних систем»	Галузь знань, спеціальність, СВО	Характеристика навчальної дисципліни
Кількість кредитів ECTS - 5	Галузь знань – 12 «Інформаційні технології»	Статус дисципліни – вибіркова дисципліна циклу професійної підготовки Мова навчання - українська
Кількість залікових модулів – 3	Спеціальність – 122 «Комп’ютерні науки»	Рік підготовки: 4 Семестр: <i>Денна форма – 7</i>
Кількість змістових модулів – 2	Освітньо-професійна програма: «Штучний інтелект»	Лекції: 30 год. Лабораторні: 14 год.
Загальна кількість годин - 150	Ступінь вищої освіти – бакалавр	Самостійна робота: 97 год. Тренінг: 6 год. Індивідуальна робота 3год.
Тижневих годин: Денна форма навчання – 10 год., з них аудиторних – 3 год.		Вид підсумкового контролю: залік

2. МЕТА ТА ЗАВДАННЯ ВИВЧЕННЯ ДИСЦИПЛІНИ «ТЕСТУВАННЯ ТА ВЕРИФІКАЦІЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ»

2.1. Мета вивчення дисципліни

Метою вивчення дисципліни «Тестування та верифікація інтелектуальних систем» є набуття теоретичних і практичних знань з верифікації, валідації і тестування; володіння прийомами тестування на різних фазах розробки програмного продукту; використання інструментів для тестування основних характеристик системи.

2.2. Завдання вивчення дисципліни:

- оволодіння основними поняттями тестування та якості програмних продуктів;
- ознайомлення із основними поняттями тестування: термін "тестування", відладка, фази і технології тестування, проблеми тестування;
- характеристика критеріїв вибору тестів: структурні, функціональні, стохастичні, мутаційний;
- опис різновидів тестування: модульне, інтеграційне, системне, регресійне, автоматизація тестування;
- виявлення особливостей процесу і технології тестування: планування тестування, підходи до розробки тестів, особливості ручної розробки і генерації тестів, автоматизація тестового циклу;
- оформлення документації тестування.

2.3. Результати навчання

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. Відрізнити баг від фічі. Описувати баг-репорти в системі обліку помилок Mantis Bug Tracker. Отримати практику в тестуванні веб-сайтів. Отримати практику в тестуванні мобільних додатків.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «ТЕСТУВАННЯ ТА ВЕРИФІКАЦІЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ»

ЗМІСТОВИЙ МОДУЛЬ 1. ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тема 1. Поняття якості програмного забезпечення

Основні поняття та визначення. Моделі якості програмного забезпечення. Види якості програмного забезпечення.

Тема 2. Життєвий цикл програмного забезпечення

Типи та призначення моделей життєвого циклу програмного забезпечення. Моделі послідовного виконання (каскадна, V-подібна). Моделі ітераційного виконання (ітераційна, еволюційна, спіральна). Методології Scrum, Kanban, Agile.

Тема 3. Фундаментальний етап тестування

Стадії циклу розробки програмного забезпечення у розрізі тестування. Фундаментальний процес тестування. Проблеми, фази, типи, рівні та методи тестування.

Тема 4. Аналіз вимог

Характеристики вимог. Техніки тестування документації та вимог. Ціна помилок.

ЗМІСТОВИЙ МОДУЛЬ 2. ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тема 5. Основні поняття тестування

Опис переваг роботи тестувальником та шляхи розвитку. Правила оформлення звітів про дефекти. Життєвий цикл багів. Інструменти для зняття скріншотів. Оформлення звітів про дефекти.

Тема 6 Тестування Web-проектів

Етапи тестування Web-проектів. Вічне коло тестування. Анатомія web-сторінки. Чекліст для тестування web-сайту. Тестування верстки. Кросбраузерне тестування. Інструменти тестування дизайну web-сайту.

Тема 7. Підходи до функціонального тестування

Тестування Web-форм. Пошук функціональних багів . Техніки тестування. Еквівалентне розбиття. Граничні сценарії. Чек-лист для перевірки функціоналу сайту. Тестування без вимог. Неформальні техніки тестування .

Тема 8. Життєвий цикл тестування програмного забезпечення

Поняття тест-плану. Стратегії тестування. Інструменти для планування. Jira Software. Ролі в процесі розробки програмного забезпечення.

Тема 9. Тест-дизайн та тест-кейси

Поняття техніки тест-дизайну. Тест-кейси та їх атрибути. Критерії якості тест-кейсів. Інструменти для тест-дизайну (TestLink). Життєвий цикл тест-кейсу.

Тема 10. Тестування мобільних додатків

Сучасні мобільні платформи. Етапи розробки мобільних додатків. Особливості тестування додатків на мобільних пристроях. Створення копій екрану на мобільних пристроях. Використання iOS, Android емуляторів. Особливості оформлення баг-репортів.

Тема 11. Тестування ігор

Ігрові жанри та механіки. Виконання замірів продуктивності в іграх. Функціональне тестування ігор. Тестування сумісності. Локалізаційне тестування та особливості опису баг-репортів в іграх.

Тема 12. Автоматизація процесу тестування

Огляд інструментів автоматизації тестування: Rational Robot, TestComplete, Borland Silk Test та інші.

4. СТРУКТУРА ЗАЛІКОВОГО КРЕДИТУ З ДИСЦИПЛІНИ «ТЕСТУВАННЯ ТА ВЕРИФІКАЦІЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ»

Денна форма навчання

Тема	Кількість годин					
	Лекції	Лабор.	ІР	Тренінг	СР	Контрольні заходи
<u>Змістовий модуль 1. ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</u>						
Тема 1. Поняття якості програмного забезпечення	2		1	2	6	Опитування під час заняття
Тема 2. Життєвий цикл програмного забезпечення	2				4	Опитування під час заняття
Тема 3. Фундаментальний етап тестування	2				8	Опитування під час заняття
Тема 4. Аналіз вимог	2	2			8	Опитування під час заняття
<u>Змістовий модуль 2. ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</u>						
Тема 5. Основні поняття тестування	2		2	4	5	Опитування під час заняття
Тема 6 Тестування Web-проектів	2	4			10	Опитування під час заняття
Тема 7. Підходи до функціонального тестування	2				10	Опитування під час заняття
Тема 8. Життєвий цикл тестування програмного забезпечення	2				8	Опитування під час заняття
Тема 9. Тест-дизайн та тест-кейси	4	2			10	Опитування під час заняття
Тема 10. Тестування мобільних додатків	2	2			10	Опитування під час заняття
Тема 11. Тестування ігор	4	2			10	Опитування під час заняття
Тема 12. Автоматизація процесу тестування	4	2			8	Опитування під час заняття
ВСЬОГО	30	14	3	6	97	

5. ТЕМАТИКА ЛАБОРАТОРНИХ РОБІТ

1. Оформлення звітів про дефекти.
2. Пошук багів по верстці.
3. Пошук функціональних багів.
4. Складання тест-плану.
5. Складання тест-кейсів для web-проєкту.

6. САМОСТІЙНА РОБОТА

Студенти розробляють блок-схему одного із процесів управління відповідно до заданої предметної області (список додається), а також складають тести для перевірки роботи процесу. Студенти можуть самостійно запропонувати тему дослідження, відповідно до власних вподобань та в рамках даного предмету і погодити її з викладачем.

Перелік тем:

- | | |
|----------------------------------|-------------------------------|
| 1. Авіакомпанія | 20. Надання кредитів |
| 2. Автовокзал | 21. Олімпіада |
| 3. Автосалон | 22. Пекарня |
| 4. Автосервіс | 23. Перукарня |
| 5. Аеропорт | 24. Поліклініка |
| 6. Аптека | 25. Порт |
| 7. Бібліотека | 26. Працевлаштування |
| 8. Вантажні перевезення | 27. Прокат автомобілів |
| 9. Ветлікарня | 28. Прокат турспорядження |
| 10. Видавництво | 29. Реабілітаційне відділення |
| 11. Вирощування та продаж квітів | 30. Ремонт автомобілів |
| 12. Відділ кадрів | 31. Розклад занять |
| 13. галерея | 32. Санаторій |
| 14. Гастроном | 33. Таксі |
| 15. Готель | 34. Театр |
| 16. Залізничний вокзал | 35. Турагенство |
| 17. Книжковий магазин | 36. Туристична база |
| 18. Меблевий магазин | 37. Фотосалон |
| 19. Музей | 38. Футбол |

7. ТРЕНІНГ З ДИСЦИПЛІНИ

№ з/п	Вид роботи	Порядок проведення тренінгу
1.	Вступна частина	Ознайомлення студентів з темою тренінгового заняття «Тестування мобільних пристроїв та їх програмного забезпечення» і видача завдання
2.	Практична частина	Виконання завдань студентами згідно з індивідуальним завданням: 1.Провести тестування власного мобільного телефону, зокрема: - Функціональне тестування: Використовується для перевірки функціональних можливостей програми відповідно до специфікації вимог - Тестування продуктивності: Виконується для тестування продуктивності клієнтських додатків, продуктивності сервера та продуктивності мережі - Тестування пам'яті: Мобільні пристрої мають обмежену пам'ять порівняно з комп'ютерами, цей тип тестування проводиться для перевірки оптимізованого використання пам'яті додатком - Тестування на переривання: Використовується для перевірки на перебої через вхідний дзвінок або SMS, попередження про низький обсяг пам'яті, попередження про низький заряд акумулятора тощо під час запуску програми - Тестування встановлення : Тестування встановлення використовується для перевірки простоти та плавності процесу встановлення, що включає також оновлення та видалення - Тестування зручності використання: Як завжди, він перевіряв ефективність, результативність та задоволеність від застосування 2.Представити результати у вигляді короткого звіту, що повинен містити копії екранів проведених тестувань. Описати хід виконання кожного завдання.
3.	Підведення підсумків	Обговорення результатів виконаних завдань

8. ЗАСОБИ ОЦІНЮВАННЯ ТА МЕТОДИ ДЕМОНСТРУВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

У процесі вивчення дисципліни «Тестування та верифікація інтелектуальних систем» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточне тестування та опитування;

- оцінювання результатів лабораторних робіт;
- оцінювання виконання завдань тренінгу;
- оцінювання виконання завдань самостійної роботи;

9. КРИТЕРІЇ, ФОРМИ ПОТОЧНОГО ТА ПІДСУМКОВОГО КОНТРОЛЮ

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

Модуль 1		Модуль 2	Модуль 3
40%	40%	5%	15%
Поточне оцінювання	Модульний контроль 1	Тренінг	Самостійна робота
Оцінка визначається як середнє арифметичне з оцінок, отриманих за виконання і захист 5 лабораторних робіт	Модульна контрольна робота (15 тестових завдань та 1 практичне)	Виконання завдань під час тренінгу (6 завдань)	Виконання завдання для самостійної роботи

Модуль №1 включає дві складові за якими оцінюють знання студента:

1. **“Поточне оцінювання”** - оцінка визначається як середнє арифметичне оцінок, отриманих за виконання та захист 5 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

ECTS, бали	Критерії оцінювання
	лабораторної роботи. Звіт відсутній або не відповідає вимогам.

2. **“Модульний контроль”** - оцінка визначається за результатами складання тесту. Тест включає 15 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 №2 - “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 №3 - “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

Шкала оцінювання

За шкалою університету	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.	Інструмент для тест-дизайну TestLink	1-12
2.	Система обліку помилок Mantis	1-12

11. РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Авраменко А. С. Тестування програмного забезпечення : навч. посіб. / А. С. Авраменко, В. С. Авраменко, Г. В. Косенюк. – Черкаси :ЧНУ імені Богдана Хмельницького, 2018. – 284 с.
2. Крепич С. Я. Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник для бакалаврів галузі знань 12 "Інформаційні технології" спеціальності 121 "Інженерія програмного забезпечення" / С. Я. Крепич, І. Я. Співак. – Тернопіль : ФОП Паляниця В. А., – 2020. – 478 с.
3. Gerardus Blokdijk. Software Testing A Complete Guide. 5STARCook. 2021, 313 p.
4. Maurício Aniche. Effective Software Testing. Manning. 2022, 328 p.
5. Mukesh Sharma. Software Testing 2020: Preparing for New Roles. Productivity Press. 2016, 158 p.
6. Casey Rosenthal, Nora Jones. Chaos Engineering: System Resiliency in Practice, 1st Edition, Kindle Edition, O'Reilly Media. 2020, 329 p.
7. Chhavi Raj Dosaj. The Self-Taught Software Tester A Step By Step Guide to Learn Software Testing Using Real-Life Project. Independently published. 2020, 217 p.
8. Arnon Axelrod. Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects, 1st ed. Edition. – Apress, 2018. 558 p.