

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

ЗАТВЕРДЖУЮ:

Декан факультету комп'ютерних
інформаційних технологій

Ігор ЯКИМЕНКО



“ 30 ” 2024 р.

ЗАТВЕРДЖУЮ:

Проректор з науково-педагогічної
роботи

Віктор ОСТРОВЕРХОВ



2024 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
2024

30.09.2024

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 27 серпня 2024 р.

В.о. завідувача кафедри



Надія ВАСИЛЬКІВ

Розглянуто та схвалено групою забезпечення спеціальності 122 „Комп’ютерні науки”, протокол № 1 від 30 серпня 2024 р.

Голова групи
забезпечення спеціальності,
д-р техн. наук, професор



Мирослав КОМАР

Гарант освітньо-професійної
програми "Комп'ютерні науки",
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

У результаті вивчення курсу студент повинен знати:

- поняття верифікації, валідації і тестування;
- прийоми тестування на різних фазах розробки якісного програмного продукту;
- умови ефективного застосування інструментальних засобів в розробці якісного програмного забезпечення;
- принципи розробки тестових програм і тестових наборів в програмному проекті;
- етапи та елементи розробки проектної документації для етапу тестування;
- необхідність та ефективність сумісної роботи проектної команди, яка складається з розробників і тих, хто тестує розроблюване ПЗ.

У результаті вивчення курсу студент повинен вміти:

- розробляти документацію на систему, що тестується: опис вимог до системи, тести, тестові процедури і специфікації розробника;
- планувати процес тестування;
- розробляти різні види тестів;
- шукати дефекти системи в процесі тестування, приймати участь в їх виправленні і модернізації додатку, який проходить тестування.

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					Контрольні заходи
	Лекції	Лабор.	ІР	Тренінг	СР	
<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 | 21. Олімпіада |
| 2. Автовокзал | 22. Пекарня |
| 3. Автосалон | 23. Перукарня |
| 4. Автосервіс | 24. Поліклініка |
| 5. Аеропорт | 25. Порт |
| 6. Аптека | 26. Працевлаштування |
| 7. Бібліотека | 27. Прокат CD-дисків |
| 8. Вантажні перевезення | 28. Прокат автомобілів |
| 9. Ветлікарня | 29. Прокат відеофільмів |
| 10. Видавництво | 30. Прокат турспорядження |
| 11. Вирощування та продаж квітів | 31. Реабілітаційне відділення |
| 12. Відділ кадрів | 32. Ремонт автомобілів |
| 13. галерея | 33. Розклад занять |
| 14. Гастроном | 34. Санаторій |
| 15. Готель | 35. Таксі |
| 16. Залізничний вокзал | 36. Театр |
| 17. Книжковий магазин | 37. Турагенство |
| 18. Меблевий магазин | 38. Туристична база |
| 19. Музей | 39. Фотосалон |
| 20. Надання кредитів | 40. 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	<p>Виконання завдань студентами згідно з індивідуальним завданням:</p> <p>1. Провести тестування власного мобільного телефону, зокрема:</p> <ul style="list-style-type: none"> - 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 <p>2. Представити результати у вигляді короткого звіту, що повинен містити копії екранів проведених тестувань. Описати хід виконання кожного завдання.</p>
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 виконання 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 лабораторних робіт	Модульна контрольна робота (15 тестових завдань та 1 практичне)	Виконання завдань під час тренінгу (6 завдань)	Виконання завдання для самостійної роботи

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

Модуль “Тренінг” визначається, як середнє арифметичне з оцінок, отриманих під час тренінгу.

Модуль “Самостійна робота” визначається, як середнє арифметичне з оцінок, отриманих під час виконання завдання самостійної роботи, а саме оцінка за теоретичні знання, представлення результатів та проведених досліджень.

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

Основна література

1. Крепич С. Я. , Співак І.Я. Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник для бакалаврів галузі знань 12 "Інформаційні технології" спеціальності 121 "Інженерія програмного забезпечення". Тернопіль : ФОП Паляниця В. А., 2020. 478 с.
2. Gerardus Blokdyk. Software Testing A Complete Guide. 5STARCOOKS. 2021, 313 p.
3. Maurício Aniche. Effective Software Testing. Manning. 2022, 328 p.
4. Casey Rosenthal, Nora Jones. Chaos Engineering: System Resiliency in Practice, 1st Edition, Kindle Edition, O'Reilly Media. 2020, 329 p.
5. 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.

Додаткова література

1. István Forgács, Attila Kovács. Modern Software Testing Techniques: A Practical Guide for Developers and Testers. Apress, 2024. 266 p.
2. Bernard Homès. Fundamentals of Software Testing. Wiley-ISTE, 2024. 400 p.
3. Tom Taulli. AI-Assisted Programming: Better Planning, Coding, Testing, and Deployment. O'Reilly Media. 2024, 225p.
4. David J. Weiss, Alper Sahin. Computerized Adaptive Testing: From Concept to Implementation (Methodology in the Social Sciences Series). The Guilford Press, 2024. 360 p.
5. Fangting Huang, Yuriy Drohobytskiy, Hanna Homeniuk, Volodymyr Kochan, Roman Kochan, Mariana Levkiv, Nataliia Kochan, Pavlo Bykovyy. Ad-hoc Signal Conditioning Transducer with Self-Testing for Data Acquisition Devices. The 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 7-9 September, 2023, Dortmund, Germany, pp. 999-1004.
6. Dave Westerveld. API Testing and Development with Postman - Second Edition: API creation, testing, debugging, and management made easy. Packt Publishing, 2024. 358 p.