

Черкаський національний університет імені Богдана Хмельницького
ННІ Інформаційних та освітніх технологій
Кафедра прикладної математики та інформатики

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

1. Загальна інформація про курс

Мова викладання	Курс викладається українською мовою.
Статус дисципліни	Обов'язкова
Викладач	Дзюба Вікторія Анатоліївна, кандидат технічних наук, старший викладач кафедри прикладної математики та інформатики
Код класу /	
Корпоративна пошта / E-mail:	viktoriya.dzyuba15@vu.cdu.edu.ua
Затвердження та перегляд робочої навчальної програми	Розглянуто та затверджено на засіданні кафедри прикладної математики та інформатики 27.08.2025, протокол № 1

СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Форма навчання	Семестр у межах дисципліни	Кількість кредитів	Загальна кількість годин	Аудиторна робота				Самостійна робота			Форма підсумкового контролю
денна	перший	4	120 год	40 год / 33 %				80 год / 67 %			іспит
				лекції	лабора-торні	прак-тичні	семінар-ські	розрахункові роботи	індивіду-альні завдання	підготовка до занять	
				20 год	20 год				40 год	40 год	

МЕТА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Оволодіння основними поняттями в області програмної інженерії, пов'язаними з методами забезпечення якості програмного забезпечення (ПЗ), і навичками використання методів тестування ПЗ з акцентом на використанні формалізованих моделей поведінки. Отримання уявлення про основні поняття тестування, сучасні критерії вибору тестів, види тестування, особливості процесу і технології індустріального тестування.

ПЕРЕДУМОВИ ДЛЯ ВИВЧЕННЯ ДИСЦИПЛІНИ ТА УЗГОДЖЕННЯ З ІНШИМИ ДИСЦИПЛІНАМИ

Курс базується на поняттях, що вивчаються в дисциплінах:

- програмування;
- програмне забезпечення та інформаційно-комунікаційні технології;
- мови програмування;
- веб-програмування;
- об'єктно-орієнтоване програмування;
- бази даних та інформаційні системи;
- проектування інформаційних систем.

ОЧІКУВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

Знання основних понять у галузі забезпечення якості ПЗ, основних методів моделювання поведінки ПЗ і ситуацій його функціонування, основних методів тестування, основних методів оцінки повноти тестування ПЗ, основи індустріального тестування, методів інтеграційного тестування та особливостей його застосування.

Вміння розробляти і реалізовувати тести для програм різних типів, оцінювати повноту тестування програм різних типів, аналізувати результати виконання тестів, застосовувати формалізовані моделі при розробці тестів і оцінці їх повноти, складати звіти у стандартному форматі, зіставляти вартість і витрати на реалізацію завдання. Володіння різними видами автоматичного тестування, використанням різноманітних технологій віртуалізації, навичками розв'язання типових задач розробки тестів для ПЗ, навичками аналізу результатів виконання тестів, навичками застосування формалізованих моделей для розробки тестів.

Результатом вивчення навчальної дисципліни «Аналіз даних» є формування наступного комплексу компетентностей.

Інтегральна компетентність: здатність розв'язувати складні спеціалізовані задачі та практичні проблеми в області інформаційних систем та технологій, або в процесі навчання, що характеризуються комплексністю та невизначеністю умов, які потребують застосування теорій та методів інформаційних технологій.

Загальні компетентності:

- ЗК 1. Здатність до абстрактного мислення, аналізу та синтезу;
- ЗК 8. Здатність оцінювати та забезпечувати якість виконуваних робіт.

Фахові компетентності:

- СК 3. Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережевої структури, управління ними.;
- СК 8. Здатність управляти якістю продуктів і сервісів інформаційних систем та технологій протягом їх життєвого циклу;
- СК 10. Здатність вибору, проектування, розгортання, інтегрування, управління, адміністрування та супроводжування інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації.

ПРОГРАМНІ РЕЗУЛЬТАТИ НАВЧАННЯ

ПРН 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

- ПРН 5. Аргументувати вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.
- ПРН 7. Обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.
- ПРН 12. Вміти розробляти програмне забезпечення різного рівня складності, що входить до складу інформаційних систем та технологій, і спрямоване, зокрема, на виконання задач інтелектуального пошуку, аналізу та обробки даних з метою видобутку знань і прогнозування поведінки аналізованого об'єкта при розв'язанні прикладних науково-виробничих задач і задач бізнесу.

КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Критерієм успішного проходження здобувачем освіти підсумкового оцінювання є досягнення ним мінімальних порогових рівнів оцінок за кожним запланованим результатом навчання навчальної дисципліни, який визначається до кожного завдання через якісні критерії і трансформується в мінімальну позитивну оцінку обраної для даної дисципліни шкали. Після завершення курсу використана шкала перенормовується у накопичувальну 100-бальну системою з відповідністю у національній («відмінно», «добре», «задовільно», «незадовільно») та ЄКТС (A, B, C, D, E, FX, F) шкалах.

Контроль проводиться у чотири етапи: вхідний (у формі тестів для визначення рівня підготовки студентів з дисциплін, що забезпечують цей курс; за результатами цього контролю розробляються заходи індивідуальної допомоги студентам, коригування освітнього процесу тощо), поточний (використовується при проведенні лекцій та лабораторних занять і має на меті перевірку рівня засвоєння студентом матеріалу курсу та підготовленості студента до виконання конкретних видів навчальної діяльності), проміжний (модульний; при закінченні вивчення логічно завершеної частини навчальної дисципліни) та семестровий контроль (захист індивідуального проекту; екзамен).

ЗАСОБИ ОЦІНЮВАННЯ

Екзамен; аналітичні звіти; лістинги програм; лістинги тестів; тести; індивідуальні проекти.

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Аудиторна робота «Слухай, читай, обговорюй» (Л – лекція, Лб – лабораторне заняття)

Самостійна робота «Думай, пиши, аналізуй, досліджуй, твори» (Інд – індивідуальне завдання (міні-проект, проект), Пз – підготовка до занять)

Контроль «Захищай, відстоюй, неси відповідальність» (МК – модульний контроль; ПК – підсумковий контроль)

Тема	Тема навчального заняття	К-сть годин за денною/заочною формами	Засіб оцінювання	Максимальна кількість балів за формами навчання
Вхідний контроль			тест	---
Змістовий модуль 1. Основи процесу тестування				14 / ---
	Л1 Процеси тестування і розробки ПЗ. Поняття тестування ПЗ. Причини виникнення тестування ПЗ. Задачі тестувальника. Моделі розробки ПЗ. Життєвий цикл тестування. <i>Л-ра: 1, 5, 7</i>	2 / –	---	---
	Лб1 Модульне тестування на Python. Основи тестування. Прості тести. Пропуск тестів. Запуск вибраних тестів. Паралельний запуск тестів. Створення об'єктів, що використовуються у тестах, за допомогою фікстур. <i>Л-ра: 2, 3, 6</i>	2 / –	---	---
	Пз1 Модульне тестування на Python. Основи тестування. Прості тести. Пропуск тестів. Запуск вибраних тестів. Паралельний запуск тестів. Створення об'єктів, що використовуються у тестах, за	4 / –	звіт з лістингами тестів та лістингами програм	1 / ---

	допомогою фікстур. <i>Л-ра: 2, 3, 6</i>			
	Л12 Тестування документації та вимог. Поняття вимоги. Важливість вимог. Джерела і шляхи виявлення вимог. Рівні і типи вимог. Властивості якісних вимог. Техніки тестування вимог. Приклад аналізу і тестування вимог. Типові помилки при аналізі та тестуванні вимог. <i>Л-ра: 1, 5, 7</i>	2 / –	---	---
	Л62 Модульне тестування на Python. Запуск тестових сценаріїв. Керовані тести з пустими об'єктами. Виявлення непротестованого коду за допомогою coverage. Віртуальне оточення. Налаштування віртуального оточення. <i>Л-ра: 2, 3, 6</i>	2 / –	---	---
	Пз2 Модульне тестування на Python. Запуск тестових сценаріїв. Керовані тести з пустими об'єктами. Виявлення непротестованого коду за допомогою coverage. Віртуальне оточення. Налаштування віртуального оточення. <i>Л-ра: 2, 3, 6</i>	4 / –	звіт з лістингами тестів та лістингами програм	1 / ---
	Л3 Види і напрямки тестування. Спрощена класифікація тестування. Детальна класифікація тестування. Схема класифікації тестування. Класифікація по запуску коду на виконання, за доступом до коду та архітектурою додатку, за ступенем автоматизації, за рівнем деталізації додатку (за рівнем тестування), за ступенем важливості тестованих функцій (за рівнем функціонального тестування), за принципами роботи з додатком.	2 / –	---	---

	<i>Л-ра: 1, 5, 7</i>			
	Л63 Модульне тестування на Python. Використання virtualenv з tox. Повторне створення оточення. Використання різних версій Python. Інтеграція з іншими тестами. Політика тестування. <i>Л-ра: 2, 3, 6</i>	2 / –	---	---
	Пз3 Модульне тестування на Python. Використання virtualenv з tox. Повторне створення оточення. Використання різних версій Python. Інтеграція з іншими тестами. Політика тестування. <i>Л-ра: 2, 3, 6</i>	4 / –	звіт з лістингами тестів та лістингами програм	1 / ---
	Л4 Види і напрямки тестування. Класифікація за природою додатку, за фокусуванням на рівні архітектури додатку, за залученням кінцевих користувачів, за ступенем формалізації, за цілями і завданнями, за техніками та підходами, за моментом виконання (хронології). Альтернативні і додаткові класифікації тестування. Класифікація за методами білої та чорної скриньок. <i>Л-ра: 1, 5, 7</i>	2 / –	---	---
	Л64 Мета модульного тестування. Поточний стан модульного тестування. Мета модульного тестування. Використання метрик охоплення для вимірювання якості набору тестів. Властивості успішного набору тестів. Інтеграційні тести. <i>Л-ра: 2, 3, 6</i>	2 / –	---	---
	Пз4 Мета модульного тестування. Поточний стан модульного тестування. Мета модульного тестування. Використання метрик	4 / –	звіт з лістингами тестів та лістингами програм	1 / ---

	охоплення для вимірювання якості набору тестів. Властивості успішного набору тестів. Інтеграційні тести. <i>Л-ра: 2, 3, 6</i>			
Модульний контроль	МК1 Тест за матеріалами змістового модуля 1 <i>Л-ра: 1, 2, 3, 5, 6, 7</i>		тест	10 / ---
Змістовий модуль 2. Використання NUnit для тестування ПЗ				36 / ---
	Л5 Перевірка товарів та тест-кейсів, набори тест-кейсів. Чек-лист. Тест-кейс і його життєвий цикл. Атрибути тест-кейса. Інструментальні засоби керування тестуванням. Властивості якісних тест-кейсів. Набори тест-кейсів. Логіка створення ефективних перевірок. <i>Л-ра: 1, 5, 6, 7</i>	2 / –	---	---
	Л65 Структура тесту. Дослідження фреймворку xUnit. Повторне використання тестових фікстур. Іменування тесту. Рефакторинг до параметризованих тестів. Використання бібліотеки попереджень для подальшого вдосконалення тестів. <i>Л-ра: 2, 3, 4, 6</i>	2 / –	---	---
	Пз5 Структура тесту. Дослідження фреймворку xUnit. Повторне використання тестових фікстур. Іменування тесту. Рефакторинг до параметризованих тестів. Використання бібліотеки попереджень для подальшого вдосконалення тестів. <i>Л-ра: 2, 3, 4, 6</i>	4 / –	звіт з лістингами тестів та лістингами програм	1 / ---

	Л6 Звіти про дефекти. Поняття помилки, дефекту, збою, відмови. Звіт про дефект і його життєвий цикл. Атрибути звіту про дефект. Інструментальні засоби керування звітами про дефекти. Властивості якісних звітів про дефекти. Логіка створення ефективних звітів про дефекти. <i>Л-ра: 1, 5, 6, 7</i>	2 / –	---	---
	Л66 Чотири параметри хорошого тесту. Внутрішній зв'язок між першими двома атрибутами. Третій та четвертий параметри: швидкий зворотний зв'язок та ремонтпридатність. Поняття ідеального тесту. Вивчення відомих концепцій автоматизації тестів. <i>Л-ра: 2, 3, 4, 6</i>	2 / –	---	---
	Пз6 Чотири параметри хорошого тесту. Внутрішній зв'язок між першими двома атрибутами. Третій та четвертий параметри: швидкий зворотний зв'язок та ремонтпридатність. Поняття ідеального тесту. Вивчення відомих концепцій автоматизації тестів. <i>Л-ра: 2, 3, 4, 6</i>	4 / –	звіт з лістингами тестів та лістингами програм	1 / ---
	Л17 Оцінка трудозатрат, планування та звітність. Планування й звітність. Тест-план і звіт про результати тестування. Оцінка трудозатрат. <i>Л-ра: 1, 5, 6, 7</i>	2 / –	---	---
	Л67 Навантаження та перевірка крихкості. Стилi модульного тестування. Диференціація навантажень. Спостережувана поведінка та деталі реалізації. Зв'язок між навантаженнями та крихкістю випробувань. Стилi модульного тестування.	2 / –	---	---

	Розуміння функціональної архітектури. Перехід до функціональної архітектури та тестування на основі результатів. Розуміння недоліків функціональної архітектури. <i>Л-ра: 2, 3, 4, 6</i>			
	Пз7 Навантаження та перевірка крихкості. Стилi модульного тестування. Диференціація навантажень. Спостережувана поведінка та деталі реалізації. Зв'язок між навантаженнями та крихкістю випробувань. Стилi модульного тестування. Розуміння функціональної архітектури. Перехід до функціональної архітектури та тестування на основі результатів. Розуміння недоліків функціональної архітектури. <i>Л-ра: 2, 3, 4, 6</i>	4 / –	звіт з лістингами тестів та лістингами програм	1 / ---
	Л18 Приклади використання різних технік тестування. Позитивні і негативні тест-кейси. Класи еквівалентності та граничні умови. Доменне тестування і комбінації параметрів. Попарне тестування і пошук комбінацій. Дослідницьке тестування. Пошук причин виникнення дефектів. <i>Л-ра: 1, 5, 6, 7</i>	2 / –	---	---
	Л68 Рефакторинг на основі тестів. Ідентифікація коду для рефакторингу. Визначення результатів тестів для рефакторингу. Аналіз оптимального охоплення тестом. Обробка умовної логіки у контролерах. <i>Л-ра: 2, 3, 4, 6</i>	2 / –	---	---
	Пз8 Рефакторинг на основі тестів. Ідентифікація коду для рефакторингу. Визначення результатів тестів для рефакторингу. Аналіз	4 / –	звіт з лістингами тестів та лістингами програм	1 / ---

	оптимального охоплення тестом. Обробка умовної логіки у контролерах. <i>Л-ра: 2, 3, 4, 6</i>			
	Л9 Автоматизація тестування. Переваги і недоліки автоматизації. Области застосування автоматизації. <i>Л-ра: 1, 5, 6, 7</i>	2 / –	---	---
	Л69 Інтеграційне тестування. Поняття інтеграційного тесту. Приклад тестування інтеграції. Використання інтерфейсів для абстрактних залежностей. Передовий досвід інтеграційного тестування. Перевірка функціонування журналювання. <i>Л-ра: 2, 3, 4, 6</i>	2 / –	---	---
	Пз9 Інтеграційне тестування. Поняття інтеграційного тесту. Приклад тестування інтеграції. Використання інтерфейсів для абстрактних залежностей. Передовий досвід інтеграційного тестування. Перевірка функціонування журналювання. <i>Л-ра: 2, 3, 4, 6</i>	4 / –	звіт з лістингами тестів та лістингами програм	1 / ---
	Л10 Особливості автоматизованого тестування. Необхідні знання та навички. Особливості тест-кейсів у автоматизації. Технології автоматизації тестування. Автоматизація поза прямими завданнями тестування. <i>Л-ра: 1, 5, 6, 7</i>	2 / –	---	---
	Л610 Тестування бази даних. Передумови тестування бази даних. Керування транзакціями бази даних. Життєвий цикл даних тесту. Повторне використання коду у тестах. Загальні питання тестування бази даних.	2 / –	---	---

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

БАЗОВІ ДЖЕРЕЛА

Khorikov V. Unit Testing: Principles, Practices, and Patterns / V. Khorikov. – Manning Publications Co., 2020. – 305 p.

ДОДАТКОВІ ДЖЕРЕЛА

7. Jorgensen P. Software Testing. A Craftsman's Approach / P.C. Jorgensen. – Taylor & Francis Group, LLC, 2014. – 470 p.