

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «МОДИФІКАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»



Ступінь освіти	магістр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення
Тривалість викладання	Весняний семестр (3, 4 чверть)
Кількість кредитів	5 кредитів ЄКТС (150 годин)
Заняття:	весняний семестр
лекції:	2 год/тиждень
лабораторні заняття:	1 год/тиждень
Мова викладання	українська

Сторінка курсу в СДО НТУ «ДП»: <https://do.nmu.org.ua/course/view.php?id=4992>

Кафедра, що викладає Програмного забезпечення комп'ютерних систем



Викладач:

Бердник Михайло Геннадійович
Професор, д.т.н.

Персональна сторінка

<https://pzks.nmu.org.ua/ua/teachers/berdnykmg.php>

Е-mail:

Berdnyk.m.g@nmu.one

1. Анотація до курсу

Термін експлуатації програмного продукту неможливо передбачити на початку його створення, адже проблеми його використання можуть виникнути при виході нових операційних систем, фреймворків, нового обладнання та апаратно-технічних засобів, які диктують умови використання створюваного програмного забезпечення (ПЗ). Розробники ПЗ намагаються зберегти конкурентоздатність своєї продукції через постійну підтримку, додавання нових функцій та адаптацію наявних відповідно до нових вимог. Продуктивність та якість ПЗ суттєво залежить від ефективності модифікації та тестування. Виявлення дефектів та їх усунення під час створення ПЗ впливає на кінцеву вартість програмного продукту та загальний час на розробку, тому що залежить від етапу, на якому їх виявлено, а цей етап залежить від обраної методології командою розробників. Сучасні методи тестування ПЗ не дозволяють

однозначно і повністю виявити всі дефекти і встановити коректність функціонування програми, тому всі існуючі методи тестування діють у рамках формального процесу перевірки ПЗ, що досліджується або розробляється. Такий процес формальної перевірки може довести, що дефекти відсутні з точки зору використовованого методу. Тестування дозволяє зробити процес розробки ПЗ прозорим і керованим для всіх учасників проекту. У рамках курсу розглядаються такі питання: методи та засоби реінженерії ПЗ, рефакторинг та оптимізаційні перетворення ПЗ, поняття та види тестування, артефакти тестування, методи тестування, процес тестування ПЗ, чорний ящик - функціональне тестування, Білий ящик - структурне тестування, тестування потоків керування програми, модифікація ПЗ, тестування потоків даних програми, мутаційне тестування, аналіз умов застосування функціонального й структурного тестування, інтеграційне тестування компонентно-базованого ПЗ, оцінювання кількості тестів для інтеграційного тестування, оцінювання часу тестування на ранніх етапах життєвого циклу ПЗ.

2. Мета та завдання курсу

Мета дисципліни – є набуття знань, умінь та формування у магістрів системи компетентностей на рівні новітніх досягнень в області модифікації та тестування програмного забезпечення.

Завданнями дисципліни є:

- опанування теоретико-понятійної бази курсу;
- ознайомлення здобувачів з сучасними методами та засобами реінженерії програмного забезпечення;
- опанування загальних підходів до рефакторингу та оптимізаційних перетворень програмного забезпечення;
- ознайомлення здобувачів з особливостями та основними підходами до тестування програмного забезпечення.

3. Результати навчання

Дисциплінарні результати навчання:

- модифікувати архітектуру за допомогою методів реінженерії програмного забезпечення;
- модифікувати існуючі програмні продукти за допомогою рефакторингу програмного коду;
- забезпечувати якість програмного забезпечення за допомогою багатofакторного та багаторівневого тестування;
- здійснювати конфігурування та реінжиніринг програмного забезпечення з урахуванням вимог замовника;
- здійснювати тестування програмного забезпечення різними методами і засобами.

Дисциплінарні результати навчання сформовано на основі ПРН освітньо-професійної програми «Інженерія програмного забезпечення» другого (магістерського) рівня вищої освіти (РН08, РН10, РН11, РН13, РН15, РН16).

4. Структура курсу

Види та тематика навчальних занять	Внесок в загальну оцінку, %
ЛЕКЦІЇ	60
<p>1. Методи та засоби реінженерії програмного забезпечення Поняття Реінженерії програмного забезпечення. Етапи реінжинірингу. Цілі і завдання реінжинірингу. Аналіз та проектування. Тестування. Методи оптимізації процесу реінженерії програмних продуктів. Рентабельність реінжинірингу.</p>	
<p>2. Рефакторинг та оптимізаційні перетворення програмного забезпечення Поняття рефакторингу. Методи рефакторингу. Проблеми рефакторингу.</p>	
<p>3. Поняття та види тестування Поняття та мети тестування. Класифікація за: об'єктом тестування; рівнем знання про систему; ступенем автоматизації; ступенем ізольованості компонентів; часом проведення тестування; ознакою позитивності сценарію; ступенем підготовленості до тестування.</p>	
<i>Тестова контрольна робота №1 (за темами 1-3)</i>	20
<p>4. Артефакти тестування План тестування. Тест-кейс. Чек-лист. Баг Репорт.</p>	
<p>5. Методи тестування Тест дизайн. Класи еквівалентності. Метод кордонних (граничних) умов. Таблиця Прийняття Рішень. Тестування Станів і Переходів. Тестування сценаріїв використання. Передбачення помилки. Вичерпне тестування. Способи скорочення кількості тестових випадків: метод парного тестування; доменний аналіз; Input-Output Analysis.</p>	
<p>6. Процес тестування програмного забезпечення. Чорний ящик - функціональне тестування. Випадкове (стохастичне) тестування. Тестування за класами еквівалентності. Метод аналізу граничних умов</p>	
<p>7. Білий ящик - структурне тестування. Тестування потоків керування програми. Модифікація програмного забезпечення. Критерії тестування: покриття операторів (C0); покриття рішень (C1); покриття шляхів (C∞). Граничне тестування циклу. Внутрішнє тестування циклу. Простий критерій покриття умов. Критерій покриття умов/рішень. Модифікований критерій покриття умов/рішень. Комбінаторний критерій покриття умов/рішень.</p>	
<i>Тестова контрольна робота №2 (за темами 4-7)</i>	20

Види та тематика навчальних занять	Внесок в загальну оцінку, %
<p>8. Тестування потоків даних програми. Модифікація програмного забезпечення. Мутаційне тестування. Критерії: “all-defs”; “all p-uses”; “all c-uses / some p-uses”; “all p-uses / some c-uses”; “all uses” (всі використання); “all du-paths”. Застосування мутаційного критерію.</p>	
<p>9. Аналіз умов застосування функціонального й структурного тестування. Інтеграційне тестування компонентно-базованого програмного забезпечення. Критерії інтеграційного тестування: покриття операцій інтерфейсу; покриття викликів операцій; покриття активізації інтерфейсу; покриття послідовностей; покриття залежностей; покриття послідовностей викликів операцій; покриття послідовностей активізацій; покриття паралельних потоків.</p>	
<p>10. Оцінювання кількості тестів для інтеграційного тестування. Оцінювання часу тестування на ранніх етапах життєвого циклу програмного забезпечення. Створення тестів на основі UML діаграм варіантів використання чату. Оцінювання часу тестування за допомогою UML діаграм варіантів використання.</p>	
<p><i>Тестова контрольна робота №3 (за темами 8-10)</i></p>	20
ЛАБОРАТОРНІ ЗАНЯТТЯ	40
<p>Лабораторна робота №1. Тема: Тестова документація й артефакти тестування. Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів роботи з тестовою документацією та складання тестової документації відповідно до вимог (План тестування. Тест-кейс. Чек-лист. Баг Репорт).</p>	
<p><i>Звіт з виконання та захист лабораторної роботи №1</i></p>	5
<p>Лабораторна робота №2 Тема. Основний інструментарій для аналізу якості при роботі з програмним забезпеченням. Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів роботи з баг-трекінговими системам (Atlassian JIRA).</p>	
<p><i>Звіт з виконання та захист лабораторної роботи №2</i></p>	5
<p>Лабораторна робота №3 Тема. Процес тестування програмного забезпечення. Чорний ящик - функціональне тестування. Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів роботи з випадковим (стохастичним) тестуванням, тестуванням за класами еквівалентності, методом аналізу граничних умов.</p>	
<p><i>Звіт з виконання та захист лабораторної роботи №3</i></p>	5
<p>Лабораторна робота №4 Тема. Тестування потоків керування програми. Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів тестування за допомогою критеріїв:</p>	

Види та тематика навчальних занять	Внесок в загальну оцінку, %
тестування: покриття операторів (C0); покриття рішень(C1); покриття шляхів (C∞).	
<i>Звіт з виконання та захист лабораторної роботи №4</i>	5
Лабораторна робота №5 Тема: Методи побудови якісного процесу тестування. Визначення тестових конфігурацій. Формування тестових даних. Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів тестування за допомогою: граничного тестування циклу; внутрішнього тестування циклу; простого критерія покриття умов; критерія покриття умов/рішень; модифікованого критерія покриття умов/рішень; комбінаторного критерій покриття умов/рішень.	
<i>Звіт з виконання та захист лабораторної роботи №5</i>	5
Лабораторна робота №6 Тема: Тестування потоків даних програми. Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів тестування за допомогою критеріїв: “all-defs”; “all p-uses“; “all c-uses / some p-uses“; “all p-uses / some c-uses“; “all uses“ (всі використання); “all du-paths”.	
<i>Звіт з виконання та захист лабораторної роботи №6</i>	5
Лабораторна робота №7 Тема: Інтеграційне тестування компонентно-базованого програмного забезпечення. Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів тестування за допомогою критеріїв: покриття операцій інтерфейсу; покриття викликів операцій; покриття активізації інтерфейсу.	
<i>Звіт з виконання та захист лабораторної роботи №7</i>	5
Лабораторна робота №8 Тема: Оцінювання часу тестування на ранніх етапах життєвого циклу програмного забезпечення. Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів створення тестів на основі UML діаграм варіантів використання, оцінювання часу тестування за допомогою UML діаграм варіантів використання.	
<i>Звіт з виконання та захист лабораторної роботи №8</i>	5
Загальна кількість балів	100

5. Технічне обладнання та/або програмне забезпечення*

Активованій акаунт університетської пошти (student.i.p.@nmu.one) на Ofіс365. Використовуються лабораторії кафедри програмного забезпечення комп'ютерних систем (комп'ютерне та мультимедійне обладнання). Дистанційна платформа Moodle, MS Office 365, Microsoft Teams, пакети приладних програм MS Office. Для виконання лабораторних робіт застосовуються наступні програмні

продукти: Microsoft Visual Studio Community, JIRA або Bugzilla, FreeMat - відкритий безкоштовний клон Matlab, Microsoft Visio або StarUML.

6. Система оцінювання та вимоги

6.1. Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	відмінно
74-89	добре
60-73	задовільно
0-59	незадовільно

Загальні критерії досягнення результатів навчання відповідають описам 7-го кваліфікаційного рівня НРК.

6.2. Здобувачі вищої освіти можуть отримати **підсумкову оцінку** з навчальної дисципліни **на підставі поточного оцінювання знань** за умови, якщо набрана кількість балів з поточного тестування та виконання і захисту практичних робіт складатиме не менше 60 балів.

Теоретична частина оцінюється за результатами здачі трьох контрольних тестових робіт, кожна з яких містить тестові запитання з однією вірною відповіддю, максимальна кількість – 20 балів за кожен контрольну роботу (розподіл % за окремими контрольними роботами див. в таблиці розділу 4). Загалом за три контрольні тестові роботи отримується **максимум 60 балів**, тобто 60% від оцінки за дисципліну.

Лабораторні роботи (вісім лабораторних робіт – у вигляді індивідуального завдання з кожної, розподіл % див. в таблиці розділу 4) виконуються у письмовому вигляді (звіт з кожної роботи оцінюється в межах 5 балів, загалом лабораторні роботи враховуються, як 40% (максимум 40 балів). Лабораторні роботи захищаються у вигляді опитування. У сумі за практичну частину курсу при поточному оцінюванні отримується **максимум 40 балів**.

Отримані бали за теоретичну частину та лабораторні роботи додаються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за поточною успішністю здобувач вищої освіти може набрати **100 балів**.

Максимальне оцінювання поточного контролю в балах:

Теоретична частина	Практична частина	Разом
60	40	100

6.3. Критерії оцінювання підсумкової роботи. У випадку якщо здобувач вищої освіти за поточною успішністю отримав менше 60 балів та/або прагне поліпшити оцінку проводиться **підсумкове оцінювання (іспит)** під час сесії. **Іспит** проводиться у вигляді комплексної контрольної роботи, яка включає запитання з теоретичної та практичної частини курсу. Білет складається з **60 тестових завдань** з чотирма варіантами відповідей, одна правильна відповідь

оцінюється в 1 бал (разом 60 балів) та 10 тестових завдань з практичної частини, кожне з запитань оцінюється максимум у 4 бали (разом 40 балів), причому:

- 4 бали – відповідність еталону;
- 3 бали – відповідність еталону з незначними помилками;
- 2 бали – часткова відповідність еталону, питання повністю не розкриті;
- 1 бал – невідповідність еталону, але відповідність темі запитання;
- 0 балів – відповідь не наведена або не відноситься до теми запитання.

Отримані бали за відповіді на запитання з теоретичної та практичної частини курсу додаються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за підсумковою роботою здобувач вищої освіти може набрати 100 балів.

7. Політика курсу

7.1. Політика щодо академічної доброчесності. Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення плагіату у Національному технічному університеті "Дніпровська політехніка". <https://cutt.ly/IBesJEc> У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

7.2. Комунікаційна політика. Здобувачі вищої освіти повинні мати активовану корпоративну університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

7.3. Політика щодо перескладання. Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання підсумкового оцінювання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

7.4 Політика щодо оскарження оцінювання. Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

7.5. Відвідування занять. Для здобувачів вищої освіти денної форми відвідування занять є обов'язковим. Поважними причинами для неявки на заняття є хвороба, участь в університетських заходах, академічна мобільність, які необхідно підтверджувати документами. Про відсутність на занятті та причини відсутності здобувач вищої освіти має повідомити викладача або особисто, або через старосту.

7.6. Опитування. Наприкінці вивчення курсу та перед початком сесії здобувача вищої освіти буде запропоновано анонімно заповнити електронні анкети (Microsoft Forms Office 365), які буде розіслано на ваші університетські поштові скриньки. Заповнення анкет є важливою складовою вашої навчальної активності, що дозволить оцінити дієвість застосованих методів викладання та врахувати ваші пропозиції стосовно покращення змісту навчальної дисципліни.

8. Рекомендовані джерела інформації

1. Тестування програмного забезпечення. Навчальний посібник / А.С. Авраменко, В.С. Авраменко, Г.В. Косенюк. – Черкаси: ЧНУ імені Богдана Хмельницького, 2017. 284с.
2. Чистий код. Створення і рефакторинг за допомогою Agile / Роберт Сесіл Мартін. – «Фейбула», 2019. – 448с. ISBN 978-617-09-5285-1.
3. Рефакторинг. Поліпшення існуючого коду / М. Фаулер, К. Бек, Дж. Брант, В. Опдайк, Д. Робертс. – «діалектика», 2003. – 448с.
4. Martin Fowler. Refactoring: Improving the Design of Existing Code (2nd Edition). Addison-Wesley Professional, 2019. – 424p. ISBN 978-0134757599.
5. Scott Ambler. Refactoring Databases: Evolutionary Database Design. Addison-Wesley Signature Series, 2017.– 384 p. ISBN 978-0-321-77451-4.
6. Con Mike User Stories Applied for Agile Software Development. Addison-Wesley Professional, 2011. – 268 p. ISBN 0-321-20568-5.