

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «ЯКІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕСТУВАННЯ»



Ступінь освіти	бакалавр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення
Тривалість викладання	6 семестр (11, 12 чверть)
Кількість кредитів	4 кредити ЄКТС (120 годин)
Заняття:	
лекції:	2 години/тиждень
лабораторні заняття:	1 година/тиждень
Мова викладання	українська

Сторінка курсу в СДО НТУ «ДП»: <https://do.nmu.org.ua/>

Кафедра, що викладає Програмного забезпечення комп'ютерних систем

Викладач(і):



Бердник Михайло Геннадійович
Професор, доктор технічних наук

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

E-mail: berdnyk.m.g@nmu.one

1. Анотація до курсу

Одним із найважливіших кроків при розробці програмних продуктів є тестування. Важливими цілями тестування є відповідність розробленої програми заданим вимогам, дотримання логіки у процесах обробки даних та отримання вірних кінцевих результатів.

Етап тестування, спрямований на перевірку відповідності між очікуваними результатами та реальною поведінкою програми на спеціально підбраному наборі тестів (тестових даних), є одним із найдорожчих та трудомістких, і може займати до 40–60% від загального часу створення ПЗ. Тому розробка моделей та алгоритмів інтелектуальної підтримки етапу тестування ПЗ є актуальним завданням.

2. Мета та завдання курсу

Мета дисципліни: формування стійких знань та засвоєння базових понять, які використовують сучасні методи і моделі тестування програмного забезпечення;

вивчення загальних принципів та методів застосування сучасної теорії тестування, верифікації та валідації програмного забезпечення.

Завдання курсу:

- формування стійких знань та засвоєння базових понять тестування програмного забезпечення для реалізації вимог замовника;
- вивчення загальних принципів та методів застосування тестування програмного забезпечення для забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення.

3. Результати навчання:

Дисциплінарні результати навчання:

- застосовувати методи верифікації та валідації програмного забезпечення;
- застосовувати методи оцінки та забезпечення якості програмного забезпечення.

Дисциплінарні результати навчання сформовано на основі ПРН освітньо-професійної програми «Інженерія програмного забезпечення» першого (бакалаврського) рівня вищої освіти (ПР19, ПР20).

4. Структура курсу

Види та тематика навчальних занять	Внесок в загальну оцінку, %
ЛЕКЦІЇ	60
Тема 1. 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	
Тема 2. Артефакти тестування. План тестування. Тест-кейс. Чек-лист. Баг Репорт.	
Тема 3. 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.	
Тема 4. Процес тестування програмного забезпечення. Чорний ящик - функціональне тестування. Випадкове (стохастичне) тестування. Тестування за класами еквівалентності. Метод аналізу граничних умов	
<i>Тестова контрольна робота №1 (за темами 1-4).</i>	30
Тема 5. 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.	

Види та тематика навчальних занять	Внесок в загальну оцінку, %
<p>Тема 6. 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>Тема 7. 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>Тема 8. Оцінювання кількості тестів для інтеграційного тестування. Оцінювання часу тестування на ранніх етапах життєвого циклу програмного забезпечення. Методи верифікації та валідації 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 №2 (за темами 5-8).</i></p>	30
ЛАБОРАТОРНІ ЗАНЯТТЯ	40
<p style="text-align: center;">Лабораторна робота №1.</p> <p>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>	5
<p style="text-align: center;">Лабораторна робота №2</p> <p>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>	5
<p style="text-align: center;">Лабораторна робота №3</p> <p>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>	5
<p style="text-align: center;">Лабораторна робота №4</p> <p>Тема. Тестування потоків керування програми. Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів тестування за допомогою критеріїв: тестування: покриття операторів (C0); покриття рішень(C1); покриття шляхів (C∞).</p>	5
<p style="text-align: center;">Лабораторна робота №5</p> <p>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</p>	5

Види та тематика навчальних занять	Внесок в загальну оцінку, %
умов; критерія покриття умов/рішень; модифікованого критерія покриття умов/рішень; комбінаторного критерій покриття умов/рішень.	
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”.	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.	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.	5
РАЗОМ	100

5. Технічне обладнання та/або програмне забезпечення

Використовуються лабораторії кафедри програмного забезпечення комп'ютерних систем (комп'ютерне та мультимедійне обладнання). Дистанційна платформа Moodle, MS Office 365, Microsoft Teams, Microsoft Visual Studio Community.

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

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 лабораторних робіт складатиме не менше 60 балів.

Теоретична частина оцінюється за результатами здачі двох контрольних тестових робіт, кожна з яких містить тестові закриті запитання з однією вірною відповіддю (максимальна кількість – 30 балів за кожною тестовою роботою). Загалом за дві контрольні тестові роботи отримується **максимум 60 балів**, тобто 40% від оцінки за дисципліну.

Лабораторні роботи (вісім робіт – у вигляді індивідуального завдання з кожної, розподіл % див. в таблиці розділу 4) виконуються у письмовому вигляді (звіт з кожної роботи оцінюється в межах балів, представлених в таблиці розділу 4, загалом лабораторні враховуються як 60% (максимум 60 балів).

У сумі за практичну частину курсу при поточному оцінюванні отримується **максимум 40 балів**.

Отримані бали за теоретичну частину та лабораторні роботи додаються і є поточною успішністю 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 (екзамен)** під час сесії.

Екзамен проводиться у вигляді комплексної контрольної роботи, яка включає запитання з теоретичної та практичної частини курсу. Білет складається з **40 тестових завдань** з чотирма варіантами відповідей, одна правильна відповідь оцінюється в 1 бал (**разом 40 балів**) та **2 завдань** з практичної частини, кожне з запитань оцінюється максимум у 30 балів (**разом 60 балів**).

Отримані бали за тестові завдання та завдання з практичної частини 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". 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.

8 Рекомендовані джерела інформації.

1. Авраменко А.С. Тестування програмного забезпечення. Навчальний посібник / А.С. Авраменко, В.С. Авраменко, Г.В. Косенюк.– Черкаси: ЧНУ імені Богдана Хмельницького, 2017. 284с. 3. Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник / За ред. Крепич С.Я., Співак І.Я. / для бакалаврів галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення». – Тернопіль: ФОП Паляниця В.А., 2020. – 478с

2. Технології програмування та створення програмних продуктів: конспект лекцій /укладач О. В. Алексенко. – Суми : Сумський державний університет, 2013. – 133 с.

3. Тимошенко Ю.О., Дідковська М.В., Кобринський С.Ю. Розробка методу функціональної сегментації для тестування програмного забезпечення // Наукові вісті, №5(37), – Київ, 2004. – С. 48-56.

4. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення». – Полтава: ПолтНТУ, 2017. – 218 с.

5. Panagiotis Leloudas./ Introduction to Software Testing: A Practical Guide to Testing, Design, Automation, and Execution 1st ed. Edition/ Apress: 2023./ 232 p

6. Claude Y. Laporte, Alain April./ Software Quality Assurance./ Wiley-IEEE Computer Society Press 2018 465 pp/ ISBN: 9781118501825

Додаткова:

7. ДСТУ ISO 9000:2007. Системи управління якістю. Основні положення та словник термінів. – К.: Держспоживстандарт, 2008. – [Чинний від 2008-01- 01] – 35 с.– (Державний стандарт).
8. Дуднік А.С. Методичні рекомендації дисципліни «Якість програмного забезпечення та тестування» (для освітньо-кваліфікаційного рівня «бакалавр»). – К.: МАУП, 2019. – 19 с.
9. ДСТУ 2873-94. Системи обробки інформації. Програмування. Терміни та визначення. - К.: Держстандарт України, 1994.
10. ДСТУ 2941-94. Системи оброблення інформації. Розроблення систем. Терміни та визначення. - К.: Держстандарт України, 1994.
11. ДСТУ 4302:2004. Інформаційні технології. Настанови щодо документування комп'ютерних програм. - К.: Держстандарт України, 2004.
12. ДСТУ ISO/IEC 12119:2003. Інформаційні технології. Пакети програм тестування і вимоги до якості. - К.: Держстандарт України, 2003.
13. ДСТУ ISO/IEC 14764:2002. Інформаційні технології. Супроводження програмного забезпечення. - К.: Держстандарт України, 2002.
14. ДСТУ ISO/IEC 90003:2006. Програмна інженерія. Настанови щодо застосування ISO 9001:2000 до програмного забезпечення (ISO/IEC 90003:2004, IDT) - К.: Держстандарт України, 2006.
15. ДСТУ ISO/IEC TR 12182:2004. Інформаційні технології. Класифікація програмних засобів (ISO/IEC TR 12182:1998, IDT) - К.: Держстандарт України, 2004.
16. ДСТУ ISO/IEC 14598-1:2004. Інформаційні технології. Оцінювання програмного продукту. Частина 1. Загальний огляд (ISO/IEC 14598-1:1999, IDT) - К.: Держстандарт України, 2004.
17. ДСТУ ISO/IEC 15288:2005. Інформаційні технології. Процеси життєвого циклу системи (ISO/IEC 15288:2002, IDT) - К.: Держстандарт України, 2005.
18. ДСТУ ISO/IEC 15939:2008. Інженерія систем і програмних засобів. Процес вимірювання. - К.: Держстандарт України, 2008.
19. ДСТУ 3327-96. Методика випробування процесорів мов програмування. Загальні вимоги. - К.: Держстандарт України, 1996.
20. ДСТУ ISO/IEC TR 14369:2003. Інформаційні технології. Мови програмування, їхнє середовище та системний інтерфейс. Настанова щодо підготовки незалежних від мов специфікацій послуг. - К.: Держстандарт України, 2003.
21. ДСТУ 4072:2001. Інформаційні технології. Мови програмування, їхнє середовище та системний інтерфейс. Настанова щодо підготовки незалежних від мов виклик процедур. - К.: Держстандарт України, 2001.
22. ДСТУ ISO/IEC 2382-15:2005. Інформаційні технології. Словник термінів. Частина 15. Мови програмування (ISO/IEC 2382-15:1999, IDT) - К.: Держстандарт України, 2005.
23. Software testing and quality assurance/ Kshirasagar Naikand Priyadarshi Tripathy./ ISBN 978-0-471-78911-6 ./ John Wiley & Sons, Inc. 2008, 616 pp

Інформаційні ресурси:

1. [Електронний ресурс] “Що таке тестування програмного забезпечення та яке його значення?” <https://www.quality-assurance-group.com/shho-take-testuvannya-programnogo-zabezpechennya-ta-yake-jogo-znachennya/>.
2. [Електронний ресурс] лекція “Тестування програмного продукту” <http://lib.mdpu.org.ua/e-book/vstup/L11.htm>.
3. [Електронний ресурс] Basics of BDD intesting, by Alex McPeak <https://crossbrowsertesting.com/blog/development/what-is-bdd-cucumber/> .
4. [Електронний ресурс] The Pros and Cons of Behavior-Driven Development, by EmilieMaxie <https://www.verypossible.com/blog/the-pros-and-cons-of-behavior-drivendevelopment>.
5. [Електронний ресурс] What is BDD? An Introduction to Behavioral Driven Development, by Jithin Nair <https://blog.testlodge.com/what-is-bdd/> .
6. [Електронний ресурс] Writing BDD Test Scenarios, by Richard Holmes <https://www.departmentofproduct.com/blog/writing-bdd-test-scenarios/> .
7. [Електронний ресурс] BDD Guidelines and Best Practices, by Amir Ghahrai <https://www.testingexcellence.com/bdd-guidelines-best-practices/> .
8. [Електронний ресурс] Behavior Driven Development (BDD) and Acceptance Testing Driven Development (ATDD) Tools. <http://www.agilesoftwaretools.com/behaviordrivendevelopment.php>