

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчально-виховної роботи
Наталія ТМЄНОВА

« 29 » *серпня* 2026 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«МЕТОДИ ТЕСТУВАННЯ ТА ОЦІНКИ НАДІЙНОСТІ
ПРОГРАМНИХ СИСТЕМ»**

для студентів

галузь знань	F Інформаційні технології
спеціальність	F2 Інженерія програмного забезпечення
освітній рівень	магістр
освітня програма	Інженерія програмного забезпечення
вид дисципліни	вільного вибору студента

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	3
Кількість кредитів ECTS	5
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	залік

Викладачі: **Юрчук І.А.**, кандидат фізико-математичних наук, доцент

Пролонговано: на 20___/20___ н.р. _____ (підпис) _____ (ПІБ) «___» 20___р.

на 20___/20___ н.р. _____ (підпис) _____ (ПІБ) «___» 20___р.

КИЇВ – 2026

Розробник: **Юрчук Ірина Аркадіївна**, кандидат фізико-математичних наук, доцент, доцент кафедри програмних систем і технологій

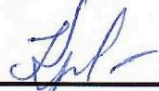
Завідувач кафедри
програмних систем і технологій


(підпис) проф. Олексій БИЧКОВ

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних технологій

Протокол № 11 від 26 червня 2026 року

Голова науково-методичної комісії 
(підпис) Ганна КРАСОВСЬКА

© І.А. Юрчук, 2026 рік
© _____, 20__ рік
© _____, 20__ рік

1. Мета дисципліни - засвоєння студентами базових знань щодо основних понять в галузі тестування програмних систем та оцінки їх надійності, опанування сучасних методів статичного й динамічного тестування та задач, моделей і методів інженерії надійності, технологічних засад розгортання ресурсно-ефективних процесів тестування й забезпечення надійності програмних систем в їх життєвому циклі – для успішної діяльності в складі груп тестування, гарантування якості, вдосконалення процесу розроблення програмних продуктів.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

Для успішного вивчення дисципліни «Методи тестування та оцінки надійності програмних систем» студенти повинні:

1. Знати:

1. Основні моделі життєвого циклу програмних систем;
2. Основні підходи та базові шаблони проектування програмних систем;
3. Основні концепції процедурного та об'єктно-орієнтованого програмування;
4. Програмні, організаційні й технологічні засоби забезпечення конкурентоздатної якості програмних систем;
5. Базові методи математичної статистики.

2. Вміти:

1. Вилучати, балансувати та документувати вимоги до програмної системи;
2. Безпечно працювати з програмним кодом у системах контролю версій;
3. Обчислювати стандартні статистичні характеристики вибірки даних (середнє, медіану, середньоквадратичне й стандартне відхилення, квантілі);
4. Застосовувати критерії згоди для перевірки статистичних гіпотез.

3. Володіти:

1. Базовими навичками застосування поширених інтегрованих середовищ розробки програмного забезпечення, зокрема Microsoft Visual Studio, J2EE;
2. Базовими навичками застосування систем контролю версій, зокрема Git.

3. Анотація навчальної дисципліни:

Дисципліна «Методи тестування та оцінки надійності програмних систем» формує вміння та навички формування стратегії тестування програмних систем на базі класифікації можливих методів та оцінки їх надійності на основі різних моделей.

4. Завдання вивчення дисципліни полягає у формуванні у студентів знань про основні методи тестування та оцінки надійності, їх математичну основу та програмні технології, що їх реалізують. Дисципліна спрямована на досягнення компетентностей щодо здатності обирати метод тестування та метод оцінки надійності програмної системи, виходячи із поставлених вимог та реалізованого функціоналу.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та / або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання ¹ (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Методи статичного (індивідуального й колективного) та динамічного (модульного, інтеграційного, функцій, системного, приймального) тестування локальних і Веб-застосунків за наявності й відсутності	лекція	виконання лабораторних робіт, індивідуально	5%

¹ Критерії оцінювання див. пункт 7.1.

	доступу до тестованого коду та його специфікації у планових, ітеративних і гнучких програмних проектах;		STEM-проект	
1.2	підходи та засоби автоматизації тестування коду різних рівнів;	лекція, лаборатор -не заняття		5%
1.3	моделі ефективних процесів тестування й забезпечення надійності та ступенів їх зрілості, а також підходи до розгортання цих процесів згідно з моделями;	лекція, лаборатор -не заняття		5%
1.4	задачі та методи встановлення, прогнозування й оцінювання показників надійності програмних систем без даних їх тестування та за їх наявності;	лекція, лаборатор -не заняття		5%
1.5	методи побудови стратегії демонстраційного тестування надійності та прогнозування оптимального моменту випуску програмної системи;	лекція, лаборатор -не заняття		5%
1.6	основні вбудовані засоби та автономні каркаси автоматизованого тестування, інструментальні засоби визначення надійності програмних систем;	лекція, лаборатор -не заняття		5%
2.1	планувати, виконувати й документувати статичне тестування в плановому і гнучкому програмному проекті;	лаборатор -не заняття	виконання лабораторних робіт, індивідуально го STEM-проекту	до 30%
2.2	планувати, проектувати, виконувати й документувати модульні, інтеграційні, системні, регресійні, приймальні тести та тести функцій у плановому і гнучкому програмному проекті з ефективним застосуванням засобів автоматизації тестування;	лаборатор -не заняття		
2.3	автоматизовано керувати життєвим циклом виявлених дефектів;	лаборатор -не заняття		
2.4	обґрунтовано вибирати модель процесу тестування в організації-розробнику, розгортати і вдосконалювати його згідно з нею, визначати необхідний рівень надійності програмної системи й відстежувати його досягнення;	лаборатор -не заняття		
2.5	визначати рівень надійності програмної системи, прийнятний для всіх зацікавлених сторін та відстежувати й перевіряти його досягнення;	лаборатор -не заняття		
3	комунікація: здатність працювати в команді, вироблення у студентів практичних навиків проведення дослідження і оптимального групового вирішення актуальних проблем із застосуванням відповідних методів і прийомів.	лаборатор -не заняття, аналітичне дослідження, дискусія, вирішення	презентація, дискурс	до 20%

4	автономність: продемонструвати розуміння особистої відповідальності за професійні та/або управлінські рішення.	<i>конкретних задач та ситуацій</i>		до 20%
---	---	-------------------------------------	--	--------

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання (необов'язково для вибіркокових дисциплін, які не входять до блоків спеціалізації)

Результати навчання дисципліни (код)	1	1	1	1	1	1	2	2	2	2	2	3	4	
Програмні результати навчання (назва)	1	2	3	4	5	6	1	2	3	4	5	3	4	
Програмні результати (Освітня програма Спеціальність 121-Інженерія програмного забезпечення. Другий (магістерський) рівень)														
ПРН 21. Вміти обирати та застосовувати методи і алгоритми інтелектуального аналізу даних та машинного навчання у задачах класифікації, прогнозування, кластерного аналізу з використанням програмних інструментів підтримки багатовимірної аналізу даних на основі технологій DataMining, TextMining, WebMining.	+		+	+			+	+	+	+		+	+	+
ПРН 21.2. Вміти розробляти складні програми різних архітектур, мотивовано обираючи наукові методи та підходи для створення і супроводження математичного та програмного забезпечення відповідно до специфікацій поставлених задач комп'ютерного зору.	+	+	+	+	+	+	+	+	+	+	+	+	+	+
ПРН 21.3. Вміти використовувати принципи і концепції передових програмних та обчислювальних технологій	+	+	+	+	+	+		+	+	+		+	+	

7. Схема формування оцінки

7.1. Форми оцінювання студентів:

- семестрове оцінювання:

- **поточний контроль** шляхом виконання та захисту лабораторних робіт;
- **поточний контроль** шляхом виконання та захисту індивідуального STEM-проєкту;
- письмова **модульна контрольна робота**;

- підсумкове оцінювання (залік):

- залік виставляється за результатами семестрового оцінювання роботи студента максимум **100 балів**.

Питома вага результатів навчання у підсумковій оцінці за умови її опанування на належному рівні така:

- ✓ результати навчання – **1.1 – 1.6 (знання)** – до 5% за кожен;
- ✓ результати навчання – **2 (вміння)** - до 30%;
- ✓ результати навчання – **3 (комунікація)** - до 20%
- ✓ результати навчання – **4 (автономність)** - до 20%

Контроль знань студентів здійснюється за модульно-рейтинговою системою. Результати навчальної діяльності студентів оцінюються за 100-бальною шкалою.

7.2. Організація оцінювання:

У змістовий модуль 1 (ЗМ1) входять теми 1-3, а у змістовий модуль 2 (ЗМ2) – теми 4-7. Обов'язковим для заліку є виконання і захист студентом лабораторних робіт та індивідуального STEM-проєкту, що передбачені робочою програмою навчальної дисципліни, а також здача модульної контрольної роботи з мінімальною кількістю балів – 12.

Оцінювання за формами контролю:

	Мін. – 60 балів	Макс. – 100 балів
Виконання та захист лабораторних робіт	18	30
Виконання та захист індивідуального STEM-проєкту	30	50
Модульна контрольна робота	12	20

Поточне семестрове оцінювання:

- оцінюванню підлягають модульна контрольна робота, звіти і захист лабораторних робіт та індивідуального STEM-проєкту;
- оцінюванню в індивідуальному STEM-проєкті підлягають програмна якість продукту, оформлена пояснювальна записка до нього та його захист відповідно до «Рекомендацій щодо оцінювання освітніх STEM-проєктів», затверджених на засіданні кафедри програмних систем і технологій, протокол № 16 від «01» травня 2026 р.;
- упродовж семестру, після завершення лекційного матеріалу орієнтовно на 14 тижні навчання при умові здачі та захисту лабораторних робіт проводиться письмова модульна контрольна робота із відкритими питаннями.

Оцінювання індивідуального проєкту:

Критерії, що оцінюються	Мін. –	Макс. –
-------------------------	--------	---------

	30 балів	50 балів
Програмна якість продукту: функціональна коректність, рівень проведеного тестування та якість коду	12	20
Захист індивідуального STEM-проєкту: відповіді на питання та демонстрація роботи продукту	9	15
Пояснювальна записка, яка містить опис призначення продукту, його архітектуру, функціонал, алгоритми роботи та інструкцію користувача у відповідності до стандартів	9	15

Перескладання лабораторної роботи чи модульної контрольної роботи з метою підвищення оцінки не допускається.

При результаті семестрового оцінювання 60 балів і вище студенту виставляється зараховано.

	Семестрова кількість балів	Підсумкова оцінка
Мінімум	60	60
Максимум	100	100

7.3. Шкала відповідності оцінок

Зараховано / Passed	60-100
Незараховано / Fail	0-59

8. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ І ЛАБОРАТОРНИХ ЗАНЯТЬ

№ п/п	Номер та назва теми	Кількість годин		
		Лекції	Лабора-торні	СР
Змістовий модуль 1. Методи тестування програмних систем				
1.	Тема 1. Основні поняття в галузі тестування програмних систем. Класифікація методів тестування.	4		10
2.	Тема 2. Мануальне тестування програмних систем	2	4	15
3.	Тема 3. Автоматизоване тестування. Роль ІІІ у процесі автоматизації	4	6	15
Змістовий модуль 2. Методи оцінки надійності програмних систем				
4.	Тема 4. Інженерія надійності програмних систем: цілі, основні поняття, задачі	2	2	12
5.	Тема 5. Моделі з фактором дискретного часу. Динамічні та статичні моделі оцінки надійності	4	8	14
6.	Тема 6. Адитивні моделі оцінки надійності програмного забезпечення	2	2	12

7.	Тема 7. Моделі оцінки надійності програмного забезпечення з фактором часу на основі ланцюгів Маркова	2	4	14
8.	Модульна контрольна робота		2	9
	ВСЬОГО	20	28	101

Загальний обсяг 150 год., в тому числі:

Лекції – 20 год.;
Лабораторні заняття – 28 год.;
Самостійна робота – 101 год;
Консультація – 1 год.

Теми лабораторних робіт

- 1-2. Тема 2. Формування стратегії мануального тестування програмної системи тестування та її реалізація.
- 3-4. Тема 3. Формування стратегії автоматизованого тестування програмної системи та її реалізація.
5. Тема 4. Оцінка надійності програмної системи експертним методом.
- 6-8. Теми 5. Моделі з фактором дискретного часу. Динамічні та статичні моделі оцінки надійності.
9. Тема 6. Адитивні моделі оцінки надійності програмного забезпечення.
10. Тема 7. Моделі оцінки надійності програмного забезпечення з фактором часу на основі ланцюгів Маркова
- 11-13. Теми 2-7. Реалізація та захист індивідуального STEM-проєкту.
14. **Модульний контроль.**

Самостійна робота студента, її зміст та обсяг

№ п/п	Зміст самостійної роботи студента	Обсяг СРС (годин)
1	Підготовка до лекційних занять (теми 1-7)	45
2	Підготовка до лабораторних занять (згідно переліку)	22
3	Підготовка та виконання індивідуального STEM-проєкту	25
4	Підготовка до модульної контрольної роботи	9
Всього		101

РЕКОМЕНДОВАНА ЛІТЕРАТУРА:

Основна:

1. Рекомендації щодо оцінювання освітніх STEM-проєктів [Електронний ресурс]. – Режим доступу: <https://pst.knu.ua/academics/bachelor/evaluation-stem-projects/>

2. Бобало Ю.Я. Математичні моделі та методи аналізу надійності радіоелектронних, електротехнічних та програмних систем / Ю.Я. Бобало та ін. – Львів, Видавництво Львівської політехніки, 2013. – 300 с.
3. Яковина В.С. Моделі, методи та засоби аналізу надійності програмних систем: монографія / Яковина В.С., Федасюк Д.В., Сенів М.М., Нитребич О.О. – Львів: Видавництво Львівської політехніки, 2015.
4. Lyu M. R. Software Reliability Engineering: A Roadmap / Michael R. Lyu // Future of Software Engineering (FOSE'07), Minneapolis, USA, May 2007. – P. 153–170.
5. Kwon Y.-M. A Markov Reward Model for Software Reliability // The Next Generation Software (NGS) Workshop at International Parallel and Distributed Processing Symposium (IPDPS). – 2007.
6. Myers G. J., Sandler C. and Badgett T. The art of software testing . – Wiley, 2011. – 256 p.

Додаткова:

1. Волочій Б.Ю. Технологія моделювання алгоритмів поведінки інформаційних систем: монографія / Б.Ю. Волочій. – Львів: Вид-во Нац. ун-ту “Львівська політехніка”, 2004. – 220 с.
2. Чабанюк Я.М. Побудова і дослідження моделі надійності програмного забезпечення з індексом величини проекту / Я.М. Чабанюк, В.С. Яковина, Д.В. Федасюк, М.М. Сенів, У.Т. Хімка // Інженерія програмного забезпечення. – № 1 (2010). – С. 24–29.
3. Goel A.L. Software Reliability Models: Assumptions, Limitations and Applicability // IEEE Trans. on Software Engineering / A.L. Goel // V. SE-11. – N12. – 1985. – P. 1411-1423.
4. Lin Huang. Lifetime Reliability for Load-Sharing Redundant Systems with Arbitrary Failure Distributions / Lin Huang, Qiang Xu // Reliability, IEEE Trans. on. – 2010. –Vol. 59, № 2. – P. 319–330.
5. Malaiya Y. K. Software Reliability Models: Theoretical Developments, Evaluation and Applications / Malaiya Y.K., Srimani P.K. – Los Angeles : IEEE Computer Society Press, 1990. – P. 191–221.
6. Musa J.D. Software Reliability Engineering: More Reliable Software Faster and Cheaper. / J.D. Musa – Second Ed., 2004. – 235 p.
7. Palviainen M. The reliability estimation, prediction and measuring of component-based software / M. Palviainen, A. Evesti, E. Ovaska // The Journal of Systems and Software. – 2011. – P. 120–131.
8. Walls L. and Alkali B. Advances in Mathematical Modeling for Reliability – IOS press, 2008. – 240 p.

Додаткові ресурси:

Інформаційні ресурси

1. Реєстр каркасів тестування для різних мов програмування. [Електронний ресурс]. – Режим доступу: http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks.
2. Електронна бібліотека з надійності фірми ReliaSoft (ReliaWiki). [Електронний ресурс]. – Режим доступу: <http://reliawiki.com/index.php/>.
3. Офіційний сайт проекту TestDriven.NET. [Електронний ресурс]. – Режим доступу: <http://www.testdriven.net/>