

# ЯКІСТЬ, ТЕСТУВАННЯ ТА ПІДТРИМКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## СИЛАБУС

Шифр і назва спеціальності	121 — Інженерія програмного забезпечення	Інститут / факультет	Комп'ютерних наук і програмної інженерії
Назва програми	«Інженерія програмного забезпечення».	Кафедра	програмної інженерії та інформаційних технологій управління
Тип програми	Освітньо-професійна	Мова навчання	Українська, англійська

## Викладач

**Орехов Сергій Валерійович**

Serhii.Oriekhov@khpі.edu.ua



к.т.н. доцент, доцент кафедри програмної інженерії та інформаційних технологій управління. Підготував і опублікував понад 30 публікацій, 5 статей у виданнях, індексованих у Scopus. (ідентифікатор ORCID-<https://orcid.org/0000-0002-5040-5861>).  
Провідний лектор з курсів: *Інформаційний пошук та семантичний веб (бакалаври) (англійською), Інтернет речей (бакалаври) (англійською)*

## Загальна інформація про курс

Анотація	Дисципліна «ЯКІСТЬ, ТЕСТУВАННЯ ТА ПІДТРИМКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ» є навчальною дисципліною з циклу професійної обов'язкової підготовки за спеціальністю 121 «Інженерія програмного забезпечення». Вона викладається у шостому семестрі в обсязі 90 год.(3 кредити ECTS), зокрема: лекції – 32 год., лабораторні – 32 год., самостійна робота – 84 год. У курсі передбачено два змістових модулі та одна модульна контрольна робота. Завершується дисципліна екзаменом.
Цілі курсу	Навчання студентів методології аналізу і оцінки якості програмного забезпечення, що дозволяє на етапі створення програмного забезпечення (ПЗ) вирішити такі основні завдання: обрати модель якості програмного забезпечення на рівні ІТ компанії, ІТ продукту або ІТ процесів; підготувати тест стратегію та тест план, що реалізуються в системі автоматизації тестування; обрати конкретний набір інструментів тестування та підтримки для реалізації стратегії або плану тестування.
Формат	Лекції, лабораторні роботи, самостійна робота. Підсумковий контроль — іспит.
Семестр	6

<b>Обсяг (кредити) / Тип курсу</b> (обов'язковий / вибірковий)	3 / Обов'язковий	<b>Лекції (години)</b>	32	<b>Практичні заняття (години)</b>	32	<b>Самостійна робота (години)</b>	26
---	------------------	------------------------	----	-----------------------------------	----	-----------------------------------	----

<b>Програмні компетентності</b>	<p>K05. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>K14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p> <p>K16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.</p> <p>K17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.</p> <p>K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.</p>
---------------------------------	---

<b>Результати навчання</b>	<b>Методи викладання та навчання</b>	<b>Форми оцінювання</b> (поточне оцінювання CAS, підсумкове оцінювання FAS)
<p>PR01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p>	<p>Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, метод зворотного зв'язку з боку студентів, проблемне навчання</p>	<p>Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), експрес-опитування(CAS), онлайн -тести (CAS), підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку навчального процесу (FAS)</p>
<p>PR03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p>	<p>Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, метод зворотного зв'язку з боку студентів, проблемне навчання</p>	<p>Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), онлайн -тести (CAS), підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку навчального процесу (FAS)</p>
<p>PR09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.</p>	<p>Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, дослідження, проектне навчання</p>	<p>Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), онлайн -тести (CAS), підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку навчального процесу (FAS)</p>
<p>PR14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p>	<p>Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, метод зворотного зв'язку з боку студентів, проблемне навчання</p>	<p>Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), збір даних за індивідуальними завданнями та звітування за результатами дослідження (CAS), онлайн -тести (CAS), підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку навчального процесу (FAS)</p>
<p>PR19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.</p>	<p>Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, дослідження, проектне навчання</p>	<p>Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), збір даних за індивідуальними завданнями та звітування за результатами дослідження (CAS), онлайн -тести (CAS), усна презентація результатів</p>

		проектів (CAS). підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку навчального процесу (FAS)
PR20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення	Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, дослідження, проектне навчання	Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), збір даних за індивідуальними завданнями та звітування за результатами дослідження (CAS), онлайн -тести (CAS), усна презентація результатів проектів (CAS). підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку навчального процесу (FAS)
PR23. Вміти документувати та презентувати результати розробки програмного забезпечення.	Інтерактивні лекції з презентаціями, дискусії, практичні заняття, командна робота, кейс-метод, дослідження, проектне навчання	Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), збір даних за індивідуальними завданнями та звітування за результатами дослідження (CAS), онлайн -тести (CAS), усна презентація результатів проектів (CAS). підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку навчального процесу (FAS)

### СИСТЕМА ОЦІНЮВАННЯ

Розподіл балів для оцінювання успішності аспіранта	Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	Нарахування балів	<b>100% підсумкове оцінювання</b> у вигляді іспиту (30%) та поточного оцінювання (70%). <b>30% іспит:</b> семестровий екзамен, відповідно до графіку навчального процесу <b>70% поточне оцінювання:</b> <ul style="list-style-type: none"> <li>40% оцінювання завдань на лабораторних роботах;</li> <li>30% проміжний контроль (1 модульна контрольна робота)</li> </ul>
	90-100	A	відмінно		
	82-89	B	добре		
	74-81	C			
	64-73	D	задовільно		
	60-63	E			
	35-59	FX	незадовільно з можливістю повторного складання		
	0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни		

<b>Політика курсу</b>	Студенти зобов'язані відвідувати заняття згідно розкладу та дотримуватися етики поведінки. У разі відсутності студентам необхідно буде виконати всі завдання, щоб компенсувати пропущені заняття. Участь у практичних заняттях вимагає попередньої підготовки та завчасного опрацювання всіх необхідних матеріалів для продуктивних дискусій під час заняття. Письмові завдання повинні бути подані до встановлених строків.
-----------------------	--

### Структура та зміст курсу

Тема	Опис теми	Лабораторна робота	Самостійна робота	Вивчення
Тема 1	Поняття якості та надійності програмного забезпечення	Лабораторна робота 1	Створення тестової програмної системи для подальшого експерименту з тестування на мові програмування Java, C#, Javascript, або PHP.	Вивчення моделей надійності програмного забезпечення та аналіз можливості їх впровадження.
Тема 2	Моделі якості програмного забезпечення на рівня IT компанії, процесу та IT продукту	Лабораторна робота 2	Підготовка документації для створення тест-плану у вигляді програмних вимог на мові UML та з використанням методології IDEF	Огляд існуючих моделей якості IT продукту

<b>Тема 3</b>	Характеристики якості програмного забезпечення	<b>Лабораторна робота 3</b>	Розрахунки значень типових метрик якості на основі розроблених артефактів програмного забезпечення		Аналіз метрик якості програмного забезпечення згідно моделі SWEBOOK
<b>Тема 4</b>	Місце процесів верифікації та тестування у життєвому циклі програмного забезпечення	<b>Лабораторна робота 4</b>	Використання метричної теорії програм для проведення тестових експериментів		Вивчення існуючих методів тестування „біла скриня”
<b>Тема 5</b>	Критерії тестування (структурні, функціональні, мутаційні, випадкові та інші)	<b>Лабораторна робота 5</b>	Застосування структурних критеріїв тестування		Вивчення існуючих методів тестування „чорна скриня”
<b>Тема 6</b>	Класифікація програмних помилок	<b>Лабораторна робота 6</b>	Особливості застосування критеріїв тестування в межах ООП парадигми		Аналіз особливостей та реалізація експерименту щодо тестування веб проєктів он-лайн
<b>Тема 7</b>	Тестові плани. Структура та призначення. Алгоритм побудови	<b>Лабораторна робота 7</b>	Використання компонентів для автоматизації тестування в середовищах програмування на мові Java, C#, Javascript, або PHP		Огляд систем автоматизації тестування
<b>Тема 8</b>	Системи підтримки процесів верифікації та тестування програмного забезпечення	<b>Лабораторна робота 8</b>	Створення тест-планів. Контроль усунення програмних помилок за допомогою спеціального програмного забезпечення.		Вивчення систем контролю усунення помилок он-лайн

### Література

<b>Обов'язкова</b>	<p>1. A. Mili, F. Tchier (2015). Software Testing Concepts and Operations John Wiley &amp; Sons, Inc..</p> <p>2. Y. Singh. SOFTWARE TESTING. Cambridge University Press. 2012.</p> <p>3. P. Ammann, J. Offutt. (2008) INTRODUCTION TO SOFTWARE TESTING. Cambridge University Press.</p> <p>4. D. Graham, E. Veenendaal, I. Evans, R. (2018) Black. FOUNDATIONS OF SOFTWARE TESTING. ISTQB CERTIFICATION. Thomson.</p> <p>5. M. Pezzè, M. Young. (2008) Software Testing and Analysis: Process, Principles, and Techniques.</p> <p>6. K. NAIK, P. (2008) TRIPATHY. SOFTWARE TESTING AND QUALITY ASSURANCE Theory and Practice. John Wiley &amp; Sons, Inc.</p> <p>7. M. Hutcheson (2003) Software Testing Fundamentals. Methods and Metrics. Wiley Publishing Inc..</p> <p>8 Standard for Software Verification and Validation Plans (ANSI / IEEE standard 1012-1986).</p>	<b>Додаткова</b>	<p>9. С. Куликов. (2021) Тестирование программного обеспечения. Базовый курс. EPAM Systems,</p> <p>10. Введення в програмну інженерію і управління життєвим циклом програмного забезпечення Guide to Software Engineering Base of Knowledge (SWEBOOK): Пер. з англ. С.Орлик [Електронний ресурс] Retrieved from: sorlik.blogspot.com/.</p> <p>11. Г. Майерс. (1980) Надежность программного обеспечения. Москва:</p> <p>12. Г. Майерс. (1982) Искусство тестирования программ. Москва: Финансы и статистика,</p> <p>13. К.Бек. (2003) Экстремальное программирование: разработка через тестирование. Санкт-Петербург : Питер,</p> <p>14. Б.Бейзер (2004) Тестирование черного ящика. Технологии функционального тестирования ПО и систем. Санкт-Петербург:</p> <p>15. McCabe T.J. A Complexity Measure // IEEE Transactions on Software Engineering. – V.2, № 4, 1976.</p> <p>16. С. Канер. (2001) Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес приложений. Киев: Диасофт,</p> <p>17. Л. Тамре (2003) Введение в тестирование программного обеспечения. Москва: Вильямс,</p>
--------------------	---	------------------	---

### Норми академічної етики

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність

Силабус за змістом повністю відповідає робочій програмі курсу.

