



## СИЛАБУС

<b>Базова інформація про дисципліну</b>	
<b>Назва дисципліни</b>	<b>СЕ020 / Тренінг-курс з тестування програмного забезпечення / Software Testing Training Course</b>
<b>Рівень вищої освіти / фахової передвищої освіти</b>	Фахова передвища
<b>Семестр</b>	1
<b>Факультет /відділення</b>	Інженерії програмного забезпечення
<b>Анотація курсу</b>	Предмет передбачає ознайомлення та закріплення знань стосовно базових підходів та інструментів тестування програмного забезпечення. Враховуючи широку аудиторію курсу, його тематика покриває питання управління та тестування вимог, дизайн тестів та створення тестової документації, практику обліку багів засобами JIRA та Confluence. Подальші теми розкривають практичні аспекти процесів тестування програмного забезпечення, в першу чергу, вебзастосунків, а також специфічного для різних доменів програмного забезпечення.
<b>Сторінка курсу в MOODLE</b>	<a href="http://78.137.2.119:2929/course/view.php?id=564">http://78.137.2.119:2929/course/view.php?id=564</a>
<b>Мова викладання</b>	Українська
<b>Лектор курсу</b>	Марченко С.В., спеціаліст I категорії канали комунікації: СДН «Moodle»: повідомлення в чаті E-mail: sv.marchenko1989@gmail.com
<b>Місце дисципліни в освітній програмі</b>	
<b>Перелік загальних компетентностей (ЗК)</b>	Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

	<p>Здатність застосовувати знання у практичних ситуаціях.</p> <p>Здатність працювати в команді.</p>
<b>Перелік спеціальних компетентностей (СК)</b>	<p>Здатність вільно користуватись сучасними комп'ютерними та інформаційними технологіями, прикладними та спеціалізованими комп'ютерно-інтегрованими середовищами для розробки, впровадження та обслуговування апаратних та програмних засобів комп'ютерної інженерії.</p> <p>Здатність брати участь у розробці системного та прикладного програмного забезпечення засобів комп'ютерної інженерії з використанням ефективних алгоритмів, сучасних методів і мов програмування.</p> <p>Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати прийняті рішення.</p>
<b>Перелік програмних результатів навчання</b>	<p>Тестувати, діагностувати та обслуговувати апаратні та програмні засоби комп'ютерної інженерії.</p> <p>Розробляти, тестувати, впроваджувати, експлуатувати програмне забезпечення для вбудованих і розподілених систем.</p> <p>Здійснювати пошук інформації з різних джерел для розв'язання задач комп'ютерної інженерії.</p> <p>Обґрунтовувати прийняті рішення, оцінювати, оформляти та представляти результати професійної діяльності згідно з діючою нормативною документацією.</p>
<b>Опис дисципліни</b>	
<b>Структура навантаження на студента</b>	<p>Загальна кількість годин – 180</p> <p>Кількість кредитів – 6</p> <p>Кількість лекційних годин – 30</p> <p>Кількість практичних занять – 30</p> <p>Кількість годин для самостійної роботи студентів – 120</p> <p>Форма підсумкового контролю – залік</p>

<b>Методи навчання</b>	За подачею навчального матеріалу: методи готових знань, дослідницький метод. З огляду на мету навчання: методи здобуття нових знань, метод формування умінь і навичок, метод застосування знань на практиці, методи закріплення знань, умінь і навичок, методи перевірки і оцінювання знань, умінь і навичок.
<b>Зміст дисципліни</b>	
Тема 1. Розробка та тестування вимог	Програмні помилки та їх наслідки. Тестування програмного забезпечення та його робочий процес. Знайомство з тестовою документацією та багтеркінговими системами. Специфікація вимог до програмного забезпечення. Матриця відповідності вимог. Тестування змін та імпакт-аналіз. Складання Test Plan.
Тема 2. Техніки дизайну тестів	Техніки тест дизайну: еквівалентне розділення (Equivalence Partitioning), граничні значення (Boundary Values), таблиця прийняття рішень (Decision Table), попарне тестування (pairwise testing), діаграма переходу станів (state-transition diagram), діаграма сценаріїв використання (use-case diagram), дослідницьке тестування, вгадування помилок. Test Case як основний артефакт тестувальника. Складання тестового набору. Use Case і Check List.
Тема 3. Поняття дефекту. Системи відслідковування помилок	Поняття багу. Складання Bug report. Серверіті та пріоритет дефекту. Життєвий цикл дефекту. Системи відслідковування помилок. Опис багів у багтрекінговій системі JIRA. Модульна контрольна робота 1.
Тема 4. Техніки модульного тестування	Анатомія модульних тестів. Стили модульного тестування. Покриття тестами. Тестові фреймворки для мови Python. Тестові дублери та мокінг залежностей. Ізоляційні фреймворки. Ідеї розробки через тестування.
Тема 5. Додаткові засоби функціонального тестування програмного забезпечення	Інтеграційні тести. Контрактні тести. Сервісні тести. Функціональні тести користувачького інтерфейсу. Інструменти ШІ та машинного

	навчання в автоматизованому функціональному тестуванні.
Тема 6. Нефункціональне тестування програмного забезпечення	Тестування безпеки. Тестування продуктивності. Тестування доступності. Модульна контрольна робота 2.

### Політика дисципліни

<b>Політика відвідування</b>	Регулярне відвідування всіх видів занять, своєчасність виконання самостійної роботи. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання зорганізується в он-лайн формі за погодженням із керівником курсу.
<b>Політика щодо дедлайнів та перескладання</b>	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.
<b>Академічна доброчесність</b>	У випадку недотримання політики академічної доброчесності (плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво) передбачено повторне проходження оцінювання.

### Система оцінювання

Поточний контроль здійснюється протягом семестру під час проведення практичних, семінарських та інших видів занять і оцінюється сумою набраних балів (максимальна сума – 100 балів; мінімальна сума, що дозволяє студенту отримати атестацію з предмету – 60 балів); підсумковий/ семестровий контроль, проводиться у формі заліку, відповідно до графіку навчального процесу.

Підсумкова оцінка за умови заліку виставляється як загальна сума балів, набраних за результатами поточного контролю.

### Накопичування рейтингових балів з навчальної дисципліни

Види навчальної роботи	Мах кількість балів												
<b>Аудиторна</b>													
Практичні завдання (6 тем)	60												
<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;"><b>T1</b></td> <td style="padding: 2px 10px;"><b>T2</b></td> <td style="padding: 2px 10px;"><b>T3</b></td> <td style="padding: 2px 10px;"><b>T4</b></td> <td style="padding: 2px 10px;"><b>T5</b></td> <td style="padding: 2px 10px;"><b>T6</b></td> </tr> <tr> <td style="text-align: center;">9</td> <td style="text-align: center;">15</td> <td style="text-align: center;">6</td> <td style="text-align: center;">9</td> <td style="text-align: center;">15</td> <td style="text-align: center;">6</td> </tr> </table>	<b>T1</b>	<b>T2</b>	<b>T3</b>	<b>T4</b>	<b>T5</b>	<b>T6</b>	9	15	6	9	15	6	
<b>T1</b>	<b>T2</b>	<b>T3</b>	<b>T4</b>	<b>T5</b>	<b>T6</b>								
9	15	6	9	15	6								
Тестування (2 модульні контрольні роботи по 5 балів)	10												
<b>Індивідуальна</b>													
Завдання до тем	30												

	<b>T1</b>	<b>T2</b>	<b>T3</b>	<b>T4</b>	<b>T5</b>	<b>T6</b>		
	10			20				
<b>РАЗОМ</b>							<b>100</b>	
<b>Шкала оцінювання</b>								
<b>ECTS</b>		<b>Бали</b>			<b>Зміст</b>			
<b>A</b>		90-100			Бездоганна підготовка в широкому контексті			
<b>B</b>		80-89			Повні знання, міцні вміння			
<b>C</b>		70-79			Хороші знання та вміння			
<b>D</b>		65-69			Задовільні знання, стереотипні вміння			
<b>E</b>		60-64			Виконання мінімальних вимог діяльності в стандартних умовах			
<b>FX</b>		35-59			Слабкі знання, відсутність умінь			
<b>F</b>		1-34			Необхідний повторний курс			

## Список рекомендованих джерел

### Основна література

1. 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. Witte F. Strategy, Planning and Organization of Test Processes: Basis for Successful Project Execution in Software Testing. Wiesbaden: Springer, 2022. 256с.
3. Atar A. Hands-On Test Management with Jira: End-to-end test management with Zephyr, synapseRT, and Jenkins in Jira. Birmingham: Packt Publishing, 2019. 264с.
4. Khorikov V. Unit Testing: Principles, Practices, and Patterns. NY: Manning Publications, 2020. 282p.
5. Mohan G. Full Stack Testing. A Practical Guide for Delivering High Quality Software. Sebastopol: O'Reilly Media, 2022. 382с.
6. Winteringham M. Testing Web APIs. NY: Manning Publications, 2022. 264с.
7. Okken B. Python Testing with pytest, Second Edition. Simple, Rapid, Effective, and Scalable. PragProg, 2022. 272p.

## Допоміжна література

1. Kulikov S. Software Testing. Base course. version 3.2.6. EPAM Systems, 2024. 278p.
2. Jain J. Learn API Testing: Norms, Practices, and Guidelines for Building Effective Test Automation. NY: Apress, 2022. 223с.
3. Kinsbruner E. A Frontend Web Developer's Guide to Testing: Explore leading web test automation frameworks and their future driven by low-code and AI. Birmingham: Packt Publishing, 2022. 286с.
4. Rodrigues A. G. та ін. Master Apache JMeter – From Load Testing to DevOps. Birmingham: Packt Publishing, 2019. 444с.
5. Beck K. Test-Driven Development: By Example. Boston: Addison-Wesley Professional, 2003. 220p.
6. Aniche M. Effective Software Testing: A developer's guide. NY: Manning Publications, 2022. 328с.

## Інформаційні ресурси

1. База знань [Електронний ресурс]. URL: <https://qalight.ua/baza-znaniy/zvidki-berutsya-pomilki-v-pz/>.
2. Попелюха Тестування ПЗ [Електронний ресурс]. URL: <https://www.youtube.com/@Popeliuha/playlists>.
3. Software Testing and Automation Specialization [Електронний ресурс]. URL: <https://www.coursera.org/specializations/software-testing-automation>.
4. Testing in .NET [Електронний ресурс]. URL: <https://learn.microsoft.com/en-us/dotnet/core/testing/>.