

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
Кафедра моделювання та програмного забезпечення

ЗАТВЕРДЖУЮ

Перший проректор

Владислав ЧУБАРОВ



« 02 »

06

2023 р.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Практика технологічна з тестування програмного забезпечення

(шифр і назва навчальної дисципліни)

спеціальність

121 – «Інженерія програмного забезпечення»

(шифр і назва напрямку підготовки)

факультет

Інформаційних технологій

(назва інституту, факультету, відділення)

Форма навчання	Курс	Семестр	Всього годин за планом	Кількість кредитів ECTS	Контрольний підсумок (семестр)	
					Залік (сем.)	Екзамен (сем.)
Денна	3	6	135	4,5	6	–
Денна скорочена	2	4	135	4,5	4	–
Заочна	3	6	135	4,5	6	–

Робочу програму навчальної дисципліни «Практика технологічна з тестування програмного забезпечення» для здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Інженерія програмного забезпечення» розроблено згідно з ОПП галузі знань 12 «Інформаційні технології» зі спеціальності 121 «Інженерія програмного забезпечення».

Розробник: старший викладач кафедри МПЗ Доценко І. О.

Робоча програма затверджена на засіданні кафедри моделювання та програмного забезпечення

Протокол від « 16 » лютого 2023 року № 4

Завідувач кафедри МПЗ, доцент, к.п.н.

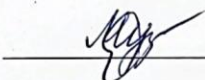


Андрій СТРЮК

Схвалено вченою радою факультету інформаційних технологій

Протокол від « 27 » лютого 2023 року № 6

Голова вченої ради

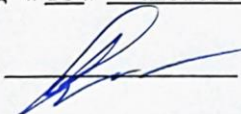


Іван МУЗИКА

Схвалено групою забезпечення ОПП

Протокол від « 16 » лютого 2023 року № 4

Гарант ОПП



Андрій СТРЮК

## ЗМІСТ

1	ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	4
2	МІСЦЕ ПРОХОДЖЕННЯ ПРАКТИКИ .....	6
3	ЗМІСТ НАВЧАЛЬНОЇ ПРАКТИКИ .....	6
4	ЗАДАЧІ НАВЧАЛЬНОЇ ПРАКТИКИ.....	6
5	КЕРІВНИЦТВО, КОНТРОЛЬ І ЗВІТНІСТЬ З ПРАКТИКИ.....	7
6	КРИТЕРІЇ ОЦІНЮВАННЯ .....	7
7	НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ.....	8
9	ІНФОРМАЦІЙНІ РЕСУРСИ .....	9
10	ТЕРМІНОЛОГІЧНИЙ СЛОВНИК.....	9
11	ЗМІНИ ТА ДОПОВНЕННЯ ДО РОБОЧОЇ ПРОГРАМИ .....	12

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

**Мета:** закріплення студентами теоретичних знань, отриманих при вивченні дисципліни «Якість програмного забезпечення та тестування» та набуття практичних навичок самостійного тестування за стандартами якості програмного забезпечення при створенні якісного програмного забезпечення для розв'язання різноманітних задач у практичній діяльності.

**Завдання:** формування у майбутніх фахівців теоретичних знань та практичних навичок володіння методами та засобами тестування програмного забезпечення, підходами проведення різноманітних видів тестування.

За результатами проходження практики здобувач повинен опанувати такі

### **компетентності:**

#### *Загальні компетентності*

ЗК01 Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02 Здатність застосовувати знання у практичних ситуаціях.

ЗК06 Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК07 Здатність працювати в команді.

#### *Фахові компетентності*

СК01 Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

СК04 Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

СК05 Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

СК10 Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

СК14 Здатність до алгоритмічного та логічного мислення.

**Програмні результати навчання** освітньої програми, яким відповідає дисципліна:

ПР01 Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПР04 Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПР09 Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПР14 Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПР16 Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.

ПР19 Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.

ПР20 Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.

У результаті вивчення навчальної дисципліни студент повинен

**знати:**

- життєвий цикл звітів, розстановку пріоритетів та їх обробку;
- загальне уявлення про інструментарій тестувальника (система управління дефектами, система управління проектами, інструменти автоматизації та інше);
- основи функціонального тестування (знати, що таке класи еквівалентності, граничні значення та інше);
- сучасні методології управління проектами.

**вміти:**

- чітко визначати мету тестування;
- створювати тестові прецеденти, а також їх виконання, спостереження та аналіз результатів;
- розробляти та працювати з необхідною тестовою документацією (тест-кейси, чек-листи, звіти про дефекти та інше) або іншої допоміжною документацією;
- аналізувати та планувати;
- проводити різні види тестування;
- тестувати веб-додатки;
- перевіряти повноту вимог, їх несуперечливість, дублювання та інші специфічні аспекти;
- встановлювати тестове оточення, налаштовувати інструментарії, здійснювати підготовку плану / стратегії тестування та автоматизації (попередньо оцінивши, чи потрібно її впровадження);
- проводити тестування нового функціоналу за розробленою документацією або методом вільного пошуку, дослідницького тестування, перевіряти різні аспекти якості.

Навчальна практика базується на опануванні дисципліни професійної підготовки:

ПП22 «Якість програмного забезпечення та тестування».

Структурно-логічне місце дисципліни в освітній програмі:

Цикл професійної підготовки ПП24.

## 2 МІСЦЕ ПРОХОДЖЕННЯ ПРАКТИКИ

Навчальна практика проводиться на базі ТОВ «Кьюатестлаб» згідно з Меморандумом про співпрацю на базі навчальної онлайн-платформи WEB100.Platform: QATestLab Univer.

Термін проходження практики – 3 тижні (135 годин/4,5 кредита).

## 3 ЗМІСТ НАВЧАЛЬНОЇ ПРАКТИКИ

**Тема 1. Техніки тест дизайну.** Визначення, еквівалентне розбиття, граничні значення, таблиці прийняття рішень, парне тестування (Pairwise Testing), передбачення помилки. Покриття тест наборів продукту.

**Тема 2. Організація процесу тестування.** Вимоги до програмного забезпечення. Тест план, стандарти, приклади. Звіт про тестування. Основні пункти звіту, графіки.

**Тема 3. Мобільне тестування** Визначення, платформи, пристрої. Мобільне тестування веб-проектів в різних браузерах, їх установка з магазинів, доповнення чек-листа.

**Тема 4. Інструменти тестування IOS, Android, Windows phone додатків** Процес установки, скріншоти, запис відео, логирование, використовувані програми (iTunes, SDK manager).

**Тема 5. Тестування ігор (game testing).** Ігрові механіки. Тестування локалізації.

## 4 ЗАДАЧІ НАВЧАЛЬНОЇ ПРАКТИКИ

- Ознайомлення зі стандартами написання тест-планів, отримання базових теоретичних та практичних навичок в написанні тест-планів.
- Ознайомлення з основними відмінностями мобільних веб-сайтів та додатків від їх аналогів на ПК, отримання базових теоретичних та практичних навичок в замірюванні швидкості роботи мобільних додатків.
- Отримання базових теоретичних та практичних навичок в написанні тестових випадків (test case) для веб-проектів, проведення тестування сайту на основі створених тестових випадків (test case).
- Отримання базових теоретичних та практичних навичок у тестуванні мобільних додатків з використанням інструментів тестування, написанні тестових випадків для тестування мобільних додатків та проведення тестування сайту на основі створених тестових випадків (test case).
- Ознайомлення з видами тестування ігор, отримання базових теоретичних та практичних навичок в розпізнаванні дефектів в іграх за видами.

## 5 КЕРІВНИЦТВО, КОНТРОЛЬ І ЗВІТНІСТЬ З ПРАКТИКИ

Керівництво і контроль за проходженням практики покладається на викладача кафедри моделювання та програмного забезпечення, який надає їм необхідну допомогу.

В період практики студенти вивчають теоретичний матеріал за програмою практики та виконують лабораторні роботи у системі WEB100 Platform: QATestLab Univer.

Впродовж практики студенти зобов'язані вести щоденник, у який заноситься графік виконання програми практики, а також робочі записи.

Для отримання диференційного заліку студент по закінченню практики повинен здати керівнику практики щоденник та звіт виконаних робіт.

При умові виконання усіх лабораторних робіт у межах програми практики та з дисципліни «Якість програмного забезпечення та тестування», а також успішного проходження тесту у системі WEB100 Platform: QATestLab Univer, при отриманні необхідної сумарної кількості балів, студент має право отримати сертифікат проходження курсу «Основи тестування програмного забезпечення».

Звіт про практику повинен містити відомості про виконану студентом роботу, а саме: тема, мета, завдання, скріншоти, а також висновки кожної виконаної роботи.

Щоденник та звіт про проходження практики надається у роздрукованому вигляді в останній день практики.

## 6 КРИТЕРІЇ ОЦІНЮВАННЯ

Оцінка звіту по практиці виставляється диференційована оцінка «відмінно», «добре», «задовільно» і «незадовільно».

Національна шкала успішності	Оцінка ECTS	Визначення ECTS	100-бальна система оцінювання
відмінно/ зараховано	<b>A</b>	ВІДМІННО - відмінне виконання лише з незначними помилками	90...100
добре/ зараховано	<b>B</b>	ДУЖЕ ДОБРЕ - вище середнього рівня з кількома помилками	80...89
	<b>C</b>	ДОБРЕ - у цілому правильно робота з певною кількістю помилок і недоліків	71...79
задовільно/ зараховано	<b>D</b>	ЗАДОВІЛЬНО - непогано, але зі значною кількістю грубих помилок	61...70
	<b>E</b>	ДОСТАТНЬО - виконання задовольняє мінімальні потреби	50...60
незадовільно/ не зараховано	<b>FX</b>	НЕЗАДОВІЛЬНО - із можливістю повторного складання	30...49
	<b>F</b>	НЕЗАДОВІЛЬНО - з обов'язковим повторним вивчення дисципліни	0...29

## 7 НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ

### 7.1 Навчальна та довідкова література

1. Положення про проведення практики студентів вищих навчальних закладів України, затверджене наказом МОУ від 8 квітня 1993р. № 93.
2. Положення про організацію освітнього процесу у Криворізькому національному університеті, затверджене наказом ректора університету від 06 лютого 2020 р., № 50 <http://www.knu.edu.ua/storage/files/2/3/117.pdf>
3. Aniche, Maurizio. Effective Software Testing: A Developer's Guide. Сполучені Штати Америки: Manning, 2022. – 348 с. <https://books.google.com.ua/books?id=U4BIEAAAQBAJ>
4. Dosaj, Chhavi Raj. The Self-Taught Software Tester A Step By Step Guide to Learn Software Testing Using Real-Life Project. Сполучені Штати Америки: Amazon Digital Services LLC - Kdp, 2020. – 217 с. <https://books.google.com.ua/books?id=Qt2QzQEACAAJ>
5. Авраменко А. С., Авраменко В. С., Косенюк Г. В. Тестування програмного забезпечення 2016 : навч. посіб. в ел. вигляді. / А. С. Авраменко, В. С. Авраменко, Г. В. Косенюк // Черкаси: ЧНУ імені Богдана Хмельницького, 2017. – 284 с. <http://eprints.cdu.edu.ua/1482/1/testyvan.pdf>
6. Software Testing Help. «Web Application Testing Complete Guide (How to Test a Website)» [Electronic resource] – Access mode: <https://www.softwaretestinghelp.com/web-application-testing/> .
7. Software Testing Help. «Website Cookie Testing & Test Cases for Testing Web Application Cookies» [Electronic resource] – Access mode: <https://www.softwaretestinghelp.com/website-cookie-testing-test-cases/>
8. Test Plan (a Real Sample) [Електронний ресурс] – Режим доступу: [https://www.softwaretestinghelp.com/wp-content/qa/uploads/2014/02/Live\\_Project\\_Test\\_Plan\\_SoftwareTestingHelp.pdf](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2014/02/Live_Project_Test_Plan_SoftwareTestingHelp.pdf)
9. Test Plan Outline (IEEE 829 Format) [Electronic resource] – Access mode: <https://www.fit.vutbr.cz/study/courses/ITS/public/ieee829.html>

### 7.2 Методична література

1. Лабораторні роботи з навчального курсу «Software testing for universities» Провідна українська компанія з тестування програмного забезпечення QATestLab Ukrainian HI-Tech Initiative. – Київ. 2023. – 125 с.



## 9 ІНФОРМАЦІЙНІ РЕСУРСИ

До складу інформаційних ресурсів навчальної дисципліни входять:

1. Бібліотека Криворізького національного університету. – Режим доступу до електронних ресурсів: <http://lib.knu.edu.ua/>
2. Сайт кафедри моделювання та програмного забезпечення КНУ <http://mpz.knu.edu.ua/novini/>
3. Персональні комп'ютери з будь-якою операційною системою.
4. Підключення до мережі Інтернет.
5. Web-ресурси на платформах:
  - доступ до системи відслідковування дефектів Mantis Bug Tracker <http://mantis.qatestlab.net;>
  - доступ до системи для управління тест-кейсами TestLink <http://tl-univer.qa-testlab.net/login.php;>
  - доступ до системи взаємодії викладач-студент: WEB100 Platform: QATestLab Univer;
  - тестові сайти, які дозволяють отримати практичний досвід у знаходженні дефектів.
6. Дистанційний курс «Якість програмного забезпечення» на платформі Google G Suite <https://classroom.google.com/c/NTQ1MDUzODYwODQx>

## 10 ТЕРМІНОЛОГІЧНИЙ СЛОВНИК

**Якість програмного забезпечення (Software quality)** – The degree to which the software meets the requirements, needs and expectations of users.

**Тестування програмного забезпечення (Software Testing)** – The process of testing and evaluating software to identify errors, defects, and deficiencies.

**Функціональне тестування (Functional testing)** – A type of testing aimed at verifying the functionality of the software in accordance with the requirements and expectations of users.

**Навантажувальне тестування (Load Testing)** – Testing that evaluates the software's response to a specific load or volume of data.

**Тестовий сценарій (Test Script)** – A detailed script or procedure that specifies the sequence of steps to perform a specific test.

**Юніт-тестування (Unit testing)** – A type of testing that checks the correct operation of individual modules or software components.

**Інтеграційне тестування (Integration Testing)** – Testing that tests the interaction between different modules or components of software.

**Системне тестування (System testing)** – Testing that checks the entire system functionality of the software and its compliance with requirements.

**Регресійне тестування (Regression Testing)** – Retesting of previously passed tests to detect new defects after software changes.

**Тест-кейс (Test Case)** – A detailed script or procedure that specifies the sequence of testing steps and the expected result.

**Автоматизоване тестування (Automated testing)** – Using special tools or programs to automate the execution of tests.

**План тестування (Test Plan)** – A document containing a description of the test approach, including scope of work, resources, schedule, and test strategy.

**Трасування дефектів (Defect Tracking)** – The process of identifying, documenting, and tracking defects or errors in software.

**Валідація програмного забезпечення (Software Validation)** – The process of verifying that the software meets the requirements and needs of the users.

**Санітарний тест (Sanitary test)** – Testing that checks the correctness of the program's interaction with various operating systems and environments.

**Тестова документація (Test Documentation)** – Documents that describe the test plan, test scenarios, test results, and other related information.

**Методології тестування (Testing Methodologies)** – Developed procedures and approaches to software testing such as Agile, Waterfall, V-model, etc.

**Статичний аналіз програмного забезпечення (Static Software Analysis)** – The process of reviewing and analyzing software code without actually executing it to identify errors and flaws.

**Тестове середовище (Test environment)** – The combination of hardware and software on which software testing is performed.

**Баг (Bug)** – Is a slang word that usually denotes an error in a program or system that produces an unexpected or incorrect result. Most bugs are caused by mistakes made by the developers of the program in its source code or in its design.

**Баг-трекінгова система (Bug tracking system)** – A bug tracking system designed to help software developers (programmers, testers, etc.) take into account and control errors (bugs) found in programs, user wishes, and also monitor the process of eliminating these mistakes and fulfillment or non-fulfillment of wishes.

**Банер (Banner)** – An electronic graphic image of an advertising nature with the possibility of switching to the page of the advertised material.

**Білд (Build)** – In the software industry, the term build refers to the process of converting the source code of a program that can be run on a specific device(s) or converted into executable code. One of the steps of creating a build is the process of compiling the source code, where the files are structured and converted into a form available for testing the application.

**Версія (Version)** – The sub-name of the software product used to move through the life cycle of the application, i.e. when the software product is corrected, its version changes so as not to change its (software product) name every time.

**Глосарій (Glossary)** (Latin glossarium – "a collection of glosses") – A dictionary of highly specialized terms in any field of knowledge with interpretation, sometimes translation into another language, comments and examples.

**Девайс (Device)** – An artificial object with an internal structure, created to perform certain functions, usually in the field of technology (technical device).

**Дефект (Defect)** – An error suggesting a specific discrepancy between the actual software product and the intended, approved version. Synonym of bug.

**Додаток (Application)** – A program designed to perform certain user tasks and designed for direct interaction with the user.

**Креш (Crash)** – This is a sudden, unexpected termination of the game/application caused by an error(s) in the software code that occurs randomly or as a result of certain user actions. Often characterized by data loss or corruption. Synonyms: emergency failure, failure.

**Крешлог (CrashLog)** – Crashlog is a file that contains information about a failure in the operation of the program or system, which led to the crash of the application or system, respectively.

**QA (Quality Assurance)** – Control and assessment of any aspect of the project, equipment or type of service in order to increase (maximize) the probability of ensuring the established (minimum) quality standards.

**Мануал (Manual)** – (English user guide or user manual), operating manual, user manual – a document whose purpose is to provide people with assistance in using a certain system. The document is part of the technical documentation of the system.

**Мокан (Mockup)** – Is a template (layout) created (assembled) by an artist (designer), by analogy with which graphic elements (perhaps carrying a functional load) must be correctly inserted (located) in a specific game area.

**Пріоритет (Priority)** – This is an attribute that indicates the priority of task execution or defect elimination. The higher the priority, the faster the defect needs to be fixed.

**Проект (Project)** (from Latin projectus – Thrown forward, coming forward) A process consisting of a set of coordinated and managed activities with start and end dates, made to achieve a goal that meets specific requirements, including time limits, cost and resources. A temporary enterprise designed to create unique products, services or results. Completion occurs when project goals are met; or it is recognized that the objectives of the project will not or cannot be achieved; or the need for the project has disappeared.

**Реліз (Release)** – Downloading the finished game to the store for further sale.

**Тестувальник (Tester)** – A specialist who conducts software testing in order to ensure and control the quality of a software product or information system.

**Фікс (Fix)** – The process of correcting errors by the developer.

**Фіча (Software feature)** – Feature – (from the English feature – feature) function, peculiarity, property. An atypical result of the program's action, which may resemble a failure or developer's shortcomings, hence the well-known expression "it's not a bug – it's a feature."

