

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра моделювання та програмного забезпечення



ЗАТВЕРДЖУЮ

Перший проректор

Владислав ЧУБАРОВ

« 02 » 06 2023 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Якість програмного забезпечення та тестування

(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	Всього аудиторних годин	Аудиторних годин, (у тому числі КЗ)		Самостійна робота (год.)	Контрольні (модульні роботи)	Контрольний підсумок (семестр)	
						Лекції	Лабораторні			Залік (сем.)	Екзамен (сем.)
Денна	3	5	90	3	32	16	16	58	1	5	–
Денна скорочена	2	3	90	3	32	16	16	58	1	3	–
Заочна	3	5	90	3	16	8	8	74	1	5	–
Заочна скорочена	2	3	90	3	16	8	8	74	1	3	–

Робочу програму навчальної дисципліни «Якість програмного забезпечення та тестування» 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

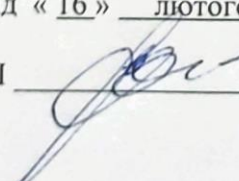


Іван МУЗИКА

Схвалено групою забезпечення ОПП

Протокол від « 16 » лютого 2023 року № 4

Гарант ОПП



Андрій СТРЮК

ЗМІСТ

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	4
2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	5
3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	7
4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	8
5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ.....	9
6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ.....	9
7. САМОСТІЙНА РОБОТА.....	10
8. МЕТОДИ НАВЧАННЯ.....	11
9. МЕТОДИ КОНТРОЛЮ.....	12
10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ.....	12
11. ШКАЛА ОЦІНЮВАННЯ.....	15
12. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ.....	16
13. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ.....	17
14. ІНФОРМАЦІЙНІ РЕСУРСИ.....	18
16. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК.....	19
17. ЗМІНИ ТА ДОПОВНЕННЯ ДО РОБОЧОЇ ПРОГРАМИ.....	21
ДОДАТОК 1 Робочий план.....	22

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, рівень вищої освіти	Характеристика навчальної дисципліни			
		денна форма навчання	заочна форма навчання	денна скорочена форма навчання	заочна скорочена форма навчання
Кількість кредитів – 3	Галузь знань 12 <u>Інформаційні технології</u> (шифр і назва)	Нормативна			
Модулів – 1	Спеціальність: 121 <u>Інженерія програмного забезпечення</u> (код та найменування спеціальності)	Рік підготовки:			
Змістових модулів – 1		3-й	3-й	2-й	2-й
Загальна кількість годин – 90		Семестр			
		5-й	5-й	3-й	3-й
Тижневих годин для денної форми навчання: аудиторних – 2 (32/16) самостійної роботи здобувача – 3,63 (58/16)	Ступінь вищої освіти: <u>бакалавр</u>	16 год.	8 год.	16 год.	8 год.
		Практичні, семінарські			
		-	-	-	-
		Лабораторні			
		16 год.	8 год.	16 год.	8 год.
		Самостійна робота			
		58 год.	74 год.	58 год.	74 год.
		Вид контролю: залік			

Примітка:

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – $32/58=0,55$

для денної скороченої форми навчання – $32/58=0,55$

для заочної форми навчання – $16/74= 0,22$

для заочної скороченої форми навчання – $16/74= 0,22$

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. Мета викладання навчальної дисципліни: Дисципліна спрямована на формування у майбутніх програмістів сучасного рівня інформаційної та програмістської культури з основ теорії якості програмного забезпечення та тестування, оволодіння базовими методами побудови тестів та автоматизованими засобами тестування, процесами управління якістю програмного забезпечення, термінологією та основами верифікації та атестації програмного забезпечення, набуття практичних навичок самостійного тестування за стандартами якості програмного забезпечення і створення якісного програмного забезпечення для розв'язання різноманітних задач у практичній діяльності.

2.2. Завдання вивчення дисципліни:

- Ознайомлення з основними поняттями якості програмного забезпечення.
- Вивчення основних методів та технік тестування. Вивчення принципів планування тестування, створення тестових сценаріїв, виконання тестів та аналізу результатів.
- Вивчення інструментів та технологій тестування, набуття вміння використовувати ці інструменти для ефективного проведення тестування.
- Розуміння ролі тестування у життєвому циклі програмного забезпечення.
- Застосування тестування для поліпшення якості програмного забезпечення, набуття вміння створення ефективних тестових сценаріїв, проведення тестування та аналіз результату для покращення якості програмного забезпечення.
- Розуміння методологій та процесів забезпечення якості.

2.3. Відповідно до освітньої програми дисципліна забезпечує наступні **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.

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.

ПР03 Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

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 здобувач 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);
- дослідне тестування (Vach);
- 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;
- проводити різні види тестування;
- тестувати веб-додатки;
- перевіряти повноту вимог, їх несуперечливість, дублювання та інші специфічні аспекти;
- встановлювати тестове оточення, налаштовувати інструментарій, 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.

2.1. Міждисциплінарні зв'язки

Дисципліна «Якість програмного забезпечення та тестування» базується на опануванні таких дисциплін:

ЗПЗ – Іноземна мова

ВК1.1 – Англійська мова (за професійним спрямуванням);

ПП4 – Архітектура та проектування програмного забезпечення;

ПП8 – Людино-машинна взаємодія;

ПП10 – Об'єктно-орієнтоване програмування;

ПП14 – Основи web-програмування;

ПП17 – Офісне програмне забезпечення.

Знання, отримані здобувачами при вивченні дисципліни, використовуються при проходженні технологічної практики з тестування програмного забезпечення, при виконанні курсових робіт, а також кваліфікаційної роботи.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1. Введення в тестування – 90 год.

Тема 1. Введення в тестування. Визначення якості програмного забезпечення (ПЗ). Характеристики якості ПЗ. Гарантія якості – QA (Quality Assurance) та контроль якості – QC (Quality Control). Верифікація і валідація. Процес тестування. Помилка, баг, збій. Визначення помилок у ПЗ. Основні терміни, які застосовуються для позначення помилок у ПЗ. Основні джерела помилок у ПЗ. Наслідки, до яких можуть призводити помилки у ПЗ. Атрибути та життєвий цикл бага. Багтреккінгові системи (Mantis, Jira, Redmine), глосарій.

Тема 2. Види тестування. Основні види тестування ПЗ. Модульне тестування ПЗ. Інтеграційне тестування ПЗ. Системне та регресійне тестування ПЗ. Навантажувальне та стресове тестування ПЗ. Приймальне тестування ПЗ. Принципи взаємодії між розробниками ПЗ і його тестувальниками. Схема тестування ПЗ.

Тема 3. Веб-тестування. Поняття веб-тестування. Анатомія веб-сторінки, верстка, дизайн. Робота з Firebug.

Тема 4. Тестування usability. Тестування практичності / перевірка на простоту використання. Чек лист та чит лист. Визначення usability testing, формування простого чек-листа. UI-дизайн, UX-тестування та аудит.

Тема 5. Кросбраузерне тестування. Види браузерів, їх версії, використання ІЕ різних версій.

Тема 6. Функціональне тестування. Регресійне тестування. Визначення, формування простого чек-листа, авторизація, реєстрація, пошук та інші основні функції.

Визначення регресійного тестування. Димне (Smoke testing), санітарне (sanity testing), інтуїтивне (ad hoc testing).

Тема 7. Тест дизайн (test design). Тест кейс (test case) та тестові набори. Атрибути тест кейсу. Визначення, життєвий цикл тестових наборів (test case), TestRail, Testlink.

Тема 8. Техніки тест дизайну. Організація процесу тестування. Визначення, еквівалентне розбиття, граничні значення, таблиці прийняття рішень, парне тестування (Pairwise Testing), передбачення помилки. Покриття тест наборів продукту.

Вимоги до програмного забезпечення. Тест план, стандарти, приклади. Звіт про тестування. Основні пункти звіту, графіки.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	денна / денна скорочена форма				заочна / заочна скорочена форма			
	усього	у тому числі			усього	у тому числі		
		л	лаб.	с.р.		л	лаб.	с.р.
1	2	3	4	5	6	7	8	9
Змістовий модуль. Якість програмного забезпечення та тестування								
Тема 1. Введення в тестування	10	2	2	6	10	1	1	8
Тема 2. Види тестування	10	2	2	6	10	1	-	9
Тема 3. Веб-тестування	10	2	2	6	10	1	1	8
Тема 4. Тестування зручності використання	12	2	2	8	12	1	-	11
Тема 5. Кросбраузерне тестування.	10	2	2	6	10	-	-	10
Тема 6. Функціональне тестування. Регресійне тестування	12	2	2	8	12	1	2	9
Тема 7. Тест дизайн (test design) та тестові набори (test case)	13	2	2	9	13	1	2	10
Тема 8. Техніки тест дизайну. Організація процесу тестування	13	2	2	9	13	2	2	9
<i>Разом за змістовим модулем</i>	90	16	16	58	90	8	8	74
Усього годин	90	16	16	58	90	8	8	74

5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Не передбачено.

6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин			
		денна	заочна	денна скорочена	заочна скорочена
1.	<i>Лабораторна робота № 1.</i> Введення в тестування. Визначення, атрибути бага; життєвий цикл бага; багтреккінгова система Mantis	2	1	2	1
2.	<i>Лабораторна робота № 2.</i> Поняття веб тестування. Анатомія веб-сторінки, верстка, дизайн	2	–	2	–
3.	<i>Лабораторна робота № 3.</i> Тестування веб-проектів. Робота з Firebug.	2	1	2	1
4.	<i>Лабораторна робота № 4.</i> Визначення usability testing, формування простого чек-листа	2	–	2	–
5.	<i>Лабораторна робота № 5.</i> Кросбраузерне тестування веб-сайту	2	–	2	–
6.	<i>Лабораторна робота № 6.</i> Функціональне тестування. Формування простого чек-листа: авторизації, реєстрації користувача, пошуку та інші основні функції	2	2	2	2
7.	<i>Лабораторна робота № 7.</i> Тест-дизайн та тест-кейси. Testlink (реєстрація, меню, створення та редагування тест-кейсу	2	2	2	2
8.	<i>Лабораторна робота № 8.</i> Техніки тест-дизайну. Еквівалентне розбиття, граничні значення, покриття тест кейсами продукту	2	2	2	2
РАЗОМ:		16	8	16	8

7. САМОСТІЙНА РОБОТА

Метою самостійної роботи здобувача є: навчитися користуватися бібліотечними фондами і каталогами, працювати з літературними джерелами, складати конспекти, аналізувати матеріал, порівнювати різні наукові концепції та робити висновки.

На самостійну роботу здобувачам денної форми навчання відведено 58 годин, заочної форми навчання – 74 годин.

Самостійна робота здобувачів при вивченні дисципліни «Якість програмного забезпечення та тестування» включає такі складові (з вказуванням нормативів):

- опрацювання лекційного матеріалу;
- підготовка до виконання, а також до захисту лабораторних робіт;
- опрацювання тем курсу, які виносяться на самостійне вивчення, за списками літератури, рекомендованими в робочій навчальній програмі дисципліни;
- підготовка до виконання контрольної-модульної роботи (тестування);
- підготовка до складання заліку за контрольними питаннями.

Розподіл годин самостійної роботи

№ з/п	Назва теми	Кількість годин			
		Денна ф. н.	Заочна ф. н.	Денна скорочена ф. н.	Заочна скорочена ф. н.
1.	Тема 1. Введення в тестування	6	8	6	8
2.	Тема 2. Види тестування	6	9	6	9
3.	Тема 3. Веб-тестування	6	8	6	8
4.	Тема 4. Тестування зручності використання	7	9	7	9
5.	Тема 5. Кросбраузерне тестування.	6	9	6	9
6.	Тема 6. Функціональне тестування. Регресійне тестування	7	9	7	9
7.	Тема 7. Тест дизайн (test design) та тестові набори (test case)	8	9	8	9
8.	Тема 8. Техніки тест дизайну. Організація процесу тестування	8	9	8	9
9.	Підготовка до контрольних заходів	4	4	4	4
	РАЗОМ	58	74	58	74

8. МЕТОДИ НАВЧАННЯ

Під час вивчення дисципліни використовуються наступні методи навчання: лекції, лабораторні заняття, самостійна робота.

Лекція – це форма навчання, під час якої викладач передає знання та інформацію здобувачам шляхом усного викладу. Під час лекції викладач розповідає про конкретну тему, пояснює основні концепції, наводить приклади та надає інформацію, яка є ключовою для розуміння предмету або дисципліни. Під час лекції використовуються різні додаткові матеріали, такі як презентації, діаграми, графіки, демонстрації програмного забезпечення, щоб проілюструвати свої пояснення та полегшити розуміння здобувачам. Лекції допомагають отримати загальне розуміння предмету, ознайомитися з ключовими термінами та принципами, а також створюють основу для подальшого детального вивчення цієї дисципліни під час самостійної роботи здобувачів.

Лабораторне заняття – це форма навчання, під час якої здобувачі активно взаємодіють з матеріалом, проводять практичні експерименти, виконують завдання та вирішують конкретні проблеми у комп'ютерних класах.

Під час лабораторних занять здобувачам надаються практичні завдання, які вимагають застосування теоретичних знань, розуміння концепцій та вміння працювати з відповідним програмним забезпеченням. Вони аналізують дані, розв'язують задачі та виконують інші завдання, що спрямовані на закріплення та практичне застосування набутих знань.

Лабораторні заняття сприяють практичному застосуванню теоретичних знань, розвитку навичок проблемного вирішення, аналітичного мислення, спільної роботи та критичного мислення. Вони допомагають здобувачам отримати практичний досвід у виконанні завдань, розвивають навички самостійності, творчого мислення та роботи з реальними проблемами своєї галузі.

Самостійна робота здобувача – це процес, під час якого здобувач незалежно вивчає, вдосконалює та закріплює набуті знання, виконує завдання та розв'язує проблеми без прямого нагляду або керівництва викладача. Це важлива складова навчального процесу, яка сприяє розвитку самостійності, відповідальності, критичного мислення та самоорганізації здобувача.

Під час самостійної роботи здобувач може здійснювати такі дії:

- Вивчення матеріалу: Здобувач самостійно читає підручники, навчальні матеріали, досліджує наукові статті та інші джерела, що стосуються предмету вивчення.
- Розв'язання завдань: Здобувач самостійно виконує завдання, надані викладачем. Це можуть бути практичні завдання, вправи, тестові завдання тощо.
- Підготовка до занять: Здобувач готується до лекцій та лабораторних занять, попередньо читаючи та осмислюючи матеріал, який буде розглядатися на занятті.
- Використання додаткових ресурсів: Здобувач шукає та використовує додаткові навчальні ресурси, такі як відеоуроки, онлайн-курси, електронні підручники, програмні засоби або веб-ресурси, щоб поглибити свої знання та розуміння предмету.
- Самооцінка та самоконтроль: Здобувач оцінює свої знання, самостійно перевіряє свої відповіді на завдання, аналізує помилки та шукає шляхи поліпшення своїх навичок.

У разі виникнення необхідності забезпечення навчального процесу в дистанційному режимі супровід та контроль знань реалізовується за допомогою дистанційного курсу, розробленого в Google Classroom. Онлайн лекції, консультації та усні відповіді на питання, захист проектів проводиться за допомогою Google Meet або Zoom.

Дистанційне навчання – Використання онлайн-ресурсів, відеоуроків, вебінарів та інтерактивних платформ для навчання з використанням офісного програмного забезпечення.

Цей метод дозволяє здобувачам вивчати матеріал у зручній для них час та темп, а також спілкуватися з викладачем та іншими здобувачами в онлайн-середовищі.

Індивідуальні консультації – Здобувачі мають можливість зустрітися з викладачем один на один для отримання додаткових пояснень, консультацій та вирішення індивідуальних питань щодо вивчення дисципліни.

Перед проведенням лабораторних робіт викладачами проводяться інструктажі: вступні, поточні, підсумкові.

9. МЕТОДИ КОНТРОЛЮ

Основними завданнями контролю знань здобувачів вищої освіти з дисципліни є оцінювання засвоєння теоретичних знань і практичних навичок, отриманих під час навчання.

Контрольні заходи мають виконувати наступні функції:

- стимулювати систематичну самостійну роботу над навчальним матеріалом;
- забезпечувати закріплення та реалізацію набутих теоретичних знань при підготовці до лабораторних занять;
- прищеплювати навички відповідального ставлення до своїх обов'язків, самостійного цілеспрямованого пошуку потрібної інформації, чіткої організації свого робочого дня.

Оцінювання знань здобувачів складається з поточного та підсумкового контролю.

Поточний контроль знань здобувачів вищої освіти передбачає оцінювання за наступними основними напрямками:

- перевірка теоретичних знань;
- перевірка виконання лабораторних робіт та контрольної-модульної роботи.

З даних компонентів складаються загальні бали, які фіксуються в журналі викладача.

Оцінювання рівня засвоєння теоретичних знань здобувачів вищої освіти проводиться за результатами захисту лабораторних робіт, а також виконання контрольної-модульної роботи. Підсумковим контролем є залік.

10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ

Для оцінювання успішності здобувачів *денної форми* навчання використовується модульно-рейтингова система, яка передбачає розподіл балів за виконання всіх запланованих видів робіт. При цьому максимальна кількість балів, яку може отримати здобувач, дорівнює 100. Ця сума складається з балів отриманих за контрольну-модульну роботу (КМР) та балів, що їх накопичив здобувач за лабораторні роботи у ході поточного контролю.

КМР у модулі відображає теоретичні знання і відповідає 20-ти відсоткам його ваги, тобто вона може дати максимально 20 балів при найвищій якості виконання. При зниженні якості КМР знижується і сума балів відповідно до шкали, що наводиться у таблиці:

Оцінювання успішності здобувачів *заочної форми* навчання оцінюється аналогічно.

Шкала оцінювання контрольної-модульної роботи

Відсоток вірних компонентів КМР	0 – 30	31 – 60	61 – 75	76 – 85	86 – 94	95 – 100
Сума балів за КМР	10	12	14	16	18	20

Оцінювання кожної лабораторної роботи та індивідуального завдання ведеться за показниками, наведеними в таблиці.

Шкала оцінювання лабораторних робіт

№ зан.	Вид роботи	Тема	Максимальна кількість балів			
			денна	заочна	денна скорочена	скорочена заочна
1	Лабораторна робота № 1	Введення в тестування. Атрибути бага, багтреккінгова системи Mantis. Створення баг-репортів	10	5	10	5
2	Лабораторна робота № 2	Веб- тестування. Тестування верстки веб-сайту	10	5	10	5
3	Лабораторна робота № 3	Тестування веб-проектів. Робота з Firebug	10	5	10	5
4	Лабораторна робота № 4	Тестування практичності / перевірка на простоту використання (usability testing) та чек-лист	10	5	10	5
5	Лабораторна робота № 5	Кросбраузерне тестування веб-сайту	10	5	10	5
6	Лабораторна робота № 6	Функціональне тестування. Формування простого чек-листа: авторизації, реєстрації користувача, пошуку та інші основні функції	10	5	10	5
7	Лабораторна робота № 7	Тест-дизайн та тест-кейси. Testlink (реєстрація, меню, створення та редагування тест-кейсу)	10	5	10	5
8	Лабораторна робота № 8	Техніки тест-дизайну. Еквівалентне розбиття, граничні значення, покриття тест кейсами продукту	10	5	10	5
Контрольно-модульна робота			20	20	20	20
Разом по модулю:			100	60	100	60
Разом за семестр:			100	60	100	60

Оцінювання кожної лабораторної роботи ведеться за наступними показниками:

1. Своєчасність практичного виконання лабораторної роботи (у тиждень згідно із графіком робіт). За несвоєчасність оцінка знижується на один бал.
2. Своєчасність захисту виконаної лабораторної роботи (у тиждень наступний за тижнем планового виконання роботи). За несвоєчасність оцінка знижується на один бал.
3. Підготовленість здобувача до захисту лабораторної роботи (його спроможність грамотно прокоментувати свою роботу та відповісти на супутні питання).

Якість роботи оцінюється наступним чином:

- робота виконана без зауважень – максимальний бал;
- робота виконана достатньо повно з деякими зауваженнями – дві третини від максимального бала;
- робота виконана не повністю – одна третина від максимального бала;
- при перевірці роботи виявлені грубі помилки – 0 балів.

Кожний із згаданих показників оцінюється певною кількістю балів, а оцінка всієї лабораторної роботи знаходиться підсумовуванням балів за кожний з показників.

Модульний контроль здійснюється 1 раз за семестр у формі тестових завдань.

Контроль проводиться на 16-ому тижні 5-го (денна ф. н.) / 3-го (денна скорочена ф. н.) семестру.

Семестровий контроль здійснюється у формі заліку. Для здобувачів *денної форми* навчання у разі виконання студентом усіх видів поточних контрольних заходів залік виставляється на підставі зарахованих балів протягом семестру. У відомість оцінка проставляється як у балах національної шкали, так і за шкалою ECTS.

Для здобувачів *заочної форми* навчання при складанні заліку враховуються результати виконання лабораторних робіт, контрольної та самостійної роботи. На залік виділяється 40 балів.

Таблиця розподілення балів для заочної форми навчання.

Змістовий модуль 1	Контрольна робота	Залік	Всього:
Лабораторні роботи	20	40	100
40			

При наявності у здобувачів **результатів неформального навчання** за освітнім компонентом «Офісне програмне забезпечення» у повному обсязі, визнання та оцінювання результатів здійснюється відповідно до «Положення про порядок визнання у Криворізькому національному університеті результатів навчання, отриманих в умовах неформальної освіти». У випадку, якщо за підсумками визнання результатів неформального навчання визнається тільки частина результатів навчання, заявнику зараховуються окремі види навчальної роботи за освітнім компонентом «Офісне програмне забезпечення».

Нижче наведені окремі види навчальної роботи, які можуть бути зараховані здобувачеві при наявності сертифікату про успішне проходження рекомендованих онлайн курсів.

№ з/п	Тема	Посилання на рекомендовані курси
1	Введення в тестування	https://courses.prometheus.org.ua/courses/course-v1:LITS+115+2017_T4/about https://training.qatestlab.com/
2	Веб-тестування	https://training.qatestlab.com/
3	Кросбраузерне тестування.	https://training.qatestlab.com/
4	Функціональне тестування. Регресійне тестування	https://training.qatestlab.com/
5	Тест дизайн (test design) та тестові набори (test case)	https://training.qatestlab.com/
6	Техніки тест дизайну. Організація процесу тестування	https://courses.prometheus.org.ua/courses/course-v1:LITS+115+2017_T4/about

11. ШКАЛА ОЦІНЮВАННЯ

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

12. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ

1. Поняття «якість ПО». Вимоги до якості програмного забезпечення.
2. Модель якості ISO 9126. Аспекти визначення якості та її атрибути.
3. Процес верифікації та валідації.
4. Модель ієрархії процесів забезпечення якості. Quality Assurance, Quality Control та тестування.
5. Процес тестування. Ключові кроки.
6. Принципи тестування.
7. Поняття збою, дефекту та відмови.
8. Ефективність тестування.
9. Звіти про дефекти. Основні атрибути баг-репорту.
10. Життєвий цикл дефекту.
11. Test Case, Тестовий набір (Test suite), Чек-лист, Чит-лист.
12. Класифікація тестування. Типи і види тестування. Рівні тестування.
13. Функціональне тестування (Functional Testing).
14. Тестування методами «білого ящика», «сірого ящика» та «чорного ящика».
15. Тестування безпеки (Security and Access Control Testing).
16. Тестування взаємодії (Interoperability Testing).
17. Навантажувальне тестування або тестування продуктивності.
18. Димове тестування (Smoke Testing) та санітарне тестування (Sanity Testing).
19. Регресійне тестування (Regression Testing).
20. Тестування зручності використання (Usability Testing).
21. Альфа-тестування і Бета-тестування програмних продуктів
22. Компонентне або Модульне тестування (Component or Unit Testing)
23. Інтеграційне тестування (Integration Testing)
24. Системне тестування (System Testing).
25. Приймальне тестування або Приймально-здаточне випробування (Acceptance Testing).
26. Техніки тест дизайну.
27. Еквівалентне розділення (Equivalence Partitioning – EP).
28. Аналіз граничних значень (Boundary Value Analysis – BVA).
29. Таблиці прийняття рішень (матриця відповідності вимог).
30. Попарне тестування (Pairwise Testing).
31. Процес розробки ПЗ (водоспадна, інтеграційна та спіральна модель розробки), методологія розробки Agile (Scrum).
32. План тестування.
33. Звіт про тестування.

13. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ

13.1. Навчальна та довідкова література

1. Aniche, Maurizio. Effective Software Testing: A Developer's Guide. Сполучені Штати Америки: Manning, 2022. – 348 с. <https://books.google.com.ua/books?id=U4BIEAAAQBAJ>
2. 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>
3. Firesmith, Donald. Common System and Software Testing Pitfalls: How to Prevent and Mitigate Them: Descriptions, Symptoms, Consequences, Causes, and Recommendations. Велика Британія: Pearson Education, 2014. – 320 с. <https://books.google.com.ua/books?id=dSaiAgAAQBAJ>
4. Graham, Dorothy., Black, Rex., Veenendaal, Erik van. Foundations of Software Testing: ISTQB Certification. Велика Британія: Cengage Learning, 2012. – 242 с. https://books.google.com.ua/books?id=_fZnXwAACAAJ
5. Myers, Glenford J., Sandler, Corey., Badgett, Tom. The Art of Software Testing. Germany: Wiley, 2011. – 256 с. <https://books.google.com.ua/books?id=GjyEFPkMCwcC>
6. Fournier, Greg. Essential Software Testing: A Use-Case Approach. Велика Британія: CRC Press, 2008. – 280 с. <https://books.google.com.ua/books?id=Q\ \ G2YCKJGsC>
7. Farrell-Vinay, Peter. Manage Software Testing. Сполучені Штати Америки: CRC Press, 2008. – 600 с. <https://books.google.com.ua/books?id=ijQEIKfBaHEC>
8. 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>
9. Is There Such a Thing As Bug-free Software? [Electronic resource] – Access mode: <https://hackernoon.com/is-there-such-a-thing-as-bug-free-software-320cd862af17> .
10. Nielsen Norman Group «Thinking Aloud: The #1 Usability Tool» [Electronic resource] – Access mode: <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/>
11. Software Testing Help. «Web Application Testing Complete Guide (How to Test a Website)» [Electronic resource] – Access mode: <https://www.softwaretestinghelp.com/web-application-testing/> .
12. 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/>
13. 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>
14. Test Plan Outline (IEEE 829 Format) [Electronic resource] – Access mode: <https://www.fit.vutbr.cz/study/courses/ITS/public/ieee829.html>
15. Web Application Testing Complete Guide (How to Test a Website) [Electronic resource] – Access mode: <https://www.softwaretestinghelp.com/web-application-testing/>
16. Web Application Testing: Step by Step Process to make it Right [Electronic resource] – Access mode: <https://tms-outsource.com/blog/posts/web-application-testing/>
17. Історія розвитку тестування програмного забезпечення. [Електронний ресурс] – Режим доступу: <http://bit.ly/2H65DJR>

18. Мобільне тестування (інфографіка) [Електронний ресурс] – Режим доступу: <https://internetdevels.ua/blog/mobile-testing-checklist>
19. Попарне тестування (pairwise testing) [Електронний ресурс] – Режим доступу: <https://training.qatestlab.com/blog/technical-articles/pairwise-testing/>
20. Портал знань [Електронний ресурс] – Режим доступу: <http://www.znannya.org/?view=software-testing-testing> .

13.2. 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. 2019. – 101 с.

14. 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:
 - Mantis Bug Tracker <http://mantis.qatestlab.net> ;
 - TestLink <http://tl-univer.qa-testlab.net/login.php> .
6. Дистанційний курс «Якість програмного забезпечення та тестування» на платформі Google G Suite <https://classroom.google.com/c/NTQ1MDUzODYwODQx>

16. 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.

Крешилог (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."

Робочий план
з дисципліни «Якість програмного забезпечення та тестування»
для здобувачів вищої освіти першого (бакалаврського) рівня вищої освіти
за спеціальністю 121 «Інженерія програмного забезпечення»

Семестр 5-й (денна ф. н.)/ 3-й (денна скорочена ф. н.)

Вид навчальної роботи	Годин у семестрі/кредити	Тиждень																Вид підсумкового контролю
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Лекційні заняття	16	2		2		2		2		2		2		2		2		
Лабораторні роботи	16		2 ПК		2 ПК		2 ПК		2 ПК		2 ПК		2 ПК		2 ПК		2 ПК	
Самостійна робота	58	3	3	3	3	3	3	4	4	3	3	4	4	4	5	4	5	
Всього годин/кредитів	90/3	5	5	5	5	5	5	6	6	5	5	6	6	6	7	6	7	Залік

ПК – поточний контроль;

Позначки:

КМР – контрольньо-модульна робота

Викладач:



І. О. Доценко