



Олександр БАРМАК
Валерія КЛІМЕНКО
Олександр МАЗУРЕЦЬ
Марина МОЛЧАНОВА
Олена СОБКО

ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ТЕОРЕТИЧНИЙ КУРС ТА
ЛАБОРАТОРНИЙ ПРАКТИКУМ:
НАВЧАЛЬНИЙ ПОСІБНИК



Бармак О. В., Кліменко В. І.,
Мазурець О. В., Молчанова М. О., Собко О. В.

**ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ
ТА ТЕСТУВАННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ.
ТЕОРЕТИЧНИЙ КУРС
ТА ЛАБОРАТОРНИЙ ПРАКТИКУМ**

Навчальний посібник

Хмельницький 2024

*Рекомендовано Вченою радою
Хмельницького національного університету
як навчальний посібник для студентів ЗВО
(лист 041/182 від 30.01.2024)*

Авторський колектив:

Бармак О. В., д-р техн. наук, проф.; *Кліменко В. І.*, викладач;
Мазурець О. В., канд. техн. наук, доц.; *Молчанова М. О.*, викладач;
Собко О. В., викладач.

Рецензенти:

- Корнига Я. І.* – д-р техн. наук, проф., декан факультету інформатики та обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»;
- Комар М. П.* – д-р техн. наук, проф., завідувач кафедри інформаційно-обчислювальних систем і управління, Західноукраїнський національний університет;
- Мартишук В. В.* – д-р фіз.-мат. наук, проф., завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій, Хмельницький національний університет.

О-75 Основи програмної інженерії та тестування програмного забезпечення. Теоретичний курс та лабораторний практикум : навч. посіб. / О. В. Бармак, В. І. Кліменко, О. В. Мазурець [та ін.]. – Хмельницький : ХНУ, 2024. – 236 с.

ISBN 978-617-7600-54-0

Навчальний посібник ставить за мету сформувати комплексне розуміння основ інженерії життєвого циклу розробки й тестування програмного забезпечення та формування відповідних практичних навичок для їх використання під час розв'язання складних спеціалізованих задач у галузі інформаційних технологій.

Для здобувачів вищої освіти, які навчаються за освітньо-професійними програмами в межах спеціальності 122 «Комп'ютерні науки».

УДК 004.4+004.9

ПЕРЕДМОВА	10
РОЗДІЛ 1. ТЕОРЕТИЧНИЙ КУРС	13
ТЕМА 1. Життєвий цикл розробки програмного забезпечення	13
1.1. Основи програмної інженерії та проблеми розробки програмного забезпечення.....	13
1.2. Цільові об'єкти програмної інженерії.....	14
1.3. Життєвий цикл розробки ПЗ. Міжнародні стандарти життєвого циклу ПЗ.....	14
1.4. Структура програмного забезпечення.....	15
1.5. Моделі та методології розробки програмного забезпечення.....	16
1.6. Методи та інструменти програмної інженерії.....	19
1.7. Розробка проєктної документації відповідно до вимог і обмежень замовника.....	19
1.8. Технічне завдання (Technical Specification).....	20
Запитання для самоконтролю.....	21
Завдання на лабораторний практикум.....	21
ТЕМА 2. Основи інженерії вимог до програмного забезпечення	22
2.1. Види вимог до ПЗ. Етапи складання вимог до ПЗ.....	22
2.2. Функціональні вимоги. Нефункціональні вимоги. Галузеві вимоги. Вимоги до програмного забезпечення.....	24
2.3. Аналіз потреб користувачів та збір вимог.....	25
2.4. Специфікація вимог до програмного забезпечення (SRS). Компоненти SRS. Тестування вимог (RTS). Способи подання вимог до ПЗ.....	25
Запитання для самоконтролю.....	27
Індивідуальне завдання для самостійної роботи.....	27
ТЕМА 3. Гнучкі методології створення програмного забезпечення та системи керування проєктами	28
3.1. Керування життєвим циклом ПЗ. Гнучкі методології створення програмних продуктів.....	28
3.2. Переваги та недоліки методології гнучкої розробки.....	30
3.3. Сфери застосування Agile.....	30
3.4. Методологія Scrum.....	31
3.5. Методологія Kanban.....	33
3.6. Інструменти керування проєктами.....	34
3.7. Середовище керування проєктами Jira.....	35
Запитання для самоконтролю.....	36

Завдання на лабораторний практикум	36
Індивідуальне завдання для самостійної роботи	36
ТЕМА 4. Системи керування версіями та сервіси для спільної роботи.....	37
4.1. Системи контролю версій	37
4.2. Типи систем контролю версій.....	38
4.3. Система керування версіями Git.....	39
4.4. Вебсервіс спільної роботи GitLab та GitHub	40
4.5. Компоненти та команди GitHub	41
4.6. Проблема конфлікту злиття версій.....	44
4.7. Робота з програмним кодом за допомогою GitHub Desktop	44
Запитання для самоконтролю	45
Завдання на лабораторний практикум	45
ТЕМА 5. Проектування інтерфейсу користувача та інструменти прототипування.....	46
5.1. Призначення та типи інтерфейсів користувача. Введення до UI/UX.	46
5.2. Етапи проектування інтерфейсу користувача (UI)	47
5.3. Дослідження потреб та очікувань користувачів для формування UX.....	48
5.4. Прототипування та креативність. Види прототипів. Засоби прототипування (Figma, Sketch, Axure RP, Adobe Photoshop)	49
5.5. Компоненти дизайну (сітка, колір, типографіка)	52
5.6. Принципи UX Flow та UI Fluidity Importance.....	55
Запитання для самоконтролю	56
Завдання на лабораторний практикум	56
ТЕМА 6. Проектування архітектури програмного забезпечення.....	57
6.1. Проектування архітектури ПЗ. Принципи проектування архітектури.	57
6.2. Послідовність проектування архітектури ПЗ	58
6.3. Критерії для оцінки спроектованої архітектури.....	59
6.4. Концептуальна модель програмного забезпечення	60
6.5. Структура системи та її властивості. Типи структур систем ПЗ	61
6.6. Шаблони проектування програмного забезпечення	64
Запитання для самоконтролю	66
Практичні завдання для самостійного опрацювання	66
ТЕМА 7. Мови та засоби моделювання програмного забезпечення.....	67
7.1. Підходи до моделювання ПЗ.....	67
7.2. Методи структурного аналізу систем.....	68

7.3. Мови моделювання (UML, SysML, ArchiMate).....	69
7.4. Засоби інформаційного моделювання (діаграми сценаріїв, взаємодії, станів, класів, компонентів, часові).....	70
7.5. Стандарт моделювання бізнес-процесів BPMN	71
7.6. Особливості аналізу та функціонального моделювання бізнес-процесів	74
Запитання для самоконтролю	75
Практичні завдання для самостійного опрацювання.....	75
ТЕМА 8. Методології, технології та засоби моделювання програмного забезпечення.....	76
8.1. Методології моделювання ПЗ.....	76
8.2. Особливості дослідження характеристик і поведінки складних об'єктів і систем	77
8.3. Моделювання потоків даних (DFD)	78
8.4. Нотації DFD-діаграм.....	79
8.5. Методи оцінювання ризиків проєктування функціональних моделей	81
8.6. Функціональне моделювання бізнес-процесів (SADT, IDEF0) ..	82
8.7. CASE-технології. Класифікація CASE-засобів. Переваги CASE-технологій.....	85
8.8. Сучасні засоби автоматизації моделювання ПЗ.....	86
Запитання для самоконтролю	87
Завдання на лабораторний практикум	87
ТЕМА 9. Методи забезпечення та контролю якості програмного забезпечення.....	88
9.1. Поняття якості ПЗ. Метрики і стандарти якості ПЗ.....	88
9.2. Стандарти ISO 9000 / IEEE	89
9.3. Завдання тестування ПЗ	90
9.4. Основні принципи тестування	90
9.5. Система моніторингу, контролю та забезпечення якості.....	91
9.6. Шляхи оцінювання та забезпечення якості ПЗ (Verification and Validation). Тестувальник і QA інженер	92
Запитання для самоконтролю	93
Завдання на лабораторний практикум	93
ТЕМА 10. Планування тестування програмного забезпечення та робота з дефектами.....	94
10.1. Тестування програмного забезпечення	94
10.2. Базова термінологія тестування ПЗ	94
10.3. Види тестування.....	95

10.4. Життєвий цикл тестування (STLC)	95
10.5. Документація з тестування ПЗ.....	96
10.6. План тестування (Test Plan)	98
10.7. Пошук та документування дефектів у ПЗ.....	98
10.8. Пріоритетність та критичність дефектів.....	100
10.9. Життєвий цикл дефектів.....	100
10.10. Інструментарій Jira для документування дефектів	100
Запитання для самоконтролю	101
Завдання на лабораторний практикум	101
ТЕМА 11. Підходи та рівні тестування	102
11.1. Підходи до тестування. Ознайомлювальне тестування, тестування за сценарієм	102
11.2. Ручне і автоматизоване тестування	103
11.3. Позитивне і негативне тестування.....	103
11.4. Функціональний та структурний підходи до тестування.	104
11.5. Рівні тестування. Модульне тестування (Unit testing). Інтеграційне тестування (Integration testing). Системне тестування (System testing). Приймальне тестування (Acceptance testing)	104
11.6. Регресивне тестування та комбінування рівнів тестування	105
11.7. Тестування продуктивності.....	105
11.8. Інструментальні засоби тестування.....	106
11.9. Класифікація помилок	106
Запитання для самоконтролю	107
Практичні завдання для самостійного опрацювання.....	107
ТЕМА 12. Засоби ручного тестування	108
12.1. Ручне тестування. Типи ручного тестування. Переваги та недоліки ручного тестування.....	108
12.2. Процес ручного тестування.....	109
12.3. Правила ручного тестування.....	109
12.4. Інструменти для ручного тестування	109
12.5. Використання Checklist для планування розробки Test Case. Приклади Checklist	110
12.6. Тестування поведінки ПЗ засобами Test Case	111
12.7. Життєвий цикл Test Case.....	113
12.8. Атрибути (поля) Test Case. Приклади Test Case	115
12.9. Test Case Suite. Види Test Case Suite.	118
12.10. Інструментальні засоби для створення Test Case.....	119

Запитання для самоконтролю	120
Завдання на лабораторний практикум	120
Індивідуальне завдання для самостійної роботи.....	120
ТЕМА 13. Засоби автоматизованого тестування	121
13.1. Автоматизація тестування. Переваги та недоліки	121
13.2. Інструменти та технології автоматизації тестування.....	122
13.3. Генератори тестів. Структура тестового набору для автоматизованого прогону. Додатки для автоматизованого прогону тестів	123
13.4. Сервера постійної інтеграції (Hudson, CruiseControl).....	124
13.5. Порівняння витрат та ефективності різних методів тестування.....	124
Запитання для самоконтролю	125
Практичні завдання для самостійного опрацювання.....	125
ТЕМА 14. Спеціалізовані фреймворки для модульного тестування.....	126
14.1. Основні принципи модульного тестування. Типи модульних тестів.....	126
14.2. Переваги та недоліки модульного тестування	126
14.3. Інструменти модульного тестування. Фреймворки для модульного тестування.....	127
14.4. Фреймворки для модульного тестування.....	127
14.5. Тестування ПЗ за допомогою Unit Test.....	128
14.6. Правила написання модульних тестів. Інструменти модульного тестування.....	130
14.7. Генерація звітів з тестування Unit Test	130
Запитання для самоконтролю	131
Завдання на лабораторний практикум	131
Індивідуальне завдання для самостійної роботи.....	131
ТЕМА 15. Тестування інтерфейсу користувача.....	132
15.1. Завдання і цілі тестування UI.....	132
15.2. Типи вимог до UI	133
15.3. Функціональне тестування UI.....	134
15.4. Тестування зручності використання UI	135
15.5. Методи тестування UI	135
15.6. Ітеративний підхід до покращення UI/UX.....	136
15.7. Проблеми тестування UI	136
Запитання для самоконтролю	137
Практичні завдання для самостійного опрацювання.....	137

ТЕМА 16. Проектування діаграм розгортання програмного забезпечення.....	138
16.1. Розгортання програмного застосунку	138
16.2. Ключові ролі у розгортанні програмного забезпечення	139
16.3. Планування розгортання. План розгортання. Переваги планування розгортання.....	140
16.4. Підготовка середовища для розгортання. Переваги підготовки середовища для розгортання.....	141
16.5. Кроки підготовки середовища для розгортання.....	141
16.6. Інсталяція програмного продукту. Етапи установки програмного продукту. Види установки програмного продукту	142
16.7. Навчання користувачів і персоналу. Мета навчання користувачів і персоналу. Типи навчання.....	142
16.8. Вимоги до розгортання. Мета вимог до розгортання	143
16.9. Діаграма розгортання програмного забезпечення. Елементи діаграми розгортання. Типи діаграм розгортання	143
16.10. Створення діаграм розгортання.....	144
16.11. Переваги діаграм розгортання	145
Запитання для самоконтролю	145
Завдання на лабораторний практикум	145
ТЕМА 17. Керування супроводом програмного забезпечення.....	146
17.1. Особливості супроводу програмного забезпечення	146
17.2. Керування супроводом ПЗ	147
17.3. План супроводу ПЗ	147
17.4. Процеси супроводу ПЗ	148
17.5. Техніки супроводу ПЗ	149
Запитання для самоконтролю	149
Практичні завдання для самостійного опрацювання.....	149
ТЕМА 18. Тестування програмного забезпечення засобами штучного інтелекту.....	150
18.1. Напрямки застосування ШІ для тестування ПЗ	150
18.2. Автоматизоване візуальне тестування ПЗ (Automated Visual Software Testing).....	151
18.3. Інтелектуальне програмування агентів тестування	151
18.4. IntelliTest, як засіб для автоматизації тестування з використанням ШІ.....	153
Запитання для самоконтролю	154
Практичні завдання для самостійного опрацювання.....	154
ПЕРЕЛІК ЛІТЕРАТУРИ ДО ТЕОРЕТИЧНОГО КУРСУ.....	155

РОЗДІЛ II. ЛАБОРАТОРНИЙ ПРАКТИКУМ.....	157
Лабораторна робота 1. Створення технічного завдання для розробки ПЗ.....	157
Лабораторна робота 2. Використання Jira для керування проєктами за методологією Scrum	165
Лабораторна робота 3. Використання GitHub та GitHub Desktop для керування версіями та спільної роботи	174
Лабораторна робота 4. Проктування та креативне прототипування UI з використанням Figma	187
Лабораторна робота 5. Прикладне моделювання потоків даних (DFD) та функціональних моделей бізнес-процесів (IDEF0)	194
Лабораторна робота 6. Пошук та документування дефектів	198
Лабораторна робота 7. Розробка Checklist, Test Case та Test Case Suite для тестування ПЗ.....	207
Лабораторна робота 8. Розробка Unit Test та тестування ПЗ за допомогою фреймворків	214
Лабораторна робота 9. Визначення вимог до розгортання та проєктування діаграми розгортання ПЗ.....	225
Самостійна робота №1. Розробка специфікації вимог до ПЗ (SRS).....	231
Самостійна робота №2. Використання Trello для керування проєктами за методологією Kanban	232
Самостійна робота №3. Розробка Checklist, Test Case та Test Case Suite для тестування ПЗ.....	233
Самостійна робота №4. Розробка Unit Test та тестування ПЗ за допомогою фреймворку NUnit	234
ПЕРЕЛІК ЛІТЕРАТУРИ ДО ЛАБОРАТОРНОГО ПРАКТИКУМУ	235

ПЕРЕДМОВА

Сучасний етап розвитку інформаційних технологій та інтенсивне впровадження систем із штучним інтелектом у різні сфери людської діяльності призводить до необхідності розробки складних спеціалізованих програмних систем і висуває нові, підвищені вимоги до підготовки здобувачів вищої освіти спеціальності 122 «Комп'ютерні науки». Зокрема, стрімкий технічний прогрес створює постійну потребу у вдосконаленні та адаптації програм, а постановка задач характеризується комплексністю та невизначеністю умов. Це підвищує роль і одночасно ускладнює задачу інженерії та тестування програмного забезпечення, тому сучасний професіонал з комп'ютерних наук повинен володіти цілим рядом компетенцій, серед яких особливе місце займають інженерія життєвого циклу розробки та життєвого циклу тестування програмного забезпечення. Набуття цих компетентностей надає можливість створювати продукти, які відповідають сучасним вимогам і легко інтегруються з новими технологіями.

Отже, в епоху інтелектуальних систем програмна інженерія та тестування стають стратегічно важливими складовими розробки програмних систем, забезпечуючи якість, безпеку та ефективність продуктів. Програмна інженерія та тестування стають ефективними інструментами для виявлення та усунення вразливостей, забезпечуючи надійний захист від можливих атак. Зміни в бізнес-моделях, зокрема перехід до цифрових стратегій, підкреслюють необхідність надійного програмного забезпечення для оптимізації бізнес-процесів та поліпшення обслуговування клієнтів. Постійне зростання вимог до якості користувацького досвіду покладає відповідальність на програмну інженерію та тестування для забезпечення зручності та ефективності програм, вони грають важливу роль у підтримці високого стандарту користувацького інтерфейсу та взаємодії. Розгортання стратегій DevOps та автоматизації в розробці програмного забезпечення підкреслює значення програмної інженерії та тестування для забезпечення швидкого випуску високоякісного програмного забезпечення. Тестування програмного забезпечення забезпечує виявлення дефектів та підвищення якості продукту шляхом аналізу програмного забезпечення.

Мета дисципліни «Основи програмної інженерії та тестування програмного забезпечення» – надання здобувачам вищої освіти теоретичних знань та формування практичних навичок з інженерії та тестування програмного забезпечення для їх використання під час розв'язання складних спеціалізованих задач та практичних проблем у галузі комп'ютерних наук, що передбачає застосування практичних навичок з інженерії життєвого

циклу розробки й тестування програмного забезпечення і характеризується комплексністю та невизначеністю умов.

Дисципліна обсягом п'ять кредитів ЄКТС, передбачає аудиторні заняття і самостійну роботу із засвоєння теоретичних положень дисципліни, набуття практичних навичок на лабораторних заняттях, виконання індивідуальних завдань та іспит.

Вивчення дисципліни у повному обсязі сприятиме поглибленню та розширенню компетентностей та програмних результатів навчання, зокрема таких як: оцінювати та забезпечувати якість виконуваних робіт, бути критичними і самокритичними при розробці ПЗ, тестуючи його та детектуючи у ньому дефекти; застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника; застосовувати теоретичні та практичні основи методології та технології моделювання потоків даних (DFD) для дослідження характеристик і поведінки складних об'єктів і систем; розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорії та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов; виконувати аналіз та функціональне моделювання бізнес-процесів (IDEF0), побудову та практичне застосування функціональних моделей організаційно-економічних і виробничо-технічних систем; володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (технічне завдання); застосовувати знання методології та CASE-засобів проєктування складних систем й методів структурного аналізу систем при розробці і дослідженні функціональних моделей (IDEF0) організаційно-економічних і виробничо-технічних систем.

Структура та вміст курсу орієнтовані на здобувачів вищої освіти спеціальності 122 «Комп'ютерні науки», відповідають сучасним світовим тенденціям в області інженерії життєвого циклу розробки і тестування програмного забезпечення, враховано актуальні міжнародні стандарти життєвого циклу програмних засобів. Знайшли широке відображення ідеї і підходи до інженерії та тестування програмного забезпечення, які на даний час стали стандартом викладання дисципліни та наведені у працях дипломованих фахівців в напрямку комп'ютерних наук світового рівня, таких як «Concise Guide to Software Testing (Undergraduate Topics in Computer

Science)» (Dr. Gerard O'Regan) [1] та «Essentials of Software Engineering» (Dr. Frank Tsui, Dr. Orlando Karam) [2].

Навчальний посібник структурно складається з передмови та двох основних розділів зі списками літератури до кожного.

Розділ «Теоретичний курс» містить основні теми: життєвий цикл розробки програмного забезпечення, основи інженерії вимог до програмного забезпечення, гнучкі методології створення програмного забезпечення та системи керування проєктами, системи керування версіями та сервіси для спільної роботи, проєктування інтерфейсу користувача й інструменти прототипування, проєктування архітектури програмного забезпечення, мови та засоби моделювання програмного забезпечення, методології та засоби моделювання програмного забезпечення, методи забезпечення та контролю якості програмного забезпечення, проєктування діаграм розгортання програмного забезпечення, керування супроводом програмного забезпечення, планування тестування програмного забезпечення та робота з дефектами, підходи та рівні тестування, засоби ручного та автоматизованого тестування, спеціалізовані фреймворки для модульного тестування, тестування інтерфейсу користувача, та тестування програмного забезпечення засобами штучного інтелекту. В кінці кожної теми наводяться запитання для самоконтролю, посилання на відповідну лабораторну роботу та завдання для самостійної роботи.

Розділ «Лабораторний практикум» містить завдання та описи ходу виконання лабораторних робіт та завдань для самостійного виконання, які закріплюють отримані теоретичні знання. Завданням лабораторних робіт та індивідуальних завдань з дисципліни є набуття практичних навичок створювати технічне завдання для розробки програмного забезпечення, використовувати середовища Jira та Trello для керування проєктами за методологіями Scrum та Kanban, використовувати засоби GitHub та GitHub Desktop для керування версіями та спільної роботи, проєктувати та креативно прототипувати UI з використанням інструментарію Figma, виконувати прикладне моделювання потоків даних (DFD) та функціональних моделей бізнес-процесів (IDEF0), виконувати пошук та документування дефектів у програмному забезпеченні, розробляти Checklist, Test Case та Test Case Suite для тестування програмного забезпечення, розробляти Unit Test та тестувати програмне забезпечення за допомогою фреймворків, визначати вимоги до розгортання програмного забезпечення та проєктувати діаграми розгортання програмного забезпечення.

Навчальне видання

*Бармак Олександр Володимирович,
Кліменко Валерія Ігорівна,
Мазурець Олександр Вікторович
та ін.*

**ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ
ТА ТЕСТУВАННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ.
ТЕОРЕТИЧНИЙ КУРС
ТА ЛАБОРАТОРНИЙ ПРАКТИКУМ**

Навчальний посібник

Підписано до друку 19.03.2024, Формат 60 × 84 1/16.
Папір офсетний. Друк цифровий. Гарн. Таймс. Ум. друк. арк. 13,58.
Тираж 100 прим. Зам. № 3012

Авторський оригінал-макет

Віддруковано: ФОП Мельник А.А.
м. Хмельницький, вул. Чорновола 37, тел. (0382) 74-32-22
Свідчення Державного комітету інформаційної політики, телебачення
та радіомовлення України про внесення суб'єкта видавничої справи
до державного реєстру видавців, виготівників та розповсюджувачів видавничої продукції
ДК №1942 від 15.09.2008



БАРМАК
Олександр Володимирович



КЛІМЕНКО
Валерія Ігорівна



МАЗУРЕЦЬ
Олександр Вікторович



МОЛЧАНОВА
Марина Олексіївна



СОБКО
Олена Віталіївна

Запропонований увазі читача навчальний посібник ставить за мету сформувати комплексне розуміння основ інженерії життєвого циклу розробки й тестування програмного забезпечення та формування практичних навичок для їх використання під час розв'язання складних спеціалізованих задач та практичних проблем у предметній області комп'ютерних наук.

У навчальному посібнику знайшли широке відображення сучасні наукові ідеї та інноваційні практики з інженерії й тестування програмного забезпечення, враховано актуальні міжнародні стандарти життєвого циклу програмних засобів. Структура та вміст посібника відповідають новітнім світовим підходам до вивчення основ розробки і тестування програмного забезпечення й сприяють якісній практичній підготовці фахівців у галузі інформаційних технологій.