

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Кафедра комп'ютерної інженерії та електромеханіки

ЗАТВЕРДЖУЮ

Декан факультету інженерії та
інформаційних технологій

Ігор ПІНАСЮК
(підпис)
« 11 » червня 2025 року



РОБОЧА ПРОГРАМА

навчальної дисципліни Проектування архітектури та забезпечення якості програмного забезпечення

Рівень вищої освіти другий (магістерський)

Ступінь вищої освіти магістр

Спеціальність F3 Комп'ютерні науки

Освітня програма Комп'ютерні науки

Факультет інженерії та інформаційних технологій

Київ
2025 рік

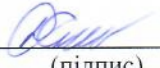
РОЗРОБЛЕНО: Київський національний університет технологій та дизайну

РОЗРОБНИКИ ПРОГРАМИ: Калашник Валерій Юрійович, к.т.н., старший викладач, кафедри комп'ютерної інженерії та електромеханіки

Схвалено вченою радою факультету інженерії та інформаційних технологій
від «11» червня 2025 року, протокол № 3

Схвалено науково-методичною радою факультету інженерії та інформаційних технологій
від «11» червня 2025 року, протокол № 2

Обговорено та рекомендовано на засіданні кафедри комп'ютерної інженерії та електромеханіки
Протокол від «5» червня 2025 року № 12

Завідувач кафедри  Дмитро СТАЦЕНКО
(підпис)

Погоджено:

Гарант ОП кафедри
комп'ютерної інженерії та електромеханіки  Геннадій МЕЛЬНИК
(підпис)

«5 » червня 2025 року

1 ОПИС ДИСЦИПЛІНИ

Найменування показників	Характеристика дисципліни	
	очна форма здобуття вищої освіти	заочна, дистанційна форма здобуття вищої освіти
Кількість годин / кредитів – 90 / 3	обов'язкова	
Змістові модулі – 1	Рік підготовки:	
Розділи – 1	1-й	1-й
Загальна кількість тижневих годин для денної форми здобуття вищої освіти: аудиторних – 2 самостійної роботи – 5,5	Семестр	
	1-й	1-й
	Лекції	
	12 год.	2 год.
	Практичні	
	12 год.	2 год.
	Самостійна робота	
	66 год.	86 год.
	Вид підсумкового контролю: екзамен (семестр 1).	

2 АНОТАЦІЯ ДИСЦИПЛІНИ

Назва дисципліни: Проєктування архітектури та забезпечення якості програмного забезпечення. Робоча програма навчальної дисципліни складається з такого змістовного модуля:

Змістовий модуль 1. Проєктування архітектури та забезпечення якості програмного забезпечення.

Мета курсу – набуття студентами комплексних теоретичних знань та практичних навичок у галузі проєктування, візуалізації, тестування та забезпечення якості сучасних архітектур програмного забезпечення, включаючи принципи розробки розподілених систем, контейнеризації та оптимізації процесів тестування.

Результати навчання:

знати: Ключові архітектурні патерни програмних систем (багаторівневі, подієво-орієнтовані, мікроядерні, мікросервісні, клієнт-серверні). Принципи роботи технологій контейнеризації та їхню роль у розгортанні архітектур. Основні концепції та можливості сучасних фреймворків для веб-додатків та мікросервісів. Стратегії та методи інтеграційного, системного та Performance тестування архітектурних рішень. Алгоритми генерації та оптимізації тестових даних, а також аналізу результатів тестування. Підходи та інструменти моніторингу, діагностики та усунення проблем ПЗ на етапі експлуатації. Концепції та стратегії реінжинірингу програмного забезпечення. Основи мов моделювання архітектури.

вміти: Аналізувати вимоги та обирати відповідний архітектурний патерн для програмної системи. Застосовувати Docker для контейнеризації та розгортання компонентів архітектури. Використовувати сучасні фреймворки для розробки компонентів веб-додатків та мікросервісів. Розробляти стратегії інтеграційного та системного тестування для складних архітектур. Застосовувати алгоритми для оптимізації процесів тестування та аналізу його результатів. Налаштовувати системи моніторингу та використовувати їх для діагностики проблем. Формулювати задачі на модифікацію або реінжиніринг ПЗ на основі аналізу проблем. Створювати діаграми архітектури.

здатен продемонструвати: Глибоке розуміння принципів проєктування масштабованих та надійних архітектур ПЗ. Здатність застосовувати сучасні технології та інструменти для реалізації архітектурних рішень. Компетентність у забезпеченні якості програмного забезпечення на всіх етапах його життєвого циклу. Критичне мислення при виборі та обґрунтуванні архітектурних рішень та методик тестування.

володіти навичками: Проєктування архітектури програмних систем. Роботи контейнеризацію та системами оркестрації контейнерів. Розробки високопродуктивних та безпечних веб-додатків/мікросервісів за допомогою сучасних фреймворків. Автоматизації тестування та інтеграції його в CI/CD процеси. Моніторингу та аналізу логів розподілених систем. Виявлення та усунення архітектурних недоліків. Моделювання архітектури за допомогою стандартних нотаций.

самостійно вирішувати: Питання, пов'язані з вибором та обґрунтуванням архітектурного патерну для конкретного проєкту. Завдання з оптимізації процесів тестування та забезпечення якості ПЗ. Проблеми, що виникають під час експлуатації програмних систем. Задачі з підтримки узгодженості та еволюції архітектурних моделей.

Програмні результати навчання:

ІК Здатність розв'язувати задачі дослідницького та/або інноваційного характеру у сфері комп'ютерних наук.

- | | |
|------|---|
| ЗК 1 | Здатність до абстрактного мислення, аналізу та синтезу. |
| ЗК 2 | Здатність застосовувати знання у практичних ситуаціях. |
| ЗК 3 | Здатність спілкуватися державною мовою як усно, так і письмово. |
| ЗК 4 | Здатність спілкуватися іноземною мовою. |
| ЗК 5 | Здатність вчитися й оволодівати сучасними знаннями. |
| ЗК 6 | Здатність бути критичним і самокритичним. |
| ЗК 7 | Здатність генерувати нові ідеї (креативність). |

ФК 2 Здатність формалізувати предметну область певного проєкту у вигляді відповідної інформаційної моделі.

- ФК 5** Здатність розробляти, описувати, аналізувати та оптимізувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення.
- ФК 6** Здатність застосовувати існуючі і розробляти нові алгоритми розв'язування задач у галузі комп'ютерних наук.
- ФК 8** Здатність розробляти і реалізовувати проєкти зі створення програмного забезпечення, у тому числі в непередбачуваних умовах, за нечітких вимог та необхідності застосовувати нові стратегічні підходи, використовувати програмні інструменти для організації командної роботи над проєктом.
- ФК 10** Здатність оцінювати та забезпечувати якість ІТ- проєктів, інформаційних та комп'ютерних систем різного призначення, застосовувати міжнародні стандарти оцінки якості програмного забезпечення інформаційних та комп'ютерних систем, моделі оцінки зрілості процесів розробки інформаційних та комп'ютерних систем.
- ПРН1** Мати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері комп'ютерних наук і є основою для оригінального мислення та проведення досліджень, критичне осмислення проблем у сфері комп'ютерних наук та на межі галузей знань.
- ПРН 2** Мати спеціалізовані уміння/навички розв'язання проблем комп'ютерних наук, необхідні для проведення досліджень та/або провадження інноваційної діяльності з метою розвитку нових знань та процедур.
- ПРН 3** Зрозуміло і недвозначно доносити власні знання, висновки та аргументацію у сфері комп'ютерних наук до фахівців і нефахівців, зокрема до осіб, які навчаються.
- ПРН 4** Управляти робочими процесами у сфері інформаційних технологій, які є складними, непередбачуваними та потребують нових стратегічних підходів.
- ПРН 6** Розробляти концептуальну модель інформаційної або комп'ютерної системи.
- ПРН 10** Проєктувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення.
- ПРН 11** Створювати нові алгоритми розв'язування задач у сфері комп'ютерних наук, оцінювати їх ефективність та обмеження на їх застосування.
- ПРН 13** Оцінювати та забезпечувати якість інформаційних та комп'ютерних систем різного призначення.
- ПРН 14** Тестувати програмне забезпечення.
- ПРН 17** Виявляти та усувати проблемні ситуації в процесі експлуатації програмного забезпечення, формулювати завдання для його модифікації або реінжинірингу.
- ПРН 18** Збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до інформаційної або комп'ютерної системи, що розробляється, експлуатується чи супроводжується.
- ПРН 19** Аналізувати сучасний стан і світові тенденції розвитку комп'ютерних наук та інформаційних технологій.

Необхідні передумови: ступень бакалавра.

Види навчальних занять: лекція, практичні, самостійні.

Методи навчання: Пояснювально-демонстраційний, репродуктивний, частково-пошуковий, дослідницький.

Методи контролю: Усний (опитування, захист практичних робіт), письмовий (звіти з практичних робіт), підсумкове тестування (іспит).

Інструменти, обладнання та програмне забезпечення (за потреби):

ПК з ПЗ, Visual Studio Code (VS Code), NestJS та npm, Docker Desktop, Postman, Archimate, Git, k6, Grafana, draw.io

Форми підсумкового контролю: екзамен.

Засоби діагностики успішності навчання: виконання та захист практичних робіт, звіт за результатом самостійної роботи, тестові завдання для підсумкового контролю.

Мова навчання: українська.

3 ПРОГРАМА ДИСЦИПЛІНИ

Тематичний план навчальної дисципліни

№ теми	Назва теми лекції, практичного, лабораторного, семінарського, індивідуального заняття	Кількість годин за формами здобуття вищої освіти:	
		очна	заочна, дистанційна
1	Тема 1. Проектування сучасних архітектурних рішень	20	20
	Лекція 1: Проектування сучасних архітектурних рішень за допомогою фреймворків	2	0,3
	Практична робота 1: Проектування та реалізація архітектури	2	0,3
	Самостійна робота.	16	19,4
2	Тема 2. Model моделювання та візуалізація архітектури програмного забезпечення	24	24
	Лекція 2: UML + C4 Model моделювання та візуалізація архітектури програмного забезпечення	2	0,3
	Практична робота 2: Моделювання та візуалізація архітектури за допомогою C4 Model	2	0,3
	Лекція 3: Archimate + Порівняння + Підтримка моделей	2	0,3
	Практична робота 3: Поглиблене моделювання та підтримка (Archimate)	2	0,3
	Самостійна робота.	16	22,8
3	Тема 3. Розробка та оцінка ефективності алгоритмів для оптимізації процесів тестування	20	20
	Лекція 4: Розробка та оцінка ефективності алгоритмів для оптимізації процесів тестування	2	0,3
	Практична робота 4: Автоматизація тестування	2	0,3
	Самостійна робота.	16	19,4
4	Тема 4. Тестування, моніторинг, реінжиніринг програмного забезпечення.	26	26
	Лекція 5: Тестування та моніторинг програмного забезпечення	2	0,3
	Практична робота 5: Тестування продуктивності та моніторинг.	2	0,3
	Лекція 6: Виявлення та усунення проблем. Реінжиніринг.	2	0,3
	Практична робота 6: Реінжиніринг та оптимізація сервісу.	2	0,3
	Самостійна робота.	18	24,8
Разом з дисципліни		90	

5 ОЦІНЮВАННЯ

5.1 Розподіл балів з дисципліни, які отримують здобувачі вищої освіти

Форма для екзамену

Поточне оцінювання та самостійна робота				Екзамен	Сума
T1	T2	T3	T4		
20	21	23	26	10	100

5.2 Розподіл балів за видами робіт

Семестр 1

Види робіт, що оцінюються в балах	T1	T2		T3	T4		Усього
Практична робота	Пр 1	Пр 2	Пр 3	Пр 4	Пр 5	Пр 6	
Виконання і захист практичної роботи	6	5	5	7	8	9	40
Самостійна робота	Самостійна робота 1			Самостійна робота 2			
Звіт за результатом самостійної роботи	25			25			50
Екзамен	10						10
Всього з дисципліни							100

5.3 Критерії оцінювання

Об'єктом оцінювання знань здобувачів є програмний матеріал освітнього компоненту, засвоєння якого перевіряється під час навчальних занять та СРС.

Поточний контроль знань і умінь здобувачів вищої освіти проводиться після вивчення логічно завершеної частини навчального матеріалу освітнього компоненту.

Оцінювання виконаних видів робіт здійснюється з урахуванням самостійності та повноти виконання, якості оформлення звіту та своєчасності представлення результатів.

Виконаним вважається практична робота, що відпрацьовано в повному обсязі, оформлено у вигляді звіту (протоколу) та своєчасно подано викладачу для оцінювання, але не пізніше 2 робочих днів до початку сесії.

Виконаною вважається самостійна робота, яка зроблена в повному обсязі та вчасно подана на перевірку викладачу, але не пізніше 2 робочих днів до початку сесії.

Критерії оцінювання видів робіт

Оцінка у відсотках від макс. кількості балів, відведених за певний вид роботи	Критерії оцінювання видів робіт
100%	Здобувач виконав весь обсяг запланованих робіт заняття і бездоганно відповідає на запитання про виконану роботу.
80%	Здобувач виконав весь обсяг запланованих робіт заняття, але при відповіді на запитання робить помилки, які виправляє самостійно після вказування на них.
70%	Здобувач виконав весь обсяг запланованих робіт заняття, але при відповіді на запитання робить помилки, які виправляє з деякою підказкою викладача.
60%	Здобувач виконав весь обсяг запланованих робіт заняття, але при відповіді на запитання робить помилки, які не виправляє самостійно

	після вказування на них. Здобувач орієнтується у всіх питаннях, що стосуються дисципліни (теми заняття).
50%	Здобувач виконав весь обсяг запланованих робіт заняття, але при відповіді на запитання робить помилки, які не виправляє самостійно після вказування на них. Здобувач орієнтується не у всіх, а у більшості питань що стосуються дисципліни (теми заняття).
40%	Здобувач виконав весь обсяг запланованих робіт заняття, але не орієнтується у питаннях, що стосуються дисципліни (теми заняття), або не виконав весь обсяг запланованих робіт, але орієнтується у запитаннях щодо виконаної частини роботи і оформлення звіту (протоколу) відповідає вимогам.
20%	Здобувач не виконав весь обсяг запланованих робіт заняття, не відповідає на запитання про виконану роботу, оформлення звіту (протоколу) не відповідає вимогам. Здобувач повинен повторно відпрацювати і захистити роботу.

Критерії оцінювання для екзамену

Студент допускається до складання екзамену, якщо виконані всі види робіт, передбачені робочою програмою та сума накопичених протягом семестру балів не менша ніж 35.

Екзамен проводиться у формі тестового контролю та складається з 20 питань на знання та розуміння теоретичних відомостей та на володіння навичками застосування теоретичних знань для розв'язання практичних задач спеціальності по 0,5 балу за кожне питання.

Всього – 10 балів.

Студент вважається таким, що склав екзамен, якщо він за результатами складання екзамену набрав не менше 6 балів

Відповідність шкал оцінок якості засвоєння навчального матеріалу

Оцінка за національною шкалою для екзамену, КП, КР / заліку	Оцінка за шкалою КНУТД	Оцінка за шкалою ECTS	Пояснення
Відмінно / зараховано	90-100	A	Відмінно (відмінне виконання лише з незначною кількістю помилок)
Добре / зараховано	82-89	B	Дуже добре (вище середнього рівня з кількома помилками)
	74-81	C	Добре (в загальному вірне виконання з певною кількістю суттєвих помилок)
Задовільно / зараховано	64-73	D	Задовільно (непогано, але зі значною кількістю недоліків)
	60-63	E	Достатньо (виконання відповідає мінімальним критеріям)
Незадовільно / не зараховано	35-59	FX	Незадовільно (з можливістю повторного складання)
	0-34	F	Незадовільно (з обов'язковим повторним вивченням дисципліни)

6 ПОЛІТИКА КУРСУ

6.1 Обов'язкове дотримання академічної доброчесності студентами, а саме:

- самостійне виконання всіх видів робіт, завдань, форм контролю, передбачених робочою програмою навчальної дисципліни;
- посилення на джерела інформації у разі використання ідей, розробок, тверджень, відомостей;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використанні методики досліджень і джерела інформації.

6.2 Виконання та захист практичних занять, СРС має відбуватися під час навчальних занять та консультацій відповідно графіку освітнього процесу.

6.3 Перенесення терміну здачі робіт/перездача:

- допускається з поважних причин (лікарняний, академічна мобільність тощо) за письмовою заявою, завіреною працівниками деканату;
- в умовах воєнного часу допускається звітування щодо виконаних лабораторних/практичних занять та курсових робіт в інші дні і часи, ніж передбачені планом занять, але лише за умов узгодженості з викладачами.

6.4 При виявленні плагіату робота студента не оцінюється, а відправляється на переробку з додатковими питаннями по роботі або надається новий варіант.

6.5 Виконані роботи повинні бути захищені на парах за оформленими вимогами наданими кожній роботі, також завантажені у відповідний модуль діяльності на сторінці дисципліни в МСОП за два робочі дні до початку сесії, в іншому випадку вони вважаються такими, які подані до перездачі;

6.6 Допускається визнання результатів навчання, отриманих у формальній, неформальній та інформальній освіті відповідно до «Положенням про порядок визнання та перезарахування результатів навчання, здобутих шляхом формальної, неформальної та/або інформальної освіти та визначення академічної різниці у Київському національному університеті технологій та дизайну».

7 МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

1. Проєктування архітектури та забезпечення якості програмного забезпечення: конспект лекцій для здобувачів вищої освіти спеціальності F3 Комп'ютерна інженерія денної та заочної форми навчання / Упор. В.Ю. Калашник, [Електронний ресурс] – К.: КНУТД, 2025. – 255 с.
2. Проєктування архітектури та забезпечення якості програмного забезпечення: методичні вказівки до виконання практичних робіт здобувачів вищої освіти спеціальності F3 Комп'ютерна інженерія / Упор. В.Ю. Калашник, [Електронний ресурс] – К.: КНУТД, 2025. – 209 с.
3. Проєктування архітектури та забезпечення якості програмного забезпечення: методичні вказівки для самостійних робіт здобувачів вищої освіти спеціальності F3 Комп'ютерна інженерія / Упор. В.Ю. Калашник, [Електронний ресурс] – К.: КНУТД, 2025. - 27 с.
4. Проєктування архітектури та забезпечення якості програмного забезпечення: методичні вказівки для здобувачів вищої освіти заочної форми навчання спеціальності F3 Комп'ютерна інженерія / Упор. В.Ю. Калашник, [Електронний ресурс] – К.: КНУТД, 2025.

8 РЕКОМЕНДОВАНА ЛІТЕРАТУРА

I. Публікації Авторів Курсу (2025)

1. Калашник, В. Ю. (2025). **Стратегії Реінжинірингу Від Монолітної До Мікросервісної Архітектури: Аналіз Ефективності Та Ризиків.** V Міжнародна науково-практична конференція. Munich, Germany. ISBN: 979-8-89814-231-5. Pp. 142-145. URL: <https://eu->

- conf.com/wp-content/uploads/2025/08/MODERN-SCIENTIFIC-PROBLEMS-AND-WAYS-TO-SOLVE-THEM.pdf
2. Калашник, В. Ю. (2025). **Інтеграція Сучасних Індустріальних Практик Проектування Архітектури Та Забезпечення Якості ПЗ У Навчальні Програми**. Abstracts of XXXIV International Scientific and Practical Conference. Krakow, Poland. ISBN: 979-8-89814-228-5. Pp. 60-62. URL: <https://eu-conf.com/wp-content/uploads/2025/08/LATEST-SCIENTIFIC-TRENDS-IN-THE-DEVELOPMENT-OF-EDUCATION.pdf>
 3. Калашник, В. Ю. (2025). **Інноваційні Підходи До Викладання Проектування Архітектури Та Забезпечення Якості Програмного Забезпечення В Контексті Сучасних Освітніх Технологій**. XXVII Міжнародна науково-практична конференція. Munich, Germany. ISBN: 979-8-89940-585-3. Pp. 138-142. URL: <https://eu-conf.com/wp-content/uploads/2025/05/CURRENT-TRENDS-IN-THE-DEVELOPMENT-OF-MODERN-EDUCATIONAL-TECHNOLOGIES.pdf>
 4. Мельник, Г. В., Калашник, В. Ю., & Стаценко, Д. В. (2025). **Еволюція Технологій Управління ІТ-Проектами: Комплексний Підхід**. *Технічні науки та технології*, 2025, 6(47), 1406-1430. DOI: [https://doi.org/10.52058/2786-6025-2025-6\(47\)-1406-1430](https://doi.org/10.52058/2786-6025-2025-6(47)-1406-1430)
 5. Астістова, Т. І., Дроменко, В. Б., & Калашник, В. Ю. (2025). **Розроблення Програмного Забезпечення Для Інформаційного Ресурсу Бібліотеки Закладів Освіти**. *Інвестиції: практика та досвід*, 2025(3.2), 7. DOI: <https://doi.org/10.32782/2663-5941/2025.3.2/07>

II. Актуальні Міжнародні Джерела (2024–2021)

6. Richards, M., & Gandhi, R. (2024). **Head First Software Architecture: A Learner's Guide to Architectural Thinking**. Sebastopol, CA: O'Reilly Media. ISBN: 978-1098134358
7. Davis, A. (2024). **Bootstrapping Microservices, Second Edition: With Docker, Kubernetes, GitHub Actions, and Terraform** (2nd ed.). Manning Publications. ISBN: 978-1633438569
8. Lock, A. (2023). **ASP.NET Core in Action** (3rd ed.). Manning Publications. ISBN: 978-1633438620
9. Burns, B. (2023). **Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services**. O'Reilly Media. ISBN: 978-1098156343
10. Kane, S., & Matthias, K. (2023). **Docker: Up & Running; Shipping Reliable Containers in Production** (3rd ed.). Sebastopol, CA: O'Reilly Media. ISBN: 978-1098131821
11. **In This AWS re:Invent 2023 Session, Balachander Keelapudi, Jonathan Luke, and Lee Gilmore Discuss How City Electric Supply (CES) Leveraged AWS Serverless for Event-Driven Architecture and Domain-Driven Design** (2023). [Відеозапис].
12. Ford, N., Parsons, R., Sadalage, P., & Kua, P. (2022). **Building Evolutionary Architectures: Automated Software Governance** (2nd ed.). Sebastopol, CA: O'Reilly Media. ISBN: 978-1492097549
13. **AWS re:Invent 2022: Dr. Werner Vogels on Asynchronous Systems and Event-Driven Architecture for Global Scale** (2022). [Відеозапис].
14. Newman, S. (2021). **Building Microservices: Designing Fine-Grained Systems** (2nd ed.). Sebastopol, CA: O'Reilly Media. ISBN: 978-1492034025
15. Bass, L., Clements, P., & Kazman, R. (2021). **Software Architecture in Practice** (4th ed.). Boston, MA: Addison-Wesley.
16. Luksa, M. (2021). **Kubernetes in Action** (2nd ed.). Manning Publications.
17. Brown, S. (2021). **The C4 Model for Software Architecture**.
18. Byars, B. (2021). **Testing Microservices with Mountebank**. Packt Publishing. ISBN: 9781617494778
19. Bird, C., et al. (2021). **Machine Learning in Software Engineering**. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART)*. DOI:10.5220/0011040600003176
20. Richards, M., Ford, N., Sadalage, P., & Dehghani, Z. (2021). **Software Architecture: The Hard Parts**. Sebastopol, CA: O'Reilly Media. ISBN: 978-1-492-08689-5

III. Класичні та Фундаментальні Джерела (2020–1994)

21. Hunt, A., & Thomas, D. (2020). **The Pragmatic Programmer: Your Journey To Mastery** (2nd ed.). Boston, MA: Addison-Wesley. ISBN: 978-0135957059
22. Rosenthal, C., & Jones, N. (2020). **Chaos Engineering: System Resiliency in Practice**. Sebastopol, CA: O'Reilly Media.
23. Richards, M., & Ford, N. (2020). **Fundamentals of Software Architecture: An Engineering Approach**. Sebastopol, CA: O'Reilly Media.
24. Martin, R. C. (2019). **Чистий Код. Створення, Аналіз І Рефакторинг**. Київ: Наш Формат. (Ориг. 2008).
25. Fowler, M. (2019). **Refactoring: Improving the Design of Existing Code** (2nd ed.). Boston, MA: Addison-Wesley. ISBN: 978-0134757599
26. Humble, J., Forsgren, N., & Kim, G. (2018). **Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations**. IT Revolution Press. ISBN: 978-1942788331.
27. Martin, R. C. (2017). **Clean Architecture: A Craftsman's Guide to Software Structure and Design**. Boston, MA: Pearson. ISBN: 978-0134494166
28. Bell, M. (2017). **Service-Oriented Modeling: Service-Oriented Architecture, BPM, and Enterprise Software Development**. Hoboken, NJ: Wiley. ISBN: 978-0470141113
29. Murphy, N. R., et al. (2016). **Site Reliability Engineering: How Google Runs Production Systems**. Sebastopol, CA: O'Reilly Media. ISBN: 978-1491929124
30. Fowler, M. (2002). **Patterns of Enterprise Application Architecture**. Boston, MA: Addison-Wesley. ISBN: 978-0321127426
31. **IEEE Std 1016-2009. IEEE Standard for Information Technology – Systems and Software Engineering – Recommended Practice for Software Design Descriptions**. IEEE. ISBN: 978-0-7381-5925-6
32. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). **Design Patterns: Elements of Reusable Object-Oriented Software**. Reading, MA: Addison-Wesley. ISBN: 0-201-63361-2

9 ІНТЕРНЕТ-РЕСУРСИ

1. <https://martinfowler.com/> .
2. <https://grafana.com/docs/k6/latest/>.
3. <https://docs.nestjs.com/> .
4. <https://docs.docker.com/>.
5. <https://c4model.com/> .
6. <https://prometheus.io/docs/>.
7. <https://grafana.com/docs/>.
8. <https://plantuml.com/>.
9. <https://www.opengroup.org/certifications/archimate-program-overview>.
10. <https://developer.hashicorp.com/consul/docs>.
11. <https://jmeter.apache.org/>.
12. <https://www.uml-diagrams.org/>.

ПЕРЕЗАТВЕРДЖЕНО на 20___/20__ н.р.

Протокол засідання кафедри від «___» _____ 20__ р. № ___

Завідувач кафедри _____
(підпис) (Власне ім'я ПРИЗВИЩЕ)

ПЕРЕЗАТВЕРДЖЕНО на 20___/20__ н.р.

Протокол засідання кафедри від «____» _____ 20__ р. № ____

Завідувач кафедри _____
(підпис) (Власне ім'я ПРИЗВИЩЕ)