

Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Спеціальність **121 Інженерія програмного забезпечення**

Освітня програма «**Інженерія програмного забезпечення**»

Рівень вищої освіти **перший (бакалаврський)**

Факультет **математики та інформатики**

2024-2025 навчальний рік

Робоча програма навчальної дисципліни «Моделювання та проектування програмного забезпечення» для здобувачів першого (бакалаврського) рівня вищої освіти зі спеціальності 121 Інженерія програмного забезпечення за освітньою програмою «Інженерія програмного забезпечення»

Мова навчання: українська

Розробник: Петренко С. В., доцент кафедри інформаційних технологій та моделювання,
канд. пед. наук, доцент

Робоча програма затверджена на засіданні кафедри інформаційних технологій та моделювання.

Протокол від 27 серпня 2024 року № 8.

Завідувач кафедри



Мороз І. П.

Робочу програму схвалено навчально-методичною комісією факультету математики та інформатики.

Протокол від 3 вересня 2024 року № 7.

Голова навчально-методичної комісії



Гнедко Н. М.

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
		Обов'язкова	Обов'язкова
Кількість кредитів:	5	Рік підготовки:	
Модулів:		3	3
Змістових модулів:	2	Семестр:	
Індивідуальне науково-дослідне завдання:		6	6
Загальна кількість годин:	150	Лекції:	
Тижневих годин:		26 год.	8 год.
аудиторних годин:	3	Практичні:	
самостійної роботи студента		-	-
	6	Лабораторні:	
		24 год.	8 год.
		Самостійна робота:	
		100	134
		Індивідуальні завдання:	
		-	-
		Вид контролю:	
		Екзамен	Екзамен

Передумови для вивчення дисципліни: «Програмування», «Алгоритми і структури даних», «Об'єктно-орієнтоване програмування», «Основи інженерії програмного забезпечення»

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна «Моделювання та проектування програмного забезпечення» відноситься до обов'язкових компонентів професійної підготовки здобувачів вищої освіти першого (бакалаврського) рівня за спеціальністю 121 Інженерія програмного забезпечення. Робоча програма навчальної дисципліни складена у відповідності до освітньо-професійної програми «Інженерія програмного забезпечення» підготовки бакалаврів за названою спеціальністю.

Мета дисципліни полягає у ознайомленні здобувачів вищої освіти з сучасними інженерними підходами до моделювання та проектування надійного та якісного програмного забезпечення; формування у здобувачів вищої освіти розуміння необхідності застосування даних архітектурних принципів та набуття теоретичних знань та практичних навичок з проектування програмних продуктів.

У результаті освоєння повного курсу навчальної дисципліни «Моделювання та проектування програмного забезпечення» у здобувачів вищої освіти мають сформуватися визначені освітньою програмою **компетентності**.

Загальні компетентності

K01. Здатність до абстрактного мислення, аналізу та синтезу.

K02. Здатність застосовувати знання у практичних ситуаціях.

Фахові компетентності

K13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

К14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

К15. Здатність розробляти архітектури, модулі та компоненти програмних систем.

3. ОЧІКУВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

У результаті освоєння повного курсу навчальної дисципліни «Моделювання та проектування програмного забезпечення» у здобувачів вищої освіти формуються глибокі, міцні і системні знання, які передбачають вільне володіння понятійним апаратом, розуміння основних задач предмету, його мети та завдання, а також здатність до практичного застосування цих знань при реалізації прикладних застосувань. Згідно з освітньо-професійною програмою мають бути досягнуті наступні **програмні результати навчання**:

ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

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

ПР10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.

ПР11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.

ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПР26. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі інженерії програмного забезпечення.

Здобувачі вищої освіти повинні

знати:

- сучасні підходи при моделюванні та проектуванні ПЗ;
- можливості, переваги і недоліки підходів проектування;
- загальні шаблони архітектури програмного забезпечення;
- основні ідеї та методи аналізу та синтезу складних систем;

вміти:

- моделювати системи та процеси, стани та поведінки складних об'єктів інформатизації в процесі проектування інформаційних систем і технологій;
- аналізувати вимоги до програмного забезпечення.
- проектувати архітектуру ПЗ з використанням існуючих патернів;
- проектувати компоненти складних систем;
- обирати оптимальні рішення при проектуванні програмного забезпечення з огляду на існуючі вимоги.

4. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1. «Концепції проектування: моделювання, патерни та декомпозиція систем»

Тема 1. Ключові принципи проектування та моделювання програмного забезпечення.

Процес архітектури ПЗ. Ознаки поганого коду. Принципи проектування та практики моделювання.

Тема 2. Патерни проектування додатків. Історія виникнення патернів. Переваги та недоліки шаблонів. Класифікація патернів: породжувальні, структурні та поведінкові.

Тема 3. Основи багаторівневої архітектури. Опис підходу. Ключові поняття. Приклад багаторівневої архітектури. Аналіз загальних архітектурних характеристик.

Тема 4. Декомпозиція складних систем: підходи та рішення. Декомпозиція, як інструмент покращення. Декомпозиція за бізнес-можливостями. Розділення за піддоменами. Автономний сервіс. Сервіс на команду.

Змістовий модуль 2. «Архітектурні рішення: від патернів до безпеки та моніторингу»

Тема 5. Основні шаблони архітектури програмного забезпечення. Огляд 10 найбільш популярних архітектурних патернів. Аналіз специфіки їх застосування. Аналіз переваг та недоліків.

Тема 6. Менеджмент даних. Database per Service та Shared Database per Service. Узгодженість даних між сервісами. Сервісна команда. Шлюз API.

Тема 7. Надійність та безпека при проєктуванні ПЗ. Open Authorization. Основні принципи роботи.

Тема 8. Моніторинг та спостереження. Поняття моніторингу, спостереження та телеметрії. Види, особливості, проблеми та інструменти.

Тема 9. Сховища даних: типи та специфіка використання. Реляційні та нереляційні БД. Особливості та специфіка використання. Кейси застосування.

5. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		лек	пр	лаб	інд	с.р.		лек	пр	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Змістовий модуль 1. «Концепції проєктування: моделювання, патерни та декомпозиція систем»												
Тема 1. Ключові принципи проєктування та моделювання програмного забезпечення.	12	2		2		8	12	1				11
Тема 2. Патерни проєктування додатків.	30	6		4		20	30	1		1		28
Тема 3. Основи багаторівневої архітектури.	12	2		2		8	12	1		1		10
Тема 4. Декомпозиція складних систем: підходи та рішення.	12	2		2		8	12			1		11
Разом за змістовим модулем 1	66	12	-	10	-	44	66	3	-	3	-	60
Змістовий модуль 2. «Архітектурні рішення: від патернів до безпеки та моніторингу»												
Тема 5. Основні шаблони архітектури програмного забезпечення.	24	4		4		16	24	1		1		22
Тема 6. Менеджмент даних.	24	4		4		16	24	1		1		22

Тема 7. Надійність та безпека при проектуванні ПЗ.	12	2		2		8	12	1		1		10
Тема 8. Моніторинг та спостереження.	12	2		2		8	12	1		1		10
Тема 9. Сховища даних: типи та специфіка використання.	12	2		2		8	12	1		1		10
Разом за змістовим модулем 2	84	14	-	14	-	56	84	5	-	5	-	74
ІНДЗ												
Усього годин	150	26	-	24	-	100	150	8	-	8	-	134

6. ТЕМИ СЕМІНАРСЬКИХ ЗАНЯТЬ

Не передбачено навчальним планом

7. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Не передбачено навчальним планом

8. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з.п.	Назва теми	Кількість год. (денна ф.н.)
1.	Ключові принципи проектування та моделювання програмного забезпечення.	2
2.	Патерни проектування додатків.	4
3.	Основи багаторівневої архітектури.	2
4.	Декомпозиція складних систем: підходи та рішення.	2
5.	Основні шаблони архітектури програмного забезпечення.	4
6.	Менеджмент даних.	4
7.	Надійність та безпека при проектуванні ПЗ.	2
8.	Моніторинг та спостереження.	2
9.	Сховища даних: типи та специфіка використання.	2
	Разом	24

9. САМОСТІЙНА РОБОТА

№ з.п.	Назва теми, питання для опрацювання	Кількість год. (денна ф.н.)
1.	Ключові принципи проектування та моделювання програмного забезпечення.	8
2.	Патерни проектування додатків.	20
3.	Основи багаторівневої архітектури.	8
4.	Декомпозиція складних систем: підходи та рішення.	8
5.	Основні шаблони архітектури програмного забезпечення.	16
6.	Менеджмент даних.	16
7.	Надійність та безпека при проектуванні ПЗ.	8
8.	Моніторинг та спостереження.	8

9.	Сховища даних: типи та специфіка використання.	8
	Разом	100

10. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Не передбачено

11. МЕТОДИ НАВЧАННЯ

Під час викладання дисципліни використовуються такі методи навчання:

- МН1 – словесний метод (лекція, дискусія, обговорення досліджуваного явища чи процесу, аналіз проблемних ситуацій);
- МН2 – практичний метод (лабораторні заняття);
- МН3 – наочний метод (ілюстрації, демонстрації);
- МН4 – робота з навчально-методичною літературою (конспектування, складання рефератів);
- МН5 – інтерактивний метод (із застосуванням аудіо, відео, новітніх інформаційних технологій та комп'ютерних засобів навчання);
- МН6 – самостійна робота (самостійний аналіз, проектування та програмна реалізація завдань).

12. МЕТОДИ ОЦІНЮВАННЯ

- МО1 – екзамен;
- МО2 – усне або письмове опитування;
- МО7 – презентації результатів виконаних завдань та досліджень;
- МО9 – захист лабораторних і практичних робіт.

13. ЗАСОБИ ДІАГНОСТИКИ РЕЗУЛЬТАТІВ НАВЧАННЯ

Засобами діагностики та методами демонстрування результатів можуть бути:

- екзамен;
- аналітичні звіти;
- презентації результатів виконаних завдань та досліджень;
- лабораторні роботи.

Види та методи навчання і оцінювання

Код компетентності (за ОПШ)	Назва компетентності	Код ПРН	Назва програмного результату навчання	Методи навчання	Методи оцінювання результатів навчання
K01	Здатність до абстрактного мислення, аналізу та синтезу.	ПР07	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9
K02	Здатність застосовувати знання у практичних ситуаціях.	ПР06	Уміння вибирати та використовувати відповідну задачу методологію створення програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9

		ПР07	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9
		ПР09	Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9
К13	Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.	ПР09	Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9
		ПР10	Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проєктування.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9
К14	Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.	ПР11	Вибирати вихідні дані для проєктування, керуючись формальними методами опису вимог та моделювання.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9
		ПР12	Застосовувати на практиці ефективні підходи щодо проєктування програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9
К15	Здатність розробляти архітектури, модулі та компоненти програмних систем.	ПР12	Застосовувати на практиці ефективні підходи щодо проєктування програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9
		ПР26	Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6	МО1, МО2, МО7, МО9

14. КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Результат освітньої діяльності здобувача вищої освіти оцінюється згідно Положення про оцінювання знань і умінь здобувачів вищої освіти РДГУ за такими критеріями оцінювання та рівнями компетентності:

Суми балів за 100-бальною шкалою	Оцінка в ЄКТС	Значення оцінки ЄКТС	Критерії оцінювання	Рівень компетентності	Оцінка за національною шкалою
90-100	A	відмінно	здобувач вищої освіти виявляє особливі творчі здібності, вміє самостійно здобувати знання,	високий (творчий)	відмінно

			без допомоги викладача знаходить і опрацьовує необхідну інформацію, вміє використовувати набуті знання і вміння для прийняття рішень у нестандартних ситуаціях, переконливо аргументує відповіді, самостійно розкриває власні здібності		
82-89	B	добре	здобувач вищої освіти вільно володіє теоретичним матеріалом, застосовує його на практиці, вільно розв'язує вправи і задачі у стандартних ситуаціях, самостійно виправляє допущені помилки, кількість яких незначна	достатній (конструктивно-варіативний)	добре
74-81	C	добре	здобувач вищої освіти вміє зіставляти, узагальнювати, систематизувати інформацію під керівництвом викладача, загалом самостійно застосовувати її на практиці; контролювати власну діяльність; виправляти помилки, з-поміж яких є суттєві, добирати аргументи для підтвердження думок		
64-73	D	задовільно	здобувач вищої освіти відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, за допомогою викладача може аналізувати навчальний матеріал, виправляти помилки, з-поміж яких є значна кількість суттєвих	середній (репродуктивний)	задовільно
60-63	E	задовільно	здобувач вищої освіти володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні		
35-59	FX	незадовільно з можливістю повторного складання семестрового контролю	здобувач вищої освіти володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу	низький (рецептивно-продуктивний)	незадовільно
1-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	здобувач вищої освіти володіє матеріалом на рівні елементарного розпізнавання і відтворення окремих фактів, елементів, об'єктів	низький (рецептивно-продуктивний)	незадовільно

Підсумкова (загальна) оцінка з навчальної дисципліни є сумою рейтингових оцінок (балів), одержаних за окремі оцінювальні форми навчальної діяльності: поточне і підсумкове

оцінювання рівня засвоєння теоретичного та практичного матеріалу під час аудиторних занять і самостійної роботи; оцінка (бали) за виконання лабораторних завдань; оцінка (бали) за індивідуальну науково-дослідну роботу; оцінка (бали) за участь у наукових конференціях, олімпіадах, підготовку наукових публікацій, рефератів тощо.

15. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ ВИЩОЇ ОСВІТИ

В університеті діє накопичувальна кредитно-трансферна система оцінювання програмних результатів навчання студентів, що реалізується в ході виконання і захисту лабораторних робіт, виконання ІНДЗ та модульного контролю, для яких визначено мінімальну кількість балів, яку слід набрати для формування рейтингового балу студента та виставлення його у залікову книжку і відомість успішності студентів з відповідними оцінками за національною та європейською кредитно-трансферною системами (ЄКТС).

Розподіл балів за видами освітньої діяльності

Поточне тестування та самостійна робота									Екзамен	Сума
Змістовий модуль 1				Змістовий модуль 2						
T1	T2	T3	T4	T5	T6	T7	T8	T9		
8	10	8	10	10	10	10	8	8		
Модульний контроль – 4				Модульний контроль – 4						
40				50					10	100

T1, T2, ..., T9 – теми змістових модулів.

Оцінювання за видами діяльності:

№ з.п.	Вид навчальної діяльності	Оціночні бали	Кількість балів
T1	Виконання завдань лабораторного заняття	4	8
	Презентації результатів виконаних завдань	4	
T2	Виконання завдань лабораторного заняття	4	10
	Презентації результатів виконаних завдань	6	
T3	Виконання завдань лабораторного заняття	4	8
	Презентації результатів виконаних завдань	4	
T4	Виконання завдань лабораторного заняття	4	10
	Презентації результатів виконаних завдань	6	
Модульний контроль		4	4
T5	Виконання завдань лабораторного заняття	4	10
	Презентації результатів виконаних завдань	6	
T6	Виконання завдань лабораторного заняття	4	10
	Презентації результатів виконаних завдань	6	
T7	Виконання завдань лабораторного заняття	4	10
	Презентації результатів виконаних завдань	6	
T8	Виконання завдань лабораторного заняття	4	8
	Презентації результатів виконаних завдань	4	
T9	Виконання завдань лабораторного заняття	4	8
	Презентації результатів виконаних завдань	4	
Модульний контроль		4	4

Екзамен	10	10
Разом	100	

16. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Самостійна робота студентів над теоретичним та практичним матеріалом навчальної дисципліни здійснюється в таких формах:

- вивчення теоретичного матеріалу, що викладений на лекційних заняттях та призначеного для самостійного опрацювання;
- індивідуальне виконання навчальних завдань, розв'язування алгоритмічних задач та завдань по розробці алгоритмів та програмуванню.

В якості навчально-методичного забезпечення самостійної роботи студентів використовується основна та допоміжна література з дисципліни (підручники, навчальні посібники, монографії, словники, довідники, енциклопедії, журнали, статті у наукових виданнях), Інтернет-ресурси, матеріал лекцій, представлений у електронному вигляді та інтерактивних презентацій, методичні рекомендації для виконання лабораторних робіт.

17. ПИТАННЯ ДЛЯ ПІДГОТОВКИ ДО ПІДСУМКОВОГО КОНТРОЛЮ

1. Життєвий цикл програмного забезпечення.
2. Що таке архітектура програмного забезпечення.
3. Значимість архітектури програмного забезпечення.
4. Ітераційні кроки для основної діяльності з проектування архітектури.
5. Архітектурні стилі.
6. Зв'язність та зв'язаність.
7. Інверсія залежності (Dependency Inversion).
8. Основні принципи проектування.
9. Специфічні проблеми дизайну.
10. Комунікація між сервісами.
11. Управління винятками.
12. Повідомлення між сервісами: ендпоінти, валідація та трансформація.
13. Дизайн REST vs SOAP.
14. Етапи проектування для рівня обслуговування.
15. Рівні додатків.
16. Компоненти, модулі та функції.
17. Рівні презентації, бізнесу та даних.
18. Сервіси та рівні.
19. Монолітна архітектура.
20. Мікросервісна архітектура.
21. Порівняння монолітної та сервісної архітектури.
22. Проблеми розподіленої архітектури.
23. Декомпозиція складних систем: за бізнес можливостями та за піддоменами.
24. Декомпозиція складних систем: за сервісами та за командами.
25. Асинхронна та синхронна комунікації. Сумісність, трансформація та валідація повідомлень.
26. Шаблон Багаторівнева архітектура.
27. Шаблон Клієнт-сервер.
28. Шаблон Майстер-робітник.
29. Шаблон Брокер.
30. Шаблон Рівний до рівного.
31. Шаблон MVC.
32. База даних на сервіс.
33. Спільна база даних.
34. Сага: оркестрація та хореографія.

35. Паттерн CQRS.
36. Паттерн Пошук подій.
37. Шаблон Автоматичний вимикач.
38. Шлюз API, access token (JWT).
39. Якість програмного забезпечення.
40. Аутентифікація та авторизація.
41. Антишаблони проектування програмного забезпечення.
42. Матриця ризиків.
43. Оцінка ризиків.
44. Моделі якості ПЗ.
45. Доступність, гнучкість та безпека в моделюванні ПЗ.
46. Логування та агрегація логів.
47. Метрики додатків.
48. Аудит.
49. Розподілене трасування.
50. Трасування помилок.
51. Перевірка здоров'я API.
52. Логування змін та деплоїв.
53. Імплементация спостереження.
54. Критерії якісно спроектованого спостереження.
55. Переваги та недоліки спостереження.
56. Відмінності між моніторингом та спостереженням.
57. Функціональні вимоги до програмного забезпечення.
58. Нефункціональні вимоги до програмного забезпечення.
59. Моделювання архітектурних зразків.
60. Моделювання кооперацій.
61. Переваги та недоліки реляційних сховищ даних.
62. Сховища даних типу "ключ-значення".
63. Документо-орієнтовані сховища даних.
64. Графові сховища даних.
65. Сховища даних типу "сімейство колонок" (Column-Family Stores).
66. Моделі розподілення даних.
67. Узгодженість даних в нереляційних сховищах.
68. NoSQL vs SQL.
69. Переваги та недоліки нереляційних сховищ даних.

18. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». Полтава: ПолтНТУ, 2017. 218 с.
2. Петрик М. Р., Петрик О. Ю. Моделювання програмного забезпечення : науково-методичний посібник. Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. 200 с.
3. Роберт М. Чиста архітектура. Фабула, 2019. 416 с.
4. Табунщик Г. В., Каплієнко Т. І., Петрова О. А. Проектування та моделювання програмного забезпечення сучасних інформаційних систем. Запоріжжя: Дике Поле, 2016. 250 с.
5. Фрімен Е. Head First. Патерни проектування. Легкий для сприйняття довідник. Фабула, 2020. 672 с.
6. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, Inc., 2017. 613 p.
7. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. O'Reilly, 2020. 418 p.

8. Sommerville I. Software engineering; tenth edition. Harlow: Pearson Education Limited, 2016. 811 p.

Допоміжна:

1. Будай А. Дизайн-патерни – просто, як двері. Google Sites: веб-сайт. URL: <https://sites.google.com/site/designpatternseasy/> (дата звернення: 01.12.2024)
2. Ford N., Parsons R., Kua P. Building Evolutionary Architectures: Support Constant Change. O'Reilly Media, Inc., 2017. 190 p.
3. Freeman E., Robson E. Head First Design Patterns, 2nd Edition. O'Reilly Media, Inc., 2020. 633 p.
4. Ford N., Richards M., Sadalage P., Dehghani Z. Software Architecture: The Hard Parts. O'Reilly Media, Inc., 2021. 446 p.

19. ІНФОРМАЦІЙНІ (ІНТЕРНЕТ) РЕСУРСИ

- <https://classroom.google.com/c/NTYyODEwODg5MTEy?cjc=u6wnxud>
- <https://microservices.io/>
- <https://www.redhat.com/architect/>
- [https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff650706\(v=pandp.10\)](https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff650706(v=pandp.10))
- <https://developer.ibm.com/technologies/web-development/articles/>
- https://www.tutorialspoint.com/software_architecture_design/architecture_models.htm
- <https://martinfowler.com/architecture/>
- <https://www.geeksforgeeks.org/fundamentals-of-software-architecture/>