

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

«ЗАТВЕРДЖУЮ»

Заступник декана

з навчально-виховної роботи

Наталія ТМЄНОВА

«29» серпня 2026 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ**

для студентів

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	бакалавр
освітня програма	Інженерія програмного забезпечення

вид дисципліни	обов'язкова
----------------	--------------------

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	8
Кількість кредитів ECTS	6
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: **Васильєв О.М., професор**

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

КИЇВ – 2026

Розробник(и): **Васильєв Олексій Миколайович**, доктор фізико-математичних наук, професор кафедри програмних систем і технологій

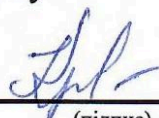
Завідувач кафедри
програмних систем і технологій


(підпис) проф. Олексій БИЧКОВ

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних технологій

Протокол № 11 від 26 червня 2026 року

Голова науково-методичної комісії 
(підпис) Ганна КРАСОВСЬКА

1. Мета дисципліни "Архітектура та проектування програмного забезпечення" полягає у формуванні у студентів навичок і знань, необхідних для визначення ефективної архітектури програмного забезпечення та вміння реалізувати таку архітектуру на практиці.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

Мати уявлення про принципи функціонування комп'ютерів, спосіб реалізації програм, загальні підходи щодо програмування.

3. Анотація навчальної дисципліни:

Навчальна дисципліна "Архітектура та проектування програмного забезпечення" є складовою циклу професійної підготовки фахівців освітньо-кваліфікаційного рівня "бакалавр". Вона належить до переліку спеціальних компетентностей випускника спеціальності "Інженерія програмного забезпечення" і надає студентам знання та уміння, необхідні для створення програмних продуктів, які мають сучасну ефективну архітектуру. Для досягнення цілей можуть використовуватись різні інструменти, технології, фреймворки, мови програмування.

Результати навчання полягають в знанні базових прийомів для визначення архітектури та базових підходів щодо створення сучасного програмного забезпечення. Методи викладання: лекції, лабораторні заняття та самостійна робота студентів. Методи оцінювання: оцінювання лабораторних робіт, контрольні роботи після основних розділів курсу, іспит. Підсумкова оцінка виставляється на основі проміжних оцінок (60%) та іспиту (40%).

4. Завдання (навчальні цілі) освоєння студентами методів використання сучасних комп'ютерних засобів, здатність студентами застосовувати знання у практичних ситуаціях для розв'язання відповідних задач. Також здатність вчитися і оволодівати сучасними знаннями з курсу "Архітектура та проектування програмного забезпечення", здатність до пошуку, оброблення та аналізу інформації з різних джерел, в тому числі, електронних ресурсів, та здатність студентів до абстрактного мислення, аналізу та синтезу матеріалу з технічних дисциплін.

Загальні компетентності

ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК-2. Здатність застосовувати знання у практичних ситуаціях.

ЗК-6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК-7. Здатність працювати в команді.

Спеціальні (фахові, предметні) компетентності

ФК-1. Здатність ідентифікувати, класифікувати та формувати вимоги до програмного забезпечення.

ФК-2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК-8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК-10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

ФК-11. Здатність реалізувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати, розуміти і застосовувати ефективні підходи щодо проектування програмного забезпечення	Лекції, лабораторні роботи, самостійна робота	Тести, звіти з лабораторних, усні опитування, контрольні роботи, залік, іспит	10
1.2	Вміння аналізувати проблеми щодо створення програмного забезпечення Мотивовано обирати мови програмування та технології для розв'язання завдань створення і програмного забезпечення			10
1.3	Мотивовано обирати мови програмування для розв'язання завдань створення і супроводження програмного забезпечення			20
2.1	Мати навички проектування програмних продуктів			20
2.2	Мати навички реалізації основних парадигм реалізації програмних продуктів			20
2.3	Мати навички тестування програмного забезпечення			20

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	1.1	1.2	1.3	2.1	2.2	2.3
Програмні результати навчання						
ПР-11. Вибирати вихідні дані для проектування, керуючись формальними методами вимог та моделювання	+		+	+		
ПР-12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення		+			+	+
ПР-25. Знати основи захисту виробничого персоналу і населення від аварій, катастроф, здійснювати моніторинг за відповідністю виробничих процесів вимогам систем охорони навколишнього середовища і безпеки життєдіяльності.	+			+		
ПР-30 Знати, розуміти, вміти застосовувати моделі математичної	+	+	+	+	+	+

теорії ігор та методи розробки алгоритмів ігрових програм із застосуванням штучного інтелекту для розробки програмного забезпечення комп'ютерних ігор, вміння застосовувати методи цифрового синтезу і обробки зорового контенту, створювати, перетворювати, оцифровувати, обробляти і виводити зображення засобами обчислювальної техніки, а також володіти інструментами та технологіями графічного дизайну, вміння творчо вирішувати композиційні, стилістичні та комерційні завдання проекту						
--	--	--	--	--	--	--

7. Схема формування оцінки

7.1. Для контролю знань студентів використовується модульно-рейтингова система. Результати навчання студентів оцінюються за 100-бальною шкалою. Семестр складається з двох змістових модулів. Підсумковий контроль (іспит) проводиться у письмовому вигляді. Максимальна оцінка за іспит – 40 балів.

У змістовий модуль 1 (ЗМ1) входять теми 1-6, у змістовий модуль 2 (ЗМ2) – теми 7-12. Обов'язковим є виконання та захист студентом лабораторних робіт, що передбачені робочою програмою навчальної дисципліни, здача двох модульних контрольних робіт за кожен семестр. При цьому сумарна мінімальна кількість балів за семестр – 36.

	ЗМ1		ЗМ2	
	Мін. – 18 балів	Макс. – 30 балів	Мін. – 18 балів	Макс. – 30 балів
Виконання та захист лабораторних робіт	12	20	12	20
Модульна контрольна робота	6	10	6	10

Для студентів, які набрали сумарно меншу кількість балів ніж критично-розрахунковий мінімум (36 балів), для одержання допуску до іспиту є Perezдача модульних контрольних робіт, виконання та захист лабораторних робіт.

Замість виконання та захисту лабораторних робіт студенти можуть запропонувати для виконання індивідуальні та колективні проекти. Умови виконання проекту (чи проектів) такі:

- Тематика та виконавці проекту попередньо узгоджуються з викладачем.
- Результати проекту мають перекривати тематику декількох лабораторних робіт (не обов'язково всіх).
- У випадку реалізації проекту він зараховується замість тих лабораторних робіт, які перекриваються результатами проекту.
- Максимальна оцінка за проект дорівнює сумарній максимальній оцінці за лабораторні роботи, які перериваються результатами проекту.
- Якщо виконавців проекту декілька, оцінка кожному з них виставляється індивідуально з урахуванням внеску в проект та підготовкою під час захисту проекту.
- Лабораторні, які не перекриваються результатами проекту, здаються в індивідуальному порядку або в рамках іншого проекту.

- семестрове оцінювання:

1. Модульна контрольна робота №1: 10 балів.

2. Модульна контрольна робота №2: 10 балів.

- підсумкове оцінювання (у формі іспиту):

- максимальна кількість балів які можуть бути отримані студентом 40 балів на іспиті за 100-бальною шкалою;
- студент не допускається до іспиту, якщо під час семестру набрав менше ніж 36 балів;
- для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою за 24 бали.

7.2. Організація оцінювання:

Упродовж семестру, після завершення лекційного матеріалу по змістовим модулям, проводиться письмова модульна контрольна робота із відкритими питаннями або у вигляді тесту.

Іспит проводиться письмово у вигляді відкритих питань або у вигляді тесту.

7.3. Шкала відповідності оцінок

За умови іспиту:

Відмінно	90-100
Добре	75-89
Задовільно	60-74
Незадовільно	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій, практичних занять та самостійних робіт

8 СЕМЕСТР

№ теми	Назва лекції	Кількість годин		
		лекції	лабораторні	Самост. робота
Змістовий модуль 1. Принципи дизайну				
1	Вступ до курсу. Основні теми курсу. Методи аналізу ефективності процесів та систем	2	2	5
2	Історія впровадження в програмування підходів із застосуванням шаблонів. Основні типи шаблонів	2	2	5
3	Основні парадигми програмування: теорія і практика	4	4	10
4	Принципи дизайну SOLID	4	4	10
5	Сумісність і зв'язність компонентів	4	2	10
6	Принципи вирішення задач проектування з використанням патернів	4	2	5
	Модульна контрольна робота 1		2	5
Змістовий модуль 2. Основні шаблони				
7	Огляд основних шаблонів. Уявлення про антишаблони	4	4	10

8	Концепція породжуючих шаблонів	4	4	10
9	Основні структури шаблони	4	4	10
10	Базові шаблони поведінки	4	2	5
11	Теорія і практика застосування шаблонів	4	2	5
12	Окремі питання реалізації архітектури програмного забезпечення: сучасний стан і перспективи	2	2	5
	Модульна контрольна робота 2		2	5
Всього		42	38	100

Загальний обсяг 180 год, в тому числі:

Лекцій – **42 год**.

Практичні заняття - **38 год**.

Самостійна робота – **100 год**.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА:

Основна

[1] Роберт Мартін. Чиста архітектура. К.: Фабула, 2020. - 368 с.

[2] Роберт Мартін. Чистий код. К.: Фабула, 2019. - 448 с.

[3] Murat Yener, Alex Theedom. Professional Java EE Design Patterns. Wrox, 2015. - 224 p.

Додаткова

[1] Бичков О.С. Основи сучасного програмування. ВПЦ “Київський університет”, 2007. - 260 с.

[2] Ерік Фрімен, Елізабет Робсон, Кеті Сьєрра, Берт Бейтс. Head First. Патерни проектування. К.: Фабула, 2020. - 672 с.

[3] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley, 2015. - 360 p.

[4] Васильєв О.М. Алгоритми. Видавництво: "Ліра-К", Київ, 2022. - 424 с.