



**МІЖНАРОДНИЙ ЄВРОПЕЙСЬКИЙ
УНІВЕРСИТЕТ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
«ЄВРОПЕЙСЬКА ШКОЛА БІЗНЕСУ»**

ЗАТВЕРДЖЕНО

Директор ННІ
«Європейська школа бізнесу»


Юлія РЕМИГА

від «11» 09 2023 р.

М.П.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Рівень вищої освіти

перший (бакалаврський)

(назва)

Спеціальність

121 Інженерія програмного забезпечення

(код і назва спеціальності)

Освітня програма

«Інженерія програмного забезпечення»

(назва освітньої програми)

Робоча програма навчальної дисципліни **«Архітектура та проектування програмного забезпечення»** складена на основі освітньо-професійної програми «Інженерія програмного забезпечення» для першого (бакалаврського рівня спеціальності 121 «Інженерія програмного забезпечення»), затвердженої Вченою радою Університету «30» травня 2023 року, протокол № 4.

Укладач програми: Шевчук Борис Вікторович – кандидат педагогічних наук, доцент

Рецензент: к.ф.м. наук, доцент Шерман З.О.

Гарант освітньої програми:  Олександр НЕСТЕРЕНКО

Робочу програму навчальної дисципліни розглянуто та схвалено на засіданні кафедри інформаційних технологій, протокол №1 від «31» серпня 2023 р.

Розглянуто і схвалено Вченою радою Навчально-наукового інституту «Європейська школа бізнесу», протокол № 1 від «11» вересня 2023 р.

ВСТУП

Програма вивчення навчальної дисципліни «Архітектура та проектування програмного забезпечення» складена відповідно до Стандарту вищої освіти України (далі – Стандарт) галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення».

Опис навчальної дисципліни (анотація). Дана навчальна дисципліна є однією з дисциплін фахової підготовки майбутніх розробників програмного забезпечення.

Таблиця 1

Найменування показників	Галузь знань, напрям підготовки, освітній рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 5	Галузь знань, 12 «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»	Нормативна	
Розділів – 2	Спеціальність: 121 «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	Рік підготовки	
Змістових розділів – 3		2023	
Індивідуальне науково-дослідне завдання:		Семестр	
		5	
		Лекції	
		22	8
Загальна кількість годин – 150		Практично-лабораторні	
		32	4
Тижневе навантаження: аудиторних – 4	Освітній рівень: бакалавр	Самостійна робота	
самостійної роботи студента – 6		96	138
		Вид контролю:	
		екзамен	екзамен

Предметом вивчення навчальної дисципліни «Архітектура та проектування програмного забезпечення» є методи та засоби проектування програмного забезпечення.

Міждисциплінарні зв'язки: «Якість програмного забезпечення та тестування», «Основи програмування», «Об'єктно-орієнтоване програмування», «Моделювання та аналіз програмного забезпечення».

1. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1.1. **Метою** навчальної дисципліни «Архітектура та проектування програмного забезпечення» є вивчення та практичне засвоєння методів та засобів проектування програмного забезпечення, вивчення шаблонів проектування, середовищ розробки і архітектур для їх застосування на процесах розробки програмних продуктів.

1.2. **Основними завданнями** вивчення навчальної дисципліни «Архітектура та проектування програмного забезпечення» є:

- оволодіння основними методами об'єктно-орієнтованого проектування програмного забезпечення, вміння застосовувати ці методи на практиці;
- набуття знання із основних існуючих архітектур, програмних засобів для реалізації, їх характеристики та особливості, а також вміння застосовувати ці особливості на практиці;
- розуміння понять архітектури програмних засобів, підходів до їх аналізу та побудови;
- застосування мови UML (Universal Modeling Language) для моделювання і проектування програмного забезпечення.

1.3. **Компетентності та результати навчання**, формуванню яких сприяє дисципліна (взаємозв'язок з нормативним змістом підготовки здобувачів вищої освіти, сформульованим у термінах результатів навчання у Стандарті).

Згідно з вимогами стандарту дисципліна забезпечує набуття студентами **компетентностей**:

Таблиця 2

Інтегральна компетентність	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, які характеризуються комплексністю і невизначеністю умов у сфері розробки програмного забезпечення.
Загальні компетентності	ЗК 02. Здатність застосовувати знання у практичних ситуаціях.
Спеціальні (фахові) компетентності	СК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. СК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування СК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення

Деталізація компетентностей відповідно до дескрипторів НРК у формі «Матриці компетентностей» наведено у табл. 3.

Таблиця 3

Матриця компетентностей

	Компетентність	Знання	Уміння / навички	Комунікація	Автономія та відповідальність
Інтегральна компетентність					
.	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, які характеризуються комплексністю і невизначеністю умов, у сфері розробки	змісту базових понять, предмету та методів курсу і основних шаблонів проектування	проводити аналіз вимог до ПЗ, що розробляється та постановку вимог до програмного забезпечення	Програмна взаємодія	Самостійне проектування і апробація на виробництві

	програмного забезпечення.				
Загальні компетентності					
	ЗК 02. Здатність застосовувати знання у практичних ситуаціях.	життєвого циклу програмного забезпечення, стадії у різних моделях життєвого циклу	оцінювати трудомісткість і вибирати адекватні підходи до розробки ПЗ	Зв'язок теоретичних і практичних знань	Спостереження процесів обробки інформації
Спеціальні (фахові, предметні) компетентності					
	СК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. СК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування СК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення	методології та технології збору основних вимог клієнтів та методи їх аналізу, основних етапів, принципів та стилів побудови складних програмних систем	проектувати архітектуру ПЗ з використанням засобів візуального моделювання та розробляти інтерфейс програмних модулів, проводити декомпозицію	Застосовування технологій паралельної роботи	Опис інформаційних процесів

Інтегративні кінцеві програмні результати навчання, формуванню яких сприяє навчальна дисципліна:

Програмні результати навчання	ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії 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.
--------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Результати навчання:

Після опанування дисципліни студент повинен

знати:

- зміст базових понять, предмету та методів курсу;
- основні шаблони проектування;
- методологію та технологію збору основних вимог клієнтів та методи їх аналізу;
- основні етапи, принципи та стилі побудови складних програмних систем;
- життєвий цикл програмного забезпечення, стадії у різних моделях життєвого циклу;
- основні архітектурні патерни та патерни проектування нижнього рівня;

уміти:

- проводити аналіз вимог до ПЗ, що розробляється;
- оцінювати трудомісткість і вибрати адекватні підходи до розробки ПЗ;
- проектувати архітектуру ПЗ з використанням засобів візуального моделювання;
- проектувати компоненти архітектурного рішення
- проводити постановку вимог до програмного забезпечення;
- приймати рішення щодо вибору архітектури програмної системи;
- розробляти інтерфейс програмних модулів, проводити декомпозицію;
- приймати рішення щодо застосування та впровадження патернів різних рівнів абстракції;
- проводити аналіз архітектури програмного забезпечення на предмет відповідності атрибутам якості.

2. ІНФОРМАЦІЙНИХ ОБСЯГ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

На вивчення навчальної дисципліни «*Архітектура та проектування програмного забезпечення*» відводиться 150 годин 5 кредитів ЄКТС.

МОДУЛЬ I. АРХІТЕКТУРА ПЗ

Змістовий модуль I. Вступ до дисципліни. Структура програмного забезпечення

Тема 1. Вступ до дисципліни «Архітектура та проектування ПЗ». Основні підходи до розробки програмного забезпечення

Змістовий модуль II. Архітектура програмного забезпечення

Тема 2. Застосування візуального моделювання з використанням мови UML в процесі розробки програмного забезпечення

Тема 3. Правила розробки та життєвий цикл програмного забезпечення. Еволюція моделей життєвого циклу ПЗ

Тема 4. Аналіз, характеристика та структура ПЗ. Архітектура програмного забезпечення

МОДУЛЬ II. ПРОЕКТУВАННЯ ПЗ

Змістовий модуль III. Проектування програмного забезпечення

Тема 5. Проектування архітектури програмного забезпечення

Тема 6. Технології проектування програмного забезпечення

Тема 7. Стратегії і методи проектування програмного забезпечення.

Стандарти та інструментальні засоби при виборі архітектури програмного забезпечення

Тема 8. Патерни в розробці програмного забезпечення

Тема 9. Архітектурні шаблони і стилі.

Тема 10. Аналіз якості та оцінка програмного дизайну. Методи аналізу архітектури

3. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№	Назви модулів і тем	Кількість годин (денна форма навчання)				
		Всього	у тому числі			
			Лекції	Практичні (семінарські)	Лабораторні	СРС
МОДУЛЬ I. АРХІТЕКТУРА ПЗ						
<i>Змістовий модуль I. Вступ до дисципліни. Структура програмного забезпечення</i>						
1.	<i>Тема 1.</i> Вступ до дисципліни «Архітектура та проектування ПЗ». Основні підходи до розробки програмного забезпечення	20	2	6	-	12
<i>Всього за змістовий модуль I:</i>		20	2	6	-	12
<i>Змістовий модуль II. Архітектура програмного забезпечення</i>						
2.	<i>Тема 2.</i> Застосування візуального моделювання з використанням мови UML в процесі розробки програмного забезпечення	18	2	6	-	10
3.	<i>Тема 3.</i> Правила розробки та життєвий цикл програмного забезпечення. Еволюція моделей життєвого циклу ПЗ	18	2	6	-	10
4.	<i>Тема 4.</i> Аналіз, характеристика та структура ПЗ. Архітектура програмного забезпечення	24	2	8	-	14
<i>Всього за змістовий модуль II:</i>		60	6	20	-	34
<i>Всього за модуль I:</i>		80	8	26	-	46
МОДУЛЬ II. ПРОЕКТУВАННЯ ПЗ						
<i>Змістовий модуль III. Проектування програмного забезпечення</i>						
5.	<i>Тема 5.</i> Проектування архітектури програмного забезпечення	10	2	2	-	6
6.	<i>Тема 6.</i> Технології проектування програмного забезпечення	14	4	2	-	8
7.	<i>Тема 7.</i> Стратегії і методи проектування програмного забезпечення. Стандарти та	10	2	2	-	6

	інструментальні засоби при виборі архітектури програмного забезпечення					
8.	<i>Тема 8.</i> Патерни в розробці програмного забезпечення	10	2	2	-	6
9.	<i>Тема 9.</i> Архітектурні шаблони і стилі.	12	2	4	-	6
10.	<i>Тема 10.</i> Аналіз якості та оцінка програмного дизайну. Методи аналізу архітектури	14	2	4	-	8
<i>Всього за змістовний модуль III:</i>		<i>70</i>	<i>14</i>	<i>16</i>	<i>-</i>	<i>40</i>
<i>Всього за модуль II:</i>		<i>70</i>	<i>14</i>	<i>16</i>	<i>-</i>	<i>40</i>
<i>Всього годин за курс:</i>		<i>150</i>	<i>22</i>	<i>42</i>	<i>-</i>	<i>86</i>

4. ТЕМИ ЛЕКЦІЙ

№ з/п	Назва модулів, тем та їх зміст
МОДУЛЬ I. АРХІТЕКТУРА ПЗ	
<i>Змістовний модуль I. Вступ до дисципліни. Структура програмного забезпечення</i>	
	<u><i>Тема 1. Вступ до дисципліни «Архітектура та проектування ПЗ». Основні підходи до розробки програмного забезпечення</i></u> <i>Зміст теми:</i> Сучасний стан сфери розроблення програмних засобів. Перспективи розвитку інформаційних технологій. Структурний та об'єктно-орієнтований принцип розроблення програмного забезпечення. Непроцедурні парадигми. Структурне програмування. Об'єктно-орієнтоване програмування. Життєвий цикл. Стратегії розроблення програмного забезпечення. Моделі життєвого циклу. Гнучке розроблення програмного забезпечення.
1.1.	
<i>Змістовний модуль II. Архітектура програмного забезпечення</i>	
	<u><i>Тема 2. Застосування візуального моделювання з використанням мови UML в процесі розробки програмного забезпечення</i></u> <i>Зміст теми:</i> Історія створення UML та її цілі. Засоби UML. Діаграми варіантів використання. Діаграми взаємодії. Діаграми послідовності. Кооперативні діаграми. Діаграми класів. Діаграми станів. Діаграми діяльності. Діаграми компонентів. Діаграма розміщення.
1.2.	
	<u><i>Тема 3. Правила розробки та життєвий цикл програмного забезпечення. Еволюція моделей життєвого циклу ПЗ</i></u> <i>Зміст теми:</i> Проблеми і правила розробки ПЗ. Розгляд технічного завдання. Поняття, процеси та моделі життєвого циклу ПЗ. Етапи життєвого циклу ПЗ. Класичний життєвий цикл. Макетування. Стратегії конструювання ПЗ. Спіральна, інкрементна та компонентно-орієнтована модель життєвого циклу. XP-процес. Швидка розробка додатків. Спіральна модель.
1.3.	
	<u><i>Тема 4. Аналіз, характеристика та структура ПЗ. Архітектура програмного забезпечення</i></u> <i>Зміст теми:</i> Вимоги до ПЗ (Software Requirements). Проектування ПЗ (software design). Конструювання ПЗ (Software Construction). Тестування ПЗ (Software Testing). Супровід ПЗ (Software maintenance). Управління конфігурацією ПЗ. Введення в архітектуру програм. Цілі вибору архітектури. Декомпозиція. Модель та зразки проектування. Використання моделей. Класифікація архітектури. Зразки
1.4.	

	проектування. Зразки проектування. Компоненти. Типи архітектури і їх моделі. Архітектури, які засновані на потоках даних. Рівневі архітектури.
МОДУЛЬ II. ПРОЕКТУВАННЯ ПЗ	
Змістовний модуль III. Проектування програмного забезпечення	
1.5.	<u>Тема 5. Проектування архітектури програмного забезпечення</u> Зміст теми: Аналіз області рішень. Архітектура ПЗ. Розробка і оцінка архітектури на основі сценаріїв. Діаграми при проектуванні архітектури ПЗ.
1.6.	<u>Тема 6. Технології проектування програмного забезпечення</u> Зміст теми: Визначення технології створення програмного забезпечення. Загальні вимоги, запропоновані до ТС ПЗ. Впровадження ТС ПЗ. Визначення потреб у ТС ПЗ. Аналіз можливостей організації. Визначення організаційних потреб та огляд ринку технологій. Визначення критеріїв успішного впровадження. Розробка стратегії впровадження ТС ПЗ. Критерії, оцінки та вибору ТС ПЗ. Виконання пілотного проектування. Характеристики пілотного проекту. Планування пілотного проекту. Особливості пілотного проекту. Вигода від використання ТС ПЗ.
1.7.	<u>Тема 7. Стратегії і методи проектування програмного забезпечення. Стандарти та інструментальні засоби при виборі архітектури програмного забезпечення</u> Зміст теми: Стратегії проектування ПЗ (Software Design Strategies). Загальні стратегії (General Strategies) Стандарт IEEE/AM81 для опису проекту. Контроль якості при виборі архітектури. Метрики для вибору архітектури. Перевірка архітектури за допомогою варіантів використання. Методи проектування ПЗ (Software Design Methods). Компонентне проектування. Функціонально-орієнтоване або структурне проектування (Function-Oriented-Structured Design). Об'єктно-орієнтоване проектування (Object-Oriented Design). Проектування на основі структур даних (Data-Structure-Centered Design). Компонентне проектування (Component-Based Design). Інструментальні засоби. IBM Rational Rose. Borland Together. Microsoft Visio. StarUML, Dia, Draw.io. Високорівневі інструментальні засоби. Низькорівневі інструментальні засоби
1.8.	<u>Тема 8. Патерни в розробці програмного забезпечення</u> Зміст теми: Вивчення та класифікація патернів. Патерни проектування в нотації мови UML. Патерн Фасад та його позначення у нотації мови UML. Патерн Спостерігач та його позначення в нотації мови UML. Шаблон проектування «Модель-подання-контролер». Модель уявлення архітектури «4+1».
1.9.	<u>Тема 9. Архітектурні шаблони і стилі</u> Зміст теми: Архітектурні шаблони і стилі. Поняття архітектурного стилю. Огляд основних архітектурних стилів. Поєднання архітектурних стилів. Архітектура клієнт/сервер. Компонентна архітектура. Проектування на основі предметної області. Багатошарова архітектура. Архітектура, заснована на шині повідомлень. N-рівнева/3-рівнева архітектура. Об'єктно-орієнтована архітектура. Сервісно-орієнтована архітектура (SOA).
1.10.	<u>Тема 10. Аналіз якості та оцінка програмного дизайну. Методи аналізу архітектури</u> Зміст теми: Атрибути якості. Аналіз якості і техніки оцінки (Quality Analysis and Evaluation Techniques). Вимірювання (Measures). Нотації проектування (Software Design Notations). Структурні описи, статичний погляд (Structural Descriptions, Static View). Поведінкові описи, динамічний погляд (Behavioral Descriptions, Dynamic

View). Метод аналізу компромісних архітектурних рішень (комплексний підхід до оцінки архітектури). Етапи методу аналізу компромісних архітектурних рішень. Метод аналізу вартості та ефективності (кількісний підхід до прийняття архітектурно-проектних рішень). Комплексне прийняття рішень. Реалізація СВАМ.

5. ТЕМИ ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

№ з/п	Теми лабораторних занять	К-сть годин
1	Збирання та аналіз вимог. Стислий опис проекту.	2
2	Планування процесу розробки програмної системи засобами Ms Project	2
3	Розроблення технічного завдання на проект системи	2
4	Проектування з використанням UML. Структурні діаграми.	2
5	Проектування з використанням UML. Діаграми поведінки.	2
6	Проектування з використанням UML. Діаграми взаємодії	2
7	Аналіз архітектури веб-додатків на основі предметної області	2
8	Аналіз архітектури веб-додатків з використання CMS	2
9	Проектування інтерфейсу веб-додатків	2
10	Аналіз структури веб-додатків.	2
11	Розгортання веб-додатків	2
12	Аналіз якості та оцінка веб-додатків	2
13	Розширення функціональності веб-додатків за допомогою модулів	2
14	Оцінка та вибір компонентів програмного забезпечення	2
15	Попереднє проектування архітектури програмного забезпечення	2
16	Модельовання та візуалізація архітектури програмного забезпечення	2
17	Аналіз архітектури програмного забезпечення	2
18	Створення та дослідження архітектурного прототипу програмного забезпечення	4
19	Реалізація та розгортання програмного забезпечення	2
20	Дослідження шаблонів проектування	2
Всього:		42

6. САМОСТІЙНА РОБОТА

№	Назва теми, що виноситься на самостійне опрацювання	К-сть годин
1.	Місце проектування в інженерії ПЗ	2
2.	Характеристики проектної діяльності	4
3.	Суттєві елементи проектування	4
4.	Характеристики слабоструктурованих задач	4
5.	Фактори, які повинні бути відображені в проектах	4
6.	Труднощі створення система на базі програмного	4
7.	Що таке модель? Учасники процесу моделювання	4
8.	Когнітивні та прагматичні, абстрактні та матеріальні моделі.	4
9.	Функції моделювання	4
10.	Вимоги до моделей	4
11.	Контекст проектування у життєвому циклі ПЗ	4
12.	Процес проектування ПЗ	4
13.	Основні етапи проектування програмного забезпечення	4
14.	Класифікація атрибутів якості ПЗ	4

15.	Ієрархічна модель якості ПЗ	4
16.	Реляційна модель якості ПЗ	4
17.	Надійність та готовність	4
18.	Здатність до супроводу та портативність	4
19.	Принцип розділу задач	4
20.	Принцип та наслідки абстракції	4
21.	Принцип та наслідки модуляризації	4
22.	Модуль та його категорії	4
Всього:		86

7. МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни *«Архітектура та проектування програмного забезпечення»* застосовуються інформаційні та практичні методи навчання, а саме: виконання та захист практичних робіт, тестування та опитування, презентації результатів опрацювання лекційних занять, опанування термінологічними поняттями засобами створення тлумачного словника, а також консультування з виконання самостійної роботи студентів.

Методи навчально-пізнавальної діяльності: пояснювально-ілюстративний метод, репродуктивний метод, метод проблемного викладу, частково-пошуковий або евристичний метод, дослідницький метод.

Методи стимулювання й мотивації навчально-пізнавальної діяльності: індуктивні і дедуктивні методи навчання, методи стимулювання і мотивації навчання.

8. МЕТОДИ КОНТРОЛЮ

Відповідно до плану вивчення дисципліни *«Архітектура та проектування програмного забезпечення»* передбачається проведення поточного та підсумкового контролю.

Основними формами проведення поточного контролю є виконання та захист лабораторних робіт. Захист лабораторних робіт представляє собою усну відповідь на запитання викладача в межах теми лабораторної роботи та надання звіту про виконану роботу. При оцінці усної відповіді враховуються: знання теоретичного матеріалу з відповідної теми; цілісність та повнота відповіді на поставлені запитання; оперування науковими визначеннями та поняттями; термінологічна та технічна грамотність відповіді; логічність та лаконічність викладу матеріалу; уміння довести свою думку; уміння супроводжувати відповідь графічними засобами.

Перевірка результатів опрацювання теоретичних питань, проводиться у формі тестових завдань або формування реферату, презентації, розробки проекту.

Контроль самостійної роботи студентів спрямований на виявлення рівня розвитку пізнавальних здібностей та творчої ініціативи студентів, самостійності, відповідальності та організованості; рівня сформованості мислення, здібностей до саморозвитку, самовдосконалення та самореалізації та опанування студентами елементів методики наукових досліджень.

9. ФОРМА ПІДСУМКОВОГО КОНТРОЛЮ УСПІШНОСТІ НАВЧАННЯ

Формою підсумкового контролю є *екзамен*, який складається очно (при особливій ситуації – у формі комп'ютерного тесту) в період призначений деканатом або за індивідуальним графіком, який затверджується навчальним планом.

10. СХЕМА НАРАХУВАННЯ ТА РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

Накопичення балів протягом семестру

№ з/п	Вид діяльності	Кількість балів за дидактичну одиницю	Кількість	Загальна кількість балів
1	Практична частина (виконання практичних робіт із захистом звіту)	2	21	42
2	Теоретична частина (контроль на лекціях та тестування)	1	11	11
3	Самостійна робота (створення тлумачного словника)	7	1	7
Максимальна кількість балів:				60

Загальна оцінка знань студентів за поточним контролем

Результати поточного контролю знань студентів в цілому оцінюються в діапазоні від **0** до **60** балів.

Студент допускається до підсумкового контролю за умови виконання вимог навчальної програми та у разі, якщо за поточну навчальну діяльність він набрав не менше **36** балів.

Підсумкове оцінювання знань студентів

Підсумкове оцінювання знань студентів проводиться у формі *екзамену*.

Розподіл балів оцінювання при підсумковому контролі з навчальної дисципліни

Оцінка в балах за поточне оцінювання	Оцінка за національною шкалою
35-40	Відмінно
21-34	Добре
10-20	Задовільно
менше 10	Незадовільно

Під час оцінювання відповіді на окреме питання під час екзамену додатково враховуються допущені недоліки та помилки, якими вважаються:

- неохайне оформлення роботи (не загальноприйняті скорочення, незрозумілий почерк, використання олівців замість чітких чорнил) (мінус 2 бали);
- неточності в назвах окремих термінів та понять (мінус 4 бали).

Критерії оцінювання відповіді на теоретичні питання:

1. Повна відповідь на питання, яка оцінюється **«відмінно»**, повинна відповідати таким вимогам:

- розгорнутий, вичерпний виклад змісту даної у питанні проблеми;
- повний перелік необхідних для розкриття змісту питання фізичних термінів та законів;
- здатність здійснювати порівняльний аналіз різних теорій, концепцій, підходів та самостійно робити логічні висновки й узагальнення;
- уміння користуватись методами наукового аналізу фізичних явищ, процесів і характеризувати їхні риси та форми виявлення;
- демонстрація здатності висловлення та аргументування власного ставлення до альтернативних поглядів на дане питання;

2. Відповідь на питання оцінюється **«добре»**, якщо:

- відносно відповіді на найвищий бал не зроблено розкриття хоча б одного з пунктів, вказаних вище (якщо він явно потрібний для вичерпного розкриття питання) або, якщо:
- при розкритті змісту питання в цілому правильно за зазначеними вимогами зроблені окремі помилки під час: використання формул.

3. Відповідь на питання оцінюється **«задовільно»**, якщо:

- відносно відповіді на найвищий бал не зроблено розкриття чотирьох чи більше пунктів, зазначених у вимогах до нього (якщо вони явно потрібні для вичерпного розкриття питання);
- одночасно присутні чотири чи більше типів недоліків, які окремо характеризують критерій оцінки питання;
- висновки, зроблені під час відповіді, не відповідають правильним чи загально визначеним при відсутності доказів супротивного аргументами, зазначеними у відповіді;
- характер відповіді дає підставу стверджувати, що особа, яка складає іспит, не зовсім правильно зрозуміла зміст питання чи не знає правильної відповіді і тому не відповіла на нього по суті, допустивши грубі помилки у змісті відповіді.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
66-73	D	задовільно	
60-65	E		
30-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1-29	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Загальна підсумкова оцінка в балах, за національною шкалою та за шкалою ECTS заноситься до заліково-екзаменаційної відомості, навчальної картки та залікової книжки студента.

11. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

- робоча навчальна програма дисципліни;
- електронний курс на платформі дистанційного навчання;
- плани лекцій, практичних занять та самостійної роботи студентів;
- тези лекцій з дисципліни;
- тестові завдання до тем лекцій;
- перелік питань до екзамену.

12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Бандура В. В. Архітектура та проектування програмного забезпечення : конспект лекцій / В. В. Бандура, Р. І. Храбатин. – Івано-Франківськ : ІФНТУНГ, 2012. – 240 с.
2. Бородкіна, І.Л. Інженерія програмного забезпечення: навч. посібник / І. Л. Бородкіна, Г. О. Бородкін ; НУБіП. — Київ : Центр учбової літ., 2020. — 204 с.
3. Кузьміна, О.О. Архітектура і програмне забезпечення ЕОМ : навч. посіб- ник / О.О. Кузьміна. - Дніпропетровськ. : РВВ ДНУ, 2002. - 64 С.
4. Кучеров, Д. П. Інженерія програмного забезпечення : навч. посіб. / Д. П. Кучеров, Є. Б. Артамонов. – Київ : НАУ, 2017. – 386 с. : іл.
5. Левус, Є.В. Життєвий цикл програмного забезпечення : навчальний посібник / Є.В. Левус, Т.А. Марусенкова, О.О. Нитребич. – Львів : Видавництво Львівської полі- техніки, 2017. - 207 С.

6. Мартін Р. Чиста архітектура: мистецтво розробки програмного забезпечення» / Роберт Мартін, Фабула, 2019. – 416 с.
7. Pattern Oriented Software Architecture: On Patterns and Pattern Languages / Buschmann F.Schmidt D.C.Henney K. - John Wiley & Sons, Inc, Vol. 5.-2007.
8. Software architecture : foundations, theory, and practice / Richard N. Taylor, Nenad Medvidović, Eric M. Dashofy. - Hoboken (N.J.) : Wiley, 2010.

Додаткова:

1. Петрик М. Р. Моделювання програмного забезпечення : науково-методичний посібник / М. Р. Петрик, О. Ю. Петрик – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с.
2. Пугачов Р. В. Системи контролю версіями : навчально-методичний посібник / Р. В. Пугачов, Н. Ю. Любченко, М. О. Соболев. – Харків: НТУ «ХП», 2019. – 130 с.
3. Табунщик Г. В., Каплієнко Т.І., Петрова О.А. Проектування та моделювання програмного забезпечення сучасних інформаційних систем / Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с.
4. Фрімен Е., Робсон Е. Патерни проектування / Е. Фрімен, Е. Робсон. –:Фабула, 2020. – 672 с.

Інформаційні ресурси:

1. Визначення архітектури додатків за допомогою Rational Software Architect. – Режим доступу: <https://www.ibm.com/developerworks/ru/library/r-define-application-architecture-rational-software-architect-2/index.html>.
2. Система дистанційного навчання Міжнародного Європейського Університету. Курс «Архітектура та проектування програмного забезпечення». Режим доступу: <https://dist.ieu.edu.ua/course/view.php?id=632>.
3. C++ Tutorial. – Режим доступу: <https://www.tutorialspoint.com/cplusplus/index.htm>.
4. Java Tutorial. – Режим доступу: <https://www.tutorialspoint.com/java/index.htm>.
5. Matplotlib – офіційна сторінка бібліотеки. – Режим доступу: <https://matplotlib.org>.
6. UML Tutorial. – Режим доступу: <https://www.tutorialspoint.com/uml/index.htm>.

