

# КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кафедра моделювання та програмного забезпечення

**ЗАТВЕРДЖУЮ**

Перший проректор

\_\_\_\_\_ Владислав ЧУБАРОВ

“ \_\_\_\_\_ ” \_\_\_\_\_ 2023 р.

## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Архітектура та проектування програмного забезпечення

(шифр і назва навчальної дисципліни)

спеціальність 121 – «Інженерія програмного забезпечення»

(шифр і назва напряму підготовки)

факультет Інформаційних технологій

(назва інституту, факультету, відділення)

Форма навчання	Курс	Семестр	Всього годин за планом	Кількість національних кредитів	Всього аудиторних годин	Аудиторних годин, (у тому числі КЗ)		Самостійна робота (год.)	Контрольні роботи (години)	Залік (сем.)	Екзамени (сем.)
						Лекції	Лабораторні				
Денна	2, 3	4, 5	300	10	102	68	34	198	2	5	4
Денна скорочена	1, 2	2, 3	300	10	102	68	34	198	2	3	4
Заочна	2, 3	4, 5	300	10	32	16	16	268	-	5	4

Кривий Ріг – 2023 рік

Робочу програму навчальної дисципліни «Архітектура та проектування програмного забезпечення» для здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Інженерія програмного забезпечення» розроблено згідно з ОПП галузі знань 12 «Інформаційні технології» зі спеціальності 121 «Інженерія програмного забезпечення».

Розробник: канд.техн.наук, доцент кафедри МПЗ Трачук А.А.

Робоча програма затверджена на засіданні кафедри моделювання та програмного забезпечення

Протокол від “ \_\_\_\_ ” \_\_\_\_\_ 2023 року № \_\_\_\_

Завідувач кафедри МПЗ, доцент, к.п.н. \_\_\_\_\_ Андрій СТРЮК

Схвалено вченою радою факультету інформаційних технологій

Протокол від “ \_\_\_\_ ” \_\_\_\_\_ 2023 року № \_\_\_\_

Голова вченої ради \_\_\_\_\_ Іван МУЗИКА

Схвалено групою забезпечення ОПП

Протокол від “ \_\_\_\_ ” \_\_\_\_\_ 2023 року № \_\_\_\_

Гарант ОПП \_\_\_\_\_ Андрій СТРЮК

## ЗМІСТ

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	4
2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	5
3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	8
4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	9
5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ.....	10
6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ.....	10
7. САМОСТІЙНА РОБОТА.....	11
8. МЕТОДИ НАВЧАННЯ.....	12
9. МЕТОДИ КОНТРОЛЮ.....	13
10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ.....	13
11. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ.....	17
12. ТЕМИ ПРОЕКТІВ.....	17
13. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ.....	18
15. ІНФОРМАЦІЙНІ РЕСУРСИ.....	18
14. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК.....	19
Додаток до робочої програми. Робочий план.....	24

## 1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни		
		денна форма навчання	заочна форма навчання	денна скорочена форма навчання
Кількість кредитів – <b>10</b>	<b>Галузь знань</b> <b>12 Інформаційні технології</b> (шифр і назва)	Нормативна		
Модулів – <b>4</b>	<b>Спеціальність</b> <b>121 Інженерія програмного забезпечення</b> (код та найменування спеціальності)	<b>Рік підготовки:</b>		
Загальна кількість годин - <b>300</b>		2,3	2,3	1,2
		<b>Семестр</b>		
		4,5	4,5	2,3
<b>Тижневих годин для денної форми навчання:</b> аудиторних – <b>3</b> самостійної роботи студента – <b>5,8</b>	Ступінь вищої освіти: <b>бакалавр</b>	<b>Лекції</b>		
		68 год.	16 год.	68 год.
		<b>Практичні, семінарські</b>		
		-	-	-
		<b>Лабораторні</b>		
		34 год.	16 год.	34 год.
		<b>Самостійна робота</b>		
		198 год.	268 год.	198 год.
<b>Вид контролю</b>				
Іспит – 4 семестр, залік – 5 семестр	Іспит – 4 семестр, залік – 5 семестр	Іспит – 2 семестр, залік – 3 семестр		

### Примітка.

Співвідношення кількості годин аудиторних занять до самостійної роботи становить:

для денної форми навчання – 0,34

для денної скороченої форми навчання – 0,34

для заочної форми навчання – 0,12

## 2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. **Метою** викладання навчальної дисципліни «Архітектура та проектування програмного забезпечення» є формування у здобувачів теоретичних знань, пов'язаних з використанням технічних і програмних засобів, що забезпечують найбільш ефективні технології проектування програмного забезпечення, у тому числі аспектів, пов'язаних з розподіленими системами і архітектурою програмного забезпечення та практичних навичок застосування проектування програмного забезпечення, шаблонів проектування, середовищ розробки і архітектури.

2.2. **Завданнями** вивчення дисципліни «Архітектура та проектування програмного забезпечення» є

- опанування теоретичних знань та практичних умінь з формування базового уявлення про галузі застосування архітектури програмного проекту;
- набуття вмінь і навичок проектування програмного забезпечення;
- опанування теоретичних і практичних основ шаблонів проектування середовищ розробки;
- оволодіння методами оцінки ризиків та якості проектування програмного забезпечення;
- дослідження існуючої архітектури проміжного програмного забезпечення;
- ознайомлення з проектуванням розподілених систем з використанням проміжного програмного забезпечення.

Відповідно до освітньої програми дисципліна забезпечує наступні **компетентності**:

### *Загальні компетентності*

ЗК01 Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02 Здатність застосовувати знання у практичних ситуаціях.

ЗК05 Здатність вчитися і оволодівати сучасними знаннями.

ЗК06 Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

### *Фахові компетентності*

СК01 Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

СК02 Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

СК03 Здатність розробляти архітектури, модулі та компоненти програмних систем.

СК05 Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

СК07 Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

СК11 Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

СК12 Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

СК13 Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

СК14 Здатність до алгоритмічного та логічного мислення.

**Програмні результати навчання** освітньої програми, яким відповідає дисципліна:

ПР01 Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПР03 Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПР04 Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПР05 Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

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.

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

ПР13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПР14 Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПР15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПР16 Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.

ПР17 Вміти застосовувати методи компонентної розробки програмного забезпечення.

ПР18 Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

ПР23 Вміти документувати та презентувати результати розробки програмного забезпечення.

В результаті вивчення дисципліни здобувачі повинні **знати:**

- базові методи створення архітектури проекту;
- принципи архітектури в проектуванні програмного забезпечення;
- базові шаблони проектування програмного забезпечення;
- методи зворотної інженерії для відновлення дизайну програмного забезпечення;
- принципи компонентного проектування;
- базові принципи проектування з урахуванням таких якостей, як продуктивність, безпека, захищеність, можливість повторного використання, надійність.

**вміти:**

- вибирати вихідні дані для проектування, керуючись формальними методами опису вимог;
- аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;
- застосовувати різні шаблони проектування та середовища розробки;
- проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування;
- проектувати і реалізовувати програмне забезпечення, використовуючи різні технології проміжного програмного забезпечення;
- модифікувати проекти, використовуючи продумані підходи до управління змінами;
- вміти застосовувати методи компонентної розробки програмного забезпечення.

#### 2.4. Міждисциплінарні зв'язки

При вивченні дисципліни використовуються знання здобувачів з дисциплін «Основи інженерії програмного забезпечення», «Організація комп'ютерних мереж», «Алгоритми та структури даних», «Об'єктно-орієнтоване програмування».

Знання, одержані здобувачами при вивченні дисципліни, використовуються при вивченні дисциплін «Сучасні технології Internet-програмування» з курсовою роботою, «Програмна інженерія розподілених інтернет-застосувань» з курсовою роботою, «Програмування мобільних пристроїв».

Вимоги до знань та умінь визначаються галузевими стандартами вищої освіти України.

### 3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

#### **Змістовий модуль 1. Проектування програмного забезпечення.**

**Тема 1.** Основні поняття проектування програмного забезпечення. Цілі проектування ПО. Місце проектування ПО в життєвому циклі ПЗ. Послідовність проектування ПО.

**Тема 2.** Постійні дані, управління пам'яттю, виняткові ситуації. Принципи проектування. Приховування інформації, зчеплення і зв'язність. Кризис програмування. Розвиток та історія проектування програмного забезпечення.

**Тема 3.** Проектування архітектури. Збільшене проектування ПО. Формування архітектури програмної системи. Оцінка якості архітектури ПС.

**Тема 4.** Аналіз вимог. Класи аналізу. Взаємодія класів без урахування станів. взаємодія класів з урахуванням станів. Діаграми станів (кінцеві автомати).

#### **Змістовий модуль 2. Стратегії проектування.**

**Тема 5.** Архітектурні стилі проектування програмного забезпечення. Архітектурні стилі, шаблони проектування, паттерни, повторне використання.

**Тема 6.** Функціонально орієнтоване проектування. Поняття та структура системи програмування.

**Тема 7.** Принципи функціонування систем програмування. Бібліотеки підпрограм.

**Тема 8.** Питання апаратного забезпечення в програмній архітектурі. Програмно-апаратні засоби для реалізації методу поелементного діагностування.

#### **Змістовий модуль 3. Об'єктно-орієнтоване проектування.**

**Тема 9.** Завдання і особливості об'єктно-орієнтованого проектування програмних засобів. Аспектно - орієнтоване проектування.

**Тема 10.** Технології реалізації компонентів реалізації. Родове програмування. Архітектурне проектування.

**Тема 11.** Використання компонент при проектуванні ПО. Проектування в конкретних класах і проектування в інтерфейсах. Проектування компонентів ПО. Принципи проектування компонентів.

**Тема 12.** Класи проектування. Уточнення класів аналізу. проектування взаємодії класів.

#### **Змістовий модуль 4. Детальне проектування.**

**Тема 13.** Уніфікована мова моделювання UML. Використання моделювання в проектуванні програмного забезпечення. Основні діаграми. Недоліки мови UML.

**Тема 14.** Проектування за умов повторного використання компонентів. Формальний аналіз проектування. Методи оцінки і вимірювання ПЗ. Аналіз ризиків. Забезпечення якості ПЗ. Управління конфігурацією програмного продукту. Конкретні методи проектування. SSA/SD, JSD, OOD. Інструментальні засоби управління проектом.

**Тема 15.** Компонентне проектування. Проектування інтерфейсу компонент і системи. Нотації проектування. Засоби підтримки проектування. Вимір атрибутів проектування. Метрики проектування.

**Тема 16.** Паттерни проектування. Поняття патерну проектування. Типи патернів проектування. Основні патерни проектування.

#### 4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	денна / денна скорочена форма				заочна форма			
	разом	у тому числі			разом	у тому числі		
		лекц.	лаб.	с.р.		лекц.	лаб.	с.р.
1	2	3	4	5	6	7	8	9
<b>Змістовий модуль 1 Проектування програмного забезпечення</b>								
Тема 1. Основні поняття ППЗ	19	4	2	13	17	-	-	17
Тема 2. Принципи проектування	19	4	2	13	18	2	2	14
Тема 3. Проектування архітектури	16	4	2	10	18	2	2	14
Тема 4. Аналіз вимог	16	4	2	10	17	-	-	17
<b>Разом за змістовим модулем 1</b>	<b>70</b>	<b>16</b>	<b>8</b>	<b>46</b>	<b>70</b>	<b>4</b>	<b>4</b>	<b>62</b>
<b>Змістовий модуль 2 Стратегії проектування</b>								
Тема 5. Архітектурні стилі	19	4	2	13	20	2	2	16
Тема 6. Функціонально орієнтоване проектування	19	4	2	13	20	-	-	20
Тема 7. Бібліотеки підпрограм	21	4	2	15	20	-	-	20
Тема 8. Апаратне забезпечення в програмній архітектурі	21	6	3	12	20	2	2	16
<b>Разом за змістовим модулем 2</b>	<b>80</b>	<b>18</b>	<b>9</b>	<b>53</b>	<b>80</b>	<b>4</b>	<b>4</b>	<b>72</b>
<b>Разом за семестр 1</b>	<b>150</b>	<b>34</b>	<b>17</b>	<b>99</b>	<b>150</b>	<b>8</b>	<b>8</b>	<b>134</b>
<b>Змістовий модуль 3 Об'єктно-орієнтоване проектування</b>								
Тема 9. Аспектно - орієнтоване проектування	19	4	2	13	17	-	-	17
Тема 10. Технології реалізації компонентів реалізації	19	4	2	13	18	2	2	14
Тема 11. Використання компонент при проектуванні ПО	16	4	2	10	18	2	2	14
Тема 12. Класи проектування	16	4	2	10	17	-	-	17
<b>Разом за змістовим модулем 3</b>	<b>70</b>	<b>16</b>	<b>8</b>	<b>46</b>	<b>70</b>	<b>4</b>	<b>4</b>	<b>62</b>
<b>Змістовий модуль 4 Стратегії проектування</b>								
Тема 13. Уніфікована мова моделювання UML	19	4	2	13	20	2	2	16

Тема 14. Проектування за умов повторного використання компонентів	19	4	2	13	20	-	-	20
Тема 15. Компонентне проектування	21	4	2	15	20	-	-	20
Тема 16. Паттерни проектування	21	6	3	12	20	2	2	16
<b>Разом за змістовим модулем 4</b>	<b>80</b>	<b>18</b>	<b>9</b>	<b>53</b>	<b>80</b>	<b>4</b>	<b>4</b>	<b>72</b>
<b>Разом за семестр 2</b>	<b>150</b>	<b>34</b>	<b>17</b>	<b>99</b>	<b>150</b>	<b>8</b>	<b>8</b>	<b>134</b>
<b>Разом</b>	<b>300</b>	<b>68</b>	<b>34</b>	<b>198</b>	<b>300</b>	<b>16</b>	<b>16</b>	<b>268</b>

## 5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Не передбачено.

## 6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин		
		ден на	заочна	Ден на скорочена
1.	Лабораторна робота №1. Принципи проектування.	2	1	2
2.	Лабораторна робота №2. Робота з використанням UML	2	1	2
3.	Лабораторна робота №3. Складання функціональних діаграм	2	1	2
4.	Лабораторна робота №4. Приховування інформації, зчеплення і зв'язність.	2	1	2
5.	Лабораторна робота №5. Проектування атрибутів якості	2	1	2
6.	Лабораторна робота №6. Проектування компонентів	2	1	2
7.	Лабораторна робота №7. Функціонально орієнтоване проектування.	2	1	2
8.	Лабораторна робота №8. Аспектно - орієнтоване проектування.	3	1	3
9.	Лабораторна робота №9. Проектування інтерфейсу компонент і системи.	2	1	2
10.	Лабораторна робота №10. Архітектура для заданої наочної області і лінійки продуктів	2	1	2
11.	Лабораторна робота №11. Використання режимів і навігації.	2	1	2
12.	Лабораторна робота №12. Локалізація і інтернаціоналізм.	2	1	2
13.	Лабораторна робота №13. Час відгуку і зворотний зв'язок.	2	1	2
14.	Лабораторна робота №14. Шаблони проектування.	2	1	2
15.	Лабораторна робота №15. Об'єктно-орієнтоване проектування .	2	1	2
16.	Лабораторна робота №16. Засоби підтримки проектування.	3	1	3
<b>Усього годин</b>		<b>34</b>	<b>16</b>	<b>34</b>

## 7. САМОСТІЙНА РОБОТА

На самостійну роботу здобувачам денної форми навчання відведено 198 годин, заочної - 268 годин.

Самостійна робота здобувачів при вивченні дисципліни «Архітектура та проектування програмного забезпечення» залучає такі складові:

- опрацювання лекційного матеріалу з кожної теми;
- опрацювання додаткової літератури по темі;
- вивчення окремих тем або питань, що передбачені для самостійного опрацювання;
- підготовка до виконання, а також до захисту лабораторних робіт;
- підготовка до проведення контрольних заходів.

### Розподіл годин самостійної роботи

№ з/п	Назва теми	Кількість годин		
		денна	заочна	Денна скорочена
1.	Історія виникнення та розвиток АППО	13	17	13
2.	Етапи проектування ПЗ	13	14	13
3.	Дані і знання, які можуть використовуватись у АППО	10	14	10
4.	Метод структурного аналізу	10	17	10
<b>Разом за змістовим модулем 1:</b>		<b>46</b>	<b>62</b>	<b>46</b>
5.	Паттерни проектування	13	16	13
6.	Мова проектування UML	13	20	13
7.	Проектування інтерфейсу компонент	15	20	15
8.	Засоби підтримки проектування.	12	16	12
<b>Разом за змістовим модулем 2:</b>		<b>53</b>	<b>72</b>	<b>53</b>
<b>Разом за семестр 1</b>		<b>99</b>	<b>134</b>	<b>99</b>
9.	Аспектно - орієнтоване проектування.	13	17	13
10.	Архітектура для заданої наочної області і лінійки продуктів	13	14	13
11.	Структуризація системи	10	14	10
12.	Модель сховища даних;	10	17	10
<b>Разом за змістовим модулем 3:</b>		<b>46</b>	<b>62</b>	<b>46</b>
13.	Модель клієнт-сервер;	13	16	13
14.	Графи потоків даних	13	20	13
15.	Динамічні структури даних	15	20	15

№ з/п	Назва теми	Кількість годин		
		ден на	заочна	Денна а скорочена
16.	Міри зчеплення модулів	12	16	12
<b>Разом за змістовим модулем 4:</b>		<b>53</b>	<b>72</b>	<b>53</b>
<b>Разом за семестр 2</b>		<b>99</b>	<b>134</b>	<b>99</b>
<b>Разом:</b>		<b>198</b>	<b>268</b>	<b>198</b>

## 8. МЕТОДИ НАВЧАННЯ

Використовуються наступні методи навчання: лекції, лабораторні заняття, самостійна робота.

Навчальна лекція – це логічне, послідовне викладання змісту навчання, яке характеризується судженнями, висновками, підсумком. Вона охоплює основний теоретичний матеріал однієї або кількох тем навчальної дисципліни. Призначенням лекції є формування у здобувачів фундаментальних знань з дисципліни, а також визначає основний зміст і характер усіх інших навчальних занять та самостійної роботи здобувачів із цієї дисципліни.

Лабораторне заняття - форма організації навчання, яку проводять за завданням і під керівництвом НПП. Основні дидактичні цілі – експериментальне підтвердження вивчених теоретичних положень навчальної дисципліни та формування вмінь й навичок їх практичного застосування. Проведення лабораторного заняття ґрунтується на попередньо підготовлених наборах завдань різної складності для розв’язання на занятті. Лабораторне заняття проводиться у навчальних лабораторіях з використанням пристосованого до умов навчального процесу устаткування.

Самостійна робота здобувача є основним способом оволодіння навчальним матеріалом у час, вільний від обов’язкових аудиторних занять. Мета виконання самостійної роботи – поглиблення, узагальнення й закріплення теоретичних знань і практичних умінь здобувачів із дисципліни шляхом вироблення вміння самостійної роботи з навчальною і фаховою літературою та інформацією в мережі Інтернет.

Самостійна робота здобувачів здійснюється у формі: підготовки до лекцій і лабораторних занять, виконанні самостійних проектів. Самостійну роботу здобувач може виконувати у бібліотеці, комп’ютерних класах, а також у домашніх умовах.

Підготовка до лекцій передбачає самостійне опрацювання теоретичного матеріалу. При цьому необхідно звернути увагу на необхідність чіткого засвоєння основних термінів та визначень, розуміння їх змісту, обов’язкового аналізу використання теоретичних положень для розв’язання наданих прикладів.

Самоперевірку засвоєння навчального матеріалу здобувач здійснює за контрольними запитаннями, що надано після кожної теми у конспекті лекцій та іншій літературі, та після кожного лабораторного заняття у відповідних методичних вказівках. Якщо на деякі запитання здобувач не може надати відповіді, то необхідно повторити вивчення навчального матеріалу, або визначити правильну відповідь з викладачем на консультації.

Під час вивчення даної дисципліни використовуються:

– мультимедійні освітні технології: інтерактивні лекції (презентації) із використанням програми MS Power Point у поєднанні з анімацією та звуковим супроводом; перегляд відеороликів за окремими пунктами тем занять, використання електронних посібників;

– діалогові технології: організація групових обговорень, використання «мозкового штурму».

Лекції проводяться з використанням технічних засобів навчання й супроводжуються демонстрацією презентацій за допомогою проектора.

У разі виникнення необхідності забезпечення навчального процесу в дистанційному режимі супровід та контроль знань реалізовується за допомогою дистанційного курсу, розробленого в Google Classroom. Онлайн лекції, консультації та усні відповіді на питання, захист проєктів проводиться за допомогою Google Meet або Zoom.

## 9. МЕТОДИ КОНТРОЛЮ

Основними завданнями контролю знань здобувачів вищої освіти з дисципліни є оцінювання засвоєння теоретичних знань і практичних навичок, отриманих під час навчання.

Контрольні заходи мають виконувати наступні функції:

- стимулювати систематичну самостійну роботу над навчальним матеріалом;
- забезпечувати закріплення та реалізацію набутих теоретичних знань при підготовці до практичних занять;
- прищеплювати навички відповідального ставлення до своїх обов'язків, самостійного цілеспрямованого пошуку потрібної інформації, чіткої організації свого робочого дня.

Оцінювання знань здобувачів складається з поточного та підсумкового контролю.

Поточний контроль знань здобувачів вищої освіти передбачає оцінювання за наступними основними напрямками:

- перевірка теоретичних знань;
- перевірка підготовки до лабораторних занять;
- перевірка виконання індивідуального проєкту.

З даних компонентів складаються загальні бали, які фіксуються в журналі викладача.

Оцінювання рівня засвоєння теоретичних знань здобувачів вищої освіти проводиться під час усної співбесіди зі здобувачами по теоретичним матеріалам, за результатами захисту проєкту й виконання самостійних робіт. Підсумковим контролем є залік.

## 10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ

Використовується модульно-рейтингова система оцінювання, яка передбачає розподіл балів за виконання всіх запланованих видів робіт. При цьому максимальна кількість балів за модуль при умові його бездоганного виконання дорівнює 100. Ця сума складається з балів, що накопичив студент у ході поточного контролю, та балів за виконання індивідуального проекту.

Успішність студентів-заочників оцінюється аналогічно.

Лабораторні роботи у модулі відображують оволодіння навичками та вміння застосовувати знання на практиці і сукупно відповідають 55-ти відсоткам ваги. При зниженні якості виконання тієї чи іншої лабораторної роботи, знижується і кількість балів, якою вона оцінюється.

### Розподіл балів за видами робіт

№ модуля	№ зан.	Вид роботи	Тема	Максимальна кількість балів		
				Денна\ Денна скороч	заочна	
1,2	1	Лабораторна робота № 1	Принципи проектування.	10	-	
	2	Лабораторна робота № 2	Робота з використанням UML	10	5	
	3	Лабораторна робота № 3	Складання функціональних діаграм	10	5	
	4	Лабораторна робота № 4	Приховування інформації, зчеплення і зв'язність.	10	-	
	<b>Контрольно-модульна робота</b>				<b>10</b>	<b>40</b>
	<b>Разом по модулю 1</b>				<b>50</b>	<b>50</b>
	5	Лабораторна робота № 5	Проектування атрибутів якості	10	5	
	6	Лабораторна робота № 6	Проектування компонентів	10	-	
	7	Лабораторна робота № 7	Функціонально орієнтоване проектування.	10	-	
	8	Лабораторна робота № 8	Аспектно - орієнтоване проектування.	10	5	
	<b>Контрольно-модульна робота</b>				<b>10</b>	<b>40</b>
	<b>Разом по модулю 2</b>				<b>50</b>	
	<b>Разом за семестр</b>				<b>100</b>	

№ модуля	№ зан.	Вид роботи	Тема	Максимальна кількість балів	
				Денна\ Денна скороч	заочн а
3,4	9	Лабораторна робота № 9	Проектування інтерфейсу компонент і системи.	10	-
	10	Лабораторна робота № 10	Архітектура для заданої наочної області і лінійки продуктів	10	5
	11	Лабораторна робота № 11	Використання режимів і навігації.	10	5
	12	Лабораторна робота № 12	Локалізація і інтернаціоналізм.	10	-
	<b>Контрольно-модульна робота</b>			<b>10</b>	<b>40</b>
	<b>Разом по модулю 3</b>			<b>50</b>	50
	13	Лабораторна робота № 13	Час відгуку і зворотний зв'язок.	10	5
	14	Лабораторна робота № 14	Шаблони проектування.	10	-
	15	Лабораторна робота № 15	Об'єктно-орієнтоване проектування .	10	-
	16	Лабораторна робота № 16	Засоби підтримки проектування.	10	5
	<b>Контрольно-модульна робота</b>			<b>10</b>	<b>40</b>
	<b>Разом по модулю 4</b>			<b>50</b>	
	<b>Разом за семестр</b>				<b>100</b>

Оцінювання кожної лабораторної роботи та індивідуального завдання ведеться за показниками, наведеними в таблиці:

№ з/п /Критерій оцінювання	Максимальна кількість балів (денна форма/заочна форма)				
	Своєчасність в виконання	Своєчасність в захисту	Якість рішень	Додаткові завдання	Всього за роботу
Лабораторна робота № 1	1/-	1/-	3/5	-	5/5
Лабораторна робота № 2	1/-	1/-	3/5	-	5/5
Лабораторна робота № 3	1/-	1/-	3/5	-	5/5

Лабораторна робота № 4	1/-	1/-	3/5	-	5/5
Лабораторна робота № 5	1/-	1/-	3/5	1	6/6
Лабораторна робота № 6	1/-	1/-	3/5	1	6/6
Лабораторна робота № 7	1/-	1/-	3/5	-	5/5
Лабораторна робота № 8	1/-	1/-	3/5	-	5/5
Лабораторна робота № 9	1/-	1/-	3/5	1	6/6
Лабораторна робота № 10	1/-	1/-	3/5	-	5/5
Лабораторна робота № 11	1/-	1/-	3/5	1	6/6
Лабораторна робота № 12	1/-	1/-	3/5	1	6/6
Лабораторна робота № 13	1/-	1/-	3/5	-	5/5
Лабораторна робота № 14	1/-	1/-	3/5	-	5/5
Лабораторна робота № 15	1/-	1/-	3/5	-	5/5
Лабораторна робота № 16	1/-	1/-	3/5	-	5/5
Всього	16/-	16/-	48/90	5	100/100

Під своєчасністю практичного виконання та своєчасністю захисту лабораторної роботи розуміється виконання та захист у тиждень згідно із графіком робіт.

Якість знайдених студентом рішень (ефективність алгоритму, доречність використання структур даних, елементів інтерфейсу, тощо) оцінюється наступним чином:

- робота виконана без зауважень - максимальний бал;
- робота виконана достатньо повно з деякими зауваженнями –дві третини від максимального бала;
- робота виконана не повністю – одна третина від максимального бала;
- при перевірці роботи виявлені грубі помилки – 0 балів.

Лабораторні роботи №№5, 6, 9, 11,12 містять додаткові завдання підвищеної складності, за якісне виконання яких можна отримати відповідну кількість балів.

Оцінка всієї роботи знаходиться підсумовуванням балів за кожний з показників.

Для допуску до підсумкового контролю студент повинен виконати графік навчального процесу, усі види запланованих завдань і протягом семестру отримати в сумі не менше 50 балів.

Семестровий контроль здійснюється у формі іспиту в четвертому семестрі для денної та заочної форм навчання, у другому семестрі для денної скороченої форми навчання, залік в п'ятому семестрі для денної та заочної форм навчання, у третьому семестрі для денної скороченої форми навчання.

У разі виконання студентом усіх видів поточних контрольних заходів залік виставляється студенту на підставі зарахованих балів протягом семестру. Результати заліку оцінюються за 100-бальною шкалою. У відомість оцінка проставляється як у балах національної шкали, так і за шкалою ECTS:

При наявності у здобувачів **результатів неформального навчання** за освітнім компонентом «Архітектура та проектування програмного забезпечення» у повному обсязі, визнання та оцінювання результатів здійснюється відповідно до «Положення про порядок визнання у Криворізькому національному університеті результатів навчання, отриманих в умовах неформальної освіти». У випадку, якщо за підсумками визнання результатів неформального навчання визнається тільки частина результатів навчання, заявнику зараховуються окремі види навчальної роботи за освітнім компонентом «Архітектура та проектування програмного забезпечення».

Нижче наведені окремі види навчальної роботи, які можуть бути зараховані здобувачеві при наявності сертифікату про успішне проходження рекомендованих онлайн курсів.

Тема	Посилання на рекомендовані курси
Функціонально орієнтоване проектування	<a href="https://courses.prometheus.org.ua/courses/course-v1:LITS+ITPM101+FREE_2021_T1/about">https://courses.prometheus.org.ua/courses/course-v1:LITS+ITPM101+FREE_2021_T1/about</a>
Стратегії проектування	<a href="https://courses.prometheus.org.ua/courses/course-v1:IRF+DV101+2016_T3/about">https://courses.prometheus.org.ua/courses/course-v1:IRF+DV101+2016_T3/about</a>
Функціонально орієнтоване проектування	<a href="https://prometheus.org.ua/prometheus-plus/pm-lits-course/">https://prometheus.org.ua/prometheus-plus/pm-lits-course/</a>
Патерни проектування	<a href="https://training.epam.ua/Training/Details/3508?lang=ua">https://training.epam.ua/Training/Details/3508?lang=ua</a>
Технології реалізації компонентів реалізації	<a href="https://training.epam.ua/Training/Details/3508?lang=ua">https://training.epam.ua/Training/Details/3508?lang=ua</a>
Використання компонент при проектуванні ПЗ	<a href="https://training.epam.ua/Training/Details/3508?lang=ua">https://training.epam.ua/Training/Details/3508?lang=ua</a>
Проектування за умов повторного використання компонентів	<a href="https://training.epam.ua/Training/Details/3508?lang=ua">https://training.epam.ua/Training/Details/3508?lang=ua</a>
Компонентне проектування	<a href="https://training.epam.ua/Training/Details/3508?lang=ua">https://training.epam.ua/Training/Details/3508?lang=ua</a>

### Шкала оцінювання

Національна шкала успішності	Оцінка ECTS	Визначення ECTS	100-бальна система оцінювання
відмінно/ зараховано	<b>A</b>	ВІДМІННО - відмінне виконання лише з незначними помилками	90...100
добре/ зараховано	<b>B</b>	ДУЖЕ ДОБРЕ - вище середнього рівня з кількома помилками	80...89

	<b>C</b>	ДОБРЕ - у цілому правильно робота з певною кількістю помилок і недоліків	71...79
задовільно/ зараховано	<b>D</b>	ЗАДОВІЛЬНО - непогано, але зі значною кількістю грубих помилок	61...70
	<b>E</b>	ДОСТАТНЬО - виконання задовольняє мінімальні потреби	50...60
незадовільно/ не зараховано	<b>FX</b>	НЕЗАДОВІЛЬНО - із можливістю повторного складання	30...49
	<b>F</b>	НЕЗАДОВІЛЬНО - з обов'язковим повторним вивчення дисципліни	0...29

## 11 ІНСТРУМЕНТИ, ОБЛАДНАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

- Аудиторія персональних комп'ютерів класу Celeron, AMD, Pentium, Core 2 Duo, Core i5, i7 (або вище) з операційною системою типу Windows 7, 8 або 10. Забезпеченість комп'ютерами – 12 шт. на 25 студентів.
- Підключення до мережі Інтернет.
- Програмне забезпечення: Rational Rose, Adobe Dreamweaver CS5, Visual Studio Code.

## 12. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ

1. Основні етапи розвитку архітектури ПЗ
2. Структурний і об'єктний підхід до програмування. Основні причини «кризи програмування» і шляхи виходу з неї.
3. Компонентний підхід і CASE-технології, технологія СОМ. Технологія створення розподілених додатків СОRВА.
4. Життєвий цикл (ЖЦ) програмного продукту. Моделі ЖЦ ПС.
5. Методи проектування зверху-вниз і знизу-вгору. Гідності й недоліки.
6. Каскадна і спіральна моделі ЖЦ, випадки застосування, переваги і недоліки.
7. Головні ризики програмних проектів та способи реагування
8. Суть процесу макетування програмного забезпечення
9. Основні принципи побудови моделей складних систем
10. Сутність аналізу вимог. Процес декомпозиції системи
11. Мета і завдання планування процесу розробки. Чотири основні фази розробки програмного продукту
12. Діаграма потоків даних і її основні об'єкти.
13. Принцип декомпозиції при побудові DFD діаграм. Словник термінів
14. Діаграми «сутність-зв'язок». Варіанти нотації при побудові діаграми "сутність-зв'язок". Суті. Типи зв'язків.
15. Метод покрокової деталізації. Функціональні діаграми. Варіанти впливів блоків в функціональних діаграмах.
16. Методологія SADT. Принципи.
17. CASE-технології і CASE-засоби
18. Цілі і зміст етапів ЖЦ і розподіл трудовитрат при класичній розробці і розробці за допомогою CASE-технології.
19. Клієнт-серверна архітектура.
20. Модель сховища даних, модель клієнт-сервер.
21. Трирівнева модель, модель абстрактної машини.
22. Різновиди моделей централізованого управління.
23. Моделі подієвого управління.
24. Уніфікована мова моделювання. Принципи.
25. Ставлення залежності. Ставлення асоціації. кратність асоціації
26. Ставлення агрегації. Ставлення композиції. Ставлення узагальнення.
27. Види діаграм UML, які використовуються для опису архітектури ПС. Статичні і динамічні діаграми UML
28. Діаграма варіантів використання. Відносини в діаграмах варіантів
29. Діаграма об'єктів
30. Діаграма класів.

## 13. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ

### 13.1 Навчальна та довідкова література

1. Тернер М. Основи Microsoft Solution Framework. - М., 2018. - 336 с.
2. Хандхаузен Р. Microsoft Visual Studio 2005 Team System/Пер. з англ. - М.: Видавничо-торговельний будинок, 2016. - 416 с.
3. Матвєєва Л.С., Волков В.А. Процес розробки програмного забезпечення. Від теорії до практики. – К., 2018. – 117 с.
4. Гамма Е., Хелм Р., Джонсон Р. Прийоми об'єктно-орієнтованого проектування. Патерни проектування, будь-яке видання.
5. R. Pressman. Software Engineering: A Practitioner's Approach, 6th Ed. - McGraw Hill, 2015. - 256 с. Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development / «Pearson», 2017. –736 с.
6. Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. «Pearson», 2017. — 464 с.

### 13.2. Методична література

1. Конспект лекцій з курсу "Архітектура та проектування програмного забезпечення" [Текст] : Метод. посібн. / Уклад. Трачук А.А., Кривий Ріг, КНУ, 2020 р. — 152 с.
2. Методичні вказівки до виконання лабораторних робіт з дисципліни «Архітектура та проектування програмного забезпечення» для усіх форм навчання спеціальності 121 «Інженерія програмного забезпечення». / Уклад. Трачук А.А– Кривий Ріг, КНУ, 2019 р. — 76 с.

## 14. ІНФОРМАЦІЙНІ РЕСУРСИ

До складу інформаційних ресурсів навчальної дисципліни входять:

1. Бібліотека Криворізького національного університету (м. Кривий Ріг, вул. Пушкіна, 37). – Режим доступу: <http://lib.knu.edu.ua/>

Internet-ресурси:

2. Google Classroom. [Електронний ресурс] – Режим доступу: <https://classroom.google.com/u/0/c/NTg3NzQ5NjEyMzIy>
3. .NET Self-paced [Електронний ресурс] – Режим доступу: <https://training.epam.ua/ua/training/3508>
4. Adobe Dreamweaver CS5 [Електронний ресурс] – Режим доступу: <https://helpx.adobe.com/ua/dreamweaver/user-guide.html>.
5. Rational Rose [Електронний ресурс] – Режим доступу: <http://cmcstuff.esyr.org/vmkbotva-r15/>.

## 15. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК

**Natural Language Processing:** The ability of a computer system to understand, interpret, and generate human language.

**Expert Systems:** Computer systems that utilize expert knowledge to solve complex problems in specific domains.

**Intelligent Information System:** A system that utilizes artificial intelligence techniques to process and analyze information for decision-making.

**Neural Networks:** Mathematical models inspired by the structure and function of biological neural networks for pattern recognition and prediction.

**Intelligent Agents:** Software entities that perceive their environment, reason, and act autonomously to achieve specific goals.

**Data Visualization:** The representation of data and information in visual forms to facilitate understanding and exploration.

**Machine Learning:** A subset of artificial intelligence that enables systems to learn and improve from data without explicit programming.

**Decision Support Systems:** Systems that assist in decision-making processes by providing relevant information and analysis.

**Genetic Algorithms:** Search and optimization algorithms inspired by the process of natural selection and genetic operations.

**Robotics:** The interdisciplinary field that focuses on the design, development, and application of intelligent machines or robots.

**Computer Vision:** The field of computer science that deals with enabling computers to "see" and interpret visual information.

**Knowledge Engineering:** The process of acquiring, organizing, and representing knowledge within intelligent information systems.

**Semantic Web:** An extension of the World Wide Web that enables machines to understand and process information more effectively.

**Stochastic Algorithms:** Algorithms that incorporate randomness or probability in their execution.

**Optimization:** The process of finding the best solution or maximizing/minimizing an objective function.

**Random Search:** An optimization technique that explores the search space randomly.

**Evolutionary Algorithms:** Algorithms inspired by biological evolution, such as genetic algorithms and evolutionary strategies.

**Simulated Annealing:** A stochastic optimization algorithm that mimics the process of slowly cooling a material to reduce defects.

**Genetic Algorithms:** Search and optimization algorithms that simulate natural selection and genetic operators like crossover and mutation.

**Particle Swarm Optimization:** An optimization method inspired by the collective behavior of bird flocks or fish schools.

**Ant Colony Optimization:** An optimization algorithm that models the foraging behavior of ants to find optimal paths or solutions.

**Markov Chain Monte Carlo:** A class of algorithms that use Markov chains to sample from complex probability distributions.

**Tabu Search:** An optimization algorithm that maintains a memory structure to avoid revisiting previously explored solutions.

**Cross-Entropy Method:** A Monte Carlo-based optimization method for solving rare event or combinatorial optimization problems.

**Differential Evolution:** A population-based optimization algorithm that evolves a population of candidate solutions.

**Harmony Search:** An algorithm inspired by the improvisation process of musicians to search for optimal solutions.

**Genetic Programming:** A form of genetic algorithms that evolves computer programs to solve problems.

**Combinatorial Optimization:** An optimization problem involving discrete variables and finding the best combination or arrangement.

**Expert Systems:** Computer systems that utilize expert knowledge to solve complex problems.

**Knowledge Base:** A repository of structured knowledge used by an expert system for decision-making.

**Rule-Based Systems:** Expert systems that employ IF-THEN rules to make decisions or perform tasks.

**Knowledge Representation:** Techniques and formalisms for representing and organizing knowledge within an expert system.

**Inference Engine:** The component of an expert system that applies rules and logic to draw conclusions.

**Knowledge Acquisition:** The process of collecting and capturing knowledge from domain experts.

**Knowledge Engineering:** The process of acquiring, organizing, and representing knowledge within an expert system.

**Fuzzy Logic:** A mathematical framework that handles uncertainty and imprecise information in expert systems.

**Ontology:** A hierarchical structure that represents concepts, entities, and relationships in a particular domain.

**Explanation Facility:** The capability of an expert system to provide explanations for its reasoning process.

**Knowledge Validation:** The process of ensuring the accuracy and reliability of the knowledge base within an expert system.

**Knowledge Inference:** The ability of an expert system to derive new knowledge or make logical conclusions based on existing knowledge.

**Case-Based Reasoning:** A problem-solving technique that uses past cases to solve new problems.

**Uncertainty Handling:** Techniques and methods for dealing with uncertain or incomplete information in expert systems.

**Expert System Shells:** Development environments or frameworks that provide tools for building and deploying expert systems.

**Knowledge Base:** A repository or database that stores structured knowledge for a specific domain or problem.

**Knowledge Management:** The process of creating, organizing, and managing knowledge within an organization or system.

**Knowledge Representation:** Techniques and methods for encoding and structuring knowledge in a form suitable for storage and reasoning.

**Ontology:** A formal representation of knowledge that defines concepts, relationships, and constraints within a domain.

**Knowledge Engineering:** The process of acquiring, organizing, and representing knowledge within a knowledge base.

**Inference:** The process of drawing logical conclusions or making deductions based on the existing knowledge in the knowledge base.

**Knowledge Acquisition:** The process of capturing or eliciting knowledge from domain experts and other sources.

**Knowledge Base Management:** The activities involved in maintaining and updating a knowledge base, including storage, retrieval, and version control.

**Knowledge Validation:** The process of ensuring the accuracy, consistency, and quality of the knowledge stored in the knowledge base.

**Knowledge Integration:** The process of combining or merging knowledge from different sources or domains into a unified knowledge base.

**Knowledge Querying:** The ability to search and retrieve specific information or knowledge from the knowledge base.

**Knowledge Sharing:** The practice of distributing and exchanging knowledge within an organization or community.

**Knowledge Elicitation:** Techniques for extracting or eliciting tacit knowledge from human experts to populate the knowledge base.

**Knowledge Visualization:** Methods and tools for representing and visualizing knowledge in a meaningful and comprehensible manner.

**Knowledge-based Reasoning:** The use of knowledge and inference rules stored in the knowledge base to solve problems or make decisions.



Завідувач кафедри \_\_\_\_\_

Завідувач кафедри \_\_\_\_\_

## ДОДАТОК 1

**Робочий план з дисципліни  
«Архітектура та проектування програмного забезпечення»**

## Семестр 1

Вид навчальної роботи	Годин у семестрі / кредити	Тиждень										Вид підсумкового контролю
		1	3	5	7	9	11	13	15	17		
Лекційні заняття	34	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	2ПК
Лабораторні заняття	17	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	1ПК
Самостійна робота	99	11	11	11	11	11	11	11	11	11	11	
Всього годин/кредитів	150/5	18	18	18	18	18	18	18	18	18	14	екзамен

## Семестр 2

Вид навчальної роботи	Годин у семестрі / кредити	Тиждень										Вид підсумкового контролю
		1	3	5	7	9	11	13	15	17		
Лекційні заняття	34	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	4ПК	2ПК
Лабораторні заняття	17	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	2ПК	1ПК
Самостійна робота	99	11	11	11	11	11	11	11	11	11	11	
Всього годин/кредитів	150/5	18	18	18	18	18	18	18	18	18	14	залік

Позначки: ПК – поточний контроль

Викладач

А.А.Трачук