


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЕКОНОМІКО-ТЕХНОЛОГІЧНИЙ ІНСТИТУТ ІМЕНІ Р.ЕЛЬВОРТИ

Кафедра інформаційних технологій

"ЗАТВЕРДЖУЮ"

Завідувач кафедри

 /О.П. Бондар/  
30 серпня 2024 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**«Об'єктно-орієнтоване програмування»**

Спеціальність: 122 «Комп'ютерні науки»

Освітня програма (освітньо-професійна): «Комп'ютерні науки»

Нормовані дані	Курс	Семестр	Всього годин за планом	Кількість кредитів ECTS	Всього аудит (год.)	Аудиторних годин, (у тому числі КЗ)			Самостійна робота (год.)	Контрольні (мод.) роботи	Контрольний підсумок (семестр)	
						Лекції	Лабораторні роботи	Практичні заняття			Залік	Екзамен
Форма навчання												
Денна	2	4	180	6	108	36	72		72	2		4

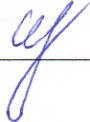
Кропивницький 2024 р.

Робочу програму складено на основі освітньо-професійної програми «Комп'ютерні науки» за спеціальністю 122 «Комп'ютерні науки»

Укладач: викладач кафедри інформаційних технологій Економіко-технологічного інституту ім. Роберта Ельворті Книшук А.В.

Робочу програму затверджено на засіданні кафедри інформаційних технологій.

Протокол № 1 від "30" серпня 2024 року

Завідувач кафедри  / О.П. Бондар /

Схвалено Вченою радою ЕТІ ім. Р.Ельворті  
Протокол № 15 від "24" вересня 2024 року

Голова Вченої ради  (Штець Т.Ф.)

## 1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни	
		Денна форма навчання	Заочна форма навчання
Кількість кредитів - 6	Галузь знань: 12 «Інформаційні технології»  Спеціальність: 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки»	Статус дисципліни: нормативна	
Залікових модулів -3		Рік підготовки	
Індивідуальне завдання студента -		2	
Загальна кількість годин - 180		Семестр	
Тижневих годин для денної форми навчання: аудиторних – 6	Ступінь вищої освіти: бакалавр)	4	
		Лекції (год.)	
		36	
		Практичні, семінарські (год.)	
		Лабораторні (год.)	
		72	
		Самостійна робота (год.)	
		72	
		Індивідуальне завдання (год.)	
		-	
Вид контролю:			
IV семестр екзамен			

**Примітка:** співвідношення кількості годин аудиторних занять та самостійної роботи становить (%): 57% до 43% .

## 2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### 2.1. Мета дисципліни:

- оволодіння проектуванням, розробкою та кодуванням програмних систем за допомогою засобів об'єктно-орієнтованого програмування (ООП) для розв'язку задач системного програмування в технічних та телекомунікаційних системах.

### 2.2. Завдання дисципліни:

- конструювання (проектування) програмних систем на основі відповідного набору абстрактних типів даних (АТД);
- проектування в термінах взаємозв'язку АТД та застосування механізму успадкування для сумісного використання коду та інтерфейсу;
- використання віртуальних функцій для динамічної обробки зв'язаних об'єктів;
- проектування конкретних прикладних задач з використанням шаблонів та власних розробок;
- користування програмними засобами, розробленими за допомогою ООП-методології.

### 2.3. Отримання компетенцій

За результатами вивчення дисципліни здобувач повинен опанувати наступні **компетентності**:

- **інтегральна** здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

#### - загальні

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК9. Здатність працювати в команді.

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління

## 2.4. Очікувані результати навчання

У результаті вивчення даної навчальної дисципліни студент повинен:

### Знати:

- основні концепції та ключові моменти ООП-парадигми.

### Вміти:

- самостійно розробляти алгоритми об'єктно-орієнтованих програм з використанням мов програмування високого рівня;
- самостійно проектувати, розробляти, відлагоджувати та використовувати програмні додатки на основі об'єктно-орієнтованого підходу.

Навчальний матеріал дисципліни структурований за модульним принципом і складається з трьох навчальних модулів. [Окремо виконується у четвертому семестрі курсова робота.](#)

**2.4.1.** У результаті засвоєння навчального матеріалу навчального модуля №1 "Побудова АТД і об'єктів" студент повинен:

### Знати:

- поняття класів та об'єктів;
- концепцію ООП – інкапсуляцію;
- практичні засоби підтримання інкапсуляції.

### Вміти:

- самостійно розробляти алгоритми, кодувати та відлагоджувати програмні додатки з використанням концепцій інкапсуляції;
- самостійно схематизувати предметну область проекту, використовуючи концепцію АТД;

**2.4.2.** У результаті засвоєння навчального матеріалу навчального модуля №2 "Зв'язки між об'єктами. Успадкування" студент повинен:

### Знати:

- концепцію успадкування;
- особливості системи вводу/виводу.
- практичні засоби підтримання інкапсуляції та успадкування.

### Вміти:

- самостійно організувати взаємозв'язки типів, використовуючи ієрархію класів;
- самостійно розробляти алгоритми, кодувати та відлагоджувати програмні додатки з використанням концепцій наслідування.

**2.4.3.** У результаті засвоєння навчального матеріалу навчального модуля №3 "Поліморфізм і розробка ОО програми" студент повинен:

### Знати:

- концепцію поліморфізму;
- практичні засоби ОО мовою програмування підтримки динамічного і параметричного поліморфізму і розробки ОО програми (віртуальні

функції, родові функції родові класи, шаблони, контейнери, ітератори, виняткові ситуації);

### **Вміти:**

- самостійно практично застосовувати концепцію статичного поліморфізму;
- практично використовувати концепцію динамічного і параметричного поліморфізму;
- створювати власні родові функції і родові класи (шаблони);
- використовувати бібліотеку стандартних шаблонів;
- проектувати, розробляти і кодувати мовою ООП – програмні додатки, комбінуючи власні розробки та стандартні шаблони і функції;

## **2.5. Програмні результати навчання**

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПР15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

## **2.6. Структурно-логічне місце дисципліни в освітній програмі**

Необхідна навчальна база перед початком вивчення дисципліни: для успішного вивчення дисципліни необхідні базові знання, отримані студентами в об'ємі шкільної програми, навички роботи за комп'ютером.

Попередні дисципліни	Наступні дисципліни:
Програмування C++	Проектування інформаційних систем
Теорія алгоритмів	Виконання кваліфікаційної роботи

Знання та вміння, отримані студентом під час вивчення даної навчальної дисципліни, використовуються в подальшому при вивченні багатьох наступних дисциплін професійної підготовки фахівця з базовою та повною вищою освітою.

### **3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

#### **Змістовий модуль №1 "Побудова АТД та об'єктів"**

##### **Тема 1.** ООП – сучасна методологія програмування.

Еволюція мов програмування. ООП – підхід в програмуванні. Інкапсуляція, успадкування, поліморфізм. Принципи проектування даних. Принципи роботи і використання ОО-систем.

##### **Тема 2.** Класи і об'єкти.

Класи. Відкриті і закриті члени класу. Об'єкти. Конструктори. Деструктори. Конструктори з параметрами. Присвоєння об'єктів. Передача об'єктів функціям. Об'єкти як значення, що повертаються функціями. Вбудовані функції в оголошенні класу.

##### **Тема 3.** Масиви об'єктів, вказівники і посилання.

Масиви об'єктів. Вказівники на об'єкти. Вказівник this. Оператори new і delete. Посилання. Передача посилань на об'єкти. Посилання як значення, що повертаються функцією. Незалежні посилання.

##### **Тема 4.** Статичні елементи класу

Статичні змінні. Визначення і ініціалізація статичних змінних-членів класу. Статичні методи

##### **Тема 5.** Константні елементи класу

Константні об'єкти класів. Константні методи класів. Константні посилання і класи. Перевантаження константних і неконстантних функцій.

##### **Тема 6.** Реалізація модульності програм

Відокремлення оголошення від реалізації. Файли з класами. Бібліотеки.

##### **Тема 7.** Дружні функції та класи

Дружні функції. Дружні функції для декількох класів. Дружні класи. Дружні методи. Анонімні об'єкти.

#### **Змістовий модуль №2 Зв'язки між об'єктами. Успадкування**

##### **Тема 8.** Типи зв'язків між об'єктами

Зв'язки між об'єктами в програмуванні. Композиція об'єктів. Реалізація та варіація композицій. Реалізація агрегації. Реалізація асоціацій. Рефлексивна асоціація. Залежність.

##### **Тема 9.** Успадкування.

Принцип успадкування. Специфікатори доступу при успадкуванні. Управління доступом до базового класу. Захищені члени класу. Конструктори і деструктори при успадкуванні.

##### **Тема 10.** Множинне успадкування

Механізми успадкування декількох базових класів. Особливості використання конструкторів і деструкторів при реалізації механізму

успадкування. Передача параметрів конструкторам базового класу. Повернення успадкованим членам класу початкової специфікації доступу

### **Тема 11.** Віртуальні базові класи

Поняття про віртуальні базові класи. Вказівники на базові класи. Вказівники на похідні класи. Абстрактні класи.

### **Змістовий модуль №3 "Поліморфізм і розробка ОО-програм"**

**Тема 12.** Статичний поліморфізм. Перевантаження функцій. Дружні функції. Перевантаження функцій. Перевантаження конструкторів. Конструктори копій. Аргументи за замовчуванням. Перевантаження та неоднозначність.

**Тема 13.** Статичний поліморфізм. Перевантаження операторів. Перевантаження унарних і бінарних операторів. Перевантаження операторів відношення. Перевантаження логічних операторів. Перевантаження унарних операторів. Дружні оператор-функції. Особливості використання оператора присвоєння. Перевантаження оператора індекса масиву [ ]. Перевантаження операторів вводу і виводу.

**Тема 14.** Динамічний поліморфізм. Віртуальні функції. Чисто віртуальні функції. Застосування віртуальних функцій. Успадкування віртуальних функцій. Суто віртуальні функції.

**Тема 15.** Параметричний поліморфізм. Шаблони. Родові функції. Застосування родових функцій і родових класів. Шаблони і успадкування. Шаблони і дружні функції.

**Тема 16.** Стандартна бібліотека шаблонів. Шаблони контейнерних класів. Контейнери. Вектор. Список. Дек. Множина і мультимножина. Бітові множини. Відображення і мультिवідображення. Стек. Черга. Пріоритетна черга. Ітератори. Типи ітераторів. Методи роботи з ітераторами. Функції і об'єкти функцій. Рядкові шаблони.

**Тема 17.** Обробка виняткових ситуацій. Поняття виняткової ситуації. Генерація виняткових ситуацій. Перехоплення виняткових ситуацій. Оброблення виняткових ситуацій, що виникають під час застосування оператора виділення пам'яті. Конструктори, деструктори і оброблення виняткових ситуацій. Виняткові ситуації та успадкування.



#### 4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин					
	усього	у тому числі				
		лк	Пр	лаб	інд	с.р.
1	2	3	4	5	6	7
<b>Модуль 1. Побудова АТД та об'єктів</b>						
<b>Тема 1.</b> ООП – сучасна методологія програмування. Еволюція мов програмування. ООП. Принципи проектування даних. Принципи роботи і використання ОО-систем	6	2		2		2
<b>Тема 2.</b> Класи і об'єкти. Класи. Відкриті і закриті члени класу. Об'єкти. Конструктори. Деструктори. Конструктори з параметрами. Присвоєння об'єктів. Передача об'єктів функціям. Об'єкти як значення, що повертаються функціями. Вбудовані функції в оголошенні класу	14	2		6		6
<b>Тема 3.</b> Масиви об'єктів, вказівники і посилання. Масиви об'єктів. Вказівники на об'єкти. Вказівник <b>this</b> . Оператори <b>new</b> і <b>delete</b> . Посилання. Передача посилань на об'єкти. Посилання як значення, що повертаються функцією. Незалежні посилання.	10	2		4		4
<b>Тема 4.</b> Статичні елементи класу Статичні змінні. Визначення і ініціалізація статичних змінних-членів класу. Статичні методи	10	2		4		4
<b>Тема 5.</b> Константні елементи класу Константні об'єкти класів. Константні методи класів. Константні посилання і класи. Перевантаження константних і неконстантних функцій.	10	2		4		4
<b>Тема 6.</b> Реалізація модульності програм Відокремлення оголошення від реалізації. Класи і заголовкові файли. Бібліотеки.	6	2		2		2
<b>Тема 7.</b> Дружні функції та класи Дружні функції. Дружні функції для декількох класів. Дружні класи. Дружні методи. Анонімні об'єкти.	10	2		4		4
Разом	66	14		26		26
<b>Модуль 2. Зв'язки між об'єктами. Успадкування</b>						
<b>Тема 8.</b> Типи зв'язків між об'єктами Зв'язки між об'єктами в програмуванні. Композиція об'єктів. Реалізація та варіація композицій. Реалізація агрегації. Реалізація асоціацій. Рефлексивна асоціація. Залежність.	10	2		4		4
<b>Тема 9.</b> Успадкування. Принцип успадкування. Специфікатори доступу при успадкуванні. Управління доступом до базового класу. Захищені члени класу. Конструктори і деструктори при успадкуванні.	14	2		6		6

<b>Тема 10.</b> Множинне успадкування Механізми успадкування декількох базових класів. Особливості використання конструкторів і деструкторів при реалізації механізму успадкування. Передача параметрів конструкторам базового класу. Повернення успадкованим членам класу початкової специфікації доступу	10	2		4		4
<b>Тема 11.</b> Віртуальні базові класи Поняття про віртуальні базові класи. Вказівники на базові класи. Вказівники на похідні класи. Абстрактні класи.	10	2		4		4
Разом	44	8		18		18
<b>Модуль 3. Поліморфізм і розробка ОО програми</b>						
<b>Тема 12.</b> Статичний поліморфізм. Перевантаження функцій. Дружні функції. Перевантаження функцій. Перевантаження конструкторів. Конструктори копій. Аргументи за замовчуванням. Перевантаження та неоднозначність.	10	2		4		4
<b>Тема 13.</b> Статичний поліморфізм. Перевантаження операторів. Перевантаження унарних і бінарних операторів. Перевантаження операторів відношення. Перевантаження логічних операторів. Перевантаження унарних операторів. Дружні оператор-функції. Особливості використання оператора присвоєння. Перевантаження оператора індекса масиву [ ]. Перевантаження операторів вводу і виводу.	14	2		6		6
<b>Тема 14.</b> Динамічний поліморфізм. Віртуальні функції. Чисто віртуальні функції. Застосування віртуальних функцій. Успадкування віртуальних функцій. Суто віртуальні функції.	10	2		4		4
<b>Тема 15.</b> Параметричний поліморфізм. Шаблони. Родові функції. Застосування родових функцій і родових класів. Шаблони і успадкування. Шаблони і дружні функції.	10	2		4		4
<b>Тема 16.</b> Стандартна бібліотека шаблонів. Шаблони контейнерних класів. Контейнери. Вектор. Список. Дек. Множина і мультимножина. Бітові множини. Відображення і мультिवідображення. Стек. Черга. Пріоритетна черга. Ітератори. Типи ітераторів. Методи роботи з ітераторами. Функції і об'єкти функцій. Рядкові шаблони.	16	4		6		6
<b>Тема 17.</b> Обробка виняткових ситуацій. Поняття виняткової ситуації. Генерація, перехоплення та обробка виняткових ситуацій, що виникають під час виділення пам'яті. Конструктори, деструктори і обробка виняткових ситуацій. Виняткові ситуації та успадкування.	10	2		4		4
Разом	70	14		28		28
<b>УСЬОГО ГОДИН</b>	<b>180</b>	<b>36</b>		<b>72</b>		<b>72</b>

## 5. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

Теми	Кількість годин
<b>Змістовий модуль 1</b>	
Складання програм із використанням структур та їх масивів.	4
Проектування класів і створення об'єктів.	4
Конструктори, деструктори. Передача об'єктів у функцію.	4
Передача посилань на об'єкти. Посилання на об'єкти як значення, що повертаються функцією.	4
Реалізація модульності при створенні ОО-програм	2
Дружні класи та дружні функції	4
<b>Разом за модуль 1</b>	<b>22</b>
<b>Змістовий модуль 2</b>	
Проектування ОО-моделі програми	4
Технологія одинарного успадкування	4
Множинне успадкування	4
Віртуальні базові класи	4
<b>Разом за модуль 2</b>	<b>16</b>
<b>Змістовий модуль 3</b>	
Проектування і реалізація програми з перевантаженням функцій	4
Проектування і реалізація програми з перевантаженням операторів	4
Проектування і реалізація програми з перевантаженням операторів	4
Віртуальні функції	4
Шаблонні функції та класи	6
Обробка виняткових ситуацій.	4
<b>Разом за модуль 3</b>	<b>34</b>
<b>РАЗОМ</b>	<b>72</b>

## 6. САМОСТІЙНА РОБОТА

Для успішного вивчення і засвоєння дисципліни «Об'єктно-орієнтоване програмування» студенти повинні володіти значним обсягом інформації, частину якої вони отримують і опрацьовують шляхом самостійної роботи. Самостійна робота полягає в опрацюванні сучасної навчальної і наукової фахової літератури.

Теми	Кількість годин
Огляд сучасних ООП-орієнтованих мов програмування. Пошук, підбір та огляд літературних джерел за заданою тематикою.	2
Передача об'єктів функціям. Об'єкти як значення, що повертаються функціями	6
Завдання на створення та вилучення об'єктів	4
Статичні елементи класу	2
Константні елементи класу	2
Підтримка модульності, простори імен, зовнішнє зв'язування	4
Зв'язки між об'єктами в програмуванні. Пошук, підбір та огляд літературних джерел за заданою тематикою.	4
Успадкування класів	4
Поліморфізм класів	4
Шаблони класів	4
Стандартні контейнерні класи	6
Абстрактні класи	4
Механізм виняткових ситуацій	4
	<b>50</b>
<b>Опрацювання лекцій та підготовка до лабораторних занять</b>	<b>22</b>
<b>РАЗОМ</b>	<b>72</b>

## 7. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Індивідуальні завдання включають:

виконання тестових завдань за темами навчальної дисципліни. За виконання індивідуальних завдань студент денної форми навчання може отримати кількість балів, зазначену в п.10.

виконання контрольної роботи. Для цього розроблені завдання, які наведені в методичних вказівках, що містять рекомендації щодо виконання роботи та рекомендовану літературу.

## **8. МЕТОДИ НАВЧАННЯ**

Лекції, лабораторні (практичні) заняття, самостійна робота з навчальною та довідковою літературою, самостійне виконання контрольних робіт, консультації.

На лекціях надається основний теоретичний матеріал з дисципліни, який ілюструється демонстраційним матеріалом з використанням мультимедійного проектора.

На лабораторних (практичних) заняттях студенти працюють, виконуючи практичні завдання. Ці завдання та методика їх виконання роз'яснюються викладачем. Основна мета цих робіт – надати студентам практичних навичок застосування набутих теоретичних знань при вирішенні конкретних завдань.

Методами навчання дисципліни є способи спільної діяльності й спілкування викладача і студентів, що забезпечують вироблення позитивної мотивації навчання, оволодіння системою професійних знань, умінь і навичок, формування наукового світогляду, розвиток пізнавальних сил, культури розумової праці майбутніх фахівців спеціальності «Комп'ютерні науки».

Залежно від джерела знань, під час навчальних занять, як практичних, так і лекційних, використовуються наступні методи навчання: словесні (пояснення, бесіда, дискусія, діалог), наочні (демонстрація, ілюстрація), практичні (рішення задач, ділові ігри).

За характером пізнавальної діяльності, при вивченні дисципліни використовуються: пояснювально-наочний проблемний виклад; частково-пошуковий та дослідницький методи.

За місцем в структурній діяльності використовуються:

- методи організації й здійснення навчальної діяльності, що поєднує словесні, наочні і практичні методи; репродуктивні й проблемно-пошукові; методи навчальної роботи під керівництвом викладача й методи самостійної роботи студентів;
- методи стимулювання й мотивації навчальної роботи, що об'єднали в собі пізнавальні ігри, навчальні дискусії, моделювання рольових ситуацій, створення ситуацій успіху в навчальній роботі, пред'явлення вимог і метод заохочення;
- методи контролю й самоконтролю за навчальною діяльністю: методи усного, письмового контролю; індивідуального й фронтального, тематичного і систематичного контролю.

## **9. МЕТОДИ КОНТРОЛЮ**

Оцінювання якості знань студентів здійснюється шляхом поточного, підсумкового (семестрового) контролю за 100-бальною шкалою оцінювання, за шкалою ECTS та національною шкалою оцінювання.

Поточний контроль знань студентів передбачає оцінювання за наступними основними напрямками:

- перевірка теоретичних знань;
- перевірка виконання індивідуальних та практичних робіт.

З даних компонентів складаються загальні бали, які фіксуються в журналі викладача. Оцінювання рівня засвоєння теоретичних знань студентів проводиться під час експрес опитування лекційного матеріалу, на практичних роботах та за результатами перевірки індивідуальних завдань.

Активна робота під час аудиторних занять також може оцінюватись викладачем певною кількістю додаткових балів по кожній темі.

Підсумкова оцінка складається з результатів усіх рівнів поточного та модульних контролів, що передбачені навчальним планом за весь термін викладання дисципліни та, якщо це передбачено, результатів складання екзамену з дисципліни.

## 10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

Для поточного та підсумкового контролю успішності здобувачів вищої освіти використовується модульно-рейтингова система, яка передбачає розподіл балів за виконання усіх запланованих видів робіт. Така система оцінювання виключає можливість суб'єктивного відношення викладача і орієнтує здобувача вищої освіти на підрахунок своїх балів за конкретні види робіт.

### Розподіл балів з дисципліни

(максимальна кількість балів, підсумковий контроль залік)

Поточне оцінювання та самостійна робота								Разом
Модуль 1								
T1	T2	T3	T4	T5	T6	T7	МКР 1	
1	3	3	2	2	2	3	5	21

Поточне оцінювання та самостійна робота					Разом
Модуль 2					
T8	T9	T10	T11	МКР 2	
3	3	3	3	5	17

Поточне оцінювання та самостійна робота							Разом
Модуль 3							
T12	T13	T14	T15	T16	T17	МКР 3	
3	3	3	3	3	2	5	22

T1, T2... T17 - теми змістових модулів.

МКР – модульна контрольна робота

Поточне оцінювання та самостійна робота	Екзамен	Разом
60	40	100

## 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	<b>A</b>	відмінно	зараховано
82-89	<b>B</b>	добре	
74-81	<b>C</b>		
64-73	<b>D</b>	задовільно	
60-63	<b>E</b>		
35-59	<b>FX</b>	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1-34	<b>F</b>	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

### 11. НАВЧАЛЬНО-МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ

1. Навчально-методичний комплекс.
2. Тексти лекцій (в електронному варіанті).

Лабораторні роботи виконуються на персональних комп'ютерах із встановленою операційною системою Windows. Програмне забезпечення: пакет Microsoft Office або LibreOffice, OpenOffice.org, середовища IDE – Code:Blocks, Microsoft Visual Studio 2020. Має бути доступ до мережі Інтернет з довільним браузером.

### 12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

#### Базова література

1. Кравець П.О. Об'єктно-орієнтоване програмування. Львів: Видавництво Львівської політехніки, 2012. – 624 с.
2. Кузнецов М. С. Об'єктно-орієнтоване програмування з використанням UML та мови C++. Дніпропетровськ: НМетАУ, 2003. – 90 с.
3. Об'єктно-орієнтоване програмування: конспект лекцій для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання / Ю. Е. Парфьонов, В. М. Федорченко, М. Ю. Лосєв, О. В. Щербаков.– Харків : Вид. ХНЕУ, 2010.– 312с.

### Додаткова література

Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. – М.: Бином, 2000. – 560 с.

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley ISBN-13: 978-0201633610

Gerbert Schildt. С++: A Beginner's Guide ISBN-13: 978-0072232158, 2012, 542 p.

Об'єктно-орієнтоване програмування. Комп'ютерний практикум: Навч. посібник. [Електронний ресурс] / Уклад.: В.М. Порєв. – Електронні текстові дані (1 файл: 2,5 МБ). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 105 с.  
<https://ela.kpi.ua/handle/123456789/51571>

### 13. ІНФОРМАЦІЙНІ INTERNET-РЕСУРСИ

1. Патерни проектування. REFACTORING GURU [Online] Available from: <https://refactoring.guru/uk/design-patterns>

2. Bjarne Stroustrup. Programming: Principles and Practice Using C++ (2nd Edition) [Online] Available from: <https://dl.icdst.org/pdfs/files3/fe0590f02fa06bb42cba558fbc9e51c.pdf>

3. С++ Standard Library including С++ 14 & С++ 17 [Online]:  
<https://www.educative.io/courses/cpp-standard-library-including-cpp-14and-cpp-17>

4. Samuel Oloruntoba. SOLID: The First 5 Principles of Object Oriented Design: [https://www.digitalocean.com/community/conceptual\\_articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design](https://www.digitalocean.com/community/conceptual_articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design)

5. Сайт з довідкою по С++: <http://www.cplusplus.com/>

6. Довідка Visual Studio 2013 С++: <https://msdn.microsoft.com/ru-RU/library/60k1461a.aspx>

7. Сайт присвячений програмуванню на С++: <http://cppstudio.com>



## ДОДАТКИ

Перелік питань до модульного контролю

### Модуль 1

1. Основні принципи ООП.
2. Характеристики ООП.
3. Засоби абстрагування у програмуванні.
4. Опис класу.
5. Опис об'єктів.
6. Вказівник `this`.
7. Використання конструкторів.
8. Робота з конструктором копіювання.
9. Використання статичних полів.
10. Використання статичних методів.
11. Дружні функції.
12. Дружні класи.
13. Використання деструкторів.

### Модуль 2

14. Зв'язки між об'єктами в програмуванні.
15. Наслідування: властивості наслідування.
16. Правила наслідування базових елементів
17. Наслідування: характеристика рівнів доступу
18. Механізми успадкування декількох базових класів.
19. Абстрактні класи. Використання абстрактних класів
20. Віртуальні базові класи

### Модуль 3

21. Перевантаження унарних операцій.
22. Перевантаження бінарних операцій.
23. Перевантаження операції присвоєння.
24. Перевантаження операції приведення типів.
25. Перевантаження операції виклику функції.
26. Перевантаження операції індексування.
27. Вказівники на методи класів.
28. Вказівники на поля класів.
29. Віртуальні методи. Правила використання віртуальних методів
30. Використання чисто віртуальних методів
31. Шаблони класів. Правила описання шаблонів
32. Параметри для шаблонних класів
33. Використання шаблонів класів
34. Конкретизація шаблонів
35. Спеціалізація шаблонів
36. Статичні елементи шаблонів, використання `typename`
37. Синтаксис виняткова ситуація

38. Обробка виняткової ситуації
39. Генерування виняткової ситуації
40. Використання виняткових ситуацій
41. Неперехопленні виняткові ситуації
42. Специфікація виняткових ситуацій