

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ЧОРТКІВСЬКИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ПІДПРИСМНИЦТВА І БІЗНЕСУ

ЗАТВЕРДЖУЮ  
Директор ЧННІПБ ЗУНУ

Надія КУЛЬЧИЦЬКА

2023 р.

ЗАТВЕРДЖУЮ

В.о. проректора з науково-педагогічної роботи  
Віктор ОСТРОВЕРХОВ

2023 р.

## РОБОЧА ПРОГРАМА

з дисципліни «Об'єктно-орієнтоване програмування»  
ступінь вищої освіти – бакалавр  
галузь знань – 01 Освіта/Педагогіка  
спеціальність – 015 Професійна освіта (Цифрові технології)  
спеціалізація – 015.39 Цифрові технології  
освітньо-професійна програма – Професійна освіта (Цифрові технології)

кафедра фундаментальних та спеціальних дисциплін

Форма навчання	Курс	Семестр	Лекції (год.)	Практ. (год.)	ІРС (год.)	Тренінг, КПЗ (год.)	Самост. робота студ. (год.)	Разом (год.)	Екзамен (сем.)
Денна	2	4	28	28	3	8	53	120	4
Заочна	2	4	8	4			108	120	4

Чортків – ЗУНУ  
2023

31.08.2023  
*[Signature]*

Робоча програма складена на основі освітньо-професійної програми підготовки бакалавра галузі знань 01 Освіта/Педагогіка спеціальності 015.039 Професійна освіта, затвердженої Вченою радою ЗУНУ (протокол № 9 від 15.06.2022 р.).

Робочу програму склав викладач Юрій ОНУФРЕЇВ

Робоча програма затверджена на засіданні кафедри фундаментальних та спеціальних дисциплін (протокол № 1 від 28.08.2023 р.)

Завідувач кафедри



Людмила ДЕРМАНСЬКА

Розглянуто та схвалено групою забезпечення спеціальності 015.39 Професійна освіта (протокол № 1 від 30.08.23 р.)

Керівник групи  
забезпечення спеціальності

  
(підпис)

Лілія РЕБУХА

# СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

## «Об'єктивно-орієнтоване програмування»

### Опис дисципліни «Об'єктивно-орієнтоване програмування»

Дисципліна «Об'єктивно-орієнтоване програмування»	Галузь знань, спеціальність, СВО	Характеристика навчальної дисципліни
Кількість кредитів – 4	галузь знань 01 Освіта/Педагогіка	<b>Статус дисципліни</b> <u>блок обов'язкових дисциплін, цикл професійної підготовки</u> <b>Мова навчання:</b> українська
Кількість залікових модулів – 4	спеціальність – 015.039 Професійна освіта	Рік підготовки: <i>Денна – 2</i> <i>Заочна - 2</i> Семестр: <i>Денна – 4</i> <i>Заочна – 3</i>
Кількість змістових модулів – 6	освітньо-професійна програма – Цифрові технології	Лекції: <i>Денна -28</i> <i>Заочна - 8</i> Практичні заняття: <i>Денна – 28</i> <i>Заочна - 4</i>
Загальна кількість годин – 120	Ступінь вищої освіти – бакалавр	Самостійна робота: <i>Денна – 53</i> <i>Заочна – 108</i> <i>Тренінг - 8</i> Індивідуальна робота: <i>Денна - 4</i>
Кількість тижневих годин – 10 Кількість аудиторних годин - 4		Вид підсумкового контролю – екзамен

## **2. Мета і завдання дисципліни «Об'єктивно-орієнтоване програмування»**

### **2.1. Мета вивчення дисципліни.**

Основною метою дисципліни “Об'єктивно-орієнтоване програмування” – ознайомлення студентів з основними принципами технології об'єктивно-орієнтованого програмування (ООП) та оволодіння ними з використанням мови програмування Java, методами проектування та створення програм згідно сучасних технологій.

**Предметом дисципліни** “Об'єктивно-орієнтоване програмування” є основні елементи, способи та прийоми програмування мовою Java.

**2.2. Завдання вивчення дисципліни:** є основи проектування програмного забезпечення; вивчити технології об'єктивно-орієнтованого програмування; вивчити прийоми роботи з візуальними середовищами програмування; набуття навичок розробки й тестування програмних продуктів функціонуючих під керуванням сучасних операційних систем; формування у студентів абстрактного мислення, яке повинне допомогти рішенню прикладних задач, пов'язаних з різноманітними галузями знань.

**2.3. Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни:**

Опанувати об'єктивно-орієнтований підхід для розробки та розвитку програмних систем; розуміти методологію об'єктивно-орієнтованого проектування, оволодіти нею і використовувати її впродовж життєвого циклу програмного забезпечення; розробляти програмне забезпечення за допомогою сучасних інструментальних засобів створення програмного забезпечення

### **2.4. Передумови для вивчення дисципліни.**

“Об'єктивно-орієнтоване програмування” взаємопов'язана з такими дисциплінами як „Алгоритмізація та програмування”, „Вступ до фаху”, „Інформатика”, „Технології Інтернет”, „Web-дизайн та Web-програмування”, „Технології проектування та адміністрування БД і СД”.

### **2.5. Програмні результати навчання.**

Студенти, після вивчення курсу повинні вміти самостійно:

- використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктивно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

– аргументувати вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.

– демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.

– обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.

**3. Програма навчальної дисципліни:**  
«Об'єктно-орієнтоване програмування»

***Змістовий модуль 1. Основні парадигми ООП.***

**Тема 1. Вступ в об'єктно–орієнтоване програмування.**

Історія об'єктно-орієнтованого програмування. Основні поняття. Абстракція даних. Наслідування, інкапсуляція та поліморфізм.

**Література:** 1,2,10.

**Тема 2. Об'єкти та класи.**

Поняття об'єкта в ООП. Стан та поведінка об'єкта. Життєвий цикл об'єкта. Відношення між об'єктами. Поняття класу. Типи класів. Відношення між класами. Інстанціювання.

**Література:** 1,2,9,10.

**Тема 3. Об'єктно-орієнтований аналіз і декомпозиція складних об'єктів.**

Об'єктно-орієнтований аналіз і декомпозиція складних об'єктів. Об'єктно-орієнтований синтез інформаційних моделей. Використання мови об'єктно-орієнтованого моделювання UML.

**Література:** 1,2,9,10.

***Змістовий модуль 2. Реалізація принципів ООП мовою C++.***

**Тема 4. Класи та об'єкти в C++.**

Класи, члени класу. Ініціалізація. Інтерфейс та реалізація. Конструктори і деструктори. Конструювання і знищення об'єктів. Вказівник this. Вказівники на члени класу. Статичні змінні класу. Статичні методи класу. Поняття дружності.

**Література:** 2,3,6,7,17.

**Тема 5. Наслідування.**

Похідні класи. Модифікатори наслідування. Розширення та обмеження спеціалізації класів. Конструктори та деструктори похідних класів. Виклик конструкторів похідних класів. Виклик конструкторів класів ієрархії. Множинне наслідування. Виклик конструкторів при множинному наслідуванні. Передача параметрів конструктору базового класу.

**Література:** 2,3,6,7,17

**Тема 6. Віртуальні функції та поліморфізм.**

Віртуальні функції та поліморфізм. Поліморфні об'єкти та контейнери. Чисто віртуальні функції та абстрактні класи. Механізм пізнього зв'язування.

**Література:** 2,3,6,7,17.

**Тема 7. Перевантаження операторів.**

Перевантаження операторів. Створення операторної функції члена. Створення префіксної і постфіксної форм операторів інкрементації і декрементації. Перевантаження скорочених операторів присвоєння. Обмеження на перевантаження операторів. Перевантаження операторів new і delete. Перевантаження індикаторів. Перевантаження оператор «()». Перевантаження оператора «->». Перевантаження оператора «,».

**Література:** 2,3,6,7,17.

**Тема 8. Шаблони.**

Простий шаблон. Шаблони-списки. Параметризовані функції. Правила ототожнення перевантажених параметризованих функцій. Наслідування і шаблони. Узагальнені класи. Застосування аргументів за умовчанням в шаблонних класах. Явні спеціалізації класів.

**Література:** 2,3,6,7,17.

### **Тема 9. Обробка виняткових ситуацій.**

Обробка помилок. Розпізнавання ситуацій. Класифікація ситуацій. Використання ресурсів. Генерація виняткових ситуацій. Повторне генерування виняткової ситуації. Функції terminate() і unexpected(). Обробники, пов'язані з функціями terminate() і unexpected(). Класи exception і bad\_exception. перехоплення класів виняткових ситуацій. Застосування декількох операторів catch. Обробка похідних виняткових ситуацій.

**Література: 2,3,6,7,17.**

### ***Змістовий модуль 3. Бібліотека стандартних шаблонів STL.***

#### **Тема 10. Вступ в STL.**

Склад STL. Предикати та функції-об'єкти. Алокатори. Ітератори. Поняття ітератора. Література: 2,3,6,7,17. Тема 11. Контейнерні класи. Поняття контейнера. Типи контейнерів. Контейнер vector. Контейнер deque. Асоціативні контейнери. Контейнер Set. Контейнер Map.

**Література: 2,3,6,7,17.**

#### **Тема 12. Алгоритми STL.**

Модифікуючі та немодифікуючі операції. Алгоритми сортування. Алгоритми роботи з множинами.

**Література: 2,3,6,7,17.**

#### **Тема 13. Оптимізація обчислень в STL.**

Шаблон класу valarray. Перетин масивів.

**Література: 2,3,6,7,17.**

### ***Змістовий модуль 4. Основи C # і платформа .NET.***

#### **Тема 14. Мова C # і платформа .NET**

Створення C #. Зв'язок C # з середовищем .NET Framework. Загальномовне виконуюча середу CLR. Загальна система типів CTS. Загальномовне специфікація CLS. Збірки. Проміжний мова CIL. Класи і простору імен. Загальномовна інфраструктура CLI.

**Література: 12,13,14,23,24.**

#### **Тема 15. Основи C #**

Проста програма на C #. Коментарі. Змінні. Область видимості змінних, константи. Типи даних. Літерали. Перетворення типів. Арифметичні оператори. Оператори відносини і логічні оператори. Оператор присвоювання. Порозрядні оператори. Тернарний оператор. Умовні оператори. Циклічні оператори: for, while, do while і foreach. Оператори переходу.

**Література: 12,13,14,23,24.**

#### **Тема 16. Масиви, кортежі і рядки**

Масиви. Багатовимірні масиви. Ступінчасті (зубчасті) масиви масивів. Клас Array. Масиви в якості параметрів. Кортежі. Рядки. Клас StringBuilder. Форматуючі рядки. Регулярні вирази.

**Література: 12,13,14,23,24.**

### ***Змістовий модуль 5. Реалізація принципів ООП мовою C#.***

#### **Тема 17. Класи і об'єкти в C#.**

Класи. Клас Object. Створення об'єктів. Методи. Конструктори. Збірка сміття і деструктори. Ключове слово this. Доступ до членів класу. Модифікатори параметрів. Необов'язкові та іменовані аргументи. Рекурсія. Ключове слово static. Індексатори. Властивості. Модифікатори доступу в аксесорах.

**Література: 12,13,14,23,24.**

#### **Тема 18. Перевантаження методів класу**

Перевантаження методів. Перевантаження конструкторів. Перевантаження індексаторів. Основи перевантаження операторів. Перевантаження операторів відношення і операторів true – false. Перевантаження логічних операторів. Оператори перетворення.

**Література:** 12,13,14,23,24.

**Тема 19.** Успадкування та поліморфізм Основи успадкування. Захищений доступ і виключення успадкування. Конструктори та успадкування. Успадкування та приховування імен. Посилання на базовий клас і об'єкти похідних класів. Віртуальні методи, властивості і індексатори. Абстрактні класи.

**Література:** 12,13,14,23,24.

**Тема 20.Інтерфейси, структури та перерахування**

Інтерфейси. Інтерфейсні посилання. Інтерфейсні властивості і індексатори. Спадкування інтерфейсів. Явна реалізація інтерфейсу. Структури. Перерахування.

**Література:** 12,13,14,23,24.

*Змістовий модуль 6. Програмування обробки винятків та подій в C#.*

**Тема 21. Обробка винятків в C#**

Основи обробки винятків. Приховати винятків. Клас Exception. Конфігурування стану виключення. Винятки рівня системи та програми. Обробка численних винятків. Оператори throw і finally. Винятки, пов'язані з пошкодженим станом (Corrupted State Exceptions). Ключові слова checked і unchecked.

**Література:** 12,13,14,23,24.

**Тема 22.Делегати та лямбда-вирази**

Делегати. Груповий виклик і адресація делегованих методів. Коваріантність і контраваріантних делегатів. Делегати Action і Func . Анонімні методи. Лямбда-вирази.

**Література:** 12,13,14,23,24.

**Тема 23. Події**

Події. Аксесор подій. Обробка подій в середовищі .NET Framework.

**Література:** 12,13,14,23,24.

**Тема 24. Узагальнення**

Огляд узагальнень. Узагальнені класи. Обмежені типи. Обмежені класи. Обмежені інтерфейси і конструктори. Обмеження посилального типу і типу значення. Ієрархії узагальнених класів. Засоби узагальнень. Узагальнені методи. Узагальнені структури. Узагальнені делегати. Узагальнені інтерфейси. Модифікація узагальнених методів. Коваріантність і контраваріантних в узагальненнях.

**Література:** 12,13,14,23,24

**4. Структура залікового кредиту  
з дисципліни «Об'єктивно-орієнтоване програмування»  
(денна форма навчання)**

	<i>Кількість годин</i>					Контрольні заходи
	Лекції	Практичні заняття	СРС	ІРС	Тренінг	
<b>Змістовий модуль 1. Основні парадигми ООП.</b> Тема 1. Вступ в об'єктно-орієнтоване програмування.	1	1	2	3	1	Поточне опитування Презентація, тести
Тема 2. Об'єкти та класи	1	1	2			
Тема 3. Об'єктно-орієнтований аналіз і декомпозиція складних об'єктів	1	1	2			
<b>Змістовий модуль 2. Реалізація принципів ООП мовою C++</b> Тема 4. Класи та об'єкти в C++	1	1	2		1	
Тема 5. Наслідування	1	1	2			
Тема 6. Віртуальні функції та поліморфізм.	1	1	2			
Тема 7. Перевантаження операторів	1	1	2			
Тема 8. Шаблони	1	1	2			
Тема 9. Обробка виняткових ситуацій	1	1	2			
<b>Змістовий модуль 3. Бібліотека стандартних шаблонів STL</b> Тема 10. Вступ в STL	1	1	2		1	Питання, кейси  Поточне опитування
Тема 11. Контейнерні класи	1	1	2			
Тема 12. Алгоритми STL	1	1	2			
Тема 13. Оптимізація обчислень в STL	1	1	2		1	
<b>Змістовий модуль 4. Основи C # і платформа .NET.</b> Тема 14. Мова C # і платформа .NET	1	1	2		2	Презентація, тести  Питання, кейси
Тема 15. Основи C #	1	1	2			
Тема 16. Масиви, кортежі і рядки	1	1	2			
<b>Змістовий модуль 5. Реалізація принципів ООП мовою C#.</b> Тема 17. Класи і об'єкти в C#	1	1	2			
Тема 18. Перевантаження методів класу	1	1	2			
Тема 19. Успадкування та поліморфізм	1	1	2			
Тема 20. Інтерфейси, структури та перерахування	1	1	2			
<b>Змістовий модуль 6. Програмування обробки винятків та подій в C#.</b> Тема 21. Обробка винятків в C#	2	2	3		2	
Тема 22. Делегати та лямбда-вирази	2	2	3			
Тема 23. Події	2	2	3			
Тема 24. Узагальнення	2	2	4		Ректорська контрольна робота	
<b>Разом</b>	<b>28</b>	<b>28</b>	<b>53</b>	<b>3</b>	<b>8</b>	<b>екзамен</b>



## (заочна форма навчання)

	<i>Кількість годин</i>			
	Лекції	Практичні заняття	СРС	Контрольні заходи
<b>Змістовий модуль 1. Основні парадигми ООП.</b> Тема 1. Вступ в об'єктно-орієнтоване програмування.			5	Поточне опитування тести
Тема 2. Об'єкти та класи	2		5	
Тема 3. Об'єктно-орієнтований аналіз і декомпозиція складних об'єктів			5	
<b>Змістовий модуль 2. Реалізація принципів ООП мовою C++</b> Тема 4. Класи та об'єкти в C++	2	2	5	
Тема 5. Наслідування			5	
Тема 6. Віртуальні функції та поліморфізм.			5	
Тема 7. Перевантаження операторів			5	
Тема 8. Шаблони			5	
Тема 9. Обробка виняткових ситуацій			5	
<b>Змістовий модуль 3. Бібліотека стандартних шаблонів STL</b> Тема 10. Вступ в STL	2		5	Питання, кейси
Тема 11. Контейнерні класи			5	
Тема 12. Алгоритми STL			5	
Тема 13. Оптимізація обчислень в STL			5	
<b>Змістовий модуль 4. Основи C # і платформа .NET.</b> Тема 14. Мова C # і платформа .NET		2	5	
Тема 15. Основи C #			5	
Тема 16. Масиви, кортежі і рядки			5	
<b>Змістовий модуль 5. Реалізація принципів ООП мовою C#.</b> Тема 17. Класи і об'єкти в C#	2		5	Питання, кейси
Тема 18. Перевантаження методів класу			5	
Тема 19. Успадкування та поліморфізм			5	
Тема 20. Інтерфейси, структури та перерахування			5	
<b>Змістовий модуль 6. Програмування обробки винятків та подій в C#.</b> Тема 21. Обробка винятків в C#	2		2	
Тема 22. Делегати та лямбда-вирази			2	
Тема 23. Події			2	
Тема 24. Узагальнення			2	
Разом	<b>8</b>	<b>4</b>	<b>108</b>	<b>Екзамен</b>

## 5. Тематика лабораторних занять

### Лабораторна робота № 1

Тема: Об'єктно-орієнтований аналіз і декомпозиція складних об'єктів.

Мета: Навчитись будувати моделі складних об'єктів з використанням мови об'єктно-орієнтованого моделювання UML.

Література: 1, 2.

### Лабораторна робота № 2

Тема: Конструювання об'єктів конкретних класів.

Мета: Засвоєння методики створення конкретних класів з використанням мови програмування C++.

Література: 2, 4, 12, 17.

### Лабораторна робота № 3

Тема: Реалізація наслідування.

Мета: Засвоєння методики реалізації наслідування з використанням мови програмування C++.

Література: 2, 10, 12, 17.

### Лабораторна робота № 4

Тема: Поліморфізм.

Мета: Засвоєння методики реалізації поліморфізму з використанням мови програмування C++.

Література: 2, 10, 12, 17.

### Лабораторна робота № 5

Тема: Перевантаження операторів.

Мета: Засвоєння методики перевантаження унарних та бінарних операторів для класів користувача.

Література: 2, 6, 11, 17.

### Лабораторна робота № 6

Тема: Розробка шаблонів.

Мета: Навчитись використовувати можливості параметричного поліморфізму.

Література: 2, 10, 12, 17.

### Лабораторна робота № 7

Тема: Обробка виняткових ситуацій.

Мета: Засвоєння методики відлагодження програм та обробки програм на етапі виконання засобами об'єктно-орієнтованого програмування.

Література: 4, 10, 12, 17.

### Лабораторна робота № 8

Тема: Використання STL.

Мета: Засвоєння принципів роботи з STL.

Література: 7, 10, 12, 17.

### Лабораторна робота № 9

Тема: Основи C# та .NET.

Мета: Навчитись створювати консольні додатки мовою C# на платформі .NET.

Література: 8, 9, 12, 16.

### Лабораторна робота № 10

Тема: Використання масивів, кортежів і рядків у C#.

Мета: Навчитись використовувати масиви, кортежі і рядки у C#.

Література: 2, 3, 11, 17

### **Лабораторна робота № 11**

Тема: Конструювання об'єктів Класів у С#.

Мета: Засвоєння методики створення конкретних класів з використанням мови програмування С#.

Література: 4, 8, 12, 13.

### **Лабораторна робота № 12**

Тема: Реалізація наслідування.

Мета: Засвоєння методики реалізації наслідування з використанням мови програмування С#.

Література: 2, 10, 12, 17.

### **Лабораторна робота № 13**

Тема: Поліморфізм.

Мета: Засвоєння методики реалізації поліморфізму з використанням мови програмування С#.

Література: 3, 11, 12, 15.

### **Лабораторна робота № 14**

Тема: Перевантаження методів на С#. Використання делегатів.

Мета: Засвоєння методики перевантаження унарних та бінарних операторів для класів користувача.

Література: 2, 10, 12, 17.

## **6. Комплексне практичне індивідуальне завдання**

Індивідуальні завдання з дисципліни виконується самостійно кожним студентом на основі заданого алгоритму. КППЗ охоплює усі основні теми дисципліни.

Метою виконання КППЗ є оволодіння навичками застосування методів побудови об'єктно-орієнтованої моделі предметної області та її програмної реалізації методами об'єктно-орієнтованого програмування. КППЗ оформлюється відповідності з встановленими вимогами. Виконання КППЗ з одним із обов'язкових складових модулів залікового кредиту з дисципліни.

Варіанти КППЗ з дисципліни “Об'єктно-орієнтоване програмування”:

1. Реалізація об'єктно-орієнтованої програми для імітації роботи ліфта.
2. Реалізація об'єктно-орієнтованої програми для імітації роботи музичного центру.
3. Реалізація об'єктно-орієнтованої програми для імітації роботи холодильника.
4. Реалізація об'єктно-орієнтованої програми для імітації роботи пральної машина.
5. Реалізація об'єктно-орієнтованої програми для імітації роботи охоронної системи приміщення (в тому числі - протипожежна).
6. Реалізація об'єктно-орієнтованої програми для імітації роботи телефонного апарату.
7. Реалізація об'єктно-орієнтованої програми для імітації роботи системи забезпечення безпеки руху.
8. Реалізація об'єктно-орієнтованої програми для імітації роботи автоматичної системи контролю стану систем літака.
9. Реалізація об'єктно-орієнтованої програми для імітації роботи автоматичної системи контролю стану систем автомобіля.
10. Реалізація об'єктно-орієнтованої програми для імітації роботи банкомата.
11. Реалізація об'єктно-орієнтованої програми для імітації роботи автомата по продажу цукерок.
12. Реалізація об'єктно-орієнтованої програми для імітації роботи телевізора.
13. Реалізація об'єктно-орієнтованої програми для імітації роботи системи опрацювання текстів.

14. Реалізація об'єктно-орієнтованої програми для імітації роботи мікрохвильової печі.

15. Реалізація об'єктно-орієнтованої програми для імітації роботи турнікета метрополітену.

### 7. Тренінг з дисципліни

Мета тренінгу з дисципліни «Об'єктно-орієнтоване програмування» – сформувати у майбутніх фахівців повне і цілісне уявлення про майбутню професійну діяльність і особистість професіонала; розвивати адекватне розуміння самого себе як майбутнього професіонала і зміцнити професійну самооцінку; сформувати чітке уявлення про професійне майбутнє, оптимізувати життєві плани студентів; сприяти усвідомленню студентами своїх особистісних особливостей і творчих можливостей, унікальності власної Я-концепції.

Успішне проходження тренінгу сприяє посиленню практичної спрямованості у підготовці фахівців за ступенем вищої освіти «бакалавр».

Проведення тренінгу дозволяє:

- Забезпечити засвоєння теоретичних знань, отриманих у процесі вивчення дисципліни «Об'єктно-орієнтоване програмування»;
- Розвинути у студентів навички пропонування обґрунтованих рішень використання теоретичних знань для розв'язання практичних завдань та змістовного інтерпретування отриманих результатів.

Організація і порядок проведення тренінгу

1. Вступна частина. Актуалізація теми тренінгового заняття та структуризація процесу його проведення. Ознайомлення студентів з метою тренінга, його завданнями, процедурою проведення, очікуваними результатами. Представлення програми тренінгу.

2. Організаційна частина. Встановлення правил проведення тренінгу, формування робочих груп студентів, визначення завдань та розподіл ролей. Забезпечення учасників тренінгу роздатковими матеріалами: таблицями, бланками документів, алгоритмами проведення, інструкціями.

3. Практична частина. Виконання тренінгових завдань із використанням базових та інноваційних методів проведення тренінгу за визначеною темою (проблемою). Підготовка презентаційних матеріалів за результатами виконання тренінгового завдання.

4. Підведення підсумків. Презентація практичної роботи в групах. Обговорення результатів виконання завдань, обмін думками з проблематики теми тренінгу, підведення підсумків, оцінка результативності роботи в групах та досягнення поставлених цілей тренінгу.

Рекомендується проведення тренінгу за наступною темою:

1. Принципи об'єктно-орієнтованого програмування.
2. Типових задачі об'єктно-орієнтованого програмування.

### 8. Самостійна робота

№ п/п	Тема	Кількість годин	
		ДФН	ЗФН
1.	Об'єктний підхід в об'єктно-орієнтованому проектуванні.	3	5
2.	Класифікація в об'єктно-орієнтованому аналізі.	3	5
3.	Об'єктно-орієнтоване проектування. Система позначень.	3	5
4.	Діаграми класів.	3	5
5.	Діаграми переходів станів.	3	5
6.	Діаграми об'єктів.	3	5
7.	Часові діаграми.	3	5
8.	Модульні діаграми.	3	5
9.	Діаграми процесів.	3	5
10.	Ідентифікація зв'язків між класами та об'єктами.	3	5
11.	Об'єктно-орієнтоване програмування в життєвому циклі розробки програми.	3	5
12.	Переваги та недоліки об'єктно-орієнтованого програмування.	3	5

13.	Відношення між об'єктами.	3	5
14.	Відношення між класами.	3	5
15.	Взаємозв'язки класів та об'єктів.	3	5
16.	Перевантаження операторів.	2	6
17.	Наслідування.	2	6
18.	Віртуальні функції.	2	7
19.	Абстрактні класи.	1	7
20.	Контроль доступу до членів класу.	1	7
	<b>Разом:</b>	<b>53</b>	<b>108</b>

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 опрацьованні навчальної і наукової фахової літератури

### 9. Методи навчання

У навчальному процесі застосовуються: лекції, практичні заняття, консультації, самостійна робота, індивідуальна робота, робота у групах, метод опитування, тестування, ситуативне моделювання, ділові ігри, реферування, виконання КППЗ, есе, підготовка і презентація проектів.

### 10. Засоби оцінювання та методи демонстрування результатів навчання

У процесі вивчення дисципліни «Об'єктно-орієнтоване програмування» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- стандартизовані тести;
- поточне опитування;
- залікове модульне тестування та опитування;
- аналітичні звіти, реферати, есе;
- розрахункові та розрахунково-графічні роботи;
- презентації результатів виконаних завдань та досліджень;
- оцінювання результатів КППЗ;
- студентські презентації та виступи на наукових заходах;
- розрахункові роботи;
- завдання на лабораторному обладнанні, тренажерах, реальних об'єктах тощо;
- ректорська контрольна робота;
- комплексний іспит;
- екзамен;
- інші види індивідуальних та групових завдань.

### 11. Політика оцінювання

*Політика щодо дедлайнів і перескладання.* Для виконання усіх видів завдань студентами і проведення контрольних заходів встановлюються конкретні терміни. Перескладання модулів проводиться у встановленому порядку.

*Політика щодо академічної доброчесності.* Списування під час проведення контрольних заходів заборонені. Під час контрольного заходу студент може користуватися лише дозволеними допоміжними матеріалами або засобами, йому забороняється в будь-якій формі обмінюватися інформацією з іншими студентами, використовувати, розповсюджувати, збирати варіанти контрольних завдань.

*Політика щодо відвідування.* Відвідування занять є обов'язковим. За об'єктивних причин (наприклад, карантин, військовий стан, хвороба, закордонне стажування) навчання може відбуватись в дистанційній формі за погодженням із керівником курсу з дозволу дирекції інституту.

## 12. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Об'єктивно-орієнтоване програмування» визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

Заліковий модуль 1	Заліковий модуль 2	Заліковий модуль 3	Заліковий модуль 4
20%	20%	20%	40%
Опитування під час заняття (теми 1-12) – 3 бали за тему – макс. 35 балів. Модульна робота – макс. 65 балів.	Опитування під час заняття (теми 13-24) – 3 бали за тему – макс. 35 балів. Модульна робота – макс. 65 балів.	Підготовка КПІЗ – макс. 40 балів. Захист КПІЗ – макс. 40 балів. Виконання завдань під час тренінгу – макс. 20 балів.	Тестові завдання (10 тестів по 2 бали за тест) – макс. 20 балів. Задачі (2 задачі) – по 30 балів, макс. 60 балів. Теоретичне питання – макс. 20 балів.

### Шкала оцінювання:

За шкалою ЗУНУ	За національною шкалою	За шкалою ECTS
90–100	відмінно	A (відмінно)
85–89	добре	B (дуже добре)
75-84		C (добре)
65-74	задовільно	D (задовільно)
60-64		E (достатньо)
35-59	незадовільно	FX (незадовільно з можливістю повторного складання)
1-34		F (незадовільно з обов'язковим повторним курсом)

## 13. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1.	Мультимедійний проектор	1-24
2.	Проекційний екран	1-24
3.	Комунікаційне програмне забезпечення (Internet Explorer, Google Chrome, Firefox)	1-24
4.	Комунікаційне програмне забезпечення (Zoom) для проведення занять у режимі он-лайн (за необхідності)	1-24
5.	Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності)	1-24
6.	Програмне забезпечення: ОС Windows	1-24
7.	Інструменти Microsoft Office (Word, Excel, Power Point і т.д.)	1-24
8.	Вихідні дані для обробки на ПК	1-24

## РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Геннадій Галисєєв. Системне програмування: Видавництво Університет «Україна», 2019.
2. Джордж Хайнеман, Гері Полліс, Стенлі Селков. Алгоритми. Довідник з прикладами на C, C ++, Java і Python. Діалектика, 2017. 432 с.
3. Марченко А.Л. Навчальний посібник C#. Введення у програмування. URL: <http://ict.edu.ru/ft/005123/marchenko.pdf>
4. Пекарський Б.Г. Основи програмування: Навчальний посібник. Кондор, 2018. 364 с. Сайт з найновішою довідкою по C++ . URL: <http://www.cplusplus.com/>
5. Сайт з ресурсами по вивченню C++. URL: <http://cppstudio.com> Andrew Stellman. Head First C#. 4th edition, O'Reilly Media, Inc. 2020.
6. C# OOP (Object-Oriented Programming) W3Schools. URL: [https://www.w3schools.com/cs/cs\\_oop.php](https://www.w3schools.com/cs/cs_oop.php)
7. C/C++ language and standard libraries reference. URL: <https://msdn.microsoft.com/en-us/library/hh875057.aspx>.
8. C++ Crash Course: A Fast-Paced Introduction.. Lospinoso Josh. ISBN 1593278885. 2019. 792с.
9. International Standard ISO/IEC 14882:2014(E) Programming Language C++, ISBN13: 978-0321563842: URL: <https://isocpp.org/std/thestandard>.
10. Joe Mayo. C# Cookbook: Modern Recipes for Professional Developers. 1st Ed.
11. Josph Albahari. C# 8.0 Pocket Reference. 1st edition, O'Reilly LCF Publishing. C#. 1st edition, Learn Coding Fast
12. Learn C#. Free courses, tutorials, videos, and more about learning the programming language C#. URL: <https://dotnet.microsoft.com/en-us/learn/csharp>
13. Mark J. Price C# 8.0 and .NET Core 3.0. 4th edition. Packt Publishing
14. Mikael Olsson. C# 10 Quick Syntax Reference. 4th Ed. Apress. 2022
15. Object-Oriented programming (C#). URL: <https://learn.microsoft.com/enus/dotnet/csharp/fundamentals/tutorials/oo>
16. Syntax Reference. 4th Ed. Apress. 2022. URL: <http://cppstudio.com> Andrew Stellman. Head First C#. 4th edition, O'Reilly Media
17. Stroustrup Bjarne. A Tour of C++, Second Edition. July 20, 2018