

Чернівецький національний університет імені Юрія Федьковича
Навчально-науковий інститут фізико-технічних та комп'ютерних наук
Кафедра комп'ютерних наук

«ЗАТВЕРДЖУЮ»
директор навчально-наукового
інституту фізико-технічних та
комп'ютерних наук,
д.ф.м.н., професор **О.В. Ангельський**
«28» вересня 2023р.



РОБОЧА ПРОГРАМА
навчальної дисципліни
Об'єктно-орієнтоване програмування
(назва навчальної дисципліни)
обов'язкова
(вказати: обов'язкова / вибіркова)

Освітньо-професійна програма «Інтелектуальний аналіз даних в комп'ютерних інформаційних системах»
(назва програми)
Спеціальність 122 «Комп'ютерні науки»
(вказати: код, назва)
Галузь знань 12 «Інформаційні технології»
(вказати: шифр, назва)
Рівень вищої освіти: перший (бакалаврський)
(вказати: перший бакалаврський / другий магістерський)
навчально-науковий інститут фізико-технічних та комп'ютерних наук
(назва факультету / навчально-наукового інституту,
на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)
мова навчання: українська
(вказати, на якій мові читається навчальна дисципліна)

Робоча програма навчальної дисципліни «Об’єктно-орієнтоване програмування» складено відповідно до освітньо-професійної програми «Інтелектуальний аналіз даних в комп’ютерних інформаційних системах» спеціальності 122 «Комп’ютерні науки» галузі знань 12 «Інформаційні технології», затвердженої вченою радою Чернівецького національного університету імені Юрія Федьковича (Протокол №10 від «26» вересня 2023 року)

Розробник: Томка Ю.Я., доцент кафедри комп’ютерних наук, к.ф.-м.н., доцент.

Затверджено на засіданні кафедри комп’ютерних наук

Протокол № 3 від “27” вересня 2023 року

Завідувач кафедри



Ю.О.Ушенко

Схвалено методичною радою навчально-наукового інституту фізико-технічних та комп’ютерних наук

Протокол № 2 від “28” вересня 2023 року

Голова методичної ради інституту
фізико-технічних та комп’ютерних наук



Я.М. Струк

©ЧНУ, 2023 рік

©ЧНУ, 2024 рік

1. Мета навчальної дисципліни:

Метою викладання дисципліни «Об'єктно-орієнтоване програмування» є: засвоєння необхідних знань із основ об'єктно-орієнтованого програмування, отримання навичок використання класів, механізмів наслідування, інкапсуляції та поліморфізму; отримання практичних навичок самостійно будувати програми середнього рівня складності з використанням структурно-модульного та об'єктно-орієнтованого методів програмування; застосування здобутих знань та підходів для розв'язання практичних задач різного рівня складності.

2. Результати навчання

Дисципліна «Об'єктно-орієнтоване програмування» є частиною освітньо-професійної програми «Інтелектуальний аналіз даних в комп'ютерних інформаційних системах» першого рівня вищої освіти за спеціальністю №122 «Комп'ютерні науки» галузі знань №12 «Інформаційні технології».

Під час вивчення даної дисципліни студенти набудуть:

I. Загальних компетентностей:

- 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.
- ЗК5.Здатність спілкуватися іноземною мовою.
- ЗК6.Здатність вчитися і оволодівати сучасними знаннями.
- ЗК7.Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК8.Здатність генерувати нові ідеї (креативність).
- ЗК10.Здатність бути критичним і самокритичним.
- ЗК11.Здатність приймати обґрунтовані рішення.
- ЗК12.Здатність оцінювати та забезпечувати якість виконуваних робіт.

II. Спеціальних (фахових, предметних) компетентностей:

- СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.
- СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальнення, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління.
- СК12. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення

III. Програмними результатами навчання є:

- ПР01. 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.
- ПР05. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основ застосування формальних моделей алгоритмів та обчислюваних функцій;
- ПР09. 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;
- ПР14. 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

Основними завданнями курсу «Об'єктно-орієнтоване програмування» є оволодіння кожним студентом процесом проектування задачі і її реалізації як системи взаємодіючих класів, що діють завдяки інкапсуляції, спадкуванню та поліморфізму, а також іншим засобам мови програмування C#.

В результаті вивчення курсу студент повинен:

знати:

- концепції та принципи організації програмного коду;
- рівні абстракції даних при проектуванні програмного забезпечення (ПЗ);
- основи роботи багатопоточних систем (multi threading)
- вимоги до організації об'єктної моделі і методи її забезпечення;
- загальні принципи побудови об'єктно-орієнтованої моделі ПЗ і принципи її функціонування;
- принципи організації структур збереження даних і методи доступу до даних;
- відмінності між об'єктно-орієнтованим і функціональним програмуванням;
- етапи проектування об'єктно-орієнтованої моделі
- принципи взаємодії об'єктів, які виконані на мові високого рівня C#
- основи роботи в середовищі розробки програмного забезпечення Microsoft Visual Studio

вміти:

- формулювати вимоги до об'єктно-орієнтованої моделі і забезпечувати її властивості;
- розробляти об'єктно-орієнтовану модель у процесі технічного проектування на основі оціночних варіантів програмного

забезпечення, вимог користувачів, аналізу технічних, економічних, функціональних, сервісних характеристик ПЗ, використовуючи науково-технічну, довідкову інформацію;

- розробляти логічну структуру об'єктної моделі ПЗ у процесі технічного проектування за допомогою методу нормалізації відношень, використовуючи рівні абстракції даних, вимоги технічного завдання;
- розробляти класи і зв'язок між ними в умовах технічного проектування за допомогою відповідного технічного і програмного забезпечення, використовуючи майстер створення нових сутностей середовища розробки програмного забезпечення Microsoft Visual Studio;
- створювати віконне та консольне ПЗ;
- створювати окремі компоненти ПЗ;
- створювати робочі області (namespaces), класи (classes), конструктори (constructors), властивості (properties), методи (methods в умовах розробки ПО за допомогою середовища розробки програмного забезпечення Microsoft Visual Studio, використовуючи візуальні інструменти інтегрованих оболонок;
- розподіляти області видимості сутностей класу.

3. Опис навчальної дисципліни

3.1. Загальна інформація

Назва навчальної дисципліни ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ												
Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю
			кредитів	годин	змістових модулів	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	2	3	5	150	3	30	-	-	45	75	-	екзамен

3.2. Дидактична карта навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7
Теми лекційних занять	Змістовий модуль 1. Введення в C# та .NET платформу. Головні програмні конструкції					
Тема 1. Введення в C# та .NET платформу	9	2	-	2	-	5
Тема 2. Головні конструкції програмування на мові C#	14	2	-	2	-	10

Разом за 3М 1	23	4	-	4	-	15
Теми лекційних занять	Змістовий модуль 2. Основи об'єктно-орієнтованого програмування. Поглиблені можливості мови C#					
Тема 3. Основні поняття про клас, методи та конструктори	9	2	-	4	-	3
Тема 4. Інкапсуляція	7	2		2	-	3
Тема 5. Поняття наслідування та поліморфізм	7	2	-	2	-	3
Тема 6. Обробка виключень	7	2	-	2	-	3
Тема 7. Робота із інтерфейсами	9	2	-	4	-	3
Тема 8. Колекції та дженеріки	7	2	-	2	-	3
Тема 9. Делегати, події та лямбда-вирази. Івенти	7	2	-	2	-	3
Тема 10. Рефлексія типів, пізнє зв'язування та програмування із використанням атрибутів. Regex	9	2	-	4	-	3
Тема 11. LINQ to Objects	9	2	-	4	-	3
Тема 12. Багатопотокове, паралельне та асинхронне програмування засобами C#	7	2	-	2	-	3
Разом за 3М2	78	20	-	28	-	30
Теми лекційних занять	Змістовий модуль 3. Об'єктно-орієнтовані концепції роботи із базами даних					
Тема 13. SOLID принципи об'єктно-орієнтованого програмування	9	1	-	-	-	5
Тема 14. Вступ до Entity Framework	8	1	-	2	-	5
Тема 15. Створення моделей у EntityFramework	8	1	-	2	-	5
Тема 16. Відношення між моделями	10	1	-	4	-	5
Тема 17. Запити Linq to Entities	11,5	1,5	-	5	-	5
Тема 18. Робота із провайдерами баз даних.	5,5	0,5	-	-	-	5
Разом за 3М3	49	6	-	13	-	30
ВСЬОГО	150	30	-	45	-	75

3.3. Теми лабораторних занять

№	Назва теми
01.	C#. Introduction to Programming/Вступ до програмування на мові C#
02.	Arrays/Масиви
03.	Defining Classes/Визначення класів
04.	Working with Abstraction/робота з абстракціями
05.	Encapsulation/Інкапсуляція
06.	Inheritance/Наслідування
07.	Interfaces And Abstraction/Інтерфейси та абстракції
08.	Generics/Дженеріки
09.	Iterators and Comparators/Ітератори та компаратори
10.	Reflection And Attributes/Рефлексія та атрибути
11.	Communications and Events/Комунікація та події
12.	Регулярні вирази/Regular Expressions
13.	Списки, словники та LINQ/Strings Dictionaries Lambda LINQ
14.	Розширені можливості LINQ/ LINQ

15.	Робота із виключеннями
16.	Робота із файлами та фаловою системою.
17.	SOLID
18.	ENTITY FRAMEWORK.Introduction to Entity Framework
19.	ENTITY FRAMEWORK.Code-First
20.	ENTITY FRAMEWORK.Relations
21.	ENTITY FRAMEWORK.Relations Advanced Relations
22.	ENTITY FRAMEWORK.Querying. Advanced Quering/

3.4. Самостійна робота

№	Назва теми
1	Введення в C# та .NET платформу - Введення в будівельні блоки платформи .NET (CLR, CTS та CLS) - Огляд збірок .NET - Розуміння загальної системи типів - Розуміння загальномовної специфікації .NET додатки на Windows/Non-Windows OS
2	Головні конструкції програмування на C# - Структура простої програми на C# - Клас System.Console - Системні типи даних та відповідні ключові слова на C# - Робота із рядкам - Службові та розширюючі перетворення типів даних - Поняття неявно типізованих локальних змінних - Ітераційні конструкції C# - Конструкції прийняття рішень та операції рівності/порівняння - Масиви C# - Методи та модифікатори параметрів - Тип enum - Типи структур - Типи значень та зсилочні типи даних - Поняття типів, що допускають null - Кортежі
3	Основні поняття про клас, методи та конструктори - Класи та об'єкти - Конструктори, ініціалізатори та деконструктори - Клас Program та метод Main. Програми верхнього рівня - Структури - Типи значень та типи посилань - Область видимості (контекст) змінних - Простір імен - Глобальні простори імен - Підключення просторів за промовчанням - Створення бібліотеки класів у Visual Studio - Створення бібліотеки класів за допомогою .NET CLI - Модифікатори доступу - Властивості - Перевантаження методів - Статичні члени та модифікатор static - Встановлення пакетів Nuget - Константи, поля та структури для читання

	• Null та посилання типи • Null та значущі типи • Перевірка на null, оператори?. та ?? • Псевдоніми типів та статичний імпорт
4	Інкапсуляція • Знайомство з типом класу C# • Поняття конструктора • Роль ключового слова this • Поняття ключового слова static • Основні принципи ООП • Модифікатори доступу C# • Перший принцип: служби інкапсуляції C# • Поняття автоматичних властивостей • Поняття синтаксису ініціалізації об'єктів • Робота з даними константних полів • Поняття часткових типів
5	Поняття наслідування та поліморфізм • Базовий механізм наслідування • Другий принцип ООП: деталі наслідування • Реалізація моделі включення / делегації • Третій принцип ООП: підтримка поліморфізму у C# • Розуміння принципів створення базового класу/успадкованого класу • Головний батьківський клас: System.Object
6	Структурована обробка виключень • Роль обробки виключень .NET • Конфігурування стану виключення • Виключення рівня системи (System.ApplicationException) • Обробка декількох виключень
7	Робота з інтерфейсами • Поняття інтерфейсних типів • Визначення спеціальних інтерфейсів • Реалізація інтерфейсу • Виклик членів інтерфейсу на рівні об'єкту • Інтерфейси в якості параметрів • Інтерфейси в якості повертаємих значень • Реалізація інтерфейсів у Visual Studio • Інтерфейси IEnumerable та IEnumerator • The ICloneable Interface • The IComparable Interface
8	Колекції та узагальнення • Причини для створення класів колекцій • Проблеми пов'язані із неузагальненими колекціями • Роль параметрів узагальнених типів • The System.Collections.Generic Namespace • The System.Collections.ObjectModel Namespace • Створення спеціальних узагальнених методів • Створення спеціальних узагальнених структур та класів • Обмеження параметрів типу
9	Делегати, події та лямбда-вирази • Поняття типу делегату .NET • Приклад найпростішого делегату • Відправка повідомлень про стан об'єкту з використанням делегатів • Поняття узагальнених делегатів

	- Розуміння подій у C# - Розуміння анонімних методів у C# - Розуміння лямбда-виразів
10	Рефлексія типів, пізнє зв'язування та програмування із використанням атрибутів - Необхідність у метаданих типів - Розуміння рефлексії - Побудова Custom Metadata Viewer - Роль атрибутів у .NET - Побудова спеціальних атрибутів
11	LINQ to Objects - LINQ-специфічні програмні конструкції - Розумінні ролі LINQ - Застосування LINQ запитів до примітивних масивів - Повернення результату запиту - Застосування LINQ запитів до колекції об'єктів - Дослідження операторів запиту LINQ у C# - Внутрішнє представлення операторів запиту LINQ
12	Багатопоточне, паралельне та асинхронне програмування - Відношення між процесором, доменом додатку, контекстом та потоком - Короткий огляд делегатів .NET - Асинхронна природа делегатів - Асинхронний виклик методу - Простір імен System.Threading - Клас System.Threading.Thread - Ручне створення вторичних потоків - Проблеми паралелізму - Parallel LINQ Queries (PLINQ) - Асинхронні виклики
13	SOLID принципи об'єктно-орієнтованого програмування - Принцип єдиного обов'язку (SRP) - Принцип відкритості/закритості (OCP) - Принцип підстановки Лісков (LSP) - Принцип розділення інтерфейсу (ISP) - Принцип інверсії залежностей (DIP)
14	Вступ до Entity Framework - Що таке Entity Framework Core - Перший додаток на EF Core - Підключення до існуючої бази даних - Основні операції із даними. CRUD - Конфігурація підключення - Логування операцій - Провайдери логування
15	Створення моделей у EntityFramework - Моделі, Fluent API та анотації даних - Управління схемою БД та міграції - Визначення моделей - Властивості моделей - Зіставлення таблиць та стовпців - Налаштування ключів - Налаштування індексів - Генерація значень властивостей та стовпців - Обмеження властивостей - Типи даних

	• Конфігурація моделей • Ініціалізація бази даних початковими даними
16	Відношення між моделями • Зовнішні ключі та навігаційні властивості • Каскадне видалення • Завантаження пов'язаних даних. Метод Include • Explicit loading • Lazy loading • Ставлення один до одного • Ставлення один до багатьох • Ставлення багато хто до багатьох • Комплексні типи
17	Запити Linq to Entities • Введення в LINQ to Entities • Вибірка та фільтрація • Сортування та проекція з бази даних • З'єднання та угруповання таблиць • Операції з безліччю: об'єднання, перетин, різниця • Агрегатні операції • Відстеження об'єктів та AsNoTracking • Виконання запитів • IEnumerable та IQueryable • Фільтри запитів рівня моделі
18	Робота із провайдерами баз даних. • Виконання SQL-запитів • Функції, що зберігаються • Збережені процедури • PostgreSQL • MySQL

3.5. Тематика індивідуальних завдань

№	Назва теми
1.	Додаткові можливості ООП у C# • Визначення операторів • Перевантаження операцій перетворення типів • Індексатори • Змінні-посилання та повернення посилання • Методи розширення • Часткові класи та методи • Анонімні типи • Кортежі • Records
2	Робота з датами та часом • Структура DateTime • Форматування дат та часу • DateOnly та TimeOnly
3	Додаткові класи та структури .NET • Відкладена ініціалізація та тип Lazy • Математичні обчислення та клас Math • Перетворення типів та клас Convert • Клас Array та масиви • Span • Індeksi та діапазони

4	Багатопоточність • Введення у багатопоточність. Клас Thread • Створення потоків. Делегат ThreadStart • Потоки з параметрами та ParameterizedThreadStart • Синхронізація потоків • Монітори • Клас AutoResetEvent • М'ютекси • Семафори
5	Паралельне програмування та бібліотека TPL • Завдання та клас Task • Робота з класом Task • Завдання продовження • Клас Parallel • Скасування завдань та паралельних операцій. CancellationToken
6	Асинхронне програмування. Task-based Asynchronous Pattern • Асинхронні методи, async та await • Повернення результату з асинхронного методу • Послідовне та паралельне виконання. Task.WhenAll та Task.WhenAny • Обробка помилок у асинхронних методах • Асинхронні стрими
7	Робота з JSON. Робота з XML C# • Сериалізація у JSON. JsonSerializer • XML-Документи • Робота з XML за допомогою класів System.Xml • Зміна документа XML • XPath • Linq to Xml. Створення XML-документа • Вибір елементів у LINQ to XML • Зміна XML-документа в LINQ to XML • Сериалізація у XML. XmlSerializer
8	Збірка сміття, управління пам'яттю та вказівники • Збірка сміття в C# • Фіналізовані об'єкти. Метод Dispose • Конструкція using • Вказівники • Вказівники на структури, члени класів та масиви
9	Робота з файловою системою • Робота з дисками • Робота з каталогами • Робота із файлами. Класи File та FileInfo • FileStream. Читання та запис файлу • Читання та запис текстових файлів. StreamReader та StreamWriter • Бінарні файли. BinaryWriter та BinaryReader • Архівація та стиснення файлів
10	Практичні аспекти. SOLID принципи об'єктно-орієнтованого програмування • Принцип єдиного обов'язку (SRP) • Принцип відкритості/закритості (OCP) • Принцип підстановки Лісков (LSP) • Принцип розділення інтерфейсу (ISP) • Принцип інверсії залежностей (DIP)

4. Критерії оцінювання результатів навчання з навчальної дисципліни

Контроль та оцінювання результатів навчання здійснюється згідно Положення про контроль і систему оцінювання результатів навчання здобувачів вищої освіти у Чернівецькому національному університеті імені Юрія Федьковича (https://www.chnu.edu.ua/media/geupxdun/polozhennia-pro-kontrol-i-systemu-otsiniuvannia_2020.pdf).

Поточний модульний контроль (ПМК) здійснюється під час проведення лекційних, практичних та індивідуально-консультативних занять з метою перевірки рівня засвоєння теоретичних знань та практичних навичок студента. ПМК проводиться у формі написання письмових робіт, проміжних тестувань та активності й влучності обговорення відповідних тем під час навчальних занять. Згідно з навчальним планом семестровий контроль з дисципліни «Об'єктно-орієнтовне програмування» відбувається у формі іспиту.

Відвідування занять. Відсутність на аудиторному занятті не передбачає нарахування штрафних балів, оскільки фінальний рейтинговий бал студента формується виключно на основі оцінювання результатів навчання. Разом з тим, обговорення результатів виконання тематичних завдань, а також презентація / публічний виступ та участь у обговореннях та доповнення на практичних заняттях оцінюватимуться під час аудиторних занять.

Оцінювання пропущених контрольних заходів. Кожен студент має право відпрацювати пропущені з поважної причини (лікарняний, мобільність тощо) заняття за рахунок самостійної роботи.

Процедура оскарження результатів контрольних заходів оцінювання. Студент може підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами Положенням про апеляцію на результати підсумкового семестрового контролю знань студентів Чернівецького національного університету імені Юрія Федьковича (<https://www.chnu.edu.ua/media/h0fn0fgh/polozhennia-pro-apeliatsiiu.pdf>).

Академічна доброчесність. Політика та принципи академічної доброчесності визначені Етичним кодексом Чернівецького національного університету імені Юрія Федьковича (<https://www.chnu.edu.ua/media/xel1ulcg/etychnyi-kodeks-chernivetskoho-natsionalnoho-universytetu.pdf>).

Інклюзивне навчання. Засвоєння знань та умінь в ході вивчення дисципліни «Об'єктно-орієнтовне програмування» може бути доступним для більшості осіб з особливими освітніми потребами, окрім здобувачів з серйозними вадами зору, які не дозволяють виконувати завдання за допомогою персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

Навчання іноземною мовою. У ході виконання завдань студентам може бути рекомендовано звернутися до англomовних джерел.

Підготовка до практичних занять та контрольних заходів здійснюється під час самостійної роботи студентів з можливістю консультування з викладачем у визначений час консультацій або за допомогою електронного листування (електронна пошта, месенджери).

Неформальна освіта. Регулюється Положенням про взаємодію формальної та неформальної освіти (<https://www.chnu.edu.ua/media/3aykf41y/polozhennia-pro-vzaiemodiiu-formalnoi-ta-neformalnoi-osvity.pdf>). При наявності сертифікатів про проходження професійно-спрямованих психолого-педагогічних курсів, тренінгів, майстер-класів з неформальної освіти, участь у неформальній освіті студентам може бути перезараховано до 35% змістового матеріалу, що відповідає прослуханому матеріалу, за умови підготовки ними презентацій та нотаток за матеріалами прослуханого курсу, чи веб-заходу та їх публічного захисту на практичних заняттях.

Також, як можливості неформальної освіти студентам під час вивчення курсу «Об'єктно-орієнтовне програмування» пропонується проходження курсів з отриманням сертифікатів, як індивідуальне завдання (ІНДЗ). На один модуль не більше 10 балів при 100% (86 – 80-99%, 66 – 60-79% тощо) проходженні курсів на віртуальних платформах (Prometheus, Coursera та інші).

Дуальна освіта. За умови роботи в компанії чи на підприємстві: теоретична частина дисципліни слухається та оцінюється на кафедрі; практична частина дисципліни перезараховується при отриманні сертифікату/ів про проходження професійно-спрямованих, відповідно дисципліни, курсів на підприємстві чи інших підтверджуючих документів.

Критерії оцінювання результатів навчання з навчальної дисципліни

Екзамен проводиться у формі, визначеній кафедрою. Екзаменаційний білет включає теоретичні і практичні завдання та містить 4 завдання (2 теоретичних питання, 2 практичні задачі). Результати екзамену оцінюються за національною чотирибальною шкалою. Максимальна кількість балів на екзамені за шкалою ВНЗ становить 40 балів.

Загальна підсумкова оцінка з дисципліни складається з суми балів за результатами ПМК та за виконання завдань, що виносяться на екзамен.

Якщо студент на екзамені отримав незадовільну оцінку, то це вважається як академічна заборгованість і набрані бали не заносяться до відомості. За графіком деканату студент перескладає екзамен і його результати заносяться до окремої відомості.

За результатами складання екзамену студент отримує:

27,0-30,0 бала – дана розгорнута вичерпна відповідь на теоретичні питання та правильно виконане практичне завдання;

24,0-26,0 бала – студентом допущені незначні помилки у відповіді на теоретичні питання чи практичне завдання;

21,0-23,0 бала – студент допустив значні помилки у відповіді на одне з теоретичних питань чи в практичному завданні;

18,0-20,0 бала – студент, допустивши значні помилки, не дав чіткої відповіді на теоретичні питання, не в повному обсязі виконав практичне завдання;

15,0-17,0 бала – студент не дав відповіді на одне із теоретичних питань, практичне завдання виконане не в повному обсязі;

10,5-14,0 бала – студент не дав відповіді на одне із теоретичних питань, не виконане практичне завдання.

0,1-10,3 бала – студент виконав частину одного з теоретичних питань, практичне завдання не виконане.

0 балів – студент не з'явився на екзамені.

Шкала оцінювання результатів екзамену

Підсумкова кількість балів за екзамен	Оцінка за національною шкалою	Оцінка за шкалою ECTS
27 - 30	5 (відмінно)	A (відмінно)
24 – 26	4 (добре)	B (дуже добре)
21 – 23		C (добре)
18 – 20	3 (задовільно)	D (задовільно)
15 – 17		E (достатньо)
10,5 – 14	2 (незадовільно) з можливістю повторного складання	FX
0 – 10,4	2 (незадовільно) з обов'язковим повторним курсом	F

Шкала оцінювання результатів навчальних досягнень при вивченні дисципліни

Кількість балів за 100 бальною шкалою (<i>max- 100 балів</i>)	Підсумкова оцінка за національною шкалою (<i>max- 5 балів</i>)	Підсумкова оцінка за шкалою ECTS
90-100	5 (відмінно)	A
80-89	4 (добре)	B
70-79	4 (добре)	C
60-69	3 (задовільно)	D
50-59	3 (задовільно)	E
35-49	2 (незадовільно) (з можливістю складання іспиту)	FX
1-34	2 (незадовільно) (з додатковим вивченням дисципліни)	F

Розподіл балів, які отримують студенти

Поточне оцінювання (аудиторна та самостійна робота)									ІНДЗ	Кількість балів (залік)	Загальна кількість балів	
Змістовий модуль № 1		Змістовий модуль № 2			Змістовий модуль № 3					-	30	100
T1	T2	T4, T5, T6, T8, T9, T12	T3, T7, T10, T11		T13	T14	T15	T16	T17			
5	5	по 3б	по 6б		-	3	3	6	6			
10б		42б			18б							

*T1, T2 ... T15 – теми змістових модулів.

5. Засоби оцінювання

Засобами оцінювання та демонстрування результатів навчання можуть бути:

- контрольні роботи;
- стандартизовані тести;
- проєкти (наскрізні проєкти; індивідуальні та командні проєкти; дослідницько-творчі тощо);
- аналітичні звіти;
- реферати;
- розрахункові, графічні, розрахунково-графічні роботи;
- презентації результатів виконаних завдань;
- студентські презентації та виступи на наукових заходах;
- виконання завдання на лабораторному обладнанні, тренажерах, реальних об'єктах тощо;
- інші види індивідуальних та групових завдань.

6. Форми поточного та підсумкового контролю

Формами поточного контролю є усна або письмова відповідь у формі :

- тестування,
- есе,
- реферат,
- творча робота,
- звіт про виконання лабораторної роботи
- тощо.

7. Рекомендована література

7.1. Базова (основна)

1. Fundamentals: Start building websites and services with ASP.NET Core 7, Blazor, and EF Core 7 / Mark J. Price. – Birmingham, United Kingdom: Packt Publishing Limited. – 818 p.
2. Mark J. Price. Apps and Services with .NET 7: Build practical projects with Blazor, .NET MAUI, gRPC, GraphQL, and other enterprise technologies / Mark J. Price. – Birmingham, United Kingdom: Packt Publishing. – 814 p.

3. Andrew Stellman. Head First C#: A Learner's Guide to Real-World Programming with C# and .NET Core / Andrew Stellman, Jennifer Greene, 2021. – 800 p.
4. Andrew Troelsen. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming / Andrew Troelsen, Phil Japikse. – Berkley, United States: Apress, 2022. – 1705 p.
5. Joseph Albahari. C# 10 in a Nutshell: The Definitive Reference / Joseph Albahari., 2022. – 1058 p.
6. Ian Griffiths. Programming C# 10: Build Cloud, Web, and Desktop Applications / Ian Griffiths, 2022. – 833 p.
7. Benjamin Perkins. Beginning C# and .NET / Benjamin Perkins, Jon D. Reid., 2021. – 824 p.
8. Edmund Krogman. Learning Oop With C#: Tutorial And Explains Deep Core Concepts Of Oop / Edmund Krogman, 2022. – 408 p.
9. Raihan Taher. Hands-On Object-Oriented Programming with C#: Build maintainable software with reusable code using C# / Raihan Taher., 2019. – 290 p.
10. Troelsen A. Pro C# 8 with .NET Core 3 Foundational Principles and Practices in Programming / A. Troelsen, J. Japikse. – Berkley, United States: Apress, 2020. – 1160 p.
11. Freeman A. Pro ASP.NET Core 3 (Develop Cloud-Ready Web Applications Using MVC 3, Blazor, and Razor Pages) / Adam Freeman., 2020. – 1400 p.
12. Bipin J. Beginning Database Programming Using ASP.NET Core 3: With MVC, Razor Pages, Web API, jQuery, Angular, SQL Server, and NoSQL / Joshi Bipin. – Berkley, United States: Apress, 2019. – 481 p.
13. Baptista G. Hands-On Software Architecture with C# 8 and .NET Core 3: Architecting software solutions using microservices, DevOps, and design patterns for Azure Cloud / Gabriel Baptista. – Birmingham, UK: Packt Publishing, 2019. – 598 p.
14. J. Price M. C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and ML.NET using Visual Studio Code / Mark J. Price. – Birmingham, UK: Packt Publishing, 2019. – 818 p.
15. Adam Tibi. Pragmatic Test-Driven Development in C# and .NET: Write loosely coupled, documented, and high-quality code with DDD using familiar tools and libraries / Adam Tibi. – Birmingham, United Kingdom: Packt Publishing, 2022. – 372 p.
16. Mark Reed. C#: 3 books in 1 - The Ultimate Beginner, Intermediate & Advanced Guides to Master C# Programming Quickly with No Experience (Computer Programming) / Mark Reed., 2022. – 344 p.
17. Dan Clark. Beginning C# Object-Oriented Programming (Expert's Voice in .NET) / Dan Clark., 2013. – 408 p.
18. Joseph Albahari. C# 8.0 Pocket Reference: Instant Help for C# 8.0 Programmers / Joseph Albahari., 2019. – 241 p.

7.2. Допоміжна

1. Об'єктно-орієнтоване програмування мовою C# / Ю.Я. Томка. Чернівці: Технодрук, 2022. 504с. <https://archer.chnu.edu.ua/xmlui/handle/123456789/6704>

2. Збірник тестових завдань з дисциплін «Об'єктно-орієнтоване програмування» та «Розробка додатків на платформі .NET». Частина 1. / Ю.Я. Томка. Чернівці: Технодрук, 2022. 264с. <https://archer.chnu.edu.ua/xmlui/handle/123456789/6706>
3. Збірник тестових завдань з дисциплін «Об'єктно-орієнтоване програмування» та «Розробка додатків на платформі .NET». Частина 2. / Ю.Я. Томка. Чернівці: Технодрук, 2022. 184с. <https://archer.chnu.edu.ua/xmlui/handle/123456789/6707>
4. Senthilvel G. Enterprise Application Architecture with .NET Core: An architectural journey into the Microsoft .NET open source platform / G. Senthilvel, O. Khan, H. Qureshi. – Birmingham, UK: Packt Publishing, 2017. 566 с.
5. C. Martin R. Clean Architecture: A Craftsman's Guide to Software Structure and Design, First Edition / Robert C. Martin. London, England: Pearson, 2017. 428 с.
6. Gammelgaard C. Microservices in .NET Core: with examples in Nancy / Christian Horsdal Gammelgaard. Shelter Island, New York: Manning Publications, 2017. 344 с.
7. Mohapatra B. Microservices by examples using .NET Core: A book with lot of practical and architectural styles for Microservices using .NET Core / B. Mohapatra, B. Banerjee, G. Aroraa. India: BPB PUBLICATIONS, 2019. 239 с.
8. Himschoot P. Blazor Revealed: Building Web Applications in .NET / Peter Himschoot. – Berkley, United States: Apress, 2019. 247 с.
9. Litvinavicius T. Exploring Blazor Creating Hosted, Server-side, and Client-side Applications with C# / Taurius Litvinavicius. Berkley, United States: Apress, 2019. 199 с.

8. Інформаційні ресурси

1. The Unified Modeling Language. [Електронний ресурс] – Режим доступу: <https://www.uml-diagrams.org/>
2. Использование GitHub с GitHub Desktop. [Електронний ресурс] – Режим доступу: <https://o7planning.org/ru/10283/using-github-with-github-desktop>
3. База знань Microsoft Developer Network (MSDN) [Електронний ресурс]. – Режим доступу : <http://msdn.microsoft.com/ru-ru>.