

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА**

**Кафедра інформаційних технологій**

**«ЗАТВЕРДЖУЮ»**

Гарант освітньої програми

РМ Роман ЛПЦУК

“ 08 ” 08 2024 р.

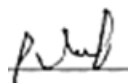
**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
**"Об'єктно-орієнтоване програмування і моделювання"**

**Освітній рівень:** Другий рівень вищої освіти  
**Галузь знань:** 12 "Інформаційні технології"  
**Спеціальність:** 122 "Комп'ютерні науки"  
**Освітня програма:** "Комп'ютерні науки"  
**Факультет:** Економіки і підприємництва

**Умань – 2024**

Робоча програма навчальної дисципліни "Об'єктно-орієнтоване програмування і моделювання" для здобувачів вищої освіти спеціальності 122 "Комп'ютерні науки" освітньо-професійної програми "Комп'ютерні науки" другого (магістерського) рівня вищої освіти. Умань: Уманський національний університет садівництва, 2024. 18 с.

Розробник: Роман ЛІЩУК, к.т.н., доцент

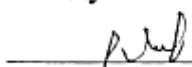


Роман ЛІЩУК

Робоча програма затверджена на засіданні  
кафедри інформаційних технологій

Протокол № 1 від «08» 08 2024 року

Завідувач кафедри інформаційних технологій

 Роман ЛІЩУК

"08" 08 2024 року

Схвалено науково-методичною комісією факультету економіки і підприємництва

Протокол № 1 від «8» серпня 2024 року

Голова  Руслан МУДРАК

«8» серпня 2024 року

© УНУС, 2024 рік

© Ліщук Р.І., 2024 рік

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітній рівень, назва освітньої програми	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 6	Галузь знань 12 "Інформаційні технології"	<i>Обов'язкова</i>	
	Спеціальність 122 "Комп'ютерні науки"		
Модулів – 1	Освітній рівень: Другий рівень вищої освіти  Освітньо-професійна програма 122 "Комп'ютерні науки"	<b>Рік підготовки</b>	
Змістових модулів – 2		1-й	1-й
Загальна кількість годин – 180 год.		<b>Семестр</b>	
		1-й	1-й
Тижневих годин для денної форми навчання: аудиторних – 1 самостійної роботи студента – 3		<b>Лекції</b>	
		32 год.	8 год.
		<b>Практичні, семінарські</b>	
		<b>Лабораторні заняття</b>	
		42 год.	4 год.
		<b>Самостійна робота</b>	
	106 год.	168 год.	
	Вид контролю: іспит		

## 2. Мета та завдання навчальної дисципліни

Робоча програма навчальної дисципліни "Об'єктно-орієнтоване програмування і моделювання" розроблена відповідно до Положення про методичне забезпечення освітнього процесу в Уманському національному університеті садівництва (схвалено Вченою радою Уманського НУС, протокол №1 від 08.10.2020, із змінами та доповненнями від 11.07.2024, протокол № 8).

Навчальна дисципліна "Об'єктно-орієнтоване програмування і моделювання" належить до обов'язкових дисциплін, вивчення яких передбачено освітньо-професійною програмою "Комп'ютерні науки" підготовки фахівців другого (магістерського) рівня вищої освіти за спеціальністю 122 "Комп'ютерні науки" галузі знань 12 "Інформаційні технології".

**Мета вивчення дисципліни** – проектування, розробка та кодування програмних систем за допомогою засобів об'єктно-орієнтованого програмування для розв'язку задач системного програмування в технічних та комп'ютерних системах, а також основним принципам реалізації об'єктно-орієнтованого програмування та моделювання.

**Завдання дисципліни** - конструювання (проектування) програмних систем на основі відповідного набору абстрактних типів даних; моделювання в термінах взаємозв'язку абстрактних типів даних та застосування механізму успадкування для сумісного використання коду та інтерфейсу.

**Предметом дисципліни** є принципи та методи розробки програмного забезпечення, засновані на об'єктно-орієнтованому підході.

**Місце навчальної дисципліни в структурно-логічній схемі освітньо-професійної програми:** вивчення змісту дисципліни базується на освоєнні курсів другого (магістерського) рівня вищої освіти: "Технології проектування ІС".

Вивчення навчальної дисципліни "Об'єктно-орієнтоване програмування і моделювання" передбачає формування та розвиток у здобувачів компетентностей і програмних результатів навчання відповідно до освітньо-професійної програми "Комп'ютерні науки" спеціальності 122 "Комп'ютерні науки" галузі знань 12 "Інформаційні технології" (табл. 1).

**Матриця компетентностей і програмних результатів навчання, що формуються під час вивчення навчальної дисципліни "Об'єктно-орієнтоване програмування і моделювання"**

Шифр компетентності	Компетентності	Шифр програмних результатів навчання	Програмні результати навчання
<b>Загальні компетентності (ЗК)</b>			
<b>ЗК 05</b>	Здатність вчитися й оволодівати сучасними знаннями.	<b>РН 2</b>	Мати спеціалізовані уміння/навички розв'язання проблем комп'ютерних наук, необхідні для проведення досліджень та/або провадження інноваційної діяльності з метою розвитку нових знань та процедур.
		<b>РН 11</b>	Створювати нові алгоритми розв'язування задач у сфері комп'ютерних наук, оцінювати їх ефективність та обмеження на їх застосування.
		<b>РН 16</b>	Виконувати дослідження у сфері комп'ютерних наук.
		<b>РН 19</b>	Аналізувати сучасний стан і світові тенденції розвитку комп'ютерних наук та інформаційних технологій.
<b>ЗК 06</b>	Здатність бути критичним і самокритичним.	<b>РН 3</b>	Зрозуміло і недвозначно доносити власні знання, висновки та аргументацію у сфері комп'ютерних наук до фахівців і нефахівців, зокрема до осіб, які навчаються.
		<b>РН 5</b>	Оцінювати результати діяльності команд та колективів у сфері інформаційних технологій, забезпечувати ефективність їх діяльності.
<b>Спеціальні (фахові) компетентності (СК)</b>			
<b>СК02</b>	Здатність формалізувати предметну область певного проєкту у вигляді відповідної інформаційної моделі.	<b>РН 2</b>	Мати спеціалізовані уміння/навички розв'язання проблем комп'ютерних наук, необхідні для проведення досліджень та/або провадження інноваційної діяльності з метою розвитку нових знань та процедур.
		<b>РН 6</b>	Розробляти концептуальну модель інформаційної або комп'ютерної системи.
		<b>РН 18</b>	Збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до інформаційної або комп'ютерної системи, що розробляється, експлуатується чи супроводжується.
<b>СК06</b>	Здатність застосовувати існуючі і розробляти нові алгоритми розв'язування задач у галузі комп'ютерних наук.	<b>РН 11</b>	Створювати нові алгоритми розв'язування задач у сфері комп'ютерних наук, оцінювати їх ефективність та обмеження на їх застосування.

<b>СК07</b>	Здатність розробляти програмне забезпечення відповідно до сформульованих вимог з урахуванням наявних ресурсів та обмежень.	<b>РН 9</b>	Розробляти алгоритмічне та програмне забезпечення для аналізу даних (включно з великими).
		<b>РН 14</b>	Тестувати програмне забезпечення.

Методи навчання та засоби діагностики, що відповідають визначеним результатам навчання за навчальною дисципліною "Об'єктно-орієнтоване програмування і моделювання", наведено в табл. 2, 3.

Таблиця 2

**Результати, методи навчання та методи контролю за навчальною дисципліною "Об'єктно-орієнтоване програмування і моделювання"**

Результати навчання за навчальною дисципліною		Методи навчання	Методи контролю
<b>1</b>	<b>Знання:</b>		
1.1	загальні поняття мов і методів програмування, покоління мов програмування, об'єктно-орієнтовані мови, об'єктно-орієнтований аналіз, дизайн і проектування, парадигми програмування.	лекція, лабораторне заняття, дискусія, виконання конкретних завдань, самонавчання через Moodle	усне опитування, експрес-контроль, участь у дискусії, виконання індивідуальних і командних завдань, підготовка та представлення презентацій, контрольна (модульна) робота, підсумковий контроль
1.2	поняття класу, специфікатори доступу, оператор new, поняття метода класу, приватні члени класу, передавання параметрів у метод, параметри виводу, перевантаження метода.		
1.3	принципи об'єктно-орієнтованого програмування, інкапсуляція з використанням традиційних методів доступу і зміни, базовий механізм спадкування, поняття поліморфного інтерфейсу.		
1.4	клас Object, основні методи класу Object, основні контейнерні класи, огляд, основні класи вводу-виводу, огляд, основні класи для організації потоків, засоби обробки виключень.		
<b>2</b>	<b>Уміння/навички:</b>		
2.1	проводити об'єктно-орієнтований аналіз предметної області і визначати абстракції з мінімальними необхідними властивостями.	лекція, лабораторне заняття, дискусія, виконання конкретних завдань, самонавчання через Moodle	усне опитування, участь у дискусії, виконання індивідуальних і командних завдань, підготовка та представлення

2.2	використовувати можливості середовища розробки для створення та подальшого запуску простого проекту.		презентацій, контрольна (модульна) робота, підсумковий контроль
<b>3</b>	<b>Комунікація:</b>		
3.1	переконаливе донесення власних знань по об'єктно-орієнтованому моделюванню та програмуванню у комп'ютерних системах.	лабораторне заняття, дискусія, виконання конкретних завдань	представлення презентацій, виконання конкретних завдань, підсумковий контроль
3.2	презентація сучасних методів та теоретичних положень, притаманних інформатиці та програмній інженерії, та їхнього застосування при побудові алгоритмів та програм для сучасних комп'ютерів.		
3.3	ведення діалогу про сучасні технології об'єктно-орієнтованого аналізу, проектування та програмування об'єктно-орієнтованої моделі різними мовами програмування.		
<b>4</b>	<b>Відповідальність і автономія</b>		
4.1	застосування об'єктно-орієнтованих технологій при розробці програм.	лабораторне заняття, дискусія, аналітична робота, вирішення конкретних завдань	представлення презентацій, виконання конкретних завдань, підсумковий контроль
4.2	Здатність самостійно розробляти й тестувати програмні продукти функціонуючі під керуванням сучасних операційних систем.		

**Методи навчання та методи контролю програмних результатів навчання з навчальної дисципліни "Об'єктно-орієнтоване програмування і моделювання"**

<b>Програмний результат навчання</b>		<b>Метод навчання</b>	<b>Методи контролю</b>
<b>РН 7</b>	Розробляти та застосовувати математичні методи для аналізу інформаційних моделей.	Лекція, практичне заняття, дискусія, виконання завдань, самонавчання за допомогою конспектів, навчальних посібників, Moodle	Усне опитування, участь у дискусії, виконання індивідуальних і командних завдань, підготовка та представлення презентацій, контрольна робота, підсумковий контроль
<b>РН 9</b>	Розробляти алгоритмічне та програмне забезпечення для аналізу даних (включно з великими).	Лекція, практичне заняття, дискусія, виконання завдань, самонавчання за допомогою конспектів, навчальних посібників, Moodle	Усне опитування, участь у дискусії, виконання індивідуальних і командних завдань, підготовка та представлення презентацій, контрольна робота, підсумковий контроль
<b>РН 11</b>	Створювати нові алгоритми розв'язування задач у сфері комп'ютерних наук, оцінювати їх ефективність та обмеження на їх застосування.	Лекція, практичне заняття, дискусія, виконання завдань, самонавчання за допомогою конспектів, навчальних посібників, Moodle	Усне опитування, участь у дискусії, виконання індивідуальних і командних завдань, підготовка та представлення презентацій, контрольна робота, підсумковий контроль



### 3. Програма навчальної дисципліни

#### Змістовий модуль 1. Основи об'єктно-орієнтованого програмування і моделювання

##### **Тема 1. Еволюція методології програмування.**

Основні тенденції розвитку мов і методів програмування. Покоління мов програмування. Зародження об'єктної моделі. Об'єктно-орієнтовані мови. Об'єктно-орієнтований аналіз, дизайн і проектування. Парадигми програмування. Основні принципи ООП

##### **Тема 2. Класи і об'єкти**

Поняття класу. Специфікатори доступу. Оператор *new*. Публічні члени класу. Різниця між структурою і класом.

##### **Тема 3. Методи класу**

Поняття метода класу. Приватні члени класу. Передавання параметрів у метод. Параметри виводу. Перевантаження метода

##### **Тема 4. Конструктори**

Поняття конструктора. Роль стандартного конструктора. Визначення спеціальних конструкторів. Роль ключового слова *this*. Побудова ланцюжка викликів конструкторів з використанням *this*. Огляд потоку конструктора. Необов'язкові аргументи. Поняття ключового слова *static*. Визначення статичних полів даних. Визначення статичних методів. Визначення статичних конструкторів. Визначення статичних класів. Поняття синтаксису ініціалізації об'єктів.

#### Змістовий модуль 2. Принципи ООП

##### **Тема 5. Інкапсуляція**

Поняття інкапсуляції. Інкапсуляція з використанням традиційних методів доступу і зміни. Використання властивостей. Використання властивостей всередині класу. Властивості, доступні тільки для читання і тільки для запису. Визначення статичних властивостей. Поняття автоматичних властивостей. Взаємодія з автоматичними властивостями. Зауваження щодо автоматичних властивостей і стандартних значень.

##### **Тема 6. Спадкування**

Базовий механізм спадкування. Вказання батьківського класу для існуючого класу. Типи спадкування. Зауваження щодо множини базових класів. Ключове слово *sealed*. Зміна діаграм класів *Visual Studio*. Подробиці про спадкування. Виклик конструкторів базового класу. Управління створенням базового класу за допомогою ключового слова *base*. Додавання запечатаного класу. Реалізація моделі включення/делегації. Визначення вкладених типів

##### **Тема 7. Поліморфізм**

Поняття поліморфізму. Ключові слова *virtual* і *override*. Перевизначення віртуальних членів в IDE-середовищі *Visual Studio*. Запечатування віртуальних членів. Абстрактні класи. Поняття поліморфного інтерфейсу. Приховування членів. Правила приведення до базового і похідного класу. Ключове слово *as*. Ключове слово *is*. Головний батьківський клас *System.Object*. Перевизначення *System.Object.ToString()*. Перевизначення *System.Object.Equals()*. Перевизначення *System.Object.GetHashCode()*. Статичні члени *System.Object*.

### **Змістовий модуль 3. Додаткові засоби програмування на мові С#**

#### **Тема 8. Інтерфейси**

Поняття інтерфейсу. Виклик учасників інтерфейсу на рівні об'єкта. Реалізація декількох інтерфейсів. Поводження з неоднозначністю при реалізації декількох інтерфейсів. Явна реалізація інтерфейсу. Перевірка реалізації інтерфейсу. Використання ключового слова *as*. Використання ключового слова *is*. Реалізація багаторівневого інтерфейсу. Реалізація властивостей інтерфейсу. Реалізація інтерфейсу структурою. Передача та повернення інтерфейсу. Передача інтерфейсу як параметра. Повернення інтерфейсу.

#### **Тема 9. Перевантаження операторів**

Поняття перевантаження операторів. Перевантаження бінарного оператора. Додавання константи до об'єкта. Додавання двох комплексних чисел. Перенавантаження унарного оператора. Перевантаження операторів *true* і *false*. Перезавантаження операторів порівняння. Перевантаження операторів перетворення типів даних. Створення оператора неявного перетворення. Створення оператора явного перетворення.

#### **Тема 10. Делегати і події**

Поняття делегата. Оголошення делегата. Створення об'єкта делегата. Виклик методу, на який посилається делегат. Увімкнення групового виклику. Видалення цілей зі списку викликів делегата. Синтаксис групових перетворень методів. Поняття узагальнених делегатів. Узагальнені делегати *Action<>* та *Func<>*.

Поняття подій С#. Ключове слово *event*. Прослуховування вхідних подій. Спрощена реєстрація подій з використанням *Visual Studio*. Створення спеціальних аргументів подій. Узагальнений делегат *EventHandler<T>*. Поняття анонімних методів С#.

Поняття лямбда-виразів. Аналіз лямбда-виразу. Обробка аргументів всередині множини операторів. Лямбда-вирази з кількома параметрами і без параметрів.

#### **Тема 11. Узагальнення і колекції**

Причини створення класів колекцій. Простір імен *System.Collections*. Неузагальнені колекції. Призначення і основні класи простору імен *System.Collections.Specialized*. Проблеми, пов'язані з неузагальненими колекціями.

Поняття узагальнених колекцій. Роль параметрів узагальнених типів. Вказування параметрів типу для: узагальнених класів і структур, узагальнених членів, узагальнених інтерфейсів.

Простір імен *System.Collections.Generic*. Синтаксис ініціалізації колекцій. Робота з класами: *List<T>*, *Stack<T>*, *Queue<T>*, *SortedSet<T>*.

Простір імен *System.Collections.ObjectModel*. Робота з *ObservableCollection*. Створення спеціальних узагальнених методів. Виведення параметрів типу. Створення спеціальних узагальнених структур і класів. Обмеження параметрів типу.

#### 4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	Денна форма						Заочна форма					
	Усього	у тому числі					Усього	у тому числі				
		л	п	лаб	інд.	с.р.		л	п	лаб	інд.	с.р.
Змістовий модуль 1. Основи об'єктно-орієнтованого програмування і моделювання												
Тема 1. Еволюція методології програмування	8	2				6	16					16
* Тема 2. Класи і об'єкти	18	4		4		10	16					16
Тема 3. Методи класу	16	2		4		10	16					16
Тема 4. Конструктори	16	2		4		10	16	2				14
Тема 5. Інкапсуляція	16	2		4		10	18	2		2		14
Тема 6. Спадкування	16	2		4		10	16	2				14
Тема 7. Поліморфізм	16	2		4		10	18	2		2		14
Тема 8. Інтерфейси	18	4		4		10	16					16
Тема 9. Перевантаження операторів	18	4		4		10	16					16
Тема 10. Делегати і події	18	4		4		10	16					16
** Тема 11. Узагальнення і колекції [Topic 11. Generalizations and collections]	20	4		6		10	16					16
Усього	180	32	0	42	0	106	180	8	0	4	0	168
Усього годин	180	32	0	42	0	106	180	8	0	4	0	168

\*залучені стейкхолдери для спільного проведення аудиторних занять

\*\*лекційне та лабораторне заняття проводяться на англійській мові

#### 5. Теми практичних занять (не передбачено)

#### 6. Теми семінарських занять (не передбачено)

#### 7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин	
		денна	заочна
1.	Лабораторна робота №1. Класи і об'єкти	4	
2.	Лабораторна робота №2. Методи класу	4	
3.	Лабораторна робота №3. Конструктори	4	
4.	Лабораторна робота №4. Інкапсуляція	4	2
5.	Лабораторна робота №5. Спадкування	4	
6.	Лабораторна робота №6. Поліморфізм	4	2
7.	Лабораторна робота №7. Інтерфейси	4	
8.	Лабораторна робота №8. Перевантаження операторів	4	
9.	Лабораторна робота №9. Делегати і події	4	
10.	Лабораторна робота №10. Узагальнення і колекції	6	
	Всього	32	4

### 8. Самостійна робота (денна форма)

№ з/п	Назва теми	Кількість годин
1.	Еволюція методології програмування	6
2.	Розробка системи класів для створення і маніпулювання об'єктами обраної предметної області	10
3.	Програмування функціоналу кожного класу завдяки конструкторам і методам	10
4.	Забезпечення інкапсуляції даних класів шляхом створення властивостей полів	10
5.	Створення дочірніх класів, які використовують функціонал батьківських класів і розширюють його завдяки новим полям/властивостям і методам	10
6.	Надання поліморфності розробленим класам шляхом використання техніки перевантаження методів	10
7.	Створення інтерфейсів і їх власна реалізація різними класами з урахуванням специфіки поведінки об'єктів різної природи	10
8.	Перевантаження ряду операторів для виконання арифметичних і логічних операторів над об'єктами різних типів - розрахунків, перетворень, порівнянь	10
9.	Розробка делегатів і створення їх об'єктів для забезпечення виконання різних методів (послідовностей методів) залежно від умов, що складаються у процесі роботи програми; маніпулювання послідовностями методів об'єкта делегата	10
10.	Створення подій і програмування реакції програми на них	10
11.	Створення множин об'єктів у вигляді колекцій та узагальнень, а також проведення маніпуляцій з елементами створених контейнерів	10
	Всього	106

### Самостійна робота (заочна форма)

№ з/п	Назва теми	Кількість годин
1.	Еволюція методології програмування	16
2.	Розробка системи класів для створення і маніпулювання об'єктами обраної предметної області	16
3.	Програмування функціоналу кожного класу завдяки конструкторам і методам	16
4.	Забезпечення інкапсуляції даних класів шляхом створення властивостей полів	14
5.	Створення дочірніх класів, які використовують функціонал батьківських класів і розширюють його	14

	завдяки новим полям/властивостям і методам	
6.	Надання поліморфності розробленим класам шляхом використання техніки перевантаження методів	14
7.	Створення інтерфейсів і їх власна реалізація різними класами з урахуванням специфіки поведінки об'єктів різної природи	14
8.	Перевантаження ряду операторів для виконання арифметичних і логічних операторів над об'єктами різних типів - розрахунків, перетворень, порівнянь	16
9.	Розробка делегатів і створення їх об'єктів для забезпечення виконання різних методів (послідовностей методів) залежно від умов, що складаються у процесі роботи програми; маніпулювання послідовностями методів об'єкта делегата	16
10.	Створення подій і програмування реакції програми на них	16
11.	Створення множин об'єктів у вигляді колекцій та узагальнень, а також проведення маніпуляцій з елементами створених контейнерів	16
	Всього	168

## 9. Самостійна робота

### *Заочна форма навчання*

Самостійна робота здобувачів заочної форми навчання передбачає набуття практичних умінь та навичок управління якістю в ІТ-проектах, створення прототипу проекту програмного забезпечення. Це завдання вимагає від студентів інтегрувати та використовувати свої знання для визначення елементів та характеристик проекту, необхідних для управління ним.

## 10. Методи навчання

В освітньому процесі використовуються наступні методи навчання: тематичні лекції, лабораторні заняття, дискусія, поточний контроль знань, індивідуальні заняття із підготовкою рефератів або презентацій, виконання поточних завдань, самонавчання на основі конспектів, посібників та іншої рекомендованої літератури, самонавчання за допомогою модульного об'єктно-орієнтованого динамічного навчального середовища Moodle (табл. 2).

Матеріали курсу "Об'єктно-орієнтоване програмування і моделювання" розміщені на платформі Moodle.

В умовах дистанційної освіти проведення лекцій і практичних занять відбувається у форматі відеоконференцій. Для організації освітнього процесу використовуються такі сервіси як Zoom, Moodle та електронна пошта.

## 11. Методи контролю

Для забезпечення оцінювання студентів проводиться поточний (модульний) і підсумковий (іспит) контроль.

Модульний контроль передбачає перевірку стану засвоєння визначеної системи елементів знань і вмінь студентів з того чи іншого модулю.

При контролі систематичності та активності роботи на лабораторних заняттях оцінюванню в балах підлягають: рівень знань, необхідний для виконання лабораторних робіт, що передбачені завданнями для самостійного опрацювання; повнота, якість і вчасність їх виконання та результати захисту; рівень знань, продемонстрований у відповідях на лабораторних заняттях; активність при обговоренні питань.

При виконанні модульних (контрольних) завдань оцінюванню в балах підлягають теоретичні знання і практичні уміння, яких набули студенти після опанування певного модуля, модульний контроль проводиться письмово.

Повторне виконання модульних контрольних робіт на вищу кількість балів дозволяється, як виняток, з поважних причин за погодженням викладача, який викладає дисципліну, з дозволу декана факультету до початку підсумкового контролю (заліку).

У разі невиконання певних завдань поточного контролю з об'єктивних причин, студенти мають право, з дозволу викладача, скласти їх до останнього заняття. Час і порядок складання визначає викладач. У разі, коли студент не з'явився на проведення модульної контрольної роботи без поважних причин, він отримує нуль балів. Перездача модульного контролю допускається у строки, які встановлюються викладачем.

Знання студента з певного модуля вважаються незадовільними, за умови коли сума балів його поточної успішності та за модульний контроль складають менше 61% від максимально можливої суми за цей модуль. У такому випадку можливе повторне перескладання модуля у терміни встановлені викладачем.

Рейтингова сума балів з навчальної дисципліни після складання модулів і підсумкового контролю виставляється як сума набраних студентом балів протягом семестру та балів набраних студентом на підсумковому контролі. До підсумкового контролю допускаються студенти, які виконали всі модульні контролі, передбачені для даної навчальної дисципліни і за рейтинговим показником набрали не менш як 35 балів.

Підсумковий контроль забезпечує оцінку результатів навчання студентів на заключному етапі вивчення дисципліни і проводиться відповідно до навчального плану у вигляді заліку в термін, встановлений графіком навчального процесу та в обсязі навчального матеріалу, визначеному даною робочою програмою навчальної дисципліни. Зміст і структура контрольних завдань, білетів і критерії оцінювання визначаються рішенням кафедри.

Якщо у підсумку студент отримав за рейтинговим показником оцінку "FX", то він допускається до повторного складання підсумкового контролю з дисципліни. Студент, допущений до повторного складання підсумкового контролю зобов'язаний у терміни, визначені деканатом, передати невиконані (або виконані на низькому рівні) завдання поточно-модульного контролю, виконати модульні контролі і скласти підсумковий контроль. Рейтинговий показник студента з навчальної дисципліни при цьому визначається за результатами повторного складання підсумкового контролю і не впливає на загальний рейтинг студента.

## 12. Розподіл балів, які отримують студенти

В основу рейтингового оцінювання знань студента закладена спеціальна 100-бальна шкала оцінювання (максимально можлива сума балів, яку може набрати студент за всіма видами контролю знань з дисципліни з урахуванням поточної успішності, самостійної роботи, науково-дослідної роботи, підсумкового контролю тощо). Кількість балів, які можна набрати у ході вивчення курсу дисципліни розподіляються наступним чином:

### Розподіл балів, які отримують студенти при вивченні дисципліни "Об'єктно-орієнтоване програмування і моделювання" (денна форма навчання)

Поточний (модульний) контроль													Бали за додаткову роботу	Підсумковий контроль	Сума
Кількість балів за модуль	Змістовий модуль 1 (60 балів)														
Кількість балів за теми	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	МК 1			
в т.ч. за видами робіт:	1	5	5	5	5	5	5	5	5	5	5	9	10	30	100
лабораторні заняття		4	4	4	4	4	4	4	4	4	4				
виконання СРС	1	1	1	1	1	1	1	1	1	1	1				

### Розподіл балів які отримують студенти при вивченні дисципліни "Об'єктно-орієнтоване програмування і моделювання" (заочна форма навчання)

Поточний (модульний) контроль													Бали за додаткову роботу	Підсумковий контроль	Сума
Кількість балів за модуль	Змістовий модуль 1 (60 балів)														
Кількість балів за теми	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	МК 1			
в т.ч. за видами робіт:	1	5	5	5	5	5	5	5	5	5	5	9	10	30	100
лабораторні заняття					4		4								
виконання СРС	1	5	5	5	1	5	1	5	5	5	5				

#### **Поточний контроль.**

Об'єктами *поточного контролю* знань студентів є активність і систематичність роботи на лабораторних заняттях, виконання завдань для самостійної роботи студентів, розв'язання модульних завдань.

При контролі на *лабораторних заняттях* оцінці підлягають: рівень знань, продемонстрований у відповідях і виступах; активність при обговоренні заявлених на занятті питань; результати опитування та письмового або тестового контролю знань.

Під час контролю виконання завдань для *самостійної роботи* оцінюванню підлягають: правильність і повнота врахування усіх складових завдання; обґрунтованість відповіді.

При контролі виконання *модульних завдань* оцінці підлягають теоретичні знання та практичні навички, яких набули студенти після опанування тем змістового модуля.

Максимальна сума балів поточного контролю з дисципліни "Об'єктно-орієнтоване програмування і моделювання" – 100. Бали розподіляються наступним чином:

1. Систематичність та активність роботи на лабораторних заняттях оцінюється в 10 балів:

а) відповідь / виконання завдань – до 4 балів:

б) змістовні доповнення при обговоренні питань – 1 бал.

2. Виконання завдань для самостійної роботи студентів денної форми навчання оцінюється в 0-1 бал. Виконання завдань для самостійної роботи студентів заочної форми навчання оцінюється максимум в 5 балів.

3. Модульний контроль, для денної та заочної форм навчання, містить 30 питань, де можна набрати бали відповідно до кількості (%) правильних відповідей:

≥ 90% правильних відповідей – 9 балів; 75 – 89% - 7 балів; 60 – 74% - 5 балів; 50 – 59% - 3 бали; 49% ≤ правильних відповідей - 0 балів.

Бали за додаткову роботу – представлення результатів науково-дослідних робіт: участь у студентських олімпіадах, конкурсах наукових робіт, грантах, науково-дослідних проектах – 1–10 балів; публікація наукових статей, тез доповіді на конференції– 1–10 балів.

Виконання студентами завдання повинно носити виключно самостійний характер. Тому, за використання заборонених джерел (шпаргалок, засобів зв'язку та ін.) чи підказок студент одержує нульову оцінку. Списування під час контрольних заходів заборонені (в т.ч. із використанням мобільних пристроїв).

### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
		для заліку
90 – 100	<b>A</b>	відмінно
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



**Оцінка "відмінно" (90 – 100 балів).** Здобувач має систематичні та глибокі знання навчального матеріалу, вміє без помилок виконувати практичні завдання, які передбачені програмою курсу, засвоїв основну й ознайомився з додатковою літературою, викладає матеріал у логічній послідовності, робить узагальнення й висновки, наводить практичні приклади у контексті тематичного теоретичного матеріалу.

**Оцінка "добре" (74 – 89 балів).** Здобувач повністю засвоїв навчальний матеріал, знає основну літературу, вміє виконувати практичні завдання, викладає матеріал у логічній послідовності, робить певні узагальнення й висновки, але не наводить практичних прикладів у контексті тематичного теоретичного матеріалу або допускає незначні помилки у формулюванні термінів, категорій, невеликі помилки у розрахунках при вирішенні практичних завдань.

**Оцінка "задовільно" (60 – 73 бали).** Здобувач засвоїв матеріал не у повному обсязі, дає неповну відповідь на поставлені теоретичні питання, припускається грубих помилок при вирішенні практичного завдання.

**Оцінка "незадовільно" (менше 60 балів).** Здобувач не засвоїв навчальний матеріал, дає неправильні відповіді на поставлені теоретичні питання, не володіє основними методами наукових досліджень при виконанні практичних завдань. Здобувач не допускається до складання іспиту, якщо кількість балів одержаних за результати успішності під час поточного та модульного контролю (відповідно змістовому модулю) впродовж семестру в сумі не досягла 35 балів.

### 13. Методичне забезпечення

1. Ліщук Р.І. Методичні матеріали для виконання лабораторних робіт з дисципліни "Об'єктно-орієнтоване програмування і моделювання" для здобувачів другого (магістерського) рівня вищої освіти спеціальності 122 "Комп'ютерні науки" ОПП "Комп'ютерні науки". Умань: УНУС, 2024. 24 с.

2. Ліщук Р.І. Методичні матеріали, завдання і вказівки для виконання самостійної роботи з дисципліни "Об'єктно-орієнтоване програмування і моделювання" для здобувачів другого (магістерського) рівня вищої освіти спеціальності 122 "Комп'ютерні науки" ОПП "Комп'ютерні науки". Умань: УНУС, 2024. 12 с.

3. Електронний курс в Moodle:  
<https://moodle.udau.edu.ua/course/view.php?id=11>

### 14. Рекомендована література

#### *Базова*

1. Troelsen A. Japikse P. Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming: 10th edition, Apress, 2021, 1411 p.

2. Коноваленко І.В. Програмування мовою C# 7.0 : навчальний посібник / Коноваленко І.В., Марущак П.О., Савків В.Б. – Тернопіль:Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 300 с.

3. Price M. J. C# 9 and .NET 5 – Modern Cross-Platform Development: Build intelligent apps, websites, and services with Blazor, ASP.NET Core, and Entity Framework Core using Visual Studio Code: 5th Edition, Packt Publishing, 2020, 822 p.

4. Щербаков О. В. Основи об'єктно-орієнтованого програмування [Електронний ресурс] : навчальний посібник / О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко. – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 237 с.

5. Сидоров М. О. Основи програмування. Навчальний посібник, Київ, 2018.

6. Mark J. Price C# 8.0 and .NET Core 3.0. – 4th edition. Packt Publishing

7. Andrew Stellman. Head First C#. – 4th edition, O'Reilly Media, Inc. – 2020.

8. Mikael Olsson. C# 10 Quick Syntax Reference. – 4th Ed. Apress. – 2023

9. Joe Mayo. C# Cookbook: Modern Recipes for Professional Developers. 1st Ed.

### *Допоміжна*

10. Clark D., Sanders J. Beginning C# object-oriented programming. Apress, 2011. 362 p.

11. Ivor Horton & Peter Van Weert. Beginning C++17 From Novice to Professional. 2018, 788 P.

12. Алхімова С. М. Об'єктно-орієнтоване програмування : підручник. У 2-х ч. Ч. 2. Об'єктно-орієнтований підхід до розробки програмного забезпечення / С. М Алхімова. - Київ: КПІ і.м.Ігоря Сікорського, Вид-во «Політехніка», 2019.- 192 с.

## **15. Інформаційні ресурси**

13. C# OOP (Object-Oriented Programming) - W3Schools. [Електронний ресурс]. – Режим доступу: [https://www.w3schools.com/cs/cs\\_oop.php](https://www.w3schools.com/cs/cs_oop.php)

14. Learn C#. Free courses, tutorials, videos, and more about learning the programming language C#. [Електронний ресурс]. – Режим доступу: <https://dotnet.microsoft.com/en-us/learn/csharp>

15. Object-Oriented programming (C#). [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/oop>.

## **16. Перезарахування та визнання результатів навчання**

Перезарахування та визнання результатів навчання з дисципліни "Об'єктно-орієнтоване програмування і моделювання" або окремого її елемента відбувається відповідно до Положення про порядок визнання в Уманському національному університеті садівництва результатів навчання, отриманих у неформальній та/або інформальній освіті (<https://www.udau.edu.ua/ua/file/4n0x>).

Здобувачі вищої освіти мають право на визнання результатів навчання в неформальній та інформальній освіті (курси навчання в центрах освіти, курси

інтенсивного навчання, семінари, конференції, олімпіади, конкурси наукових робіт, літні чи зимові школи, бізнес-школи, тренінги тощо) в обсязі, що загалом не перевищує 25 % освітньої програми.

## **17. Політика академічної доброчесності**

У процесі навчання з дисципліни "Об'єктно-орієнтоване програмування і моделювання", студенти повинні дотримуватися встановлених правил академічної доброчесності, визначених Кодексом доброчесності Уманського національного університету садівництва (<https://www.udau.edu.ua/ua/file/4dH7>). При підготовці рефератів, виконанні індивідуальних науково-дослідних завдань, а також під час проведення контрольних заходів очікується, що всі роботи подані студентами будуть їхніми оригінальними дослідженнями та міркуваннями.

Будь-які види порушення академічної доброчесності, зокрема плагіат, неправомірне використання чужих ідей, фальсифікація даних чи співучасть у таких діяннях, є абсолютно неприпустимими і не толеруються. Виявлення ознак академічної недоброчесності у письмовій роботі студента є підставою для її незарахування викладачем, незалежно від обсягу порушення.

З метою запобігання порушенням і підвищення якості академічних робіт, студентам настійно рекомендується користуватися належними академічними ресурсами та інструментами для перевірки робіт на плагіат, а також звертатися за консультаціями з питань правильного цитування і академічного письма.

## **18. Зміни у робочій програмі на 2024/2025 навчальний рік**