

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

**Кафедра інформаційних технологій**

**«ЗАТВЕРДЖУЮ»**

Гарант освітньої програми

Р.І. Ліщук Р.І. Ліщук

« 30 » 08 2021 року


**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

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

<b>Освітній рівень:</b>	Перший рівень вищої освіти
<b>Галузь знань:</b>	12 «Інформаційні технології»
<b>Спеціальність:</b>	122 «Комп'ютерні науки»
<b>Освітня програма:</b>	«Комп'ютерні науки»
<b>Факультет:</b>	економіки і підприємництва



Умань – 2021 рік

Робоча програма навчальної дисципліни «Об'єктно-орієнтоване програмування» для здобувачів вищої освіти спеціальності 122 «Комп'ютерні науки». – Умань: Уманський НУС, 2021 р. – 13 с.

Розробник: к.т.н., доцент, доцент кафедри інформаційних технологій  
 А.Є. Соколов

Робоча програма затверджена на засіданні кафедри інформаційних технологій

Протокол № 1 від 30 серпня 2021 року

Завідувач кафедри, к.т.н., доцент  Р.І. Ліщук  
« 30 »  2021 року

Схвалено науково-методичною комісією факультету економіки і підприємництва

Протокол № 1 від 31 серпня 2021 року

Голова  Л.В. Смолій  
« 31 »  2021 року

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, спеціалізація, рівень вищої освіти	Характеристика дисципліни	
		денна форма	заочна форма
Кількість кредитів 6,5	Галузь знань 12 «Інформаційні науки»	Обов'язкова	
Змістових модулів 2	Спеціальність 122 «Комп'ютерні науки»	<b>Рік підготовки:</b>	
Індивідуальне науково-дослідне завдання		2-й	
Загальна кількість годин 195		<b>Семестр</b>	
Тижневих годин для денної форми навчання: аудиторних - 6 самостійної роботи студента - 7		3-й	
	Освітній рівень: Перший рівень вищої освіти Освітньо-професійна програма 122 «Комп'ютерні науки»	<b>Лекції</b>	
		32 год.	
		<b>Практичні, семінарські</b>	
		<b>Лабораторні</b>	
		56 год.	
		<b>Самостійна робота</b>	
	107 год.		
		<b>Індивідуальні завдання:</b> курсний проект	
		Вид контролю: екзамен	

## **2. Мета та завдання навчальної дисципліни**

**Мета:** Метою вивчення дисципліни є 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 принципом реалізації об'єктно-орієнтованого програмування на мовах програмування.

### **Завдання:**

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

### **Місце дисципліни у структурно-логічній схемі підготовки здобувачів вищої освіти:**

Теоретичною базою вивчення навчальної дисципліни є такі дисципліни: «Вища математика», «Алгоритмізація і програмування», «Теорія ймовірностей і математична статистика».

Навики отримані при вивченні дисципліни можуть бути використанні при вирішенні задач з таких дисциплін: «Технології розподільних систем та паралельних обчислень», «Управління ІТ-проектами» та «Комп'ютерні мережі».

### **Компетентності:**

- знання та розуміння предметної області та розуміння професійної діяльності;
- здатність вчитися й оволодівати сучасними знаннями;
- здатність до пошуку, оброблення та аналізу інформації з різних джерел.;
- 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;
- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями,

методами й алгоритмами обчислень, структурами даних і механізмами управління.

**Програмні результати:**

-проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій;

-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;

-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.

### **3. Програма навчальної дисципліни**

#### **Змістовий модуль 1. Основи об'єктно-орієнтованого програмування (ООП)**

##### ***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/делегатії. 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. Перевантаження 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.

### **Тема 12. Управління помилками і виключеннями**

Поняття помилок і виключень. Ієрархія виключень. Ключові слова управління виключеннями. Застосування множини блоків *try*. Застосування множини блоків *catch*. Вкидання виключення. Вкладені блоки *try*. Створення власного виключення. Застосування оператора *checked/unchecked*.

### **Тема 13. Ланцюжки (нитки, треди)**

Поняття ланцюжка (треда). Використання головного тредя. Створення дочірніх тредів. Життєвий цикл тредя. Багатотредовість. Забезпечення ексклюзивних операцій за допомогою моніторів. Застосування оператора замикання (*lock*). Переривання ланцюжка.

### **Тема 14. Потоки (стріми)**

Поняття потоку. Управління файлами. Випадкове читання файлів. Читання та запис примітивних типів у двійковому форматі. Виконання вводу/виводу файлів на основі символів. Читання цілого файлу. Тимчасове утримання даних за допомогою *MemoryStream*. Керування каталогом за допомогою *DirectoryInfo*.

### **Тема 15. Основи ADO.NET і запити LINQ**

Головні компоненти ADO.NET. Підключення або створення бази даних. Доступ до вмісту бази даних. Відображення табличних даних з використанням *GridView*. Додавання рядків у таблицю. Використання параметрів з застосуванням об'єкта *Command*. Виклик збережених процедур. Оновлення таблиці.

Поняття LINQ (*Language Integrated Query*). Неявно типізовані локальні змінні. Конструктори об'єктів і колекцій. Застосування лямбда-виразів. Методи розширення. Анонімні типи. Категорії LINQ.



#### 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.
<b>Змістовий модуль 1. Основи об'єктно-орієнтованого програмування (ООП)</b>												
Тема 1. Еволюція методології програмування	8	2				6	9	1			8	
Тема 2. Класи і об'єкти	14	4		4		6	12	1		1	10	
Тема 3. Методи класу	12	2		4		6	9				9	
Тема 4. Конструктори	12	2		4		6	11	1		1	9	
<b>Разом за змістовим модулем 1</b>	<b>46</b>	<b>10</b>	<b>0</b>	<b>12</b>	<b>0</b>	<b>24</b>	<b>41</b>	<b>3</b>		<b>2</b>	<b>36</b>	
<b>Змістовий модуль 2. Принципи ООП</b>												
Тема 5. Інкапсуляція	12	2		4		6	12	1		1	10	
Тема 6. Спадкування	12	2		4		6	12	1		1	10	
Тема 7. Поліморфізм	12	2		4		6	13	1			12	
<b>Разом за змістовим модулем 2</b>	<b>36</b>	<b>6</b>	<b>0</b>	<b>12</b>	<b>0</b>	<b>18</b>	<b>37</b>	<b>3</b>		<b>2</b>	<b>32</b>	
<b>Модуль 2.</b>												
<b>Змістовий модуль 3. Додаткові засоби програмування на мові C#</b>												
Тема 8. Інтерфейси	14	2		4		8	11	1			10	
Тема 9. Перевантаження операторів	14	2		4		8	8				8	
Тема 10. Делегати і події	14	2		4		8	13	1			12	
Тема 11. Узагальнення і колекції	14	2		4		8	8				8	
Тема 12. Управління помилками і виключеннями	14	2		4		8	8				8	
Тема 13. Ланцюжки (нитки, треди)	14	2		4		8	8				8	
Тема 14. Потоки (стріми)	14	2		4		8	6				6	
Тема 15. Основи ADO.NET і запити ьшо	15	2		4		9	10				10	
<b>Разом за змістовним модулем 3</b>	<b>113</b>	<b>16</b>	<b>0</b>	<b>32</b>	<b>0</b>	<b>65</b>	<b>72</b>	<b>2</b>			<b>70</b>	
ЩДЗ												
<b>Усього годин</b>	<b>195</b>	<b>32</b>	<b>0</b>	<b>56</b>	<b>0</b>	<b>107</b>	<b>150</b>	<b>8</b>		<b>4</b>	<b>138</b>	

#### 5. Темі лабораторних занять

№ з/п	Назва теми	Кількість годин	
		денна форма	заочна форма
1	Створення об'єктів	4	1
2	Методи класу	2	
3	Конструктори	2	1
4	Інкапсуляція	2	1
5	Спадкування	2	1
6	Поліморфізм	4	
7	Застосування інтерфейсів	2	
8	Перевантаження операторів	2	
9	Делегати і події	2	
10	Узагальнення і колекції	2	
11	Ланцюжки	2	
12	Потоки	2	

13	Основи ADO.NET	2	
14	Запити LINQ	2	
	Разом	32	4

### 6. Самостійна робота

№ з/п	Назва теми	Кількість годин	
		денна форма	заочна форма
1	Еволюція методології програмування	6	18
2	Розробка системи класів для створення і маніпулювання об'єктами обраної предметної області	6	18
3	Програмування функціоналу кожного класу завдяки конструкторам і методам	6	10
4	Забезпечення інкапсуляції даних класів шляхом створення властивостей полів	6	10
5	Створення дочірніх класів, які використовують функціонал батьківських класів і розширюють його завдяки новим полям/властивостям і методам	6	12
6	Надання поліморфності розробленим класам шляхом використання техніки перевантаження методів	6	10
7	Створення інтерфейсів і їх власна реалізація різними класами з урахуванням специфіки поведінки об'єктів різної природи	6	8
8	Перевантаження ряду операторів для виконання арифметичних і логічних операторів над об'єктами різних типів - розрахунків, перетворень, порівнянь	8	6
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	8	6
10	Створення подій і програмування реакції програми на них	8	8
11	Створення множин об'єктів у вигляді колекцій та узагальнень, а також проведення маніпуляцій з елементами створених контейнерів	8	8
12	Забезпечення програми надійними засобами обробки помилок і виключень, що виникають (можуть виникнути) у процесі виконання програми	8	14
13	Організувати управління потоками для запису/читання даних у файли, а також роботу з папками файлової системи комп'ютера	8	4
14	Створити просту реляційну базу даних об'єкта управління і підключити її до створеної програми з метою забезпечення розрахунків збереженими даними	8	6
№ з/п	Назва теми	Кількість годин	
		денна форма	заочна форма
15	Застосувати оператори мови LINQ для вибірки/запису даних з бази даних безпосередньо з програми на мові C#	9	5
	Разом	107	138

## 7. Методи навчання

Навчання студентів відбувається із використанням наступних методів:

1. Словесний - навчальні лекції, доповнені опорним конспектом лекцій.
2. Наочні методи навчання - показ слайдів, що ілюструють відповідні теми; представлення моделей стандартних алгоритмів.
3. Частково-пошуковий - практична реалізація роботи над темами НДРС, окремі розділи яких підготовлені викладачем або іншими студентами.
4. Дослідницький - самостійне опрацювання літератури, дослідний пошук за темами НДРС.
5. Дедуктивний - спочатку повідомляються загальні положення, оператори та стандартні підпрограми, а після цього поступово наводяться приклади, більш конкретні задачі.
6. Індуктивний - в процесі виконання лабораторних робіт спочатку повторюється теоретичний матеріал, потім створюються алгоритми, потім на основі алгоритмів розробляються програми. За результатами роботи студенти формулюють самостійні висновки, які оформлюють у вигляді звітів.
7. Репродуктивний - перелік питань в методичних вказівках до виконання лабораторних робіт з дисципліни після кожної роботи, що створений на основі конспекту лекцій та індивідуальних завдань.

## 8. Методи контролю

Для визначення рівня засвоєння студентами навчального матеріалу використовують такі форми та методи контролю і оцінювання знань:

- *Поточний контроль* - оцінюються усні та письмові відповіді, результати комп'ютерного тестування, виконання дослідницьких завдань, захист підготовлених презентацій;

- *Модульний контроль* проводиться протягом семестру письмово або у вигляді комп'ютерного тестування, захисту творчих проектів тощо; і має на меті оцінку результатів знань студента після вивчення матеріалу з логічно завершеної частини дисципліни.

- *Семестровий контроль (залік)* виставляється з урахуванням поточної успішності та результатів модульних робіт у терміни, передбачені графіком навчального процесу.

Форми контролю:

- усне та письмове опитування;
- тестові завдання в тому числі комп'ютерне тестування;
- звіти з лабораторних робіт.

Дисципліна має два змістовних модулі, які охоплюють матеріал з усіх тем.

## 9. Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота									Екзамен	Сума
Змістовий модуль 1					Змістовий модуль 2					100
T1	T2	T3	T4	МК	T5	T6	T7	МК	30	
4	10	10	8	7	8	8	8	7		

T1, T2 ... T7 - теми змістових модулів.

МК - модульний контроль

### 10. Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

### 11. Методичне забезпечення

1. Соколов А.Є. Інструктивно-методичні матеріали для виконання лабораторних робіт студентами першого рівня вищої освіти спеціальності 122 «Комп'ютерні науки» з дисципліни «Об'єктно-орієнтоване програмування». Умань: УНУС, 2021. 30 с.
2. Соколов А.Є. Методичні вказівки для виконання самостійної роботи студентами першого рівня вищої освіти спеціальності 122 «Комп'ютерні науки» з дисципліни «Об'єктно-орієнтоване програмування». Умань: УНУС, 2021. 25 с.

### 12. Рекомендована література

#### Основна

1. 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 с.
2. 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 с.

3. Freeman A. Pro ASP.NET Core 3 (Develop Cloud-Ready Web Applications Using MVC 3, Blazor, and Razor Pages) / Adam Freeman., 2020. - 1400 с.
4. 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 с.
5. 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 с.
6. Об'єктні технології C++11 Пелешко Д. Д., Теалюк В. М./Навчальний посібник. Львів: Видавництво Львівської політехніки, 2013. - 360 с.
7. Парфьонов Ю.Е. Об'єктно-орієнтоване програмування: конспект лекцій для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання / Парфьонов Ю.Е., Федорченко В.М., Лосев М.Ю. та ін. - Х.: Вид. ХНЕУ, 2018. - 312 с.
8. Решевська К. С., Лісняк А. О., Борю С. Ю. Об'єктно-орієнтоване програмування : навчальний пошбник для здобувачів ступеня вищої освіти бакалавра спеціальності "Комп'ютерні науки" освітньо-професійної програми "Комп'ютерні науки". Запоріжжя : ЗНУ, 2020. 94 с.

#### **Додаткова**

1. Д. В. Настенко, А. Б. Нестерко, Об'єктно-орієнтоване програмування, Київ: НТУУ «КПІ», 2016. – 76 с.
2. Чубук В.В. Об'єктно-орієнтоване програмування у питаннях і відповідях: [навч. посібн.] / В. В. Чубук, Р. М. Чен, Л. А. Павленко та ін. - Х.: Вид. ХНЕУ, 2019. - 288 с.
3. Жуковський С.С., Вакалюк Т.А., Об'єктно-орієнтоване програмування мовою C++. Навчально-методичний посібник для студентів. - Житомир: Вид-во ЖДУ, 2016. 100 с.