

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ  
УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

**ЗАТВЕРДЖЕНО**

на засіданні кафедри  
економічної кібернетики та системного  
аналізу

Протокол № 1 від 22.08.2023 р.



навчально-методичної

Каріна НЕМАШКАЛО

**ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ**

робоча програма навчальної дисципліни (РПНД)

Галузь знань **12 "Інформаційні технології"**  
Спеціальність **126 "Інформаційні системи та технології"**  
Освітній рівень **перший (бакалаврський)**  
Освітня програма **"Інформаційні системи та технології"**

Статус дисципліни **обов'язкова**  
Мова викладання, навчання та оцінювання **українська**

Розробник:  
к.е.н., доц.

Підписано КЕП

Роман ЯЦЕНКО

Завідувач кафедри  
інформатики та комп'ютерної  
техніки

Лідія ГУР'ЯНОВА

Гарант програми

Ольга ТЮТЮНИК

Харків

2023

## ВСТУП

У теперішній час, найбільш поширеним методом вирішення складності розроблення програмного забезпечення є – об'єктно-орієнтований підхід до його розроблення. Такий підхід потребує від відповідних фахівців чіткого уявлення концепцій об'єктно-орієнтованого програмування, що дає можливість їхнього практичного використання у процесі розроблення застосунків будь-якою мовою програмування.

Навчальна дисципліна "Об'єктно-орієнтоване програмування" є обов'язковою навчальною дисципліною та вивчається згідно з навчальним планом підготовки здобувачів вищої освіти по спеціальності 126 "Інформаційні системи та технології" за освітньою програмою "Інформаційні системи та технології" першого (бакалаврського) рівня.

Метою навчальної дисципліни є формування компетентностей з використання основних елементів об'єктно-орієнтованого підходу та сучасних мов програмування, необхідних для розроблення відповідних програмних застосунків

Об'єкт навчальної дисципліни є основні елементи об'єктно-орієнтованого підходу до розроблення програмного забезпечення.

Предметом навчальної дисципліни є принципи об'єктно-орієнтованого програмування та основи об'єктно-орієнтованої технології.

Завдання навчальної дисципліни – формування об'єктно-орієнтованого мислення; володіння головними концепціями об'єктно-орієнтованого проектування, аналізу та програмування; володіння об'єктно-орієнтованими мовами програмування; формування навичок у розробленні об'єктно-орієнтованих програмних систем; удосконалення навичок у використанні середовищ програмування.

Навчальна програма включає навчання у формі лекцій, лабораторних занять та самостійної роботи здобувачів вищої освіти.

Для практичного засвоєння основних тем дисципліни проводяться лабораторні заняття, індивідуальні роботи та консультації з використанням персональних комп'ютерів, локальних мереж та Інтернету у комп'ютерних класах ХНЕУ С. Кузнеця.

З метою підвищення ефективності вивчення дисципліни здобувачі вищої освіти мають можливість використовувати систему дистанційного навчання ХНЕУ. С. Кузнеця.

У процесі навчання здобувачі вищої освіти отримують необхідні знання під час проведення аудиторних занять: лекційних та лабораторних. Також велике значення в процесі вивчення та закріплення знань має самостійна робота здобувачів вищої освіти.

Результати навчання та компетентності, які формує навчальна дисципліна визначено в табл. 1.

**Результати навчання та компетентності, які формує навчальна дисципліна**

Результати навчання	Компетентності, якими повинен оволодіти здобувач вищої освіти
ПР 3	КЗ 1, КЗ 2, КЗ 3, КЗ 8, КС 4

де, КЗ 1. Здатність до абстрактного мислення, аналізу та синтезу.

КЗ 2. Здатність застосовувати знання у практичних ситуаціях.

КЗ 3. Здатність до розуміння предметної області та професійної діяльності.

КЗ 8. Здатність оцінювати та забезпечувати якість виконуваних робіт.

КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

ПР 3. 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 на мовах високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

## ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### **Змістовий модуль 1. Основи об'єктно-орієнтованої парадигми**

#### **Тема 1. Основи об'єктно-орієнтованої мови програмування.**

**1.1.** Програмні платформи Microsoft .NET та Java SE: архітектура, компіляція та виконання програм, система типізації, стандартні бібліотеки класів, інструментальні засоби розроблення програм.

**1.2.** Загальні відомості про мови C# та Java: алфавіт, типи даних, операції, оператори, структура програми, основи використання стандартних бібліотек класів Microsoft .NET та Java SE.

#### **Тема 2. Поняття об'єктно-орієнтованого аналізу, проектування та програмування.**

**2.1.** Об'єктно-орієнтована декомпозиція. Принципи об'єктно-орієнтованого підходу: абстракція, інкапсуляція, ієрархія, поліморфізм.

**2.2.** Поняття об'єкта. Характеристики об'єкта. Поняття класу. Співвідношення між класом та його об'єктом.

**2.3.** Об'єктно-орієнтований аналіз та його мета. Головні види вимог до програмної системи. Об'єктно-орієнтоване проектування. Об'єктно-орієнтоване програмування.

**2.4.** UML-діаграми класів. Відношення на діаграмі класів. CASE-засоби.

**2.5.** Класи та структури. Елементи класу. Особливості використання статичних елементів. Доступ до елементів класу, модифікатори доступу.

**2.6.** Поняття про створення, ініціалізацію та використання об'єктів класу. Посилання this.

**2.7.** Життєвий цикл об'єктів. Послідовність створення та ініціалізації об'єкта. Конструктори. Конструктор за замовчуванням. Основні властивості конструкторів. Перевантаження конструкторів. Звільнення пам'яті. Система "збирання сміття".

**Змістовий модуль 2. Технологія об'єктно-орієнтованого програмування**

**Тема 3. Повторне використання класів.**

**3.1.** Поняття про асоціацію. Відношення композиції та агрегації як види асоціації. Реалізація композиції та агрегації в C# та Java.

**3.2.** Відношення успадкування. Реалізація відношення успадкування в C# та Java. Ініціалізація об'єкта базового класу. Використання конструкторів під час успадкування. Варіанти використання успадкування. Перевизначення методів.

**3.3.** Раннє та пізнє зв'язування. Віртуальні методи. Реалізація принципу поліморфізму в C# та Java. Рядкове подання об'єкта. Абстрактні класи та методи. Реалізація поліморфної поведінки на базі абстрактного класу. Інтерфейси. Реалізація поліморфної поведінки на базі інтерфейсу.

**Тема 4. Принципи об'єктно-орієнтованого проектування класів.**

**4.1.** Система принципів SOLID. Принцип єдиної відповідальності.

**4.2.** Загальні відомості про шаблони проектування. Застосування основних шаблонів проектування.

**Змістовий модуль 3. Обробка винятків і бібліотеки класів**

**Тема 5. Оброблення виняткових ситуацій.**

**5.1.** Бібліотеки та їхнє використання. Статичні та динамічні бібліотеки.

**5.2.** Види помилок у програмах. Проблеми традиційного підходу до оброблення помилок.

**5.3.** Механізм оброблення винятків. Класи винятків стандартних бібліотек Microsoft .NET та Java SE. Синтаксис оброблення винятків.

**Тема 6. Стандартні бібліотеки класів середовищ розробника програм.**

**6.1.** Загальні відомості про колекції. Основні структури даних стандартних бібліотек колекцій Microsoft .NET та Java SE. Типізовані колекції.

**6.2.** Джерела та споживачі даних. Загальні відомості про потоки введення-виведення даних. Алгоритми роботи потоків введення-виведення даних. Основні класи стандартних бібліотек Microsoft .NET та Java SE для підтримки введення-виведення даних.

**6.3.** Призначення та застосування регулярних виразів. Підтримка регулярних виразів на платформах Microsoft .NET та Java SE. Спеціальні символи, які використовують у регулярних виразах.

Перелік лабораторних занять за навчальною дисципліною наведено в табл. 2

Таблиця 2

### Перелік лабораторних занять

Назва теми та завдання	Зміст
Тема 1. Лабораторна робота 1	Розроблення застосунків з використанням базових елементів об'єктно-орієнтованого проектування з використанням платформи Microsoft .NET та Java SE
Тема 2. Лабораторна робота 2	Розроблення застосунків з використанням агрегації та композиції. Використання коментарів. Налаштування програм у середовищі програмування
Тема 3. Лабораторна робота 3	Застосування успадкування та поліморфізму. Ознаки складних програмних систем. Переваги об'єктно-орієнтованого підходу до розроблення програмних систем
Тема 4. Лабораторна робота 4	Використання шаблонів проектування. Основні властивості конструктора. Конструктор за замовчуванням
Тема 5. Лабораторна робота 5	Використання основних бібліотек Microsoft .NET Framework та Java SE
Тема 6. Лабораторна робота 6	Використання регулярних виразів. Хеш-таблиці та хеш-функції. Сортування вмісту контейнера. Розроблення користувацьких контейнерів

Перелік самостійної роботи за навчальною дисципліною наведено в табл. 3.

Таблиця 3

### Перелік самостійної роботи

Назва теми	Зміст
Тема 1 - 6	Вивчення лекційного матеріалу
Тема 1 - 6	Підготовка до лабораторних робіт
Тема 1 - 6	Підготовка до екзамену

Кількість годин лекційних, практичних (семінарських) та / або лабораторних занять та годин самостійної роботи наведено в робочому плані (технологічній карті) з навчальної дисципліни.

## МЕТОДИ НАВЧАННЯ

У процесі викладання навчальної дисципліни для набуття визначених результатів навчання, активізації освітнього процесу передбачено застосування таких методів навчання, як:

- словесні (лекції за темами 1-6), елементи проблемних лекції (за темами 1-6 навчальної дисципліни);
- наочні (демонстрації включено в усі лекційні та практичні матеріали);
- практичні (лабораторні заняття за темами 1-6 навчальної дисципліни).

В умовах змішаної форми навчання подання лекційного матеріалу та/або проведення лабораторних занять та групових та індивідуальних консультацій відбувається з використанням платформи Zoom, в умовах звичайної аудиторної форми заняття проводяться очно, в аудиторіях та комп'ютерних залах.

## ФОРМИ ТА МЕТОДИ ОЦІНЮВАННЯ

Університет використовує 100 бальну накопичувальну систему оцінювання результатів навчання здобувачів вищої освіти.

**Поточний контроль** здійснюється під час проведення лекційних та лабораторних занять і має на меті перевірку рівня підготовленості здобувача вищої освіти до виконання конкретної роботи і оцінюється сумою набраних балів:

- для дисциплін з формою семестрового контролю екзамен (іспит): максимальна сума – 60 балів; мінімальна сума, що дозволяє здобувачу вищої освіти скласти екзамен (іспит) – 35 балів

**Підсумковий контроль** включає семестровий контроль.

**Семестровий контроль** проводиться у формі семестрового екзамену (іспиту). Складання семестрового екзамену (іспиту) здійснюється під час екзаменаційної сесії.

Максимальна сума балів, яку може отримати здобувач вищої освіти під час екзамену (іспиту) – 40 балів. Мінімальна сума, за якою екзамен (іспит) вважається складеним – 25 балів.

**Підсумкова оцінка за навчальною дисципліною** визначається:

– для дисциплін з формою семестрового контролю екзамен (іспит) – сумуванням балів за поточний та підсумковий контроль.

Під час викладання навчальної дисципліни використовуються наступні контрольні заходи:

Поточний контроль: Лабораторні роботи за варіантом (45 балів), письмові контрольні роботи (15 балів).

Семестровий контроль: Екзамен (40 балів)

Більш детальну інформацію щодо системи оцінювання наведено в робочому плані (технологічній карті) з навчальної дисципліни.

Приклад екзаменаційного білета та критерії оцінювання для навчальної дисципліни.

### **Приклад екзаменаційного білета**

Харківський національний економічний університет імені Семена Кузнеця  
Перший (бакалаврський) рівень вищої освіти  
Спеціальність "Інформаційні системи та технології"  
Освітньо-професійна програма "Інформаційні системи та технології"  
Навчальна дисципліна "Об'єктно-орієнтованого проектування"

#### **Завдання 1 (евристичне). (30 балів)**

Розробити консольну програму, що складається з двох класів – Room (кімната) та MainClass.

Клас Room має явно визначені поля Площа, Висота стелі, Кількість вікон, Наявність меблів, Вартість оренди; конструктор з параметрами та перевизначає метод ToString() класу System.Object (або метод toString() класу java.lang.Object) для формування репрезентації об'єкта класу Room у вигляді рядку Площа#Висота стелі#Кількість вікон#Наявність меблів#Вартість оренди.

Клас MainClass повинен містити статичний метод для виведення до текстового файлу значень полів об'єкту класу Room шляхом неявного виклику перевизначеного методу ToString() та головний метод програми, в якому виконується введення даних кімнати з консолі, створення одного об'єкта класу Room, його ініціалізація з використанням зазначеного конструктора та виклик методу виведення даних.

## **Завдання 2 (діагностичне ). (10 балів)**

Розробити окрему консольну програму, яка дозволяє зчитати дані з текстового файлу в форматі Завдання 1 та вивести їх на консоль у вигляді таблиці.

Програма має містити один клас з трьома статичними методами: головним методом програми, методом для введення даних із зазначеного файлу та методом для виведення даних на консоль.

Кількість стовпців таблиці має дорівнювати кількості полів класу Room із Завдання 1. В її клітинках повинні відображатися значення полів без роздільника. Таблиця повинна створюватися з використанням засобів форматного виведення даних (зокрема, всі числові значення повинні виводитися з певною кількістю знаків після коми).

### **Критерії оцінювання**

**Підсумкові бали за екзамен** складаються із суми балів за виконання всіх завдань, що округлені до цілого числа за правилами математики. Алгоритм вирішення кожного завдання включає окремі етапи, які відрізняються за складністю, трудомісткістю та значенням для розв'язання завдання. Тому окремі завдання та етапи їх розв'язання оцінюються відокремлено один від одного таким чином:

**Завдання 1.** Може бути виставлено від 30 до 0 балів:

30 балів виставляється у випадку, якщо завдання виконано у повному обсязі. Програма працює правильно. Вихідний код програми задовольняє вимогам, вказаним у завданні;

29-25 балів – якщо завдання виконано. Програма працює правильно, але вона розроблена з порушенням однієї з вимог, вказаних у завданні;

24-20 балів – якщо завдання головним чином виконано. Програма працює, але вона розроблена з порушенням двох з вимог, вказаних у завданні;

19-16 балів – якщо завдання виконано з недоліками. Програма працює, але не виконано одну з вимог, вказаних у завданні, або вона розроблена з порушенням трьох вимог, вказаних у завданні;

15-10 балів – якщо завдання виконано з суттєвими недоліками. Програма працює, але не виконано дві з вимог, вказаних у завданні, або вона розроблена з порушенням чотирьох вимог, вказаних у завданні;

9-5 балів – якщо завдання практично не виконано. Програма запускається, але не виконано більше двох з вимог, вказаних у завданні, або більше чотирьох вимог реалізовано з порушеннями;

4-1 бал – якщо завдання не виконано. Програма не запускається, або завершується аварійно, але є програмний код, який частково відповідає поставленому завданню;



0 балів – якщо завдання не виконано, програма відсутня, або програмний код не відповідає поставленому завданню.

**Завдання 2.** Може бути виставлено від 10 до 0 балів:

10 балів виставляється у випадку, якщо завдання виконано у повному обсязі. Програма працює правильно. Вихідний код програми задовольняє вимогам, вказаним у завданні;

9 балів – якщо завдання виконано. Програма працює правильно, але вона розроблена з порушенням однієї з вимог, вказаних у завданні;

8 балів – якщо завдання головним чином виконано. Програма працює, але вона розроблена з порушенням двох з вимог, вказаних у завданні;

7-6 балів – якщо завдання виконано з недоліками. Програма працює, але не виконано одну з вимог, вказаних у завданні, або вона розроблена з порушенням трьох вимог, вказаних у завданні;

5-4 бали – якщо завдання виконано з суттєвими недоліками. Програма працює, але не виконано дві з вимог, вказаних у завданні, або вона розроблена з порушенням чотирьох вимог, вказаних у завданні;

3-2 бали – якщо завдання практично не виконано. Програма запускається, але не виконано більше двох з вимог, вказаних у завданні, або більше чотирьох вимог реалізовано з порушеннями;

2-1 бал – якщо завдання не виконано. Програма не запускається, або завершується аварійно, але є програмний код, який частково відповідає поставленому завданню;

0 балів – якщо завдання не виконано, програма відсутня, або програмний код не відповідає поставленому завданню.

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### Основна

1. Основи об'єктно-орієнтованого програмування [Електронний ресурс] : навч. посіб. / О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко ; Харківський національний економічний університет ім. С. Кузнеця. - Електрон. текстові дан. (2,13 МБ). – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 236 с. [Електронний ресурс]. – Режим доступу : <http://www.repository.hneu.edu.ua/handle/123456789/23847>

2. Об'єктно-орієнтоване програмування : підручник. У 2-х ч. Ч. 2. Об'єктно-орієнтований підхід до розробки програмного забезпечення / С. М. Алхімова. – Київ: КГП ім. Ігоря Сікорського, Вид-во "Політехніка", 2019. – 192 с.

3. Основи об'єктно-орієнтованого програмування : навч. посібник / Гришанович Т. О., Глинчук Л. Я.; ВНУ імені Лесі Українки. Електронні текстові дані (1 файл: 998 КБ). – Луцьк : ВНУ імені Лесі Українки, 2022. – 120 с.

4. Troelsen A. Pro C# 9 with .NET 5 : Foundational Principles and Practices in Programming / A. Troelsen, P. Japikse – Berkly : Apress, 2021. – 1353 p.

#### **Додаткова**

5. Об'єктно-орієнтоване програмування. Лабораторний практикум: навчальний посібник / Л.Л.Омельчук. – Київ: 2021. - 265 с.

6. Об'єктно-орієнтоване програмування мовою Python: навчальний посібник для студентів напрямів підготовки 122 Комп'ютерні науки та 121 Інженерія програмного забезпечення / О.В. Мнушка, В.М. Савченко, О.Б. Мацій – Харків : ХНАДУ, 2021. – 228

7. Lewis J. Java Foundations. Introduction to Program Design and Data Structures / J. Lewis, P. DePasquale, J. Chase – Lancing : Pearson, 2020. – 1104 p.

#### **Інформаційні ресурси**

8. Сайт персональних навчальних систем ХНЕУ ім. С. Кузнеця [Електронний ресурс]. – Режим доступу : <https://pns.hneu.edu.ua/enrol/index.php?id=879>

9. What are UML diagrams, and how can you use them? [Electronic resource]. – Access mode: <https://miro.com/blog/uml-diagram/>

10. C# documentation [Electronic resource]. – Access mode: <https://docs.microsoft.com/en-us/dotnet/csharp/>

11. C# Tutorial [Electronic resource]. – Access mode: <https://www.tutorialspoint.com/csharp/index.htm>

12. Java Tutorial [Electronic resource]. – Access mode: <https://www.w3schools.com/java/default.asp>

13. JDK 17 Documentation [Electronic resource]. – Access mode: <https://docs.oracle.com/en/java/javase/17/>