

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

**ЗАТВЕРДЖЕНО**

на засіданні кафедри  
інформаційних систем.  
Протокол № 1 від 22.08.2023 р.

**ПОГОДЖЕНО**

Проректор з навчально-методичної роботи

  
Каріна НЕМАШКАЛО



**ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ**

робоча програма навчальної дисципліни (РПНД)

Галузь знань	12 "Інформаційні технології"
Спеціальність	121 "Інженерія програмного забезпечення"
Освітній рівень	перший (бакалаврський)
Освітня програма	"Інженерія програмного забезпечення"

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

Розробник:  
к.т.н., с.н.с.

підписано КЕП

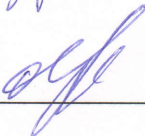
Юрій ПАРФЬОНОВ

Завідувач кафедри  
інформаційних систем

  
\_\_\_\_\_

Дмитро БОНДАРЕНКО

Гарант програми

  
\_\_\_\_\_

Олег ФРОЛОВ

Харків  
2024

## ВСТУП

У наш час найбільш поширеним методом боротьби зі складністю програмного забезпечення є об'єктно-орієнтований підхід до його розроблення. Це 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, яку вивчають, згідно з навчальним планом підготовки фахівців освітнього ступеню "бакалавр" за спеціальністю 121 "Інженерія програмного забезпечення". Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота здобувачів із технічною літературою та сучасними середовищами розроблення програм.

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

Завданнями навчальної дисципліни є формування об'єктно-орієнтованого мислення; володіння головними концепціями об'єктно-орієнтованого проектування, аналізу та програмування; володіння об'єктно-орієнтованими мовами програмування; формування навичок з розроблення об'єктно-орієнтованих програмних систем; удосконалення навичок використанні середовищ програмування.

Об'єктом 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.

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

Таблиця 1

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

Результати навчання	Компетентності, якими повинен оволодіти здобувач вищої освіти
PH7	СК10
PH8	СК1, СК10
PH13	СК2, СК3, СК7
PH15	СК10, СК11, СК13
PH17	ЗК7

де, PH7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

PH8. Вміти розробляти людино-машинний інтерфейс.

PH13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

PH15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

PH17. Вміти застосовувати методи компонентної розробки програмного забезпечення.

СК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

СК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

СК3. Здатність розробляти архітектури, модулі та компоненти програмних систем.

СК7. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

СК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

СК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

СК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

ЗК7. Здатність працювати в команді.

## **ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

### **Зміст навчальної дисципліни**

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

##### **Тема 1. Основи .NET та Java SE**

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

##### **Тема 2. Основи об'єктно-орієнтованої мови програмування**

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

2.2. Одновимірні та багатовимірні масиви в C# та Java: створення, ініціалізація, оброблення, підтримка масивів у стандартних бібліотеках .NET та Java SE.

2.3. Методи в C# та Java: визначення, механізми передачі параметрів, використання масиву як параметра, повертання масиву з методу, виклик методу.

2.4. Перевантаження методів класу. Принципи перевантаження операцій.

**Тема 3. Поняття об'єктно-орієнтованого аналізу, проєктування та програмування**

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

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

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

3.4. Моделювання програмного забезпечення. Основні поняття візуального моделювання. Основи мови UML. UML-діаграми. UML-діаграми класів. Відношення на діаграмі класів. CASE-засоби.

## **Змістовий модуль 2. Технологія ООП**

### **Тема 4. Абстрагування даних та інкапсуляція**

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

4.2. Поняття про створення, ініціалізацію та використання об'єктів класу. Посилання [this](#).

4.3. 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".

### **Тема 5. Повторне використання коду**

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

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

5.3. 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 6. Основи використання системи контролю версій Git**

6.1 Основні поняття систем контролю версій. Локальні, централізовані та розподілені системи контролю версій. Основи Git. Використання локального та віддаленого репозиторіїв.

### **Тема 7. Принципи об'єктно-орієнтованого проектування SOLID**

7.1. Система принципів SOLID. Принцип єдиної відповідальності. Принцип відкриття-закриття. Принцип інверсії залежностей. Принцип заміщення Барбара Ліскова. Принцип ізоляції інтерфейсу.

### **Тема 8. Вступ до шаблонів проектування**

8.1. Загальні відомості про шаблони проектування. Переваги та недоліки використання шаблонів проектування. Елементи шаблону проектування.

8.2. Класифікація шаблонів проєктування GoF. Застосування основних шаблонів проєктування GoF.

8.3. Огляд шаблонів розподілу відповідальностей GRASP.

### **Тема 9. Бібліотеки класів**

9.1. Бібліотеки та їхнє використання. Статичні та динамічні бібліотеки.

9.2. Розроблення бібліотек на платформі Java SE. DLL-бібліотеки. Розроблення DLL-бібліотек на платформі .NET.

## **Змістовий модуль 3. Оброблення винятків і бібліотеки класів**

### **Тема 10. Оброблення виняткових ситуацій**

10.1. Види помилок у програмах. Проблеми традиційного підходу до оброблення помилок.

10.2. Механізм оброблення винятків. Класи винятків стандартних бібліотек .NET та Java SE. Синтаксис оброблення винятків.

### **Тема 11. Стандартні бібліотеки класів середовищ розробника програм**

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

11.2. Використання лямбда-виразів

11.3. Використання LINQ та Stream API

11.4. Особливості реалізації рядкового типу даних на платформах .NET та Java SE. Класи стандартних бібліотек .NET та Java SE для подання рядків та особливості їхнього використання.

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

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

11.7. Збереження та відновлення стану об'єктів на платформах .NET та Java SE. Серіалізація та десеріалізація. «Граф» об'єктів при серіалізації. Створення класів, об'єкти яких можливо серіалізувати. Процеси серіалізації та десеріалізації. Формати серіалізації.

11.8. Загальні відомості про об'єктно-реляційне відображення даних. Поняття про ORM-фреймворки для платформ .NET та Java SE. Вступ до моделей. Класи моделей. Налаштування моделей. Міграції бази даних.

11.9. Розподілені програмні системи. Програмні клієнти та сервери. Загальні відомості про TCP-сокети. Основи використання TCP-сокетів на платформах .NET та Java SE.

Перелік лабораторних занять, а також питань та завдань до самостійної роботи наведено в табл. 2.

**Перелік лабораторних занять**

Назва теми	Зміст
Тема 2. Лабораторне заняття 1	Основи використання мов програмування C# та Java
Тема 3. Лабораторне заняття 2	Проектування класів з використанням мови UML
Тема 4. Лабораторне заняття 3	Розроблення застосунків з використанням базових елементів ООП
Тема 5. Лабораторне заняття 4	Застосування успадкування та поліморфізму
Тема 6. Лабораторне заняття 5	Застосування системи контролю версій Git
Тема 8. Лабораторне заняття 6	Використання шаблонів проектування
Тема 11. Лабораторне заняття 7	Використання колекцій
Тема 11. Лабораторне заняття 8	Використання лямбда-виразів, LINQ та Stream API
Тема 11. Лабораторне заняття 9	Використання регулярних виразів
Тема 11. Лабораторне заняття 10	Використання введення-виведення даних
Тема 11. Лабораторне заняття 11	Використання об'єктно-реляційного відображення даних
Тема 11. Лабораторне заняття 12	Використання TCP-сокетів

Перелік самостійної роботи за навчальною дисципліною наведено в табл. 3

**Перелік самостійної роботи**

Назва теми	Зміст
Теми 1 - 11	Вивчення лекційного матеріалу
Теми 1 - 11	Підготовка до лабораторних занять
Теми 1 - 11	Підготовка до екзамену

Кількість годин лекційних, 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 (лекція (Теми 2, 4, 5, 6, 8 - 11), проблемні лекції (Теми 1, 3, 7)).

Наочні (демонстрація (Теми 1 - 11)).

Практичні (лабораторні заняття, інтерактивне дистанційне навчання, коучинг, презентація, рольові ігри (Теми 1 - 11); кейс-метод (Теми 6, 11)).

## 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.

– для дисциплін з формою семестрового контролю залік: максимальна сума – 100 балів; мінімальна сума – 60 балів.

**Підсумковий контроль** включає семестровий контроль та атестацію здобувача вищої освіти.

**Семестровий контроль** проводиться у формах семестрового екзамену (іспиту), диференційованого заліку. 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 залік – сумуванням всіх балів, отриманих під час поточного контролю.

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

Під час викладання навчальної дисципліни використовуються наступні контрольні заходи:

### 1 семестр

Поточний контроль: захист лабораторних робіт (83 бали), письмова контрольна робота (17 балів).

Семестровий контроль: Диференційований залік

### 2 семестр

Поточний контроль: захист лабораторних робіт (50 балів), письмова контрольна робота (10 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 ім. С. Кузнеця

Освітній ступінь: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Семестр: 4

Навчальна дисципліна: Об'єктно-орієнтоване програмування

### ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1

Завдання 1.

Розробити клас Point3D (точка у тривимірному просторі) з полями X, Y, Z, що є координатами точки в декартовій системі. Цей клас також повинен мати метод для розрахунку відстані від початку координат до точки за формулою  $(X^2 + Y^2 + Z^2)^{1/2}$ , та властивості **set** та **get** для встановлення координат точки та одержання їх поточних значень відповідно.

Розробити ще один клас, який містить метод Main(), у якому виконується:

1. Введення з консолі координат однієї точки, створення відповідного об'єкту класу Point3D та його ініціалізація з використанням властивостей **set**.
2. Виведення на консоль координат цієї точки та відстані від неї до початку координат з використанням створеного об'єкту класу Point3D та властивості **get**.

Завдання 2.

Створити три класи: Group (група), Student (студент), MainClass (містить головний метод програми). Класи Group та Student мають бути зв'язані відношенням **агрегації**, при якому **група складається з трьох студентів**.

Клас Student як мінімум має поле LastName – прізвище студента. Клас Group як мінімум має містити метод AddStudent з параметрами, призначений для додавання студента до групи, та метод PrintList для виведення на консоль прізвищ студентів групи.

У головному методі програми як мінімум має виконуватися:

1. Створення одного об'єкту класу Group.
2. Введення з консолі початкових даних щодо студентів та виклик методу AddStudent.
3. Виклик методу PrintList.

Затверджено на засіданні кафедри "Інформаційних систем", протокол № \_\_\_ від \_\_.08.20\_\_ р.

Завідувач кафедру \_\_\_\_\_ доц. Дмитро БОНДАРЕНКО

Екзаменатор \_\_\_\_\_ доц. Юрій ПАРФЬОНОВ



## Критерії оцінювання

Екзаменаційний білет складається з двох завдань. Кожне завдання оцінюється від 0 до 6 умовних балів відповідно до наступної шкали:

6 балів	Завдання виконано в повному обсязі. Програма працює правильно. Інтерфейс та вихідний код програми задовольняють вимогам, вказаним в завданні.
5 балів	Завдання виконано. Програма працює правильно, але одна з її функціональних можливостей реалізована з порушенням вимог, вказаних в завданні.
4 бали	Завдання в основному виконано. Програма працює, але дві з її функціональних можливостей реалізовані з порушенням вимог, вказаних в завданні.
3 бали	Завдання виконано, але не в повному обсязі. Програма працює, але не реалізована одна з функціональних вимог, вказаних в завданні, або три з функціональних можливостей програми реалізовані з порушенням вимог, вказаних в завданні.
2 бали	Завдання не виконано. Програма запускається, але не реалізовані дві з функціональних вимог, вказаних в завданні, або чотири з функціональних можливостей програми реалізовані з порушенням вимог, вказаних в завданні.
1 бал	Програма запускається та частково відповідає постановці завдання, але не реалізовані більше двох з функціональних вимог, вказаних в завданні, або більш чотирьох з функціональних можливостей програми реалізовані з порушенням вимог, вказаних в завданні. Програма не запускається або завершується аварійно, але є програмний код, розроблений здобувачем, який відповідає постановці завдання.
0 балів	Програма відсутня. Програма не містить програмного коду, розробленого здобувачем. Програма не відповідає постановці завдання. Програма має явні технічні ознаки несамостійності її розробки.

*Невиконання або суттєве порушення будь-якої з загальних вимог до програм зніжує кількість умовних балів по завданню на 1 (один).*

### **Загальні вимоги до програм**

- 1. Користувацький інтерфейс консольної програми, розробленої відповідно до завдання білету, має складатися з текстових повідомлень українською (англійською) мовою, які відносяться до введення та виведення основних та допоміжних даних. Використання транслітерації заборонено.*
- 2. Додержання принципу інкапсуляції щодо рівнів доступу до полів, властивостей та методів класів.*
- 3. Вихідний код кожного з класів програми має знаходитися в окремому файлі.*

Підсумкова оцінка за іспит визначається набраною сумою балів за виконання двох завдань:

Набрана сума умовних балів	Підсумкова оцінка за іспит за 40-бальною шкалою
12	40
11	38 - 39
10	36 - 37
9	34 - 35

8	32 - 33
7	30 - 31
6	28 - 29
5	26 - 27
4	24 - 25
3	16 - 23
2	8 - 15
1	1 - 7
0	0

## **РЕКОМЕНДОВАНА ЛІТЕРАТУРА**

### **Основна**

1. Booch G. Object-Oriented Analysis and Design with Applications / G.Booch, R.A.Maksimchuk et. al. – Boston : Addison-Wesley, 2019. – 717 p.
2. Troelsen A. Pro C# 9 with .NET 5 : Foundational Principles and Practices in Programming / A. Troelsen, P. Japikse – Berkly : Apress, 2021. – 1353 p.

### **Додаткова**

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

### **Інформаційні ресурси**

4. What are UML diagrams, and how can you use them. [Electronic resource]. – Access mode: <https://miro.com/blog/uml-diagram/>
5. C# documentation [Electronic resource]. – Access mode: <https://docs.microsoft.com/en-us/dotnet/csharp/>
6. C# Tutorial [Electronic resource]. – Access mode: <https://www.tutorialspoint.com/csharp/index.htm>
7. Java Tutorial [Electronic resource]. – Access mode: <https://www.w3schools.com/java/default.asp>
8. JDK 20 Documentation [Electronic resource]. – Access mode: <https://docs.oracle.com/en/java/javase/20/>
9. Щербаков, О. В. Основи об'єктно-орієнтованого програмування : навч. посіб. / О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко - Харків : ХНЕУ ім. С. Кузнеця, 2019. - 236 с. [Електронний ресурс]. - Режим доступу: <http://repository.hneu.edu.ua/handle/123456789/23847>
10. Об'єктно-орієнтоване програмування. Методичні рекомендації до лабораторних робіт для студентів спеціальності 121 "Інженерія програмного забезпечення" першого (бакалаврського) рівня / уклад. Ю. Е. Парфьонов, О. В. Щербаков - Харків : ХНЕУ ім. С. Кузнеця, 2021. - 92 с. [Електронний ресурс]. - Режим доступу: <http://repository.hneu.edu.ua/handle/123456789/27252>

11. Персональна навчальна система "Об'єктно-орієнтоване програмування (121, осінній семестр)" [Електронний ресурс]. - Режим доступу: <https://pns.hneu.edu.ua/course/view.php?id=9127>

12. Персональна навчальна система "Об'єктно-орієнтоване програмування (121, весняний семестр)" [Електронний ресурс]. - Режим доступу: <https://pns.hneu.edu.ua/course/view.php?id=5377>