

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

ЗАТВЕРДЖЕНО

на засіданні кафедри
інформаційних систем
Протокол № 8 від 17.01.2025 р.

ПОГОДЖЕНО

Проректор з навчально-методичної роботи


Каріда ДЕМАШКАЛО

ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

робоча програма навчальної дисципліни (РПНД)

Галузь знань	12 "Інформаційні технології"
Спеціальність	122 "Комп'ютерні науки"
Освітній рівень	перший (бакалаврський)
Освітня програма	"Комп'ютерні науки"

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

Дмитро ГОЛУБНИЧИЙ

Харків
2025

ВСТУП

У наш час найбільш поширеним методом боротьби із 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 "Основи об'єктно-орієнтованого програмування" є обов'язковою навчальною дисципліною, яку вивчають, згідно з навчальним планом підготовки фахівців освітнього ступеню "бакалавр" за спеціальністю 122 "Комп'ютерні науки". Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота здобувачів із технічною літературою та сучасними середовищами розроблення програм.

Метою навчальної дисципліни "Основи об'єктно-орієнтованого 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 використання середовищ програмування.

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

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

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
ПР1.	ЗК1, ЗК4, СК17.
ПР9.	ЗК1, ЗК2, ЗК3, ЗК6, ЗК9, ЗК10, СК7.
ПР11.	ЗК1, ЗК2, ЗК6, ЗК9, ЗК10, СК9.
ПР13.	ЗК1, ЗК2, ЗК4, ЗК5, ЗК9, ЗК10, СК11.
ПР14.	ЗК2, ЗК10, ЗК13, СК10.
ПР16.	ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК8, ЗК9, ЗК13, СК1, СК14.
ПР18.	СК17.

де, ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР9. 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.

ПР11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

ПР13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

ПР14. Виконувати установку та налаштування сервісів (служб) для управління та діагностики роботи комп'ютерних мереж різних рівнів (глобальних, корпоративних, локальних), знати стандарти дротових та бездротових мереж, використовувати інструментальні засоби для проектування та моделювання комп'ютерних мереж.

ПР16. Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних.

ПР18. Розробляти рекомендації та заходи, організовувати та професійно брати участь в реалізації ІТ-рішень в діяльності щодо підвищення ефективності тестування.

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК5. Здатність спілкуватися іноземною мовою.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК9. Здатність працювати в команді.

ЗК10. Здатність бути критичним і самокритичним.

ЗК13. Здатність діяти на основі етичних міркувань.

СК1. Здатність до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтовування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування.

СК7. Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних

об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.

СК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт- сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

СК11. Здатність до інтелектуального аналізу даних на основі методів обчислювального інтелекту включно з великими та погано структурованими даними, їхньої оперативної обробки та візуалізації результатів аналізу в процесі розв'язування прикладних задач.

СК14. Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.

СК17. Здатність до організації роботи з використанням сучасних засобів забезпечення якості програмного забезпечення шляхом використання сучасної технології тестування інформаційних систем та їх компонентів.

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Зміст навчальної дисципліни

Змістовий модуль 1. Основи об'єктно-орієнтованої парадигми

Тема 1. Основи об'єктно-орієнтованої мови програмування

1.1. Програмні платформи .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 .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 ООП

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 .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 .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 .NET та Java SE для підтримки введення-виведення даних.

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

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

Таблиця 2

Перелік лабораторних занять

Назва теми	Зміст
Тема 2.	Розроблення застосунків з використанням базових елементів ООП
Тема 3.	Розроблення застосунків з використанням агрегації та композиції
Тема 3.	Застосування успадкування та поліморфізму
Тема 4.	Використання шаблонів проєктування
Тема 6.	Використання основних бібліотек .NET Framework та Java SE
Тема 6.	Використання регулярних виразів

Перелік самостійної роботи за навчальною дисципліною наведено в табл. 3

Таблиця 3

Перелік самостійної роботи

Назва теми	Зміст
Теми 1 - 6	Вивчення лекційного матеріалу
Теми 1 - 6	Підготовка до лабораторних занять
Теми 1 - 6	Підготовка до екзамену

Кількість годин лекційних та 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, 3, 5, 6), проблемні лекції (Теми 1, 4)).

Наочні (демонстрація (Теми 1 - 6)).

Практичні (лабораторні заняття, інтерактивне дистанційне навчання, коучинг, презентація, рольові ігри (Теми 1 - 6); кейс-метод (Теми 2, 6)).

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 та атестацію здобувача вищої освіти.

Семестровий контроль проводиться у формах 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: 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

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ

Завдання 1

Розробити консольну програму, що складається з двох класів – Employee та MainClass.

Клас Employee має явно визначені поля Прізвище, Ім'я, По батькові, Посада; конструктор з параметрами та методи, кожний з яких повертає поточне значення відповідного поля.

Клас MainClass повинен містити статичний метод для виведення до текстового файлу значень полів об'єкта класу Employee (кожне значення має знаходитися в окремому рядку) та головний метод програми (введення даних

студента з консолі, створення одного об'єкта класу Employee, його ініціалізація з використанням зазначеного конструктора, виклик методу виведення даних).

Завдання 2

Розробити окрему консольну програму (без використання жодного коду із програми Завдання 1), 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 Student із Завдання 1. Таблиця повинна створюватися з використанням засобів форматного виведення даних (зокрема, всі числові значення повинні виводитися з певною кількістю знаків після коми).

Критерії оцінювання виконання екзаменаційних завдань

Екзаменаційний білет містить два комплексних евристичних завдання (Завдання 1 є обов'язковим). Результатам виконання кожного із завдань присвоюється рейтинг від 0 до 6 відповідно до наступної шкали:

6	Завдання виконано в повному обсязі. Програма працює правильно. Вихідний код програми задовольняє вимогам, вказаним в завданні.
5	Завдання виконано. Програма працює правильно, але вона розроблена з порушенням однієї з вимог, вказаних в завданні.
4	Завдання головним чином виконано. Програма працює, але вона розроблена з порушенням двох з вимог, вказаних в завданні.
3	Завдання виконано із суттєвими недоліками. Програма працює, але не виконано одну з вимог, вказаних в завданні, або три з таких вимог реалізовано з порушеннями
2	Завдання практично не виконано. Програма запускається, але не виконано дві з вимог, вказаних в завданні, або чотири з таких вимог реалізовано з порушеннями
1	Завдання не виконано. Програма запускається та частково відповідає постановці завдання, але не виконано більше двох з вимог, вказаних в завданні, або більш чотирьох з таких вимог реалізовано з порушенням. Чи програма не запускається або завершується аварійно, але є програмний код, розроблений студентом, який відповідає постановці завдання. Частина файлів відповідно до структури проєкту відсутня. Надані студентом результати унеможливають перевірку виконання зазначених в екзаменаційних завданнях вимог.
0	Всі файли відповідно до структури проєкту відсутні; проєкт не містить вихідного коду, розробленого студентом; розроблена програма не відповідає постановці завдання; проєкт має ознаки, які свідчать про розроблення його поза межами часу іспиту або про несамотійність такого розроблення;

Невиконання або суттєве порушення будь-якої з загальних вимог до програм зніжує рейтинг результатів виконання завдання 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. Дотримання принципу інкапсуляції щодо рівнів доступу до полів, властивостей та методів класів.

3. Вихідний код кожного з класів програми має знаходитися в окремому файлі.

За виконання обов'язкового Завдання 1 екзаменаційного білету можна отримати до 90% від максимальної оцінки.

Завдання 2 екзаменаційного білету є додатковим, за його виконання можна отримати до 10% від максимальної оцінки.

Оцінка за 40-бальною шкалою за виконання кожного з завдань визначається таким чином:

Рейтинг результатів виконання завдання	Оцінка за виконання Завдання 1	Оцінка за виконання Завдання 2
6	36	4
5	33 - 35	3
4	30 - 32	2
3	26 - 29	
2	20 - 25	1
1	1 - 19	
0	0	0

Підсумкова оцінка за іспит за 40-бальною шкалою визначається сумою оцінок за виконання завдань екзаменаційного білету.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Booch G. Object-Oriented Analysis and Design with Applications / G.Booch, R.A.Maksimchuk et. al. – Boston : Addison-Wesly, 2019. – 717 p.

2. Troelsen A. Pro C# 10 with .NET 6 : Foundational Principles and Practices in Programming / A. Troelsen, P. Japikse – Berkly : Apress, 2022. – 1705 p.

3. 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>

Додаткова

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

5. Об'єктно-орієнтоване програмування. Методичні рекомендації до лабораторних робіт для студентів спеціальності 122 "Комп'ютерні науки", 126 "Інформаційні системи та технології" першого (бакалаврського) рівня / уклад. Ю. Е. Парфьонов - Харків : ХНЕУ ім. С. Кузнеця, 2021. - 56 с. [Електронний ресурс]. - Режим доступу: <http://repository.hneu.edu.ua/handle/123456789/25309>

Інформаційні ресурси

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

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

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

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

10. JDK 23 Documentation [Electronic resource]. – Access mode:
<https://docs.oracle.com/en/java/javase/23/>

11. Персональна навчальна система "Основи об'єктно-орієнтованого програмування" [Електронний ресурс]. - Режим доступу: <https://pns.hneu.edu.ua/course/view.php?id=879>