Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота № 4

з дисципліни «Мова програмування Java»

Тема: «Generics»

Варіант 119

Виконала:

Студентка групи ІА-31

Соколова Поліна

Київ – 2025

Мета роботи – засвоїти принципи та підходи використання узагальненого програмування; знайомство з модульним тестуванням за допомогою фреймворку JUnit.

Завдання 4.2

Реалізувати із застосування узагальненого програмування ієрархію Java-класів для вольєрів для різних видів тварин.

Хід роботи

Клас Animal - абстрактний клас, який описує спільні властивості всіх тварин. Зберігає ім’я тварини (name) та використовується як базовий тип для всіх нащадків.

Реалізовані класи:

Mammal — абстрактний клас для ссавців;

Bird — абстрактний клас для птахів;

Lion, Zebra, Giraffe, Eagle — конкретні види тварин.

Клас Cage<T extends Animal> - узагальнений клас вольєра для зберігання обмеженої кількості тварин певного типу.

Generics:
Оголошення <T extends Animal> означає, що параметр типу T повинен бути підтипом Animal, що унеможливлює створення вольєра для інших об’єктів.

Основні поля:

- int capacity — максимальна кількість тварин;
- List<T> occupants — список тварин у вольєрі.

Методи:

- addAnimal(T animal) — додає тварину, при переповненні ініціює CageFullException;

- removeAnimal(T animal) — вилучає тварину, при відсутності тварини кидає AnimalNotFoundException;
- getOccupiedCount() — повертає кількість зайнятих місць.

Вольєр є універсальним для будь-яких нащадків Animal, тому створюються підкласи для конкретних груп.

Спеціалізовані класи вольєрів:

- LionCage extends Cage<Lion> — дозволяє лише левів.
- BirdCage extends Cage<Bird> — дозволяє лише птахів.
- HoofedCage extends Cage<Hoofed> — дозволяє лише зебр і жирафів.

Кожен підклас фіксує тип параметра Generics і забезпечує типобезпеку на етапі компіляції.

Клас Zoo реалізує колекцію вольєрів і загальні функції зоопарку, зокрема підраховує загальну кількість тварин.

Використано підстановочний тип (Wildcard) Cage<? extends Animal>, що дозволяє створювати вольєри різних типів (Cage<Lion>, Cage<Bird>, Cage<Hoofed>), об’єднувати їх у спільну колекцію List<Cage<? extends Animal>> зберігаючи при цьому типову безпеку при додаванні елементів.

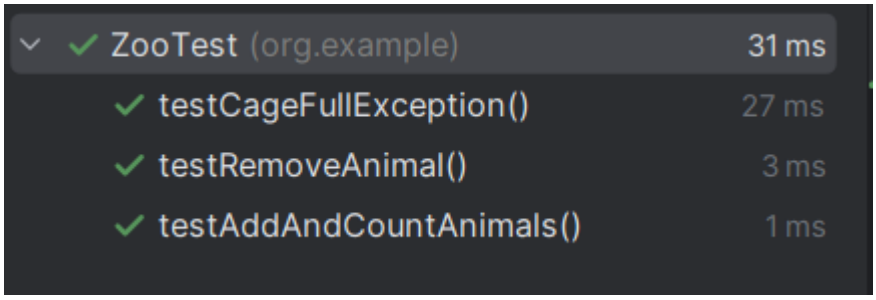
Для обробки виняткових ситуацій реалізовано два класи винятків: CageFullException — при переповненні вольєра і AnimalNotFoundException — при спробі вилучити відсутню тварину.

Також реалізоване модульне тестування з використанням JUnit 5.
Перевіряються такі аспекти:

1. Додавання допустимих тварин у відповідні вольєри і підрахунок загальної кількості тварин у зоопарку.
2. Перевищення місткості — генерується CageFullException.

3. Видалення тварин і генерація AnimalNotFoundException при відсутності.

Тести пройшли успішно:



Висновок: у результаті виконання лабораторної роботи я побудовала ієрархію класів тварин і вольєрів. Реалізовано узагальнені класи з обмеженням типів, типобезпечні спеціалізації вольєрів, використано wildcard зберігання вольєрів різних типів у зоопарку, додано перевірку виняткових ситуацій і модульне тестування.