

## Task 4 Generics

### Варіант 1 (advanced)

Добавить поддержку generics к task 2 advanced.

Обязательно найти место для применения wildcards.

Пример добавления wildcard для List: `boolean addAll(Collection<? extends E> c);`

### Или варианты из стандартного набора

**Варіант 2.** Реалізувати із застосування узагальненого програмування ієрархію Java-класів для транспортних засобів, які можуть перевозити різні типи пасажирів.

Є наступні транспортні засоби: автобус, таксі, пожежна машина, міліцейська машина. У цих транспортних засобах можуть їздити наступні види пасажирів: звичайний пасажир, пожежник, міліціонер.

Ієрархія вказаних об'єктів подана на рис. 1. В ході виконання роботи ієрархія може бути доопрацьована.

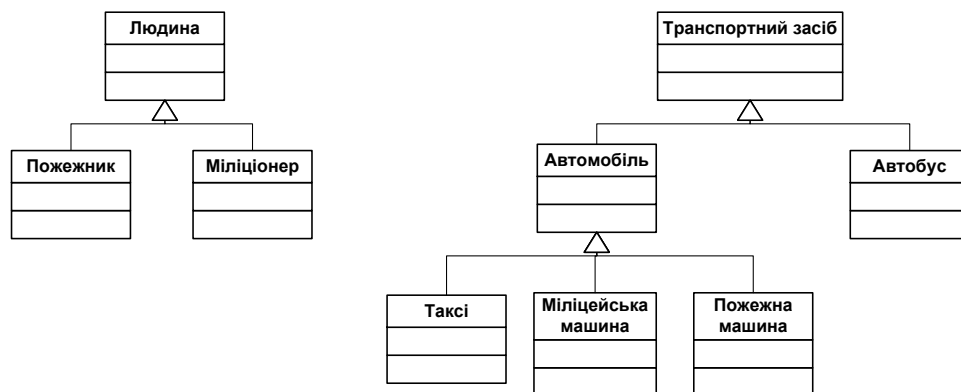


Рис. 1.

Автобус та таксі можуть перевозити будь-яких пасажирів, пожежна машина – тільки пожежників, міліцейська машина – тільки міліціонерів. Реалізувати на основі узагальненого програмування (generics) вказані обмеження щодо перевезки пасажирів.

Для класів транспортних засобів реалізувати наступні функції:

- Кожний транспортний засіб має обмежену кількість місць. Реалізувати функцію для отримання максимальної кількості місць та функцію для отримання кількості зайнятих місць.
- Посадка пасажирів у транспортний засіб. Якщо всі місця вже зайнято, функція повинна ініціювати виключну ситуацію.
- Висадка пасажирів із транспортного засобу. Функція повинна ініціювати виключну ситуацію, якщо вказаний пасажир «не сидить» у транспортному засобі.

Додатково реалізувати функцію підрахунку кількості людей, які перевозяться на автомобілями на певній ділянці дороги. Варіант коду наданий нижче. Дописати код до працездатного. Обов'язково використовувати **generics** та **wildcard** (де це потрібно).

```
public class Road {
    public List<Vehicle> carsInRoad = new ArrayList<>();
    public int getCountOfHumans(){....}
    public void addCarToRoad( .... ){ ... }
}
```

Реалізувати модульні тести з наповнення транспортних засобів різними типами пасажирів.

**Варіант 3.** Реалізувати із застосування узагальненого програмування ієрархію Java-класів для вольєрів для різних видів тварин.

У зоопарку є наступні види тварин: птахи (орли) та ссавці (леви, зебри, жирафи).

Для цих тварин передбачено вольєри: вольєри для левів, вольєри для копитних (зебр та жирафів), вольєри для птахів.

Ієрархія вказаних об'єктів подана на рис. 2. В ході виконання роботи ієрархія може бути доопрацьована.

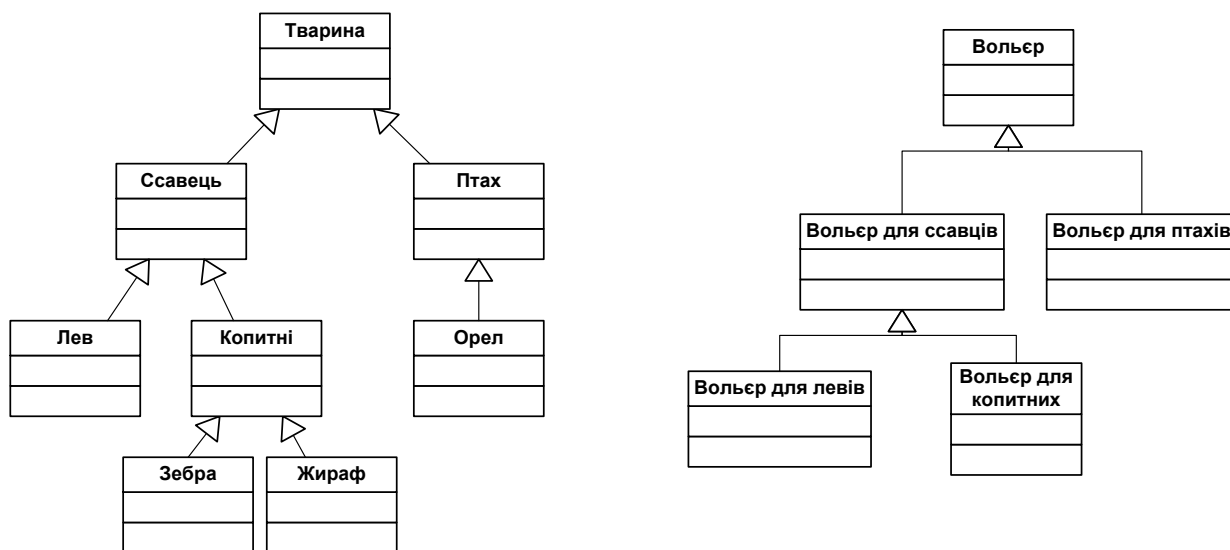


Рис. 2.

У вольєрі для левів можна розмістити тільки левів. У вольєрі для птахів – тільки птахів. К вольєрі для копитних – тільки зебр та жирафів. Реалізувати на основі узагальненого програмування (generics) вказані обмеження щодо розміщення тварин.

Для класів вольєрів реалізувати наступні функції:

Кожен вольєр має максимальну ємність (кількість тварин). Реалізувати функцію для отримання максимальної кількості місць та функцію для отримання кількості зайнятих місць.

Розміщення тварини у вольєрі. Якщо всі м'ясця у вольєрі зайнято, функція повинна ініціювати виключну ситуацію.

Вилучення тварини із вольєра. Функція повинна ініціювати виключну ситуацію, якщо вказаної тварини немає у вольєрі.

**Додатково реалізувати функцію підрахунку кількості тварин, які знаходяться у вольєрах зоопарку.**

**Варіант коду наданий нижче. Дописати код до працездатного. Обов'язково використовувати generics та wildcard (де це потрібно).**

```
public class Zoo {  
    public List<Cage> cages = new ArrayList<>();  
    public int getCountOfAnimals () {...}  
    public void addCage( .... ){ ... }  
}
```

Реалізувати модульні тести з наповнення вольєрів.