

# ЛАБОРАТОРНА РОБОТА №7

## ОБ'ЄКТНО-ОРІЄНТОВАНА ДЕКОМПОЗИЦІЯ

Мета: Використання об'єктно-орієнтованого підходу для розробки об'єкта предметної (прикладної) галузі.

### 1 ВИМОГИ

#### 1.1 Розробник

- Котенко Сергій Миколайович;
- Студент групи КІТ 102.8(а);
- 19-12-2019р..

#### 1.2 Загальне завдання

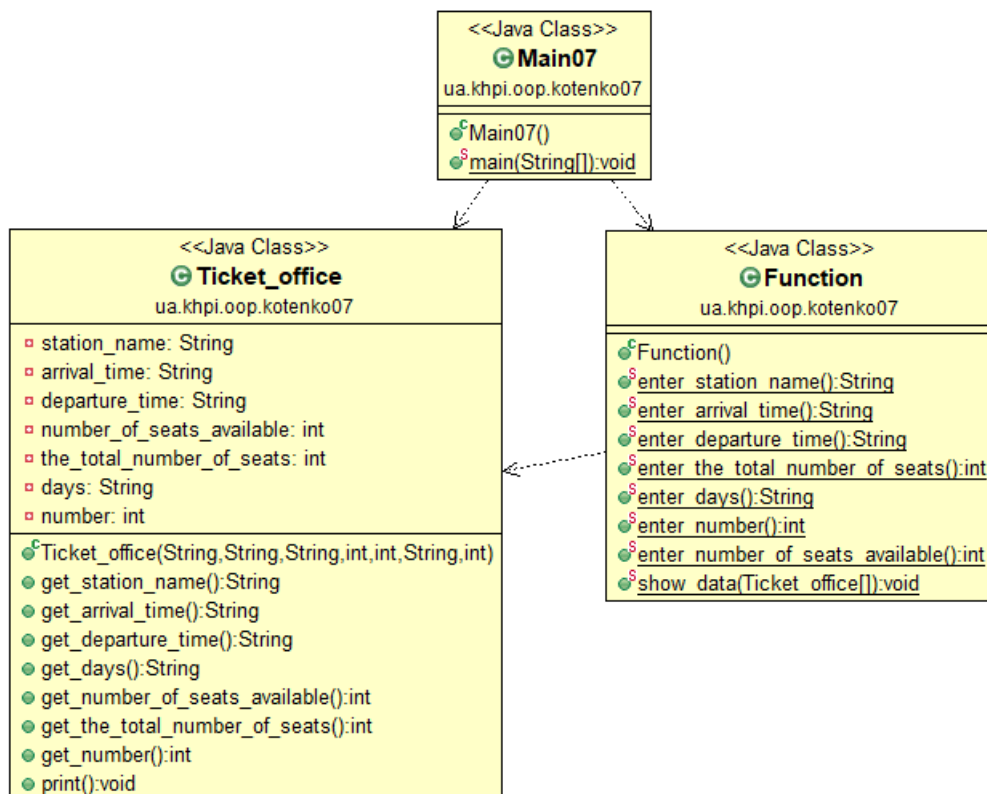
Використовуючи об'єктно-орієнтований аналіз, реалізувати класи для представлення сутностей відповідно прикладної задачі - domain-об'єктів.

Забезпечити та продемонструвати коректне введення та відображення кирилиці.

Продемонструвати можливість управління масивом domain-об'єктів.

### 2 ОПИС ПРОГРАМИ

#### 2.1 Опис логічної структури



## 2.2 Фрагменти коду

```
public class Ticket_office {  
    private String station_name; // Название станции  
    private String arrival_time; // Время прибытия  
    private String departure_time; // Время отправки  
    private int number_of_seats_available; // Кол-во свободных мест  
    private int the_total_number_of_seats; // Мест всего  
    private String days; // День недели  
    private int number; // Номер рейса
```

Рисунок 2.1 – Створене клас

```
    public String get_station_name() {  
        return station_name;  
    }  
  
    public String get_arrival_time() {  
    }  
  
    public String get_departure_time() {  
    }  
  
    public String get_days() {  
    }  
  
    public int get_number_of_seats_available() {  
    }  
  
    public int get_the_total_number_of_seats() {  
    }  
  
    public int get_number() {  
    }
```

Рисунок 2.2 – Створені гетери

```
public void print() {  
    System.out.println("Route:");  
    System.out.printf("Station name: " + this.station_name + '\n'  
        + "Flight number: " + this.number + '\n'  
        + "Days: " + this.days + '\n'  
        + "Time: " + this.departure_time + " - " + this.arrival_time + '\n'  
        + "The total number of seats: " + this.the_total_number_of_seats + '\n'  
        + "Number of seats available: " + this.number_of_seats_available);  
}
```

Рисунок 2.3 – Вивід даних

```
public static String enter_station_name() {  
    @SuppressWarnings("resource")  
    Scanner in = new Scanner(System.in);  
    System.out.print("Enter station name: ");  
    String station_name = in.nextLine();  
    return station_name;  
}
```

Рисунок 2.4 – Введення даних

## 3 ВАРІАНТИ ВИКОРИСТАННЯ

### 3.1 Ілюстрація роботи програми

```
Enter the size of array: 1
Enter station name: Kharkiv
Enter arrival time: 16:00
Enter departure time: 13:00
Enter the total number of seats: 45
Enter flight number : 30445
Enter week days: Tuesday
Enter number of seats available: 5
Route:
Station name: Kharkiv
Flight number: 5
Days: Tuesday
Time: 13:00 - 16:00
The total number of seats: 30445
Number of seats available: 45
```

Рисунок 3.1 – Результат роботи програми (Введення та вивід даних)

## ВИСНОВОК

В інтегрованому середовищі *Eclipse* розроблена програма мовою Java. Виконання програми дозволяє продемонструвати роботу об'єктно-орієнтованого підходу для розробки об'єкта предметної (прикладної) галузі.