

Лабораторна робота 7. Об'єктно-орієнтована декомпозиція

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

ВИМОГИ

- 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.

9. Квиткова каса

Дані про маршрут: маршрут - необмежений набір значень у вигляді "назва станції, час прибуття (для проміжних і кінцевої), час відправлення (для початкової та проміжних), кількість вільних місць"; загальна кількість місць; дні тижня; номер рейсу.

1.1. Розробник: Капелька Ярослав Іванович, КІТ-119а, варіант №9.

2. ОПИС ПРОГРАМИ

2.1. Засоби ООП: клас, метод класу, поле класу.

2.2. Ієрархія та структура класів: один публічний клас Main та публічний клас RouteClass, у полях якого є назва станції, час прибуття (для проміжних і кінцевої), час відправлення (для початкової та проміжних), кількість вільних місць"; загальна кількість місць; дні тижня; номер рейсу, гетери, сетери, конструктор класу та метод виведення даних класу.

2.3. Важливі фрагменти програми:

```
public class RouteClass
{
    private String[] station_name;
    private String[] departure_time;
    private String[] arrival_time;
    private String[] starting_station;
    private String[] terminal_station;
    private String[] intermediate_stations;
    private String[] departure_time_of_intermediate_stations;
    private String[] arrival_time_of_intermediate_stations;
    private int[] number_of_free_seats;
    private int total_number_of_seats;
    private ArrayList<GregorianCalendar> days_of_the_week;
    private int flight_number;

    public String[] getStation_name()
    {
        return station_name;
    }
}
```

```

    }
    public void setStation_name(String[] station_name)
    {
        this.station_name = station_name;
    }
    public String[] getDeparture_time()
    {
        return departure_time;
    }
    public void setDeparture_time(String[] departure_time)
    {
        this.departure_time = departure_time;
    }
    public String[] getArrival_time()
    {
        return arrival_time;
    }
    public void setArrival_time(String[] arrival_time)
    {
        this.arrival_time = arrival_time;
    }
    public String[] getStarting_station()
    {
        return starting_station;
    }
    public void setStarting_station(String[] starting_station)
    {
        this.starting_station = starting_station;
    }
    public String[] getTerminal_station()
    {
        return terminal_station;
    }
    public void setTerminal_station(String[] terminal_station)
    {
        this.terminal_station = terminal_station;
    }
    public String[] getIntermediate_stations()
    {
        return intermediate_stations;
    }
    public void setIntermediate_stations(String[] intermediate_stations)
    {
        this.intermediate_stations = intermediate_stations;
    }
    public String[] getDeparture_time_of_intermediate_stations()
    {
        return departure_time_of_intermediate_stations;
    }
    public void setDeparture_time_of_intermediate_stations(String[]
departure_time_of_intermediate_stations)
    {
        this.departure_time_of_intermediate_stations =
departure_time_of_intermediate_stations;
    }
    public String[] getArrival_time_of_intermediate_stations()
    {
        return arrival_time_of_intermediate_stations;
    }
    public void setArrival_time_of_intermediate_stations(String[]
arrival_time_of_intermediate_stations)
    {
        this.arrival_time_of_intermediate_stations =
arrival_time_of_intermediate_stations;
    }
    public int[] getNumber_of_free_seats()

```

```

    {
        return number_of_free_seats;
    }
    public void setNumber_of_free_seats(int[] number_of_free_seats)
    {
        this.number_of_free_seats = number_of_free_seats;
    }
    public int getTotal_number_of_seats()
    {
        return total_number_of_seats;
    }
    public void setTotal_number_of_seats(int total_number_of_seats)
    {
        this.total_number_of_seats = total_number_of_seats;
    }

    public ArrayList<GregorianCalendar> getDay_of_the_week()
    {
        return days_of_the_week;
    }
    public void setDays_of_the_week(ArrayList<GregorianCalendar> days_of_the_week)
    {
        this.days_of_the_week = days_of_the_week;
    }
    public int getFlight_number()
    {
        return flight_number;
    }
    public void setFlight_number(int flight_number)
    {
        this.flight_number = flight_number;
    }

    public RouteClass(String[] name, String[] time,String[] time1,String[]
start,String[] terminal,String[] intermediate, String[] time_of_intermediate,
String[] time1_of_intermediate,int[] number,int
total_number,ArrayList<GregorianCalendar> days,int flight)
    {
        this.station_name = name;
        this.departure_time = time;
        this.arrival_time = time1;
        this.starting_station = start;
        this.terminal_station = terminal;
        this.intermediate_stations = intermediate;
        this.departure_time_of_intermediate_stations = time_of_intermediate;
        this.arrival_time_of_intermediate_stations = time1_of_intermediate;
        this.number_of_free_seats = number;
        total_number_of_seats = total_number;
        this.days_of_the_week = days;
        flight_number = flight;
    }

    public void outputData()
    {
        System.out.println("Список названий станций: ");
        for (int i = 0; i < station_name.length; i++)
        {
            System.out.println(i+1 + ". " + station_name[i]);
        }
        System.out.println("Список времени отправления: ");
        for (int i = 0; i < departure_time.length; i++)
        {
            System.out.println(i+1 + ". " + departure_time[i]);
        }
        System.out.println("Список времени прибытия: ");
        for (int i = 0; i < arrival_time.length; i++)

```

```

    {
        System.out.println(i+1 + ". " + arrival_time[i]);
    }
    System.out.println("Список времени отправления для промежуточных
станций: ");
    for (int i = 0; i < departure_time_of_intermediate_stations.length; i++)
    {
        System.out.println(i+1 + ". " +
departure_time_of_intermediate_stations[i]);
    }
    System.out.println("Список времени прибытия для промежуточных станций:
");
    for (int i = 0; i < arrival_time_of_intermediate_stations.length; i++)
    {
        System.out.println(i+1 + ". " +
arrival_time_of_intermediate_stations[i]);
    }
    System.out.println("Список количества пустых мест: ");
    for (int i = 0; i < number_of_free_seats.length; i++)
    {
        System.out.println(i+1 + ". " + number_of_free_seats[i]);
    }
    System.out.println("Список начальных станций: ");
    for (int i = 0; i < starting_station.length; i++)
    {
        System.out.println(i+1 + ". " + starting_station[i]);
    }
    System.out.println("Список конечных станций: ");
    for (int i = 0; i < terminal_station.length; i++)
    {
        System.out.println(i+1 + ". " + terminal_station[i]);
    }
    System.out.println("Список промежуточных станций: ");
    for (int i = 0; i < intermediate_stations.length; i++)
    {
        System.out.println(i+1 + ". " + intermediate_stations[i]);
    }
    System.out.println("Общее количество мест: " + total_number_of_seats);
    System.out.println("Список дней недели: ");
    for (GregorianCalendar date : days_of_the_week)
    {
        System.out.println(date.getTime());
    }
    System.out.println("Номер рейса: " + flight_number);
}
}

```

```

public class Main07
{
    public static void main(String[] args)
    {
        String[] listOfStation_name1 = {"Лозовая-Харьков", "Харьков-Гавриловка",
"Харьков-Изюм"};
        String[] listOfStation_name2 = {"Красноград-Харьков", "Мерефа-Харьков",
"Славянск-Харьков"};
        String[] listOfdeparture_time1 = {"15:24", "15:59", "06:40"};
        String[] listOfdeparture_time2 = {"16:44", "06:58", "06:27"};
        String[] listOfarrival_time1 = {"19:13", "20:37", "10:14"};
        String[] listOfarrival_time2 = {"19:34", "7:45", "10:51"};
        String[] listOfstarting_station1 = {"Лозовая", "Харьков", "Харьков"};
        String[] listOfstarting_station2 = {"Красноград", "Мерефа", "Славянск"};
        String[] listOfterminal_station1 = {"Харьков", "Гавриловка", "Изюм"};

        String[] listOfterminal_station2 = {"Харьков", "Харьков", "Харьков"};
        String[] listOfintermediate_stations1 = {"Панютино", "Мерефа", "Маяк"};
    }
}

```

```

        String[] listOfintermediate_stations2 = {"Власовка", "Покотиловка",
"Гусаровка"};
        String[] listOfdeparture_time_of_intermediate_stations1 = {"15:35",
"16:43", "09:24"};
        String[] listOfdeparture_time_of_intermediate_stations2 = {"17:37",
"07:25", "6:56"};
        String[] listofarrival_time_of_intermediate_stations1 = {"15:36",
"16:54", "09:25"};
        String[] listofarrival_time_of_intermediate_stations2 = {"17:38",
"07:26", "6:57"};
        int[] listOfnumber_of_free_seats1 = { 35, 48, 72};
        int[] listOfnumber_of_free_seats2 = { 65, 37, 86};

        ArrayList<GregorianCalendar> List_days_of_the_week1 = new
ArrayList<GregorianCalendar>();
        GregorianCalendar days_of_the_week1 = new GregorianCalendar(2020, 10,
22);

        days_of_the_week1.set(Calendar.HOUR_OF_DAY, 15);
        days_of_the_week1.set(Calendar.MINUTE, 24);
        days_of_the_week1.set(Calendar.SECOND, 00);
        List_days_of_the_week1.add(days_of_the_week1);
        GregorianCalendar days_of_the_week2 = new GregorianCalendar(2020, 10,
25);

        days_of_the_week2.set(Calendar.HOUR_OF_DAY, 15);
        days_of_the_week2.set(Calendar.MINUTE, 59);
        days_of_the_week2.set(Calendar.SECOND, 00);
        List_days_of_the_week1.add(days_of_the_week2);
        GregorianCalendar days_of_the_week3 = new GregorianCalendar(2020, 10,
30);

        days_of_the_week3.set(Calendar.HOUR_OF_DAY, 6);
        days_of_the_week3.set(Calendar.MINUTE, 40);
        days_of_the_week3.set(Calendar.SECOND, 00);
        List_days_of_the_week1.add(days_of_the_week3);

        RouteClass routeclass1 = new RouteClass(listOfStation_name1,
listOfdeparture_time1, listofarrival_time1, listofstarting_station1,
listofterminal_station1, listOfintermediate_stations1,
listOfdeparture_time_of_intermediate_stations1,
listofarrival_time_of_intermediate_stations1, listOfnumber_of_free_seats1, 150,
List_days_of_the_week1, 1);

        ArrayList<GregorianCalendar> List_days_of_the_week2 = new
ArrayList<GregorianCalendar>();
        GregorianCalendar days_of_the_week4 = new GregorianCalendar(2020, 10,
22);

        days_of_the_week4.set(Calendar.HOUR_OF_DAY, 16);
        days_of_the_week4.set(Calendar.MINUTE, 44);
        days_of_the_week4.set(Calendar.SECOND, 00);
        List_days_of_the_week2.add(days_of_the_week4);
        GregorianCalendar days_of_the_week5 = new GregorianCalendar(2020, 10,
25);

        days_of_the_week5.set(Calendar.HOUR_OF_DAY, 6);
        days_of_the_week5.set(Calendar.MINUTE, 58);
        days_of_the_week5.set(Calendar.SECOND, 00);
        List_days_of_the_week2.add(days_of_the_week5);
        GregorianCalendar days_of_the_week6 = new GregorianCalendar(2020, 10,
30);

        days_of_the_week6.set(Calendar.HOUR_OF_DAY, 6);
        days_of_the_week6.set(Calendar.MINUTE, 27);
        days_of_the_week6.set(Calendar.SECOND, 00);
        List_days_of_the_week2.add(days_of_the_week6);

        RouteClass routeclass2 = new RouteClass(listOfStation_name2,
listOfdeparture_time2, listofarrival_time2, listofstarting_station2,
listofterminal_station2, listOfintermediate_stations2,
listOfdeparture_time_of_intermediate_stations2,

```

```

listOfarrival_time_of_intermediate_stations2, listOfnumber_of_free_seats2, 200,
List_days_of_the_week2,2);

RouteClassList Route_list = new RouteClassList();
Route_list.route[0] = new RouteClass(listOfStation_name1,
listOfdeparture_time1, listOfarrival_time1, listOfstarting_station1,
listOfterminal_station1, listOfintermediate_stations1,
listOfdeparture_time_of_intermediate_stations1,
listOfarrival_time_of_intermediate_stations1, listOfnumber_of_free_seats1, 150,
List_days_of_the_week1, 1);
Route_list.route[1] = new RouteClass(listOfStation_name2,
listOfdeparture_time2, listOfarrival_time2, listOfstarting_station2,
listOfterminal_station2, listOfintermediate_stations2,
listOfdeparture_time_of_intermediate_stations2,
listOfarrival_time_of_intermediate_stations2, listOfnumber_of_free_seats2, 200,
List_days_of_the_week2, 2);
for (RouteClass routeclass : Route_list.route)
{
    routeclass.outputData();
    System.out.println();
}
}
}

```

Результат роботи програми:

Список названий станций:

1. Лозовая-Харьков
2. Харьков-Гавриловка
3. Харьков-Изюм

Список времени отправления:

1. 15:24
2. 15:59
3. 06:40

Список времени прибытия:

1. 19:13
2. 20:37
3. 10:14

Список времени отправления для промежуточных станций:

1. 15:35
2. 16:43
3. 09:24

Список времени прибытия для промежуточных станций:

1. 15:36
2. 16:54
3. 09:25

Список количества пустых мест:

1. 35
2. 48
3. 72

Список начальных станций:

1. Лозовая
2. Харьков
3. Харьков

Список конечных станций:

1. Харьков
2. Гавриловка
3. Изюм

Список промежуточных станций:

1. Панютино
2. Мерефа
3. Маяк

Общее количество мест: 150

Список дней недели:

Sun Nov 22 15:24:00 EET 2020

Wed Nov 25 15:59:00 EET 2020

Mon Nov 30 06:40:00 EET 2020

Номер рейса: 1

```
Список названий станций:
1. Красноград-Харьков
2. Мерефа-Харьков
3. Славянск-Харьков
Список времени отправления:
1. 16:44
2. 06:58
3. 06:27
Список времени прибытия:
1. 19:34
2. 7:45
3. 10:51
Список времени отправления для промежуточных станций:
1. 17:37
2. 07:25
3. 6:56
Список времени прибытия для промежуточных станций:
1. 17:38
2. 07:26
3. 6:57
Список количества пустых мест:
1. 65
2. 37
3. 86
Список начальных станций:
1. Красноград
2. Мерефа
3. Славянск
Список конечных станций:
1. Харьков
2. Харьков
3. Харьков
Список промежуточных станций:
1. Власовка
2. Покотиловка
3. Гусаровка
Общее количество мест: 200
Список дней недели:
Sun Nov 22 16:44:00 EET 2020
Wed Nov 25 06:58:00 EET 2020
Mon Nov 30 06:27:00 EET 2020
Номер рейса: 2
```

Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з об'єктно-орієнтованою декомпозицією.

Програма протестована, виконується без помилок.