

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Об'єктно-орієнтоване програмування»

*Звіт з лабораторної роботи №15*  
*Тема: «Колекції в Java»*

Виконав:  
ст. гр. 1.КІТ102.8а  
Міщенко Д.С.

Перевірив:  
Пугачов Р.В.

Харків – 2019

# Лабораторна робота №15

## Колекції в Java

### *Мета*

- Ознайомлення з бібліотекою колекцій *Java SE*.
- Використання колекцій для розміщення об'єктів розроблених класів.

## 1 ВИМОГИ

### 1.1 Розробник

1. Міщенко Дмитро Сергійович
2. КІТ-118В
3. Варіант 13

### 1.2 Загальне завдання

#### *Вимоги*

1. Розробити консольну програму для реалізації завдання обробки даних згідно прикладної області.
2. Для розміщення та обробки даних використовувати контейнери (колекції) і алгоритми з Java Collections Framework.
3. Забезпечити обробку колекції об'єктів: додавання, видалення, пошук, сортування згідно розділу Прикладні задачі л.р. №10.
4. Передбачити можливість довготривалого зберігання даних: 1) за допомогою стандартної серіалізації; 2) не використовуючи протокол серіалізації.
5. Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах за результатом обробки параметрів командного рядка.

#### *Прикладні задачі*

13. Готель. Замовлення номера: паспортні дані; дати поселення та виселення; номер (клас, число місць); причина поселення (кількість не обмежена).

## 2 ОПИС ПРОГРАМИ

### 2.1 Засоби ООП

Поліморфізм,інкапсуляція

### 2.2 Ієрархія та структура даних

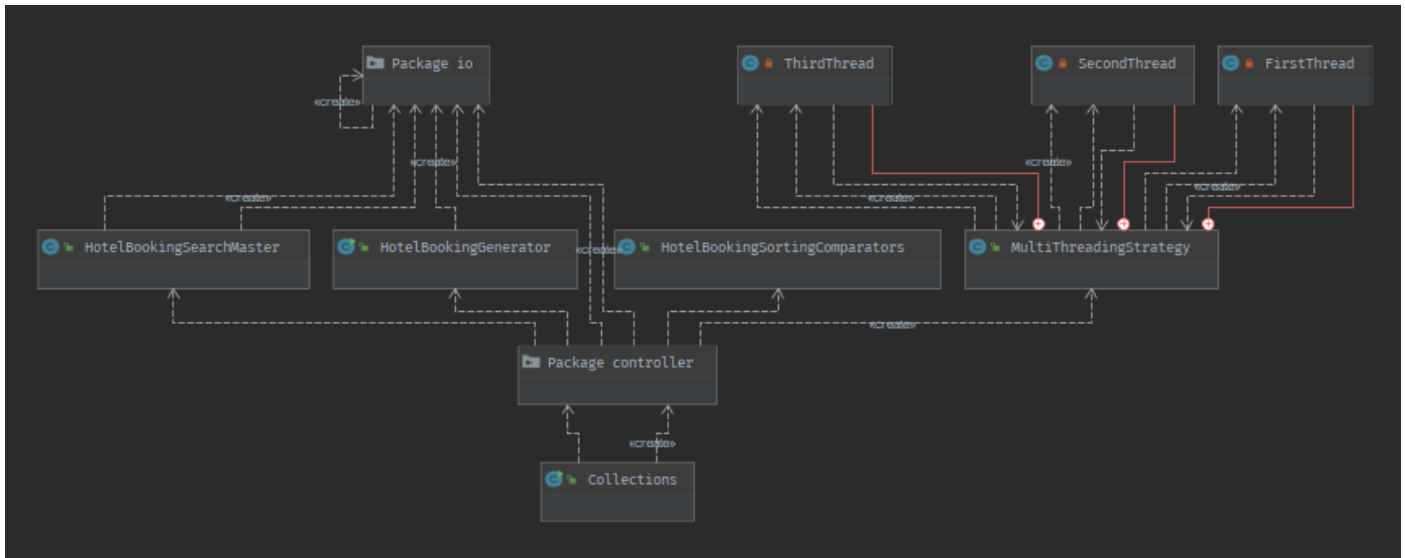


Рисунок 1 – Ієрархія класів

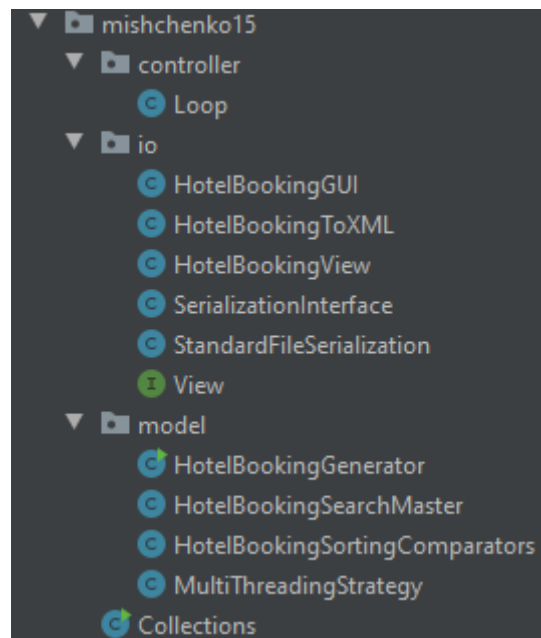


Рисунок 2 – Структура пакету

## 2.2 Важливі фрагменти програми

```
public class HotelBookingSortingComparators {

    public static void sorting(ArrayList<HotelBooking> input, Scanner in, PrintStream output) {
        boolean exit = false;
        while (!exit) {
            new HotelBookingGUI(output).sorting();
            int choose = HotelBookingInput.inInt(in);
            switch (choose) {
                case 1:
                    input.sort(Comparator.comparingInt(o -> Period.between(o.getSettlementDate(), o.getEvictionDate()).getDays()));
                    break;
                case 2:
                    input.sort(Comparator.comparing(p -> p.getHotelRoom().getClassOfRoom().getNumberClass()));
                    break;
                case 3:
                    input.sort(Comparator.comparingInt(p -> p.getHotelRoom().getCountOfPlaces()));
                    break;
                default:
                    exit = true;
                    break;
            }
        }
    }
}
```

Рисунок 3 – Сортування колекції за різними критеріями

```
public ArrayList<HotelBooking> read(String path) {
    try {
        boolean test = true;
        XMLDecoder decoder = new XMLDecoder(
            new BufferedInputStream(
                new FileInputStream(path)));
        ArrayList<HotelBooking> list = new ArrayList<>();
        while (test) {
            try {
                HotelBooking object = (HotelBooking) decoder.readObject();
                object.setSettlementReasons((ArrayList<String>) decoder.readObject());
                if (RegexCheck.checkObject(object)) list.add(object);
            } catch (Exception e) {
                test = false;
            }
        }
        return list;
    } catch (FileNotFoundException e) {
        logger.error("File not found");
    }
    return null;
}
```

Рисунок 4 –Десеріалізація за допомогою XML

```

public void write(List<HotelBooking> forwardLinkedList, String path) {
    try {
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(new FileOutputStream(path));
        objectOutputStream.writeObject(forwardLinkedList);
        objectOutputStream.close();
        logger.info("Write successful");
    } catch (IOException e) {
        logger.error("Error while writing {}", e.getMessage());
    }
}
}

```

**Рисунок 5** – Серіалізація через стандартний протокол

### 3 ВАРІАНТИ ВИКОРИСТАННЯ

```

18:06:36.115 [main] INFO  u.k.oop.mishchenko15.controller.Loop - App started
01. Add new client
02. Delete client
03. Clear list
04. Show list
05. Save
06. Load
07. Get list as string
08. Get list as array
09. Sort
10. Search
11. Thread demo
12. Parallel vs consequential demo
13. Generate random values
00. Exit
Choose option

```

**Рисунок 6** – Головне меню програми

```

Choose option
10
Choose searching criteria
1. By surname
2. By age
3. By class of room
4. By settlement reason

```

**Рисунок 7** – Пошук елементів за критерієм

```
Enter age range
20
45

-----

Passport ID: 811288328
Surname: Johnson
Name: Michael
Patronymic: Aarts
Day of birth: 01.06.1983
Date of settlement: 2020-03-16
Date of eviction: 2020-01-27
Number of room: 7465
Class of room: default
Count of places: 5
Reasons of settlement:
[business trip]

-----

-----

Passport ID: 103055863
Surname: Smith
Name: Jack
Patronymic: Fernandes
Day of birth: 15.11.1992
Date of settlement: 2020-01-27
Date of eviction: 2020-02-18
Number of room: 6772
Class of room: default
Count of places: 5
Reasons of settlement:
[hiding]

-----
```

**Рисунок 8** – Результат пошуку

## **ВИСНОВКИ**

В результаті виконання лабораторної роботи відбулось ознайомлення з бібліотекою колекцій *Java SE* та були використані колекції для розміщення об'єктів розроблених класів.