

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Об'єктно-орієнтоване програмування»

Звіт з лабораторної роботи №10
Тема: «Обробка параметризованих контейнерів»

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

Перевірив:
Пугачов Р.В.

Харків – 2019

Лабораторна робота №10

Обробка параметризованих контейнерів

Мета

Розширення функціональності параметризованих класів.

1 ВИМОГИ

1.1 Розробник

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

1.2 Загальне завдання

Вимоги

Використовуючи програму рішення завдання лабораторної роботи №9:

1. Розробити параметризовані методи (Generic Methods) для обробки колекцій об'єктів згідно прикладної задачі.
2. Продемонструвати розроблену функціональність (створення, управління та обробку власних контейнерів) в діалоговому та автоматичному режимах.
 - Автоматичний режим виконання програми задається параметром командного рядка **-auto**. Наприклад, java ClassName -auto.
 - В автоматичному режимі діалог з користувачем відсутній, необхідні данні генеруються, або зчитуються з файлу.
3. Забороняється використання алгоритмів з Java Collections Framework.

Прикладні задачі

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

Готель. Сортування за тривалістю проживання, за класом номера, за кількістю займаних місць.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Поліморфізм,інкапсуляція

2.2 Ієрархія та структура даних

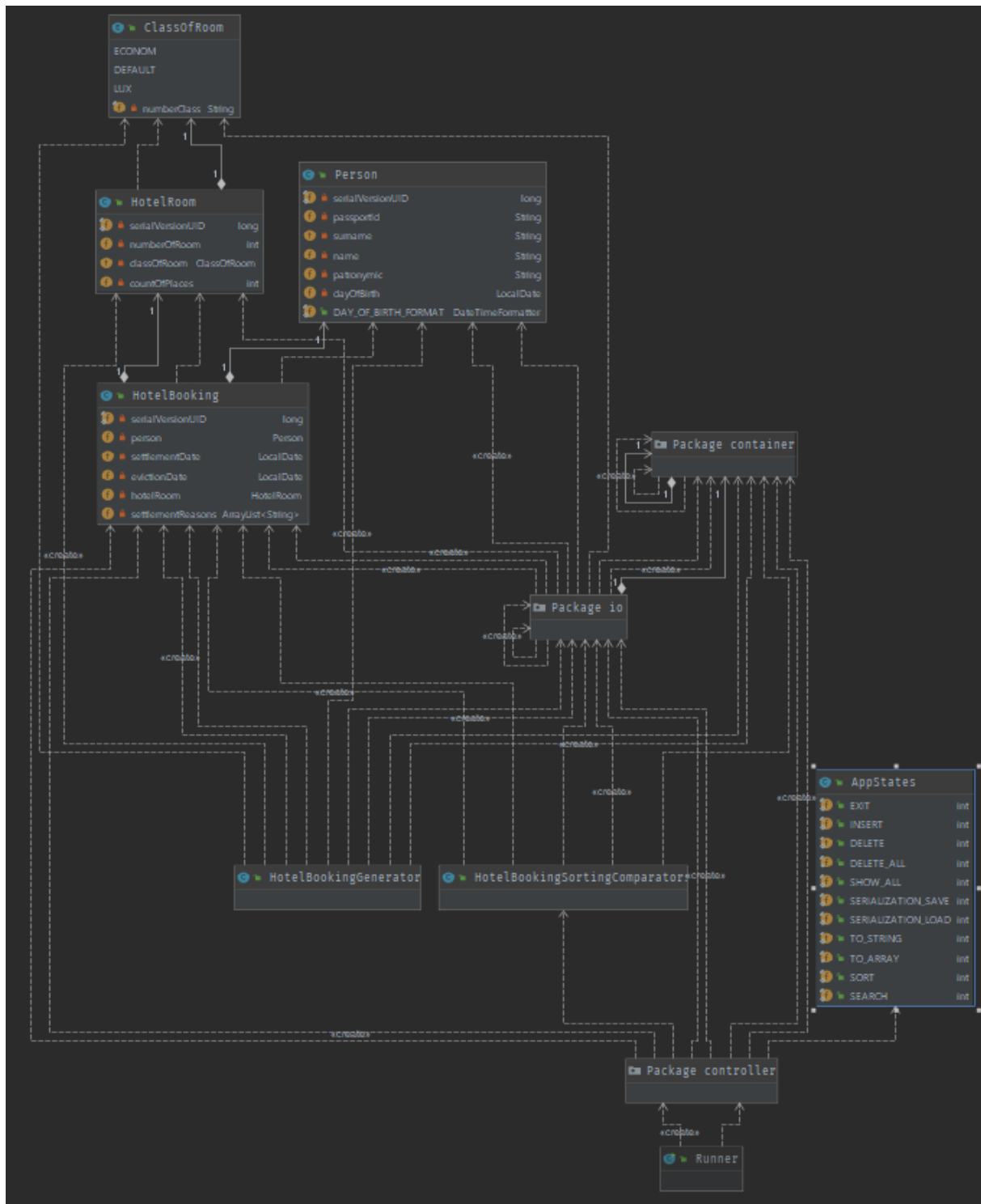


Рисунок 1 – Ієрархія класів

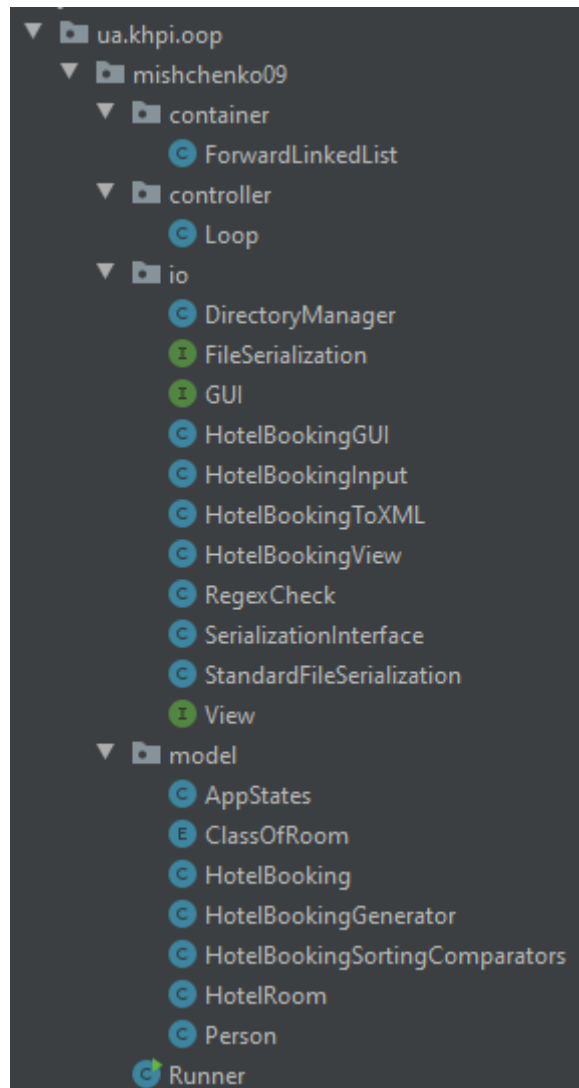


Рисунок 2 – Структура пакету

2.2 Важливі фрагменти програми

Створено метод для визначення режиму роботи програми (рис. 1)

```
private boolean checkAutoModeArg(String[] args) {

    for (String s1 : args) {
        if (s1.equals("-auto")) {
            return true;
        }
    }
    return false;
}
```

Рисунок 3 – Метод **checkAutoModeArg**

У разі присутності параметру `-auto` при запуску програми, виконується автозавантаження даних (рис. 2)

```
private ForwardLinkedList<HotelBooking> autoLoad() {
    return new HotelBookingToXML().read(Loop.class.getResource("Generated.xml").getPath());
}
```

Рисунок 4 – Автозавантаження даних

Параметризований метод сортування (рис. 3)

```
public void sort(Comparator<? super T> c, T[] arr) {
    Arrays.sort(arr, c);
    ListIterator<T> i = this.listIterator();
    for (Object e : arr) {
        i.next();
        i.set((T) e);
    }
}
```

Рисунок 5 – Метод sort

```
public static void sorting(ForwardLinkedList<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(p → Period.between(p.getSettlementDate(), p.getEvictionDate()).getDays()),
                    input.toArray(new HotelBooking[input.size()]));
                break;
            case 2:
                input.sort(Comparator.comparing(p → p.getHotelRoom().getClassOfRoom().getNumberClass()), input.toArray(new HotelBooking[input.size()]));
                break;
            case 3:
                input.sort(Comparator.comparingInt(p → p.getHotelRoom().getCountOfPlaces()), input.toArray(new HotelBooking[input.size()]));
                break;
            default:
                exit = true;
                break;
        }
    }
}
```

Рисунок 6 – Метод сортування списку

```

<?xml version="1.0" encoding="UTF-8"?>
- <java class="java.beans.XMLDecoder" version="12.0.2">
  - <object class="ua.khpi.oop.mishchenko09.model.HotelBooking">
    - <void property="evictionDate">
      - <object class="java.time.LocalDate" method="parse">
        <string>2020-03-19</string>
      </object>
    </void>
    - <void property="hotelRoom">
      - <object class="ua.khpi.oop.mishchenko09.model.HotelRoom">
        - <void property="classOfRoom">
          - <object class="java.lang.Enum" method="valueOf" id="ClassOfRoom0">
            <class>ua.khpi.oop.mishchenko09.model.ClassOfRoom</class>
            <string>ECONOM</string>
          </object>
        </void>
        - <void property="countOfPlaces">
          <int>4</int>
        </void>
        - <void property="numberOfRoom">
          <int>7225</int>
        </void>
      </object>
    </void>
    - <void property="person">
      - <object class="ua.khpi.oop.mishchenko09.model.Person">
        - <void property="dayOfBirth">
          - <object class="java.time.LocalDate" method="parse">
            <string>1995-02-12</string>
          </object>
        </void>
        - <void property="name">
          <string>James</string>
        </void>
      </object>
    </void>
  </object>
</java>

```

Рисунок 7 – Файл зі збереженими даними у форматі XML

3 ВАРІАНТИ ВИКОРИСТАННЯ

```
00:39:02.122 [main] INFO u.k.oop.mishchenko09.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 by reason
00. Exit
Choose option
```

Рисунок 8 – Головне меню програми

```
Choose option
9
Choose sorting criteria
1. By living duration
2. By class of room
3. By count of places
```

Рисунок 9 – Діалог сортування списку за різними критеріями

ВИСНОВКИ

В результаті виконання лабораторної роботи була розширена функціональність параметризованих класів за допомогою методів сортування, а також добавлений режим запуску програми –auto, при якому дані автоматично зчитуються із запрограмованого джерела.