**Лабораторна робота №6**

**Серіалізація/десеріалізація об'єктів. Бібліотека класів користувача**

**Мета:** Тривале зберігання та відновлення стану об'єктів. Ознайомлення з принципами серіалізації/десеріалізації об'єктів. Використання бібліотек класів користувача.

**1 ВИМОГИ**

1. Реалізувати і продемонструвати тривале зберігання/відновлення раніше розробленого контейнера за допомогою серіалізації/десеріалізації.
2. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення задачі л.р. №3 з іншим студентом (визначає викладач).
3. Продемонструвати послідовну та вибіркову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
4. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
5. Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.
   1. **Розробник**

* Дем’яненко Дмитро Андрійович
* Группа: КІТ-119д
* Варіант: 7

1. **ОПИС ПРОГРАМИ**
   1. **Було використано наступні засоби:**

File file = new File("C:\\Users\\d1mad\\java\\lab6.data");

oos = new ObjectOutputStream(fos);

– серіалізація;

File file = new File("C:\\Users\\d1mad\\java\\lab6.data");

FileInputStream fis = new FileInputStream(file); – десеріалізація;

* 1. **Ієрархія та структура класів**

Було створено 5 класів:

* public class Main – містить метод main;
* public class Container – клас, що містить методи для роботи з контейнером;
* public class Iterator – клас, що містить методи для роботи з ітератором.
* public class CaseMenu –клас,що містить меню
* public class Serializator –клас, що містить методи для серіалізації та для десеріалізації
  1. **Важливі фрагменти програми**

**package ua.khpi.oop.demianenko06;**

**import java.io.File;**

**import java.io.FileInputStream;**

**import java.io.FileNotFoundException;**

**import java.io.FileOutputStream;**

**import java.io.IOException;**

**import java.io.InvalidObjectException;**

**import java.io.ObjectInputStream;**

**import java.io.ObjectOutputStream;**

**public class Serializator {**

**public boolean serialization(Container s) {**

**boolean flag = false;**

**File file = new File("C:\\Users\\d1mad\\java\\lab6.data");**

**ObjectOutputStream oos = null;**

**try {**

**FileOutputStream fos = new FileOutputStream(file);**

**if (fos!=null) {**

**oos = new ObjectOutputStream(fos);**

**oos.writeObject(s);**

**flag = true;**

**}**

**} catch (FileNotFoundException e){**

**e.printStackTrace();**

**} catch (IOException e) {**

**e.printStackTrace();**

**} finally {**

**if(oos!=null) {**

**try {**

**oos.close();**

**}catch (IOException e) {**

**e.printStackTrace();**

**}**

**}**

**}**

**return flag;**

**}**

**public Container deserialization() throws InvalidObjectException{**

**File file = new File("C:\\Users\\d1mad\\java\\lab6.data");**

**ObjectInputStream ois = null;**

**try {**

**FileInputStream fis = new FileInputStream(file);**

**if (fis!=null) {**

**ois = new ObjectInputStream(fis);**

**Container s = (Container) ois.readObject();**

**return s;**

**}**

**} catch (FileNotFoundException e) {**

**e.printStackTrace();**

**} catch (ClassNotFoundException e) {**

**e.printStackTrace();**

**} catch (IOException e) {**

**e.printStackTrace();**

**} finally {**

**try {**

**ois.close();**

**}catch (IOException e) {**

**e.printStackTrace();**

**}**

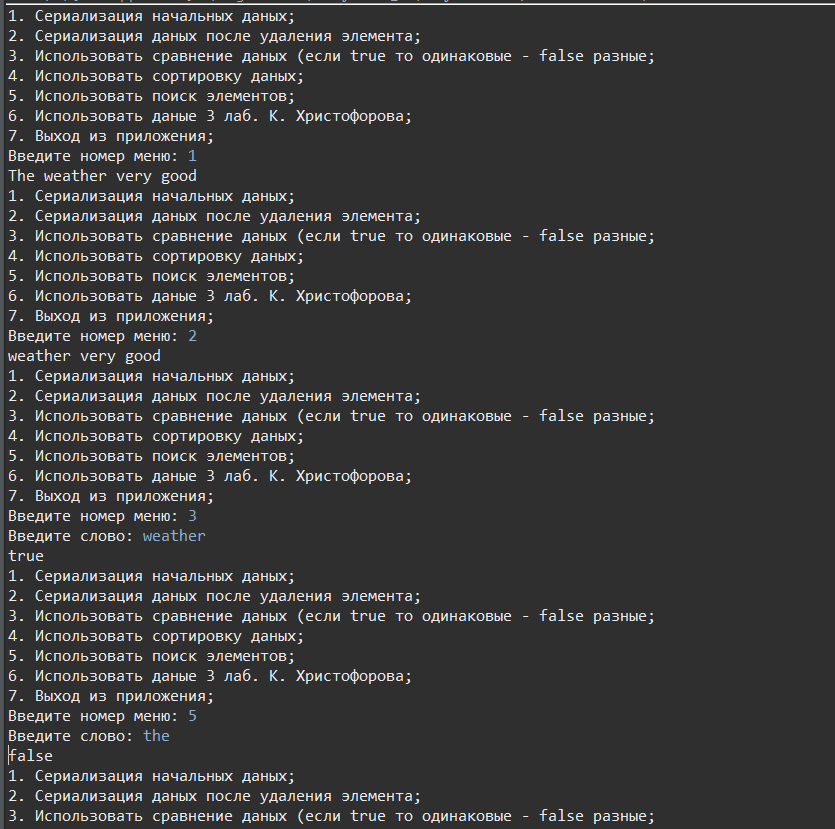
**}**

**throw new InvalidObjectException("Object fail");**

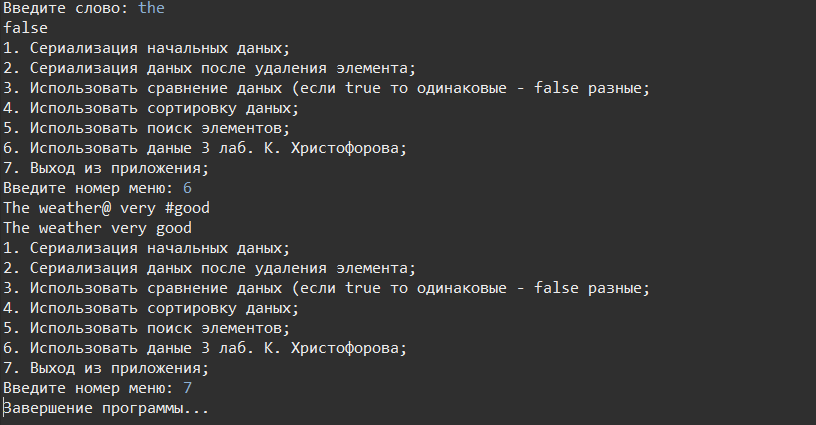
**}**

**}**

**Результати роботи програми**



а)



б)

**Висновок**

Під час виконання лабораторної роботи було набуто навичок роботи з серіалізацією\десеріалізацієюм та з розробки бібліотеки класів користувача у середовищі Eclipse IDE.