Автор: Кравчук В.П., КІТ101.8а

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.

**Вимоги:**

1. Реалізувати і продемонструвати тривале зберігання та відновлення раніше розробленого контейнера за допомогою серіалізації та 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.

ЗАВДАННЯ ДО РОБОТИ

Реалізувати і продемонструвати тривале зберігання / відновлення раніше розробленого контейнера за допомогою серіалізації / десеріалізації.

**ТЕКСТ ПРОГРАМИ**

File list.java:

package laba6;

import java.io.Serializable;

import java.util.ArrayList;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectOutputStream;

public class list implements Serializable {

ArrayList <String> newOne = new ArrayList<String>();

}

File menu.java:

**package** laba6;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Iterator;

**import** java.util.Scanner;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Iterator;

**import** java.io.Serializable;

**import** java.io.FileOutputStream;

**import** java.io.FileInputStream;

**import** java.io.ObjectInputStream;

**import** java.io.IOException;

**import** java.io.ObjectOutputStream;

**public** **class** menu

{

**public** **static** **void** main(String[] args) **throws** IOException, ClassNotFoundException

{

**int** in=0;

Scanner cin = **new** Scanner(System.***in***);

**int** size = 0;

ArrayList <String> arraylist = **new** ArrayList<String>();

String string = **new** String();

Scanner inn=**new** Scanner(System.***in***);

**do**

{

System.***out***.println("1. Заполнить контейнер");

System.***out***.println("2. Вывести на экран содержимое контейнера");

System.***out***.println("3. Функция toString()");

System.***out***.println("4. Добавить элемент в конец");

System.***out***.println("5. Очистить контейнер");

System.***out***.println("6. Удалить элемент");

System.***out***.println("7. Функция toArray()");

System.***out***.println("8. Узнать размер");

System.***out***.println("9. Contains(true, если в контейнере имеется такой элемент");

System.***out***.println("10. ContainAll(true, если в контейнере имеются все такие элементы");

System.***out***.println("11. Сериализация");

System.***out***.println("12. Десериализация");

System.***out***.println("13. Выход");

System.***out***.println("В зависимости от того, что вы хотите, введите число:");

in = cin.nextInt();

**switch** (in) {

**case** 1:

System.***out***.println("Количество элементов в контейнере: ");

size=cin.nextInt();

**for**(**int** i=0;i<size;i++)

{

string=inn.nextLine();

arraylist.add(string);

}

**break**;

**case** 2:

Iterator<String> iterator = arraylist.iterator();

**while**(iterator.hasNext())

{

System.***out***.println(iterator.next());

}

**break**;

**case** 3:

System.***out***.println(arraylist.toString());

**break**;

**case** 4:

Scanner inn1 = **new** Scanner(System.***in***);

System.***out***.println("Введите новый элемент списка");

string=inn1.nextLine();

arraylist.add(string);

**break**;

**case** 5:

arraylist.clear();

System.***out***.println("Список был очищен.");

**break**;

**case** 6:

string=inn.nextLine();

arraylist.remove(string);

**break**;

**case** 7:

String[] array = {}; // конвертируем ArrayList в массив

array = arraylist.toArray(**new** String[arraylist.size()]);

**for**(String itter : array)

{

System.***out***.println(itter);

}

**break**;

**case** 8:

System.***out***.println("Размер - " + arraylist.size());

**break**;

**case** 9:

string=inn.nextLine();

System.***out***.println(arraylist.contains(string));

**break**;

**case** 10:

ArrayList <String> arrlst = **new** ArrayList <String>();

System.***out***.println("Введите количество элементов: ");

size=cin.nextInt();

**for**(**int** i=0;i<size;i++)

{

string=inn.nextLine();

arrlst.add(string);

}

System.***out***.println(arraylist.containsAll(arrlst));

**break**;

**case** 11:

list arrlst1 = **new** list();

arrlst1.newOne = arraylist;

FileOutputStream fileOutputStream = **new** FileOutputStream("Serialized\_file");

ObjectOutputStream objectOutputStream = **new** ObjectOutputStream(fileOutputStream);

objectOutputStream.writeObject(arrlst1);

objectOutputStream.close();

**break**;

**case** 12:

{

FileInputStream fileInputStream = **new** FileInputStream("Serialized\_file");

ObjectInputStream objectInputStream = **new** ObjectInputStream(fileInputStream);

list newArrlst = (list) objectInputStream.readObject();

objectInputStream.close();

System.***out***.println(newArrlst.newOne);

}

**default**: {

System.***out***.println("До встречи.");

}

}

} **while** (in != 13);

}

}

**ВАРІАНТИ ВИКОРИСТАННЯ**

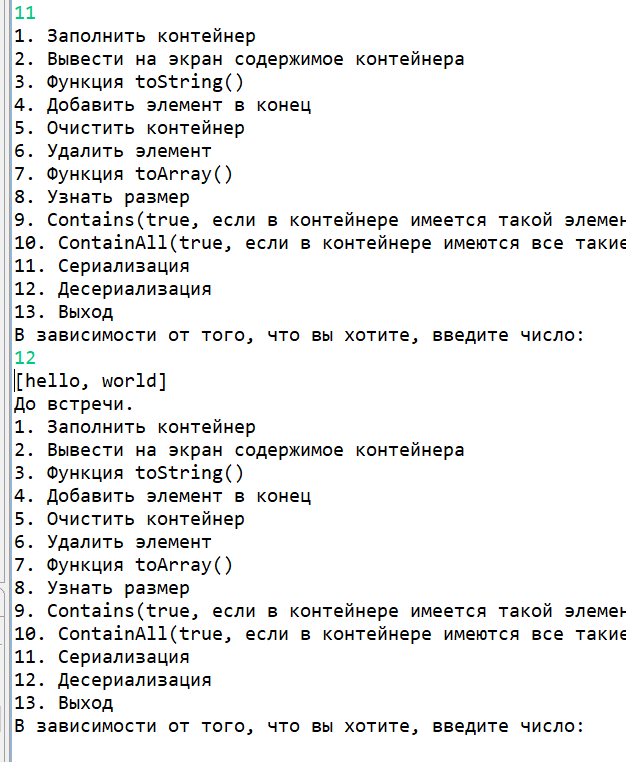
****

Рис. 6.1 – Результат роботи програми

**ВИСНОВКИ**

При виконанні лабораторної роботи набуто практичних навичок щодо реалізації тривалого зберігання та відновлення даних за допомогою серіалізації та десеріалізації. Завдання виконано. Програма працює успішно!