**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:**

FileOutputStream fos = new FileOutputStream("Serial.ser"); ObjectOutputStream oos = new ObjectOutputStream(fos); oos.writeObject(container); oos.close() – серіалізація;

FileInputStream fis = new FileInputStream("Serial.ser"); ObjectInputStream ois = new ObjectInputStream(fis); MyContainer temp = (MyContainer) ois.readObject(); ois.close() – десеріалізація;

* 1. **Ієрархія та структура класів**

Було створено 3 класи:

* public class Main – містить метод main;
* public class Container – клас, що містить методи для роботи з контейнером;
* public class MyIterator – клас, що містить методи для роботи з ітератором.

Також було використано мій utility клас, що був розроблений в лабораторній роботі №3.

* 1. **Важливі фрагменти програми**

package ua.khpi.oop.demianenko06;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.Iterator;

import java.util.Scanner;

import ua.khpi.oop.demianenko03.UtilityClass;

import ua.khpi.oop.tutomirov03.helper;

public class Main

{

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

{

Container container = new Container("The weather is very nice today.",

"I'm not going to school today.",

"Today is Saturday, which means I'm going to a picnic with my friends.",

"I hope the day goes well.");

boolean endCheck = true;

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

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

while (endCheck)

{

System.***out***.println("Menu:");

System.***out***.println("1. Enter new data");

System.***out***.println("2. Show data");

System.***out***.println("3. Add element in container");

System.***out***.println("4. Delete element from container");

System.***out***.println("5. Clear container");

System.***out***.println("6. Find element in container");

System.***out***.println("7. Sort container by length");

System.***out***.println("8. Sort container by alphabet");

System.***out***.println("9. Compare elements in container");

System.***out***.println("10. Remove all words of a given length that begin with a consonant letter.");

System.***out***.println("11. Tutomirov class");

System.***out***.println("12. Serialize container");

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

System.***out***.println("14. Exit");

System.***out***.println("Enter your option:");

int option = inInt.nextInt();

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

switch (option)

{

case 1:

container.clear();

System.***out***.println("Enter amount of sentences");

int size2 = inInt.nextInt();

String cont[] = new String[size2];

System.***out***.println("Enter new container:");

for (int i = 0; i < size2; i++)

cont[i] = inStr.nextLine();

container = new Container(cont);

System.***out***.println("\nNew container:");

container.printContainer();

break;

case 2:

if(container.size() > 0)

container.printContainer();

else

System.***out***.println("Empty container");

break;

case 3:

System.***out***.println("Enter data to add:");

container.add(inStr.nextLine());

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

container.printContainer();

break;

case 4:

System.***out***.println("Enter element to delete:");

boolean delete = container.remove(inStr.nextLine());

if (delete)

{

System.***out***.println("\nContainer:");

container.printContainer();

}

else

System.***out***.println("\nThere is no such element in container");

break;

case 5:

container.clear();

System.***out***.println("Container cleared\n");

break;

case 6:

System.***out***.println("Enter element to find:");

int pos = container.findElement(inStr.nextLine());

if (pos != -1)

System.***out***.println("Position: " + pos + "\n");

else

System.***out***.println("There is no such element\n");

break;

case 7:

System.***out***.println("1. Ascending\n2. Descending");

int sortLen = inInt.nextInt();

container.sortLength(sortLen);

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

container.printContainer();

break;

case 8:

System.***out***.println("1. Ascending\n2. Descending");

int sortAlph = inInt.nextInt();

container.sortAlphabet(sortAlph);

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

container.printContainer();

break;

case 9:

System.***out***.println("Enter positions of elements (from 1 to " + container.size() + "):");

int pos1 = inInt.nextInt();

int pos2 = inInt.nextInt();

int comp = container.compareElements(pos1,pos2);

if (comp == 1)

System.***out***.println("Elements on positions " + pos1 + " and " + pos2 + " are equal\n");

else if (comp == 0)

System.***out***.println("Elements on positions " + pos1 + " and " + pos2 + " are NOT equal\n");

else

System.***out***.println("Wrong position\n");

break;

case 10:

System.***out***.println("Enter count of letters:");

int count = inInt.nextInt();

UtilityClass.*findAndDelete*(container.toString(), count);

System.***out***.println("\n");

break;

case 11:

helper.*countVowelsAndConsonants*(container.toString());

break;

case 12:

FileOutputStream fos = new FileOutputStream("Serial.ser");

ObjectOutputStream oos = new ObjectOutputStream(fos);

oos.writeObject(container);

oos.close();

System.***out***.println("Serialization is complete\n");

break;

case 13:

FileInputStream fis = new FileInputStream("Serial.ser");

ObjectInputStream ois = new ObjectInputStream(fis);

Container temp = (Container) ois.readObject();

ois.close();

System.***out***.println("Deserialization is complete\n");

temp.printContainer();

break;

case 14:

endCheck = false;

container.clear();

inInt.close();

inStr.close();

break;

default:

System.***out***.println("Wrong command\n");

break;

}

}

System.***out***.println("End");

}

}

public Container(String... str)

{

if (str.length != 0)

{

size = str.length;

arrayStr = new String[size];

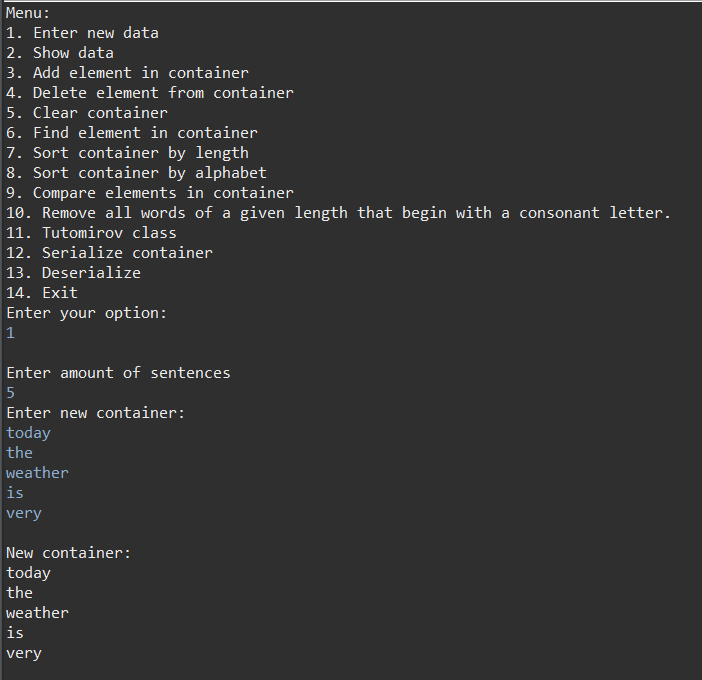
for (int i = 0; i < size; i++)

arrayStr[i] = str[i];

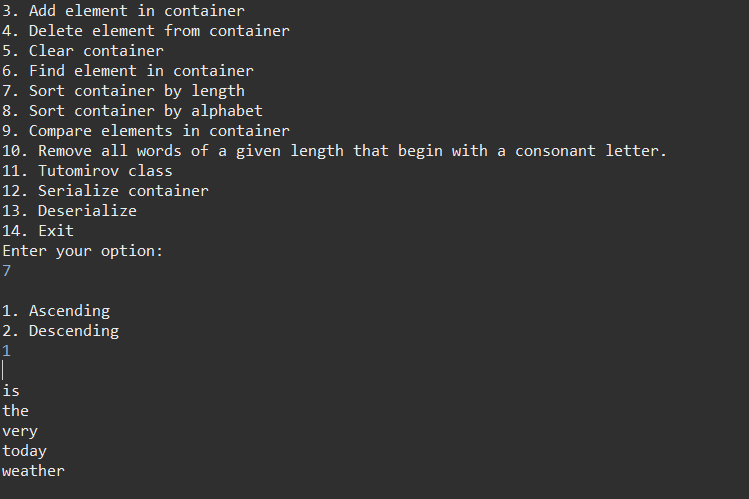
}

}

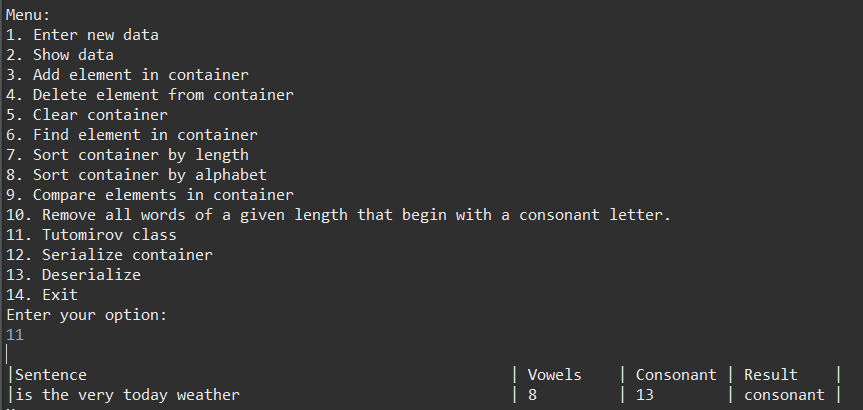
1. **Результати роботи програми**



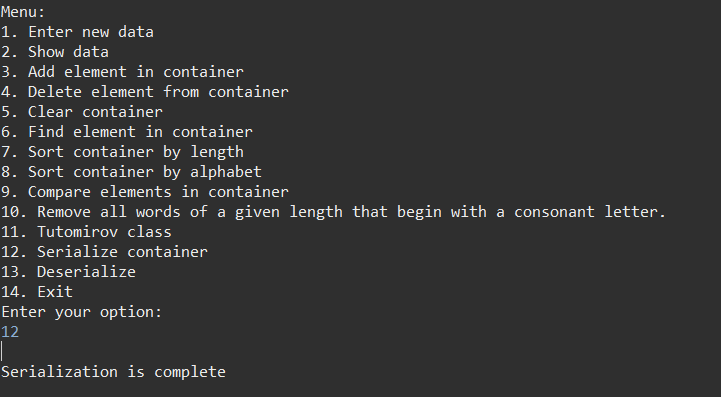
а)



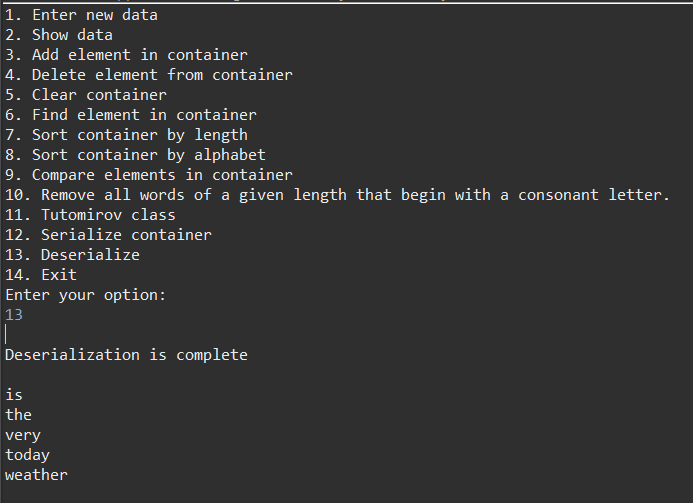
б)



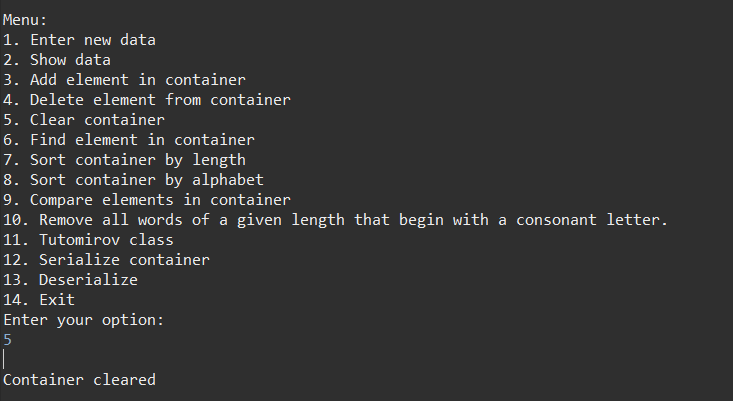
в)

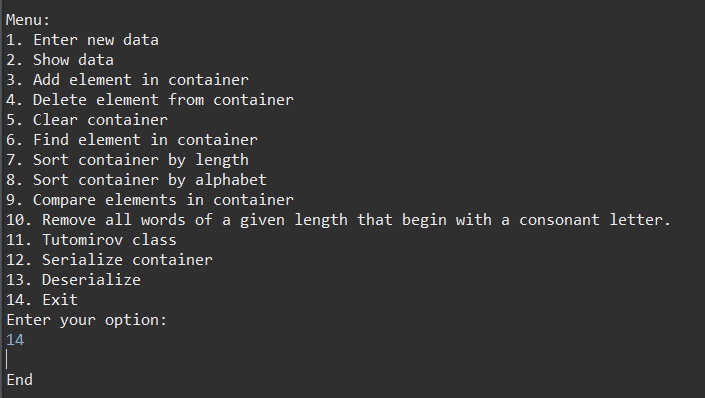


г)



д)



 е)

ж)

Рисунок 6.1 – Результат роботи програми у середовищі Eclipse

**Висновок**

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