Національний технічний університет

«Харківський політехнічний інститут»

**Лабораторна робота № 6**

з дисципліни «ООП»

Виконав студент групи КІТ - 320

Радєвич Владислав Романович

Перевірив викладач

Жилин Володимир Анатолійович

Харків 2021

**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. **Розробник**

Інформація про розробника:

Радєвич Владислав Романович;

студент групи КІТ – 320;

номер варіанту.

* 1. **Загальне завдання**

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

Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення задачі л.р. №3 з іншим студентом (визначає викладач).

Продемонструвати послідовну та вибіркову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.

Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.

Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

1. ОПИС ПРОГРАМИ
   1. **Засоби ООП**

Програма призначена для тривалого зберігання/відновлення раніше розробленого контейнера за допомогою серіалізації/десеріалізації.

* 1. **Ієрархія та структура класів**

public class Main – головний клас

public class Helper – клас для розрахунку сторки іншого користувача

class Text – власний клас для розрахунку сторки

public class Collection implements Iterable<String>, Serializab – клас контейнер

public Iterator<String> iterator() – клас ітератор

* 1. **Важливі фрагменти програми**

Файл Main.java

package ua.khpi.oop.radeivych06;

import java.io.IOException;

import java.util.Scanner;

public class Main {

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

Menu();

}

public static void Menu() {

boolean status = true;

Scanner scan = new Scanner(System.in);

Collection collection = new Collection();

Collection temp = new Collection();

while(status){

System.out.println("1. –Т–≤–µ—Б—В–Є –і–∞–љ—Ц\n"

+ "2. –Я–Њ–Ї–∞–Ј–∞—В–Є –і–∞–љ—Ц\n"

+ "3. –Т–Є–Ї–∞–љ–∞—В–Є —А–Њ–Ј—А–∞—Е—Г–љ–Ї–Є\n"

+ "4. –Т–Є–Ї–Њ–љ–∞—В–Є –њ—А–Њ–≥—А–∞–Љ—Г —Ц–љ—И–Њ–≥–Њ —Б—В—Г–і–µ–љ—В–∞\n"

+ "5. –Ч–±–µ—А–µ–≥—В–Є –і–∞–љ—Ц —Г —Д–∞–є–ї\n"

+ "6. –Ч–∞–≤–∞–љ—В–∞–ґ–Є—В–Є –і–∞–љ—Ц –Ј —Д–∞–є–ї—Г\n"

+ "7. –°–Њ—А—В–Є—А–Њ–≤–Ї–∞ –Ј–∞ –і–Њ–≤–ґ–Є–љ–Њ—О\n"

+ "8. –Я–Њ—И—Г–Ї –µ–ї–µ–Љ–µ–љ—В–∞ –≤ –Љ–∞—Б–Є–≤—Ц\n"

+ "9. –Я–Њ—А—Ц–≤–љ—П–љ–љ—П –і–∞–љ–Є—Е –Ј —Ц–љ—И–Њ–≥–Њ —Д–∞–є–ї—Г\n"

+ "0. –Ч–∞–Ї–Є–љ—З–Є—В–Є –њ—А–Њ–≥—А–∞–Љ–Љ—Г\n"

+ "–§—Г–љ–Ї—Ж—Ц—П: ");

switch (scan.nextInt()) {

case 1:

System.out.println("–Т–µ–і–Є—В—М —Б–ї–Њ–≤–Њ: ");

scan.nextLine();

collection.add(scan.nextLine());

break;

case 2:

System.out.println(collection.toString());

break;

case 3:

Text arg = new Text();

arg.text = collection.toString();

arg.DeleteNumbers();

arg.AddSpace();

arg.DeleteSpace();

collection.remove(arg.text);

collection.add(arg.outputFinal);

// VowelsOrConsonants.splitTextInSentenses(new StringBuffer(collection.toString()));

break;

case 4:

Helper.start(collection.toString());

break;

case 5:

System.out.println("–Т–µ–і–Є—В—М —Ц–Љ'—П —Д–∞–є–ї—Г: ");

scan.nextLine();

try {

collection.Save(scan.nextLine());

} catch (IOException e) {

e.printStackTrace();

}

break;

case 6:

System.out.println("–Т–µ–і–Є—В—М —Ц–Љ'—П —Д–∞–є–ї—Г: ");

scan.nextLine();

try {

collection.Download(scan.nextLine());

} catch (ClassNotFoundException | IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

break;

case 7:

collection.sort();

break;

case 8:

scan.nextLine();

System.out.println("–Т–≤–µ–і–Є—В—М —Б—В—А–Њ–Ї—Г –і–ї—П –њ–Њ—А—Ц–≤–љ—П–љ–љ—П:");

int index = collection.find(scan.nextLine());

if(index==-1)

System.out.println("–Х–ї–µ–Љ–µ–љ—В –љ–µ –Ј–љ–∞–є–і–µ–љ–Њ");

else

System.out.println("–Ш–љ–і–µ–Ї—Б –µ–ї–µ–Љ–µ–љ—В–∞ –≤ –Љ–∞—Б–Є–≤—Ц: "+index);

break;

case 9:

System.out.println("Enter file name: ");

scan.nextLine();

try {

temp.Download(scan.nextLine());

} catch (ClassNotFoundException | IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

if(collection.equals(temp))

System.out.println("–Ю–і–љ–∞–Ї–Њ–≤—Ц");

else

System.out.println("–Э–µ –Њ–і–љ–∞–Ї–Њ–≤—Ц");

break;

default:

System.out.println("–У–∞—А–љ–Њ–≥–Њ –і–љ—П");

status = false;

break;

}

}

scan.close();

}

}

Файл Helper.java

package ua.khpi.oop.radeivych06;

public class Helper {

private static float mean = 0;

public static void start(String ln) {

if(ln == null) {

throw new NullPointerException("line is null");

}

ln = clottingSpace(new StringBuilder(ln));

Collection line = findArray(ln);

findmean(line);

if(getMean() == 0) {

throw new NullPointerException("Line have`t element or size of line element is zero");

}

printline(line);

}

static private Collection findArray(String ln){

Collection line = new Collection();

StringBuilder strB = new StringBuilder();

for(int i = 0; i < ln.length();i++ ) {

if(ln.charAt(i)==' ') {

line.add(strB.toString());

strB = new StringBuilder();

}

else

strB.append(ln.charAt(i));

}

line.add(strB.toString());

// ArrayList<Integer> IndexSpace = new ArrayList<Integer>();

// int length = ln.length();

// for (int i = 0; i<length;i++) {

// if(ln.charAt(i)==' ')

// IndexSpace.add(i);

// }

// if(IndexSpace.size() == 0) {

// line.add(new String(ln));

// }

// else

// {

// line.add(new String(ln.substring(0, IndexSpace.get(0))));

// IndexSpace.add(length);

// for(int i=1;i<IndexSpace.size();i++) {

//

// if(ln.substring(IndexSpace.get(i-1)+1, IndexSpace.get(i))!="")

// line.add(new String(ln.substring(IndexSpace.get(i-1)+1, IndexSpace.get(i))));

// }

// }

return line;

}

static private void findmean(Collection line) {

float m = 0;

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

m+=line.get(i).length();

setMean(m/line.size());

return;

}

static private String clottingSpace(StringBuilder line) {

for(int i = 0; i< line.length()-1;i++)

{

if(line.charAt(i)==' '&&line.charAt(i+1)==' ') {

line.deleteCharAt(i+1);

i--;

}

}

if(line.charAt(line.length()-1)==' ')

line.deleteCharAt(line.length()-1);

if(line.charAt(0)==' ')

line.deleteCharAt(0);

return line.toString();

}

static private void printline(Collection line) {

System.out.println("Lines are greater than average: ");

for(var n : line) {

if((float)n.length() >= mean) {

System.out.println(n+"\nLine lenght:" + n.length());

}

}

System.out.println("Lines less than average: ");

for(var n : line) {

if((float) n.length() < mean) {

System.out.println(n +"\nLine lenght:" + n.length());

}

}

}

public static float getMean() {

return mean;

}

public static void setMean(float mean) {

Helper.mean = mean;

}

}

class Text {

String text;

String output1;

String output2;

String output3;

String outputFinal;

public void ViewBefore() {

System.out.printf("–С—Г–ї–Њ: \n%s\n---------\n", this.text);

}

public void DeleteNumbers() {

output1 = text;

StringBuffer strArgs = new StringBuffer(output1);

strArgs.ensureCapacity(32);

/\*

\* –Т–Є–і–∞–ї–µ–љ–љ—П —З–Є—Б–ї–∞

\*/

int length = output1.length();

for (int i = 0; i < length - 1; i++) {

while(strArgs.charAt(i) == '0' || strArgs.charAt(i) == '1' || strArgs.charAt(i) == '2'

|| strArgs.charAt(i) == '3' || strArgs.charAt(i) == '4' || strArgs.charAt(i) == '5'

|| strArgs.charAt(i) == '6' || strArgs.charAt(i) == '7' || strArgs.charAt(i) == '8'

|| strArgs.charAt(i) == '9' && strArgs.charAt(i) != ' ') {

strArgs.deleteCharAt(i);

length--;

}

}

output1 = strArgs.toString();

}

public void AddSpace() {

output2 = output1;

StringBuffer strArgs = new StringBuffer(output2);

strArgs.ensureCapacity(32);

int length = output2.length();

/\*

\* –Ф–Њ–±–∞–≤–ї–µ–љ–љ—П –њ—А–Њ–±i–ї—Г

\*/

for (int i = 0; i < length; i++) {

if (strArgs.charAt(i) != ' ' && i != 0 && strArgs.charAt(i - 1) != ' ' && i != strArgs.length()) {

strArgs.insert(i + 1, ' ');

length++;

}

}

output2 = strArgs.toString();

}

public void DeleteSpace() {

output3 = output2;

StringBuffer strArgs = new StringBuffer(output3);

strArgs.ensureCapacity(32);

int length = output3.length();

/\*

\* –Т–Є–і–∞–ї–µ–љ–љ—П –њ—А–Њ–±i–ї—Г

\*/

for (int i = 0; i < length - 1; i++) {

if (i != 0 && i < strArgs.length() && strArgs.charAt(i) == ' ' && strArgs.charAt(i + 1) == ' ') {

while (strArgs.charAt(i) == ' ' && strArgs.charAt(i + 1) == ' ') {

strArgs.deleteCharAt(i + 1);

length--;

}

}

}

output3 = strArgs.toString();

outputFinal = output3;

}

public void ViewResultDelNum() {

System.out.printf("–°—В–∞–ї–Њ –њ—Ц—Б–ї—П –≤–Є–і–∞–ї–µ–љ–љ—П —Г—Б—Ц—Е —З–Є—Б–µ–ї: %s \n", output1);

}

public void ViewResultAdSp() {

System.out.printf("–°—В–∞–ї–Њ –њ—Ц—Б–ї—П –і–Њ–і–∞–≤–∞–љ–љ—П –±—А–∞–Ї—Г—О—З–Є—Е –њ—А–Њ–±—Ц–ї—Ц–≤: %s \n", output2);

}

public void ViewResultDelSp() {

System.out.printf("–°—В–∞–ї–Њ –њ—Ц—Б–ї—П –≤–Є–і–∞–ї–µ–љ–љ—П –Ј–∞–є–≤–Є—Е –њ—А–Њ–±—Ц–ї—Ц–≤: %s \n", output3);

}

public void ViewAfter() {

System.out.printf("–Ч–∞–≥–∞–ї—М–љ—Ц–є —А–µ–Ј—Г–ї—М—В–∞—В: %s \n\n", outputFinal);

}

}

Файл Collection.java

package ua.khpi.oop.radeivych06;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.Serializable;

import java.util.Iterator;

public class Collection implements Iterable<String>, Serializable{

static final long serialVersionUID = 1L;

private String [] mass = new String[10];

private int Size = 0;

private int ActualSize = 10;

public void setMass(String[] mass) {

this.mass = mass;

this.Size = mass.length;

this.ActualSize= mass.length;

}

public String[] getMass() {

var masst = new String[Size];

for(int i = 0; i<Size; i++) {

masst[i] = mass[i];

}

return masst;

}

public void sort() {

boolean prz = true;

while(prz) {

prz = false;

for(int i = 0; i < Size-1;i++) {

if(mass[i].compareTo(mass[i + 1]) > 0) {

String temp = mass[i];

mass[i]=mass[i+1];

mass[i+1]=temp;

prz = true;

}

}

}

};

public void Save(String fileName) throws IOException {

FileOutputStream outputStream= new FileOutputStream(fileName);

ObjectOutputStream objectOutputStream = new ObjectOutputStream(outputStream);

objectOutputStream.writeObject(this);

objectOutputStream.close();

}

public void Download(String fileName) throws IOException, ClassNotFoundException{

FileInputStream inStream= new FileInputStream(fileName);

ObjectInputStream objectInStream = new ObjectInputStream(inStream);

Collection temp = (Collection) objectInStream.readObject();

this.setMass(temp.getMass());

objectInStream.close();

}

public String toString() {

StringBuilder strB = new StringBuilder();

for(var str : this)

strB.append(str+" ");

if(!(strB.length()==0))

strB.deleteCharAt(strB.length()-1);

return strB.toString();

}

public void add(String string) {

if(Size==ActualSize) {

String [] tempMass = new String[ActualSize+10];

ActualSize += 10;

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

tempMass[i]=mass[i];

mass=tempMass;

}

mass[Size++]= string;

}

public void clear() {

mass = new String[10];

Size = 0;

ActualSize = 10;

}

public boolean remove(String string) {

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

if(mass[i].equals(string)) {

delete(i);

return true;

}

return false;

}

public Object[] toArray() {

return getMass();

}

public int size() {

return Size;

}

public boolean contains(String string) {

for(var str : this) {

if(str.equals(string))

return true;

}

return false;

}

public boolean containsAll(Collection container) {

for(var str : container)

if(!contains(str)) {

return false;

}

return true;

}

private void delete(int i) {

String [] tempMass = new String[ActualSize];

for(int j = 0; j<i;j++) {

tempMass[j]=mass[j];

}

for(int j = i+1;j<Size;j++) {

tempMass[j]=mass[j];

}

mass = tempMass;

Size -= 1;

}

public int find(String str) {

for(int i = 0; i < Size; i++) {

if(mass[i].equals(str))

return i;

}

return -1;

};

public boolean equals(Collection collection) {

if(collection.size() != Size){

return false;

}

for(int i = 0; i < Size; i++) {

if(!collection.get(i).equals( mass[i])) {

return false;

}

}

return true;

};

public String get(int i) {

if(i<Size)

return mass[i];

else

return null;

}

public Iterator<String> iterator() {

Iterator<String> MyIterator = new Iterator<String>(){

private int i = 0;

public String next() {

return mass[i++];

}

public boolean hasNext() {

return i<Size;

}

public void remove() {

delete(i-1);

}

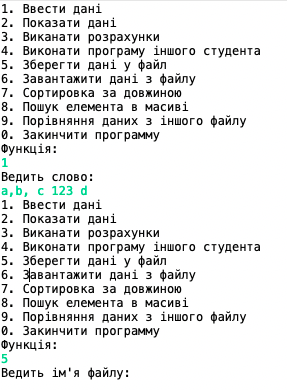
};

return MyIterator;

}

}

Демонстрація використання



ВИСНОВКИ

Протягом лабораторної роботи я навчився використовувати серіалізацію/десеріалізацію та використовувати бібліотеку класів іншого користувача.