Національний технічний університет

«Харківський політехнічний інститут»

**Лабораторна робота № 8**

з дисципліни «ООП»

Виконав студент групи КІТ - 320

Радєвич Владислав Романович

Перевірив викладач

Жилин Володимир Анатолійович

Харків 2021

**Основи введення/виведення Java SE**

Мета: оволодіння навичками управління введенням/виведенням даних з використанням класів платформи Java SE.

1 ВИМОГИ

* 1. **Розробник**

Інформація про розробника:

Радєвич Владислав Романович;

студент групи КІТ – 320;

номер варіанту.

* 1. **Загальне завдання**

Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання лабораторної роботи №7.

Забороняється використання стандартного протокола серіалізації.

Продемонструвати використання моделі Long Term Persistence.

Забезпечити діалог з користувачем у вигляді простого текстового меню.

При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.

1. ОПИС ПРОГРАМИ
   1. **Засоби ООП**

Програма призначена для управління введенням/виведенням даних з використанням класів платформи Java SE.

* 1. **Ієрархія та структура класів**

public class Main – головний клас

public class Container – клас контейнер

public class Outlet – клас предметної галузі

* 1. **Важливі фрагменти програми**

Файл Main.java

package ua.khpi.oop.radeivych08;

import java.beans.XMLDecoder;

import java.beans.XMLEncoder;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.ArrayList;

import java.util.LinkedHashMap;

import java.util.LinkedList;

import java.util.Scanner;

import ua.khpi.oop.radeivych08.Outlet;

import ua.khpi.oop.radeivych08.Container;

public class Main {

public static void main(String[] args) {

ArrayList<Integer> num1 = new ArrayList<Integer>();

num1.add(123456789);

num1.add(987654321);

num1.add(135798642);

ArrayList<Integer> num2 = new ArrayList<Integer>();

num2.add(987654321);

num2.add(135798642);

ArrayList<Integer> num3 = new ArrayList<Integer>();

num3.add(135798642);

num3.add(880053535);

LinkedHashMap<String, String> work1 = new LinkedHashMap<String, String>();

work1.put("–Я–Њ–љ–µ–і–µ–ї—М–љ–Є–Ї ", " –Њ—В 9 –і–Њ 18");

work1.put("–Т—В–Њ—А–љ–Є–Ї ", " –Њ—В 9 –і–Њ 18");

work1.put("–Я—П—В–љ–Є—Ж–∞ ", " –Њ—В 9 –і–Њ 18");

work1.put("–°—Г–±–±–Њ—В–∞ ", " –Њ—В 10 –і–Њ 15");

LinkedHashMap<String, String> work2 =new LinkedHashMap<String, String>();

work2.put("–Я–Њ–љ–µ–і–µ–ї—М–љ–Є–Ї ", " –Њ—В 8 –і–Њ 16");

work2.put("–Т—В–Њ—А–љ–Є–Ї ", " –Њ—В 8 –і–Њ 16");

work2.put("–°—А–і–µ–∞ ", " –Њ—В 9 –і–Њ 16");

work2.put("–Я—П—В–љ–Є—Ж–∞ ", " –Њ—В 10 –і–Њ 12");

LinkedHashMap<String, String> work3 = new LinkedHashMap<String, String>();

work3.put("–°–µ—А–µ–і–∞", "–Њ—В 9 –і–Њ 18");

work3.put("–І–µ—В–≤–µ—А", "–Њ—В 9 –і–Њ 18");

work3.put("–Я—П—В–љ–Є—Ж—П", "–Њ—В 9 –і–Њ 18");

work3.put("–°—Г–±–Њ—В–∞", "–Њ—В 10 –і–Њ 15");

Outlet[] array = new Outlet[3];

array[0] = new Outlet("ALLO", "–Т—Г–ї. –Ь—П—Г32", num1, "–Љ–∞–≥–∞–Ј–Є–љ —В–µ—Е–љ—Ц–Ї–Є", work1);

array[1] = new Outlet("–Ъ–ї–∞—Б—Б", "–Т—Г–ї. –Ъ–Є—Б12", num2, "—Б—Г–њ–µ—А–Љ–∞—А–Ї–µ—В", work2);

array[2] = new Outlet("AppleStore", "–≤—Г–ї. –Ь—П—Г3", num3,"–Љ–∞–≥–∞–Ј–Є–љ —В–µ—Е–љ—Ц–Ї–Є",work3);

Container cont = new Container();

cont.array = array;

int size = 3;

int i;

int number;

boolean n\_end = true;

boolean input;

boolean fileCheck = false;

String fileName = "";

String str;

String command;

String path = "/Users/whatislove/Documents/–Ф–Њ–Ї—Г–Љ–µ–љ—В—Л вАФ MacBook Air¬†вАФ –Т–ї–∞–і–Є—Б–ї–∞–≤/2 –Ї—Г—А—Б/Java";

do {

System.out.println(

"1. –°–µ—А—Ц–∞–ї—Ц–Ј–∞—Ж—Ц—П \r\n" +

"2. –Ф–µ—Б–µ—А—Ц–∞–ї—Ц–Ј–∞—Ж—Ц—П \r\n" +

"3. –Т–Є–≤–µ–і–µ–љ–љ—П –і–∞–љ–Є—Е –љ–∞ –µ–Ї—А–∞–љ \r\n" +

"4. –Ф–Њ–і–∞—В–Є –µ–ї–µ–Љ–µ–љ—В –≤ –Ї—Ц–љ–µ—Ж—М \r\n" +

"5. –Т–Є–і–∞–ї–Є—В–Є –њ–µ—А—И–Є–є –µ–ї–µ–Љ–µ–љ—В \r\n" +

"0. –Ч–∞–≤–µ—А—И–µ–љ–љ—П –њ—А–Њ–≥—А–∞–Љ–Є \r\n" +

"");

System.out.println("–Ю–±–µ—А—Ц—В—М —Д—Г–љ–Ї—Ж—Ц—О: ");

Scanner scan = new Scanner(System.in);

command = scan.nextLine();

switch (command) {

case "1":

System.out.println("–Т–µ–і–Є—В—М –љ–∞–Ј–≤—Г —Д–∞–є–ї—Г –і–ї—П –Ј–±–µ—А–µ–ґ–µ–љ–љ—П –і–∞–љ–Є—Е");

fileName = scan.nextLine();

if (fileName.indexOf(".xml") == -1) {

fileName += ".xml";

}

System.out.println("–Э–∞–Ј–≤–∞ –≤–∞—И–µ–≥–Њ XML —Д–∞–є–ї–∞ - " + fileName);

while(n\_end)

{

File directoryPath = new File(path);

String contents[] = directoryPath.list();

str = "";

if (contents == null) {

System.out.println("–Ъ–∞—В–∞–ї–Њ–≥ –љ–µ –±—Г–≤ –≤–Є–±—А–∞–љ–Є–є");

path = path.substring(0, path.length() - str.length());

continue;

}

System.out.println("–®–ї—П—Е –љ–∞ –і–∞–љ–Є–є –Љ–Њ–Љ–µ–љ—В: " + path);

System.out.println("–°–њ–Є—Б–Њ–Ї –≤—Б—Ц—Е —Д–∞–є–ї—Ц–≤ —В–∞ –Ї–∞—В–∞–ї–Њ–≥—Ц–≤ –≤ –Ј–∞–Ј–љ–∞—З–µ–љ–Њ–Љ—Г —И–ї—П—Е—Г:");

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

System.out.println(i + 1 + ") " + contents[i]);

}

System.out.println(++i + ") –Њ–±–µ—А–Є—В—М —Ж–µ–є –≤–∞—А–Є–∞–љ—В –і–ї—П –Ј–±–µ—А–µ–ґ–µ–љ–љ—П —Д–∞–є–ї—Г —В—Г—В.");

System.out.println(++i + ") –Њ–±–µ—А—Ц—В—М —Ж–µ–є –≤–∞—А–Є–∞–љ—В –і–ї—П –њ—Ц–і–љ—П—В—В—П –Ї–∞—В–∞–ї–Њ–≥–Њ–Љ –≤–Є—Й–µ");

System.out.println("–Ю–±–µ—А—Ц—В—М –Њ–і–Є–љ –Ј –Ї–∞—В–∞–ї–Њ–≥—Ц–≤ –і–ї—П –Ј–±–µ—А–µ–ґ–µ–љ–љ—П:");

number = 0;

do {

input = false;

number = scan.nextInt();

if (number > i) {

System.out.println("–Э–µ–њ—А–∞–≤–Є–ї—М–љ–Є–є –љ–Њ–Љ–µ—А, —Б–њ—А–Њ–±—Г–є—В–µ —Й–µ");

input = true;

}

} while(input);

if(number == i -1) {

n\_end = false;

}

else if (number == i) {

if (path.contains(str) && !str.equals("")) {

path = path.substring(0, path.length() - str.length());

} else

System.out.println("–Э–µ –Љ–Њ–ґ–ї–Є–≤–Њ –њ—Ц–і–љ—П—В–Є—Б—П –Ї–∞—В–∞–ї–Њ–≥–Њ–Љ –≤–Є—Й–µ");

} else {

str = "\\" + contents[number - 1];

path += str;

}

}

File dir = new File(path);

File actualFile = new File (dir, fileName);

try {

XMLEncoder e = new XMLEncoder( new BufferedOutputStream

(new FileOutputStream(actualFile)));

e.writeObject(cont.array);

e.close();

System.out.println("Serialization successful\n");

fileCheck = true;

} catch (FileNotFoundException e) {

e.printStackTrace();

}

break;

case "2":

str = "";

n\_end = true;

if (fileCheck) {

System.out.println("–Ж–Љ'—П –љ–∞ –і–∞–љ–Є–є –Љ–Њ–Љ–µ–љ—В —Д–∞–є–ї–∞ XML - " + fileName);

} else

System.out.println("XML-—Д–∞–є–ї –љ–µ –±—Г–≤ —Б—В–≤–Њ—А–µ–љ–љ—П –њ—Ц–і —З–∞—Б —Ж—М–Њ–≥–Њ —Б–µ–∞–љ—Б—Г");

path = "/Users/whatislove/Documents/–Ф–Њ–Ї—Г–Љ–µ–љ—В—Л вАФ MacBook Air¬†вАФ –Т–ї–∞–і–Є—Б–ї–∞–≤/2 –Ї—Г—А—Б/Java";

while (n\_end) {

File directoryPath = new File(path);

//List of all files and directories

String contents[] = directoryPath.list();

if (contents == null) {

System.out.println("–Э–µ –Њ–±—А–∞–љ–Љ–є –Ї–∞—В–∞–ї–Њ–≥ —В–∞ XML-—Д–∞–є–ї, —Б–њ—А–Њ–±—Г–є—В–µ —Й–µ");

path = path.substring(0, path.length() - str.length());

continue;

}

System.out.println("–®–ї—П—Е –љ–∞ –і–∞–љ–Є–є –Љ–Њ–Љ–µ–љ—В: " + path);

System.out.println("–°–њ–Є—Б–Њ–Ї –≤—Б—Ц—Е —Д–∞–є–ї—Ц–≤ —В–∞ –Ї–∞—В–∞–ї–Њ–≥—Ц–≤ –≤ –Ј–∞–Ј–љ–∞—З–µ–љ–Њ–Љ—Г —И–ї—П—Е—Г:");

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

System.out.println(i + 1 + ") " + contents[i]);

}

System.out.println(++i + ") –Њ–±–µ—А—Ц—В—М —Ж–µ–є –≤–∞—А–Є–∞–љ—В –і–ї—П –њ—Ц–і–љ—П—В—В—П –Ї–∞—В–∞–ї–Њ–≥–Њ–Љ –≤–Є—Й–µ");

System.out.println("–Ю–±–µ—А–Є—В—М –≤ —П–Ї–Є–є –Ї–∞—В–∞–ї–Њ–≥ –њ–µ—А–µ–є—В–Є –∞–±–Њ XML-—Д–∞–є–ї, —П–Ї–Є–є –љ–µ–Њ–±—Е—Ц–і–љ–Њ –њ—А–Њ—З–Є—В–∞—В–Є: ");

do {

input = false;

number = scan.nextInt();

if (number > i) {

System.out.println("–Э–µ–њ—А–∞–≤–Є–ї—М–љ–Є–є –љ–Њ–Љ–µ—А, —Б–њ—А–Њ–±—Г–є—В–µ —Й–µ");

input = true;

}

} while(input);

if (number == i) {

if (path.contains(str) && !str.equals("")) {

path = path.substring(0, path.length() - str.length());

} else

System.out.println("–Э–µ –Љ–Њ–ґ–ї–Є–≤–Њ –њ—Ц–і–љ—П—В–Є—Б—П –Ї–∞—В–∞–ї–Њ–≥–Њ–Љ –≤–Є—Й–µ");

} else {

str = "/" + contents[number - 1];

path += str;

if (contents[number - 1].contains(".xml")) {

n\_end = false;

}

}

}

dir = new File(path);

try {

XMLDecoder decoder = new XMLDecoder(new BufferedInputStream(new FileInputStream(dir)));

cont.array = (Outlet[]) decoder.readObject();

decoder.close();

cont.Show();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

break;

case "3":

for (Outlet recruitmentAgency : cont.array) {

recruitmentAgency.Show();

}

break;

case "4":

cont.add();

break;

case "5":

cont.remove();

break;

case "0":

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

break;

default:

System.out.println("–Э–µ—Ц—Б–љ—Г—О—З–∞ —Д—Г–љ–Ї—Ж—Ц—П");

break;

}

} while(!(command.equals("0")));

}

}

Файл Container.java

package ua.khpi.oop.radeivych08;

import java.util.ArrayList;

import java.util.LinkedHashMap;

import java.util.Scanner;

public class Container {

Outlet[] array = new Outlet[3];

public void add() {

ArrayList<Integer> num = new ArrayList<Integer>();

int size = array.length;

Scanner scan = new Scanner(System.in);

System.out.println("–Т–µ–і–Є—В—М –љ–∞–Ј–≤—Г —В–Њ—А–≥—Ц–≤–µ–ї—М–љ–Њ—Ч —В–Њ—З–Ї–Є: ");

String namePointOfSale = scan.nextLine();

System.out.println("–Т–µ–і–Є—В—М –∞–і—А–µ—Б—Г:");

String nameAddress = scan.nextLine();

System.out.println("–Т–µ–і–Є—В—М —Б–њ–µ—Ж—Ц–∞–ї—Ц–Ј–∞—Ж—Ц—О —В–Њ—А–≥—Ц–≤–µ–ї—М–љ–Њ—Ч —В–Њ—З–Ї–Є: ");

String specialty = scan.nextLine();

System.out.println("–Т–µ–і–Є—В—М —А–Њ–±–Њ—З—Ц –і–љ—Ц —В–Њ—З–Ї–Є: ");

String day = scan.nextLine();

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

String time = scan.nextLine();

System.out.println("–Т–µ–і–Є—В—М –љ–Њ–Љ–µ—А —В–µ–ї–µ—Д–Њ–љ—Г (—В—Ц–ї—М–Ї–Њ 9 —Ж–Є—Д—А): ");

int number = scan.nextInt();

LinkedHashMap<String, String> workTime = new LinkedHashMap<String, String>();

workTime.put(day,time);

num.add(number);

Outlet new\_el = new Outlet(namePointOfSale, nameAddress, num, specialty, workTime);

Outlet new\_arr[] = new Outlet[++size];

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

{

new\_arr[i] = array[i];

}

new\_arr[size-1] = new\_el;

array = new\_arr;

}

public void remove() {

int size = array.length;

if(size != 0) {

Outlet new\_arr[] = new Outlet[--size];

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

new\_arr[i] = array[i];

}

array = new\_arr;

} else {

System.out.println("–Ь–∞—Б–Є–≤ –њ–Њ—А–Њ–ґ–љ—Ц–є");

}

}

public void Show() {

for (Outlet check : array) {

check.Show();

}

}

}

Файл Outlet.java

package ua.khpi.oop.radeivych08;

import java.util.ArrayList;

import java.util.LinkedHashMap;

import java.util.Set;

public class Outlet {

private String namePointOfSale;

private String nameAddress;

private ArrayList<Integer> phoneNumber;

private String specialization;

private LinkedHashMap<String, String> workTime = new LinkedHashMap<String, String>();

public String getNamePointOfSale() {

return namePointOfSale;

}

public void setNamePointOfSale(String namePointOfSale) {

this.namePointOfSale = namePointOfSale;

}

public String getNameAddress() {

return nameAddress;

}

public void setNameAddress(String nameAddress) {

this.nameAddress = nameAddress;

}

public ArrayList<Integer> getPhoneNumber() {

return phoneNumber;

}

public void setPhoneNumber(ArrayList<Integer> phoneNumber) {

this.phoneNumber = phoneNumber;

}

public String getSpecialization() {

return specialization;

}

public void setSpecialization(String specialization) {

this.specialization = specialization;

}

public LinkedHashMap<String, String> getWorkTime() {

return workTime;

}

public void setWorkTime(LinkedHashMap<String, String> workTime) {

this.workTime = workTime;

}

public void Addphone\_number(int value) {

this.phoneNumber.add(value);

}

public void Addwork\_time(String key, String value) {

this.workTime.put(key, value);

}

// public String toStringPhoneNumber() {

// String string = "";

// for (Integer integer : phoneNumber) {

// string += "+" + integer.toString();

// string+="\n";

// }

// return string;

// }

//

public String toStringWorkTime() {

String string ="";

Set set = workTime.entrySet();

for (Object el: set) {

string+=el.toString();

string+="\n";

}

return string;

}

public Outlet(String namePointOfSale, String nameAddress, ArrayList<Integer> phoneNumber, String specialization,

LinkedHashMap<String, String> workTime) {

super();

this.namePointOfSale = namePointOfSale;

this.nameAddress = nameAddress;

this.phoneNumber = phoneNumber;

this.specialization = specialization;

this.workTime = workTime;

}

public Outlet() {

super();

// TODO Auto-generated constructor stub

}

public void Show()

{

System.out.println("–Э–∞–Ј–≤–∞ —В–Њ—А–≥—Ц–≤–µ–ї—М–љ–Њ—Ч —В–Њ—З–Ї–Є: "+ namePointOfSale +"\r\n" + "–Р–і—А–µ—Б–∞: " + nameAddress +

"\n–Э–Њ–Љ–µ—А–∞ —В–µ–ї–µ—Д–Њ–љ—Ц–≤: \n"+ this.getPhoneNumber() + "\r" +

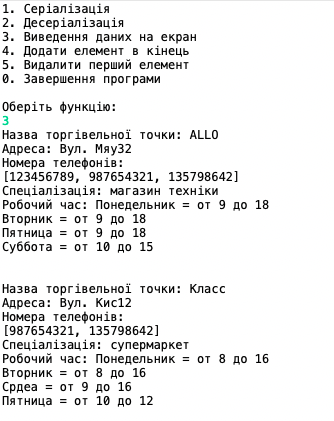
"–°–њ–µ—Ж—Ц–∞–ї—Ц–Ј–∞—Ж—Ц—П: " + specialization +

"\n–†–Њ–±–Њ—З–Є–є —З–∞—Б: "+ toStringWorkTime() + "\r\n");

}

}

Демонстрація використання



ВИСНОВКИ

Протягом лабораторної роботи я оволодів навичками управління введенням/виведенням даних з використанням класів платформи Java SE.