Автор:Горянський І.

Группа:1.КІТ101.8б

Лабораторна робота № 8

## Основи введення/виведення Java SE

**1. ЗАВДАННЯ ДО РОБОТИ**

**Загальне завдання:**.

1. Забезпечити можливість збереження і відновлення масива об’єктів рішення завдання [лабораторної роботи №7](https://oop-khpi.github.io/" \l "task_07).
2. Забороняється використання [стандартного протокола серіалізації](https://docs.oracle.com/javase/8/docs/platform/serialization/spec/serialTOC.html).
3. Продемонструвати використання моделі [Long Term Persistence](https://docs.oracle.com/javase/tutorial/javabeans/advanced/longpersistence.html).
4. Забезпечити діалог з користувачем у вигляді простого текстового меню.
5. При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.
   1. **Опис змінних:**

**private** String Address; переменная для значения

**private** String Name; переменная для значения

**private** String Special; переменная для значения

**private** String Phone; переменная для значения

**private** Date StartTime; переменная для значения

**private** Date EndTime; переменная для значения

String str1; переменная для значения

String str2; переменная для значения

String Line; переменная для значения

**private** **static** **int** *size* = 0;переменная для размера массива

**private** **static** Directory[] *array* = **new** Directory[*size*];массив объектов

Текст програми:

**package** ua.khpi.oop.gorianckiy08;

**import** java.util.Arrays;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.text.ParseException;

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**public** **class** Functions {

**private** **static** **int** *size* = 0;

**private** **static** Directory[] *array* = **new** Directory[*size*];

**void** add(Directory temp) {

Directory buf = **new** Directory(temp);

**if** (*size* == 0) {

*array* = Arrays.*copyOf*(*array*, *array*.length +1);

*array*[*array*.length -1] = buf;

} **else** {

Directory[] newArray = Arrays.*copyOf*(*array*, *array*.length+1);

newArray[*array*.length] = buf;

*array* = Arrays.*copyOf*(newArray,newArray.length);

}

}

**void** show() {

**if**(*array*.length == 0) {

System.***out***.println("No data!");

}**else** {

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

System.***out***.println("Shop centure #" + i+1);

System.***out***.println("Name of Shop centure: "+*array*[i].getName());

System.***out***.println("Address of Shop centure: "+*array*[i].getAddress());

System.***out***.println("Specialization of Shop centure: "+*array*[i].getSpecial());

System.***out***.println("Phone number of Shop centure: "+*array*[i].getPhone());

System.***out***.println("Work time of Shop centure: "+*array*[i].getTime() + "-" + *array*[i].getTime1());

}

}

}

**public** **void** ChooseMenu() {

System.***out***.println("\n\nChoose what you want to do:");

System.***out***.println("1. Add directory");

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

System.***out***.println("3. Out to File ");

System.***out***.println("4. Out from File ");

System.***out***.println("5. Xml");

System.***out***.println("6. Work with File ");

System.***out***.println("0. Exit");

System.***out***.print("Write:");

}

**public** **void** ChooseMenuFile() {

System.***out***.println("\n\nChoose what you want to do:");

System.***out***.println("1. Show directory");

System.***out***.println("2. Create directory");

System.***out***.println("3. Rename directory");

System.***out***.println("4. Create File");

System.***out***.println("5. Data output to a file ");

System.***out***.println("6. Exit");

System.***out***.print("Write:");

}

**void** toFile() {

**try** {

File file = **new** File("File.txt");

**if**(!file.exists())

file.createNewFile();

PrintWriter pw = **new** PrintWriter(file);

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

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

pw.println(*size*);

pw.println(*array*[i].getName());

pw.println(*array*[i].getAddress());

pw.println(*array*[i].getSpecial());

pw.println(*array*[i].getPhone());

pw.println(*array*[i].getTime() );

pw.println(*array*[i].getTime1());

}

pw.close();

}

**catch**(IOException e){

System.***out***.print("Error: " + e);

}

}

**void** toFile(String Path) {

**try** {

File file = **new** File(Path);

**if**(!file.exists())

file.createNewFile();

PrintWriter pw = **new** PrintWriter(file);

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

pw.println(*size*);

pw.println(*array*[i].getName());

pw.println(*array*[i].getAddress());

pw.println(*array*[i].getSpecial());

pw.println(*array*[i].getPhone());

pw.println(*array*[i].getTime() );

pw.println(*array*[i].getTime1());

}

pw.close();

}

**catch**(IOException e){

System.***out***.print("Error: " + e);

}

}

**void** FromFile() **throws** ParseException {

BufferedReader br = **null**;

Directory temp = **new** Directory();

**try** {

br = **new** BufferedReader(**new** FileReader("File.txt"));

String Line;

Line = br.readLine();

*size* = Integer.*parseInt*(Line);

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

Line = br.readLine();

temp.setName(Line);

Line = br.readLine();

temp.setAddress(Line);

Line = br.readLine();

temp.setSpecial(Line);

Line = br.readLine();

temp.setPhone(Line);

Line = br.readLine();

temp.setTime(Line);

Line = br.readLine();

temp.setTime1(Line);

*array*[i] = **new** Directory(temp);

}

}**catch**(IOException e) {

System.***out***.print("Error: " + e);

}**finally** {

**try** {

br.close();

}**catch**(IOException e ) {

System.***out***.print("Error: " + e);

}

}

}

}

**package** ua.khpi.oop.gorianckiy08;

**import** java.util.Scanner;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**import** java.text.ParseException;

**import** java.io.IOException;

**import** java.util.Date;

**import** java.text.SimpleDateFormat;

**public** **class** Directory {

**private** String Address;

**private** String Name;

**private** String Special;

**private** String Phone;

**private** Date StartTime;

**private** Date EndTime;

String str1;

String str2;

Directory(){

Address = **null**;

Name = **null**;

Special = **null**;

Phone = **null**;

StartTime = **null**;

EndTime = **null**;

}

Directory(**final** Directory obj){

Address = obj.Address;

Name = obj.Name;

Special = obj.Special;

Phone = obj.Phone;

StartTime = obj.StartTime;

EndTime = obj.EndTime;

}

**public** **void** setAddress(String Address) {

**this**.Address =Address;

}

**public** **void** setName(String Name) {

**this**.Name = Name;

}

**public** **void** setSpecial(String Special) {

**this**.Special = Special;

}

**public** **void** setPhone(String Phone) {

**this**.Phone = Phone;

}

**public** **void** setTime(**final** String d) **throws** ParseException {

**if**(checkTime(d)) {

SimpleDateFormat sdf = **new** SimpleDateFormat("HH:mm");

**this**.StartTime = sdf.parse(d);

}

}

**public** **void** setTime1(String d) **throws** ParseException{

**if**(checkTime(d)) {

SimpleDateFormat sdf = **new** SimpleDateFormat("HH:mm");

**this**.EndTime = sdf.parse(d);

}

}

**private** **boolean** checkTime(**final** String d) {

**final** String tf = "HH:mm";

**try** {

SimpleDateFormat sdf = **new** SimpleDateFormat(tf);

sdf.setLenient(**false**);

sdf.parse(d);

**return** **true**;

} **catch** (ParseException e) {

System.***out***.println("Error:" + e);

**return** **false**;

}

}

// private boolean checkNumber(String Phone) {

// Pattern pattern = Pattern.compile("[0-9]");

// Matcher matcher = pattern.matcher(Phone);

// return matcher.matches();

// }

String getAddress() {

**return** Address;

}

String getName() {

**return** Name;

}

String getSpecial() {

**return** Special;

}

String getPhone() {

**return** Phone;

}

**public** String getTime() {

SimpleDateFormat sdf = **new** SimpleDateFormat("HH:mm");

**return** sdf.format(**this**.StartTime);

}

**public** String getTime1() {

SimpleDateFormat sdf = **new** SimpleDateFormat("HH:mm");

**return** sdf.format(**this**.EndTime);

}

**public** **void** generateStore() **throws** ParseException,IOException {

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

System.***out***.println("Enter name of outlet");

Name = s.nextLine();

System.***out***.println("Enter Address");

Address = s.nextLine();

System.***out***.println("Enter Specialization");

Special = s.nextLine();

Pattern pattern = Pattern.*compile*("\\d{10,12}");

String str = **null**;

Matcher m = **null**;

**do** {

System.***out***.println("Enter Phone number");

str = s.nextLine();

m = pattern.matcher(str);

}**while**(!m.matches());

Phone = str;

**do** {

System.***out***.println("Enter begining of work");

str1 = s.nextLine();

}**while**(!checkTime(str1));

setTime(str1);

**do** {

System.***out***.println("Enter ending of work");

str1 = s.nextLine();

}**while**(!checkTime(str1));

setTime1(str1);

}

}

**package** ua.khpi.oop.gorianckiy08;

**import** org.w3c.dom.DOMImplementation;

**import** org.w3c.dom.Document;

**import** org.w3c.dom.Element;

**import** org.w3c.dom.Node;

**import** org.w3c.dom.NodeList;

**import** org.w3c.dom.Text;

**import** org.w3c.dom.ls.DOMImplementationLS;

**import** org.w3c.dom.ls.LSOutput;

**import** org.w3c.dom.ls.LSSerializer;

**import** javax.xml.parsers.DocumentBuilder;

**import** javax.xml.parsers.DocumentBuilderFactory;

**import** javax.xml.parsers.ParserConfigurationException;

**import** java.io.File;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.lang.reflect.Array;

**import** java.nio.file.\*;

**import** java.text.ParseException;

**import** java.util.Scanner;

**public** **class** main {

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

Directory MyDirectory = **new** Directory();

Functions MyFunctions = **new** Functions();

**int** loop = 1;

**while** (loop == 1) {

MyFunctions.ChooseMenu();

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

**int** choose = sc.nextInt();

**switch**(choose) {

**case** 1:

MyDirectory.generateStore();

MyFunctions.add(MyDirectory);

**break**;

**case** 2:

MyFunctions.show();

**break**;

**case** 3 :

MyFunctions.toFile();

**break**;

**case** 4:

MyFunctions.FromFile();

**break**;

**case** 5:

DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();

**try** {

DocumentBuilder builder = factory.newDocumentBuilder();

Document document = builder.newDocument();

Element root = document.createElement( "root" );

Element store = document.createElement("Store");

document.appendChild(root);

root.appendChild(store);

store.setAttribute("Name","Posad");

store.setAttribute("Addres:","Pushkinskaya 85");

store.setAttribute("Specialization","Grocery store");

store.setAttribute("Hotline:","+380665992793");

store.setAttribute("Work\_Time:","08:00 - 22:00");

DOMImplementation impl = document.getImplementation();

DOMImplementationLS implLS = (DOMImplementationLS)impl.getFeature("LS", "3.0");

LSSerializer ser = implLS.createLSSerializer();

ser.getDomConfig().setParameter("Format-pretty-print",**true**);

String str =ser.writeToString(document);

System.***out***.print(str);

LSOutput out = implLS.createLSOutput();

out.setEncoding("UTF-8");

out.setByteStream(Files.*newOutputStream*(Paths.*get*("laba.xml")));

ser.write(document,out);

}**catch** (ParserConfigurationException ex) {

ex.printStackTrace(System.***out***);

}

**break**;

**case** 6:

**int** loop1 = 1;

**while**(loop1==1) {

MyFunctions.ChooseMenuFile();

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

**int** choose1 = sc.nextInt();

**switch**(choose1) {

**case** 1:

System.***out***.println("Enter the path to file");

Scanner scan1 = **new** Scanner(System.***in***);

String myDir = scan1.next();

File dir = **new** File(myDir);

**if**(dir.isDirectory())

{

**for**(File item : dir.listFiles()){

**if**(item.isDirectory()){

System.***out***.println(item.getName() + " \t folder");

}

**else**{

System.***out***.println(item.getName() + "\t file");

}

}

}

**break**;

**case** 3:

System.***out***.println("Enter the path to file and his name ");

Scanner scan3 = **new** Scanner(System.***in***);

String newDir = scan3.next();

File dir2 = **new** File(newDir);

System.***out***.println("Enter the path to file and his new name ");

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

String str = scan.next();

File reDir = **new** File(str);

dir2.renameTo(reDir);

**break**;

**case** 2:

System.***out***.println("Enter the path to file");

Scanner scan2 = **new** Scanner(System.***in***);

String newDir1 = scan2.next();

File dir1 = **new** File(newDir1);

**boolean** created = dir1.mkdir();

**if**(created)

System.***out***.println("Folder has been created");

**break**;

**case** 4:

System.***out***.println("Enter the path to file");

Scanner scan5 = **new** Scanner(System.***in***);

String newDir2 = scan5.next();

File newFile = **new** File(newDir2);

**try**

{

**boolean** created1 = newFile.createNewFile();

**if**(created1)

System.***out***.println("File has been created");

}

**catch**(IOException ex){

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

}

**break**;

**case** 5:

Scanner scan6 = **new** Scanner(System.***in***);

String Path1 = scan6.next();

File file = **new** File(Path1);

**if**(!file.exists()) {

System.***out***.println("Please,follow point 4");

}**else** {

System.***out***.print("Please,enter the path to file");

Scanner scan4 = **new** Scanner(System.***in***);

String Path = scan4.next();

//String Path = "C:\\Users\\User\\eclipse-workspace\\laba 8\\File1.txt";

MyFunctions.toFile(Path);

}

**break**;

**case** 6:

loop1= 0;

**break**;

}

}

**case** 0 :

loop = 0;

**break**;

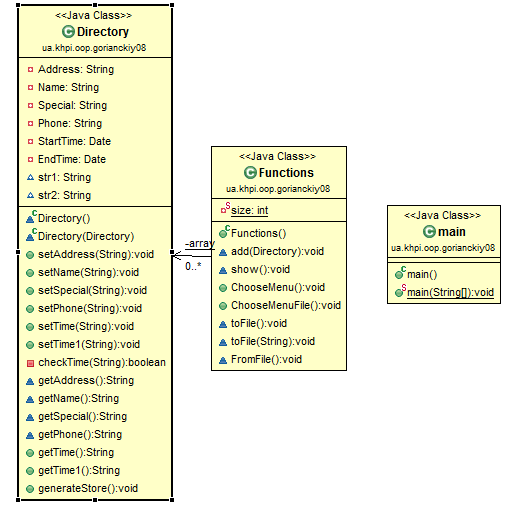
}

}

}

}

2.Діаграми



3.Результат:

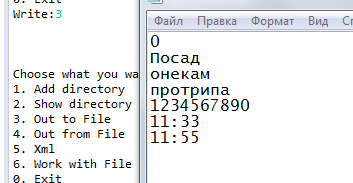


Рисунок 1- результат виведення даних в файл

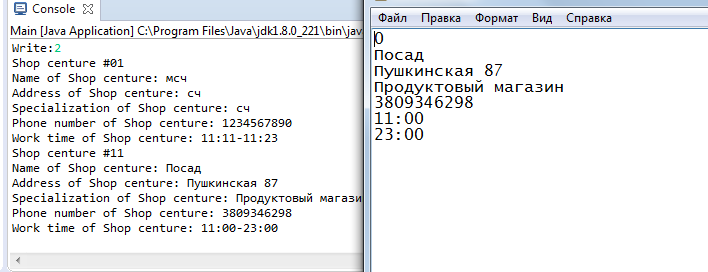


Рисунок 2 – Результат після виведення даних з файлу

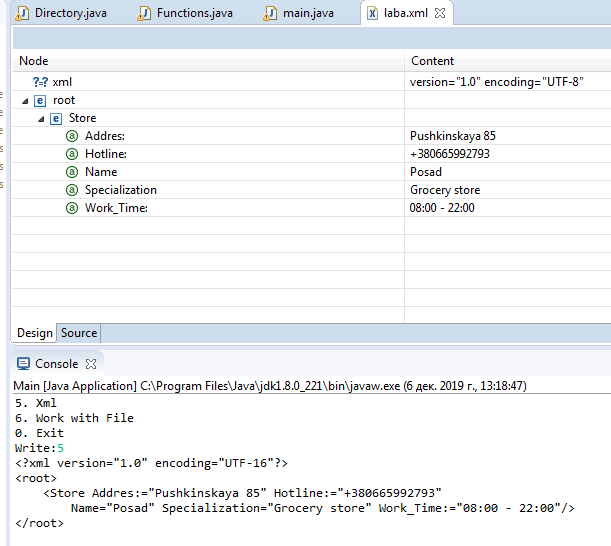


Рисунок 3 – Створення xml документу

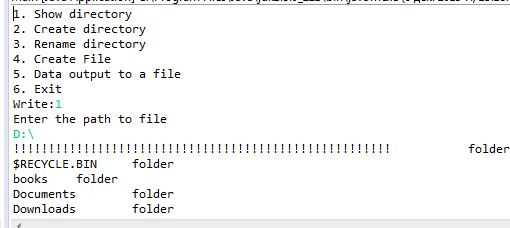


Рисунок 4 – Робота з дерикторіями

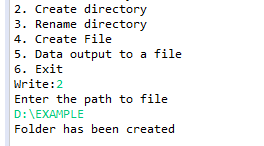
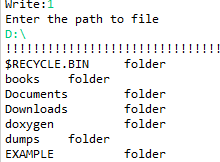
 

Рисунок 6 – Створення дерикторії

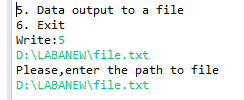
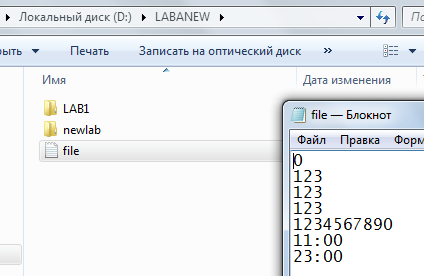
 

Рисунок 7 – Виведення даних в файл

ВИСНОВОК

При виконанні цієї лабораторної роботи були набуті навички роботи створенням документів xml та виводу даних,робота з дерик торіями,їх вмісту,створення,виводу даних з вибором місця.