Автор:Горянський І.

Группа:1.КІТ101.8б

Лабораторна робота №

## Об’єктно-орієнтована декомпозиція

**1. ЗАВДАННЯ ДО РОБОТИ**

**Загальне завдання:**.

1. Використовуючи об’єктно-орієнтований аналіз, реалізувати класи для представлення сутностей відповідно [прикладної задачі](https://oop-khpi.github.io/" \l "task_07_app) - domain-об’єктів.
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об’єктів.
   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.gorianckiy07;

**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.gorianckiy07;

**import** java.util.Arrays;

**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());

}

}

}

}

**package** ua.khpi.oop.gorianckiy07;

**import** java.io.IOException;

**import** java.text.ParseException;

**public** **class** Main {

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

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

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

MyDirectory.generateStore();

MyFunctions.add(MyDirectory);

MyDirectory.generateStore();

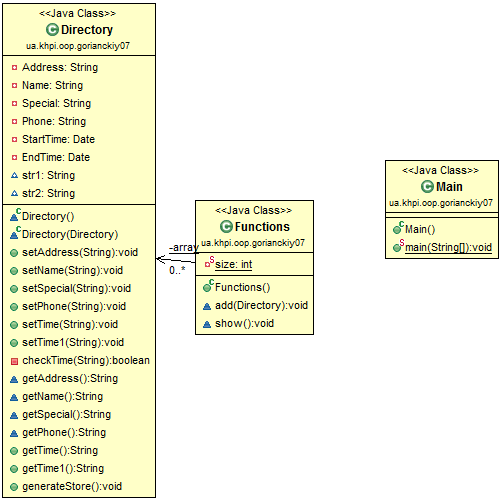
MyFunctions.add(MyDirectory);

MyFunctions.show();

}

}

2.Діаграми



3.Результат:

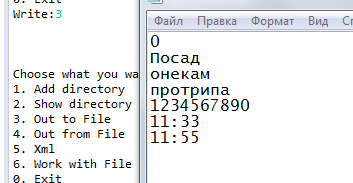


Рисунок 1- результат виведення даних в файл

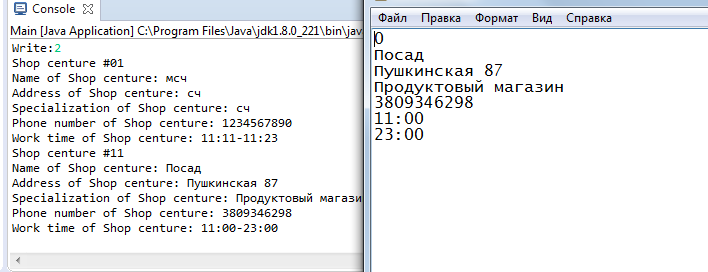


Рисунок 2 – Результат після виведення даних з файлу

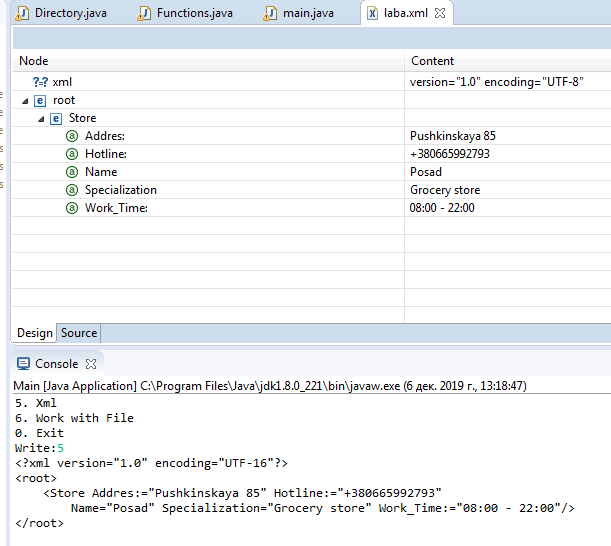


Рисунок 3 – Створення xml документу

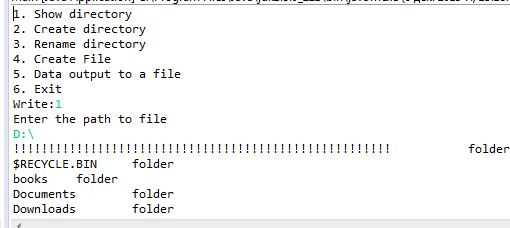


Рисунок 4 – Робота з дерикторіями

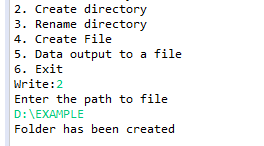
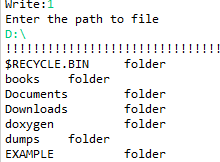
 

Рисунок 6 – Створення дерикторії

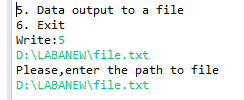
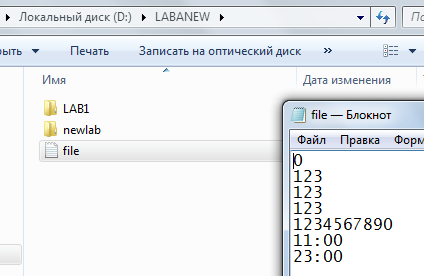
 

Рисунок 7 – Виведення даних в файл

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.