

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

# **Звіт**

## **Лабораторна робота №5**

З дисципліни:

Об’єктно орієнтоване програмування

**Виконав**

студент групи КН-111

Бойко Андрій

**Викладач:**

Грабовська Н. Р.

# Тема роботи

Використання об'єктно-орієнтованого підходу для розробки об'єкта предметної (прикладної) галузі. Оволодіння навичками управління введенням/виведенням даних з використанням класів Java SE.

## 1. Вимоги

### 1.1 Розробник

Бойко Андрій Віталійович

КН-111

3 варіант

### 1.2 Загальне завдання

Ознайомлення з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону. Ознайомлення з принципами використання регулярних виразів для обробки тексту.

### 1.3 Завдання

Продемонструвати ефективне (оптимальне) використання регулярних виразів для перевірки коректності (валідації) даних, що вводяться, перед записом в domain-об'єкти відповідно до призначення кожного поля для заповнення розробленого контейнера в попередній роботі: • при зчитуванні даних з текстового файла в автоматичному режимі; • при введенні даних користувачем в діалоговому режимі.

Використовуючи програми рішень попередніх задач, продемонструвати ефективне (оптимальне) використання регулярних виразів при вирішенні практичного завдання з Прикладні задачі. Список №2. 20 варіантів. • Передбачити можливість незначної зміни умов пошуку. • Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах.

## 2.Опис програми

### Main

```
package lab_3;

import java.beans.XMLDecoder;
import java.beans.XMLEncoder;
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.util.ArrayList;

import java.util.Scanner;

import lab_4.LinkedList;
class Menu
{
    int k;
    public int execute(Scanner in)
    {
        System.out.println("Choose what you want to do:");
        System.out.println("1. Create new person");
        System.out.println("2. Change name of the person");
        System.out.println("3. Change date of the person");
        System.out.println("4. Change numbers");
        System.out.println("5. Change adress");
        System.out.println("6. Change regdate");
        System.out.println("7. name");
        System.out.println("8. date");
        System.out.println("9. number");
        System.out.println("10. adress");
        System.out.println("11. regdate");
        System.out.println("12. Print to the file");
        System.out.println("13. Get from the file");
        System.out.println("14. Choose the person");
        System.out.println("15 . Finish the program");
        k = in.nextInt();

        return k;
    }
}

public class Main {
```

```

public static void main(String[] args) throws FileNotFoundException {

    LinkedList<AdressBook> list = new LinkedList<AdressBook>();
    ArrayList<Integer> temp=new ArrayList<>();
    ArrayList<Integer> temp1=new ArrayList<>();
    int user = 0;
    int N=0;
    Scanner in = new Scanner(System.in);
    Scanner on = new Scanner(System.in);
    Menu main = new Menu();

    while(true)
    {
        int k = main.execute(in);

        switch (k)
        {
            case 1 :
                list.add(new AdressBook());
                break;

            case 2 :
                System.out.println("Enter the name : ");
                list.get(user).setName(on.nextLine());
                break;

            case 3 :
                System.out.println("Enter the date : ");
                list.get(user).setDate(on.nextLine());
                break;

            case 4 :
                System.out.println("Enter phone number and its place : ");
                int t;
                while(true) {
                    t=in.nextInt();
                    if(t==0)
                    {
                        break;
                    }
                    else {
                        temp.add(t);
                    }
                }
                list.get(user).setNumbers(temp);

            break;

```

```

case 5 :
    System.out.println("Enter adress : ");
    list.get(user).setAddress(on.nextLine());
break;

case 6 :
    System.out.println("Enter regdate : ");
    list.get(user).setRegdate(on.nextLine());
break;

case 7:
    System.out.println(list.get(user).getname());
break;

case 8 :
    System.out.println((list.get(user).getDate()));
break;

case 9 :
    temp1=list.get(user).getNumbers();
    System.out.println(temp1);

break;

case 10 :
    System.out.println(list.get(user).getAddress());
break;

case 11 :
    System.out.println(list.get(user).getRegdate());
break;
case 12:
    XMLEncoder encoder;
    try {
        encoder = new XMLEncoder(
            new BufferedOutputStream(
                new FileOutputStream(PathGetter.getPath(in).toString())));
        encoder.writeObject(list);
        encoder.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
break;
case 13 :
    XMLDecoder decoder = new XMLDecoder(
        new BufferedInputStream(
            new FileInputStream(PathGetter.getPath(in).toString())));

```

```

        list =(LinkedList<AdressBook>) decoder.readObject();
        decoder.close();

        break;

    case 14 :
        System.out.println("Choose the person from 0 to "+(list.size()-1));
        user = in.nextInt();
        break;
    case 15 :
        in.close();
        on.close();
        return;

    }
}

}}

```

### **AdressBook**

```
package lab_3;
```

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;
```

```
public class AdressBook{
    private String name;
    private String date;
    private ArrayList<Integer> numbers;
    private String address;
    private String regdate;
    Scanner scn=new Scanner(System.in);
    public AdressBook(String name)
    {
        this.name = name;
    }
    public AdressBook()
    {
    }
    public void setName(String name)
    {
        this.name = name;
    }

    public void setDate(String date)

```

```

{
    this.date = date;
}
public void setNumbers(ArrayList<Integer> value ) {

    numbers=value;
}

public void setAddress(String address)
{
    this.adress = address;
}
public void setRegdate(String regdate)
{
    this.regdate = regdate;
}


public String getname()
{
    return name;
}
public String getDate()
{
    return date;
}


public String getAddress()
{
    return adress;
}
public String getRegdate()
{
    return regdate;
}
public ArrayList<Integer> getNumbers()
{
    return numbers;
}

}

```

### **PathGetter**

```
package lab_3;
```

```
import java.io.IOException;
import java.nio.file.*;
import java.util.ArrayList;
```

```

import java.util.Scanner;

public class PathGetter {

    public static Path getPath(Scanner input) {
        Path p = Paths.get(System.getProperty("user.home"));
        while (true) {
            System.out.println(p);
            try (DirectoryStream<Path> stream = Files.newDirectoryStream(p)) {
                int i = 1;
                ArrayList<String> foldersList = new ArrayList<>();
                for (Path file: stream) {
                    String check = file.getFileName().toString();
                    if (check.contains(".xml") || !check.contains(".")) {
                        System.out.printf("[%d] %s%n", i++, check);
                        foldersList.add(check);
                    }
                }
                System.out.printf("~[%d] return%n", i++);
                System.out.printf("~[%d] select XML file or create a new one%n", i);
                System.out.println("chose option or directory: ");
                int option = input.nextInt();
                if (option < 1 || option > i)
                    System.err.println("error");
                else if (option < i-1) {
                    if (Files.isDirectory(p.resolve(foldersList.get(option-1))))
                        p = p.resolve(foldersList.get(option-1));
                    else System.err.println("it isn't a directory");
                } else if (option == i-1)
                    p = p.getRoot().resolve(p.getParent());
                else {
                    System.out.println("enter filename in format name.xml:");
                    input.nextLine();
                    String filename = input.nextLine();
                    if (filename.contains(".xml")) {
                        return p.resolve(filename);
                    } else System.err.println("wrong file format");
                }
            } catch (IOException | DirectoryIteratorException x) {
                System.err.println("error!");
                return null;
            }
        }
    }
}

```

**Check**



```

package lab_5;

import java.util.regex.Pattern;

public class Check {

    static boolean nameCheck(String in)
    {
        if(Pattern.matches("^\\D{1,}$" , in ) )
            return true;
        else return false;
    }

    static boolean adressCheck(String in)
    {
        if(Pattern.matches("^\\w{1,}$" , in ) )
            return true;
        else return false;
    }

    static boolean regDateCheck(String in)
    {
        if(Pattern.matches("^([0-2][0-9]||3[0-1])/(0[0-9]||1[0-2])/([0-9][0-9])?[0-9][0-9]$" , in))
            return true;
        else
            return false;
    }
}

```



## Висновки

На цій лабораторній роботі, я розробив консольну програму на java для реалізації діалогового режиму роботи з користувачем. Вирішив прикладну задачу з використанням масивів , рядків , класів , ітераторів і серіалізаторів.