

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту



Лабораторна робота N5

З дисципліни

«Алгоритмізація та програмування»

Виконав:

Студент групи КН-108

Воробель Адріан

Викладач:

Грабовська Н.Р.

Мета:

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

1. Вимоги

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.

1.1 Розробник

Воробель Адріан, КН-108, номер варіанту індивідуального завдання – 5.

1.2 Задача

Опис програми:

Клас-контейнер з окремим класом для обробки текстів регулярними виразами.

2.1 Засоби ООП

Клас-контейнер виконаний LinkedList-ом, domain-клас та його підклас, Main, клас Patterns ,імплементация.

2.2 Ієрархія та структура класів

Клас LinkedList – клас з основним функціоналом, реалізує різного виду методи.

Клас TradePoint – domain-клас, та його підклас WorkTime.

Main – клас-меню.

Patterns – клас з методами для обробки тексту.

```
package com.company;

public class TradePoint {

    WorkTime time = new WorkTime();

    public String specialization;
    public String name;
    public String address;
    public String[] numbers;
    public WorkTime WorkTime;

    public String[] getNumbers() { return numbers; }
    public void setNumbers(String[] numbers) { this.numbers = numbers; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public String getAddress() { return address; }
    public void setAddress(String address) { this.address = address; }

    public String getSpecialization() { return specialization; }
    public void setSpecialization(String specialization) { this.specialization = specialization; }

    public WorkTime getWorkTime() { return WorkTime; }
    public void setWorkTime(WorkTime workTime) { WorkTime = workTime; }
}
```

```
package com.company;

public class WorkTime {

    public String MONDAY;
    public String TUESDAY;
    public String WEDNESDAY;
    public String THURSDAY;
    public String FRIDAY;
    public String SATURDAY;
    public String SUNDAY;

    public String getMONDAY() { return MONDAY; }
    public void setMONDAY(String MONDAY) { this.MONDAY = MONDAY + "MONDAY"+"\\n"; }

    public String getTUESDAY() { return TUESDAY; }
    public void setTUESDAY(String TUESDAY) { this.TUESDAY = TUESDAY + "TUESDAY"+"\\n"; }

    public String getWEDNESDAY() { return WEDNESDAY; }
    public void setWEDNESDAY(String WEDNESDAY) { this.WEDNESDAY = WEDNESDAY + "WEDNESDAY"+"\\n"; }

    public String getTHURSDAY() { return THURSDAY; }
    public void setTHURSDAY(String THURSDAY) { this.THURSDAY = THURSDAY + "THURSDAY"+"\\n"; }

    public String getFRIDAY() { return FRIDAY; }
    public void setFRIDAY(String FRIDAY) { this.FRIDAY = FRIDAY+"FRIDAY"+"\\n"; }

    public String getSATURDAY() { return SATURDAY; }
    public void setSATURDAY(String SATURDAY) { this.SATURDAY = SATURDAY+"SATURDAY"+"\\n"; }

    public String getSUNDAY() { return SUNDAY; }
    public void setSUNDAY(String SUNDAY) { this.SUNDAY = SUNDAY+ "SUNDAY"+"\\n"; }
}
```

```

package com.company;

import java.beans.XMLDecoder;
import java.beans.XMLEncoder;
import java.io.*;
import java.util.Iterator;

public class LinkedList<T> implements Iterable<T>, Serializable {

    TradePoint shop = new TradePoint();

    private class Node implements Serializable {
        private T data;
        private Node next;
        private T getData() { return data; }
        private Node getNext() { return next; }
        private void setData(T data) { this.data = data; }
        private void setNext(Node next) { this.next = next; }
    }

    private Node root;

    public LinkedList() { root = new Node(); }

    public void add(T data) {
        Node last = root;
        while (last.getNext() != null)
            last = last.getNext();
        last.setNext(new Node());
        last.getNext().setData(data);
    }

    public T get(int index) {
        Node result = root;
        for (int i = -1; i < index; i++) {
            if (result.getNext() == null)
                return null;
            else
                result = result.getNext();
        }
        return result.getData();
    }

    public boolean remove(int index) {
        Node result = root;
        if (size() != 0 && index >= 0) {
            for (int i = -1; i < index - 1; i++)
                if (result.getNext() == null)
                    return false;
            else
                result = result.getNext();
            // result.getNext().setNext(null);
        }
    }
}

```

```

Main.java x LinkedList.java x TradePoint.java x WorkTime.java x
97      System.out.println("friday");
98      shop.time.setFRIDAY(sc.nextLine() + " ");
99      System.out.println("saturday");
100     shop.time.setSATURDAY(sc.nextLine() + " ");
101     System.out.println("sunday");
102     shop.time.setSUNDAY(sc.nextLine() + " ");
103
104     shop.setWorkTime(shop.time);
105
106     System.out.print("Number:");
107
108     String n = "";
109     String[] numbers = new String[0];
110     while (!n.equals("-")) {
111         n = sc.nextLine();
112         switch (n) {
113             case "+":
114                 numbers = Arrays.copyOf(numbers, newlength: numbers.length + 1);
115                 numbers[numbers.length - 1] = sc.nextLine() + "\n";
116                 System.out.println("to add number print '+', if you inputed all numbers - print '-!');
117                 break;
118             }
119         }
120         shop.setNumbers(numbers);
121         list.add(shop);
122     }
123 }
124 break;
125 case "remove"://////////////////////////////////////
126     System.out.print("index of element:");
127     sc = new Scanner(System.in);
128     int i = sc.nextInt();
129     list.remove(i);
130     break;
131 case "clear"://////////////////////////////////////
132     list.clear();
133     break;
134 case "toArray"://////////////////////////////////////
135     list.toArray();
136     break;
137 case "toString"://////////////////////////////////////
138     System.out.println(list.toString());
139     break;
140 case "contains"://////////////////////////////////////
141     System.out.print("index of element:");
142     sc = new Scanner(System.in);
143     int ii = sc.nextInt();
144     try{
145         System.out.println("founded: "+list.get(ii));
146     }catch (NullPointerException e){
147         System.out.println("no such element");
148     }

```

```
package com.company;
import java.util.regex.*;

public class Patterns {

    boolean matchData(String data, Pattern pattern){
        Matcher matcher = pattern.matcher(data);
        if(matcher.matches())
            return true;
        else
            return false;
    }

    boolean confirmed(String string, String input){
        if(string.equals(input))
            return true;
        else
            return false;
    }
}
```

3. Варіанти використання

Програма може використовуватись для перевірки правильності вводу даних.

ВИСНОВКИ

У ході роботи я дізнався , що таке регулярні вирази, та навчився їх застосовувати на практиці.