

Звіт

Лабораторна робота 12. Регулярні вирази. Обробка тексту

Мета роботи: Ознайомлення з принципами використання регулярних виразів для обробки тексту.

ВИМОГИ

1. Використовуючи програми рішень попередніх задач, продемонструвати ефективне (оптимальне) використання регулярних виразів при вирішенні прикладної задачі.
2. Передбачити можливість незначної зміни умов пошуку.
3. Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах.

1.1. Розробник: Момот Роман Євгенійович, КІТ119а, варіант №14.

2. ОПИС ПРОГРАМИ

2.1. Засоби ООП: клас, метод класу, поле класу.

2.2. Ієрархія та структура класів: один публічний клас Main, публічний клас Event, у полях якого є час початку події, тривалість, адреса події, імена людей, опис події, гетери, сетери, конструктор класу та метод виведення даних класу. Також є клас Node, який виконує роль покажчика на елемент і клас MyContainer, який містить покажчик на головний елемент та методи обробки масиву елементів.

2.3. Важливі фрагменти програми:

```
private static MyContainer<Event> auto(MyContainer<Event> arr) {  
    Pattern pattern;  
    Matcher matcher;  
  
    System.out.println("\nSize of container: " + arr.getSize());  
    System.out.println("Adding elements...");  
  
    ArrayList<String> people = new ArrayList<String>();
```

```

        people.add("John");
        people.add("Bill");
        people.add("Івасик");

        Event event = new Event(new GregorianCalendar(2019,4,28), 120,
"Харьков, ул. Заозёрская 39",
            people, "Лучшая тусовка");
        arr.add(event);

        people = new ArrayList<String>();
        people.add("Roman");
        people.add("Dmitriy");

        event = new Event(new GregorianCalendar(2005,12,15), 30,
"Харьков, пр. Тракторостроителей",
            people, "Скучно");
        arr.add(event);

        System.out.println("Size of container: " + arr.getSize());

        System.out.println("\nOutputing data with toArray:");
        Object[] tempArr = arr.toArray();
        for (int i = 0; i < tempArr.length; i++) {
            System.out.println(i+1 + " ");
            ((Event)tempArr[i]).outputData();
            System.out.println( );
        }

        System.out.println("Is container empty?");
        System.out.println(arr.isEmpty());

```

```

        System.out.println("\nReading data from file...");

        try(BufferedReader reader = new BufferedReader(new
InputStreamReader(new FileInputStream("data.txt"), "UTF-8"))){

            int i = 0;

            String str;

            String[] data;

            String[] patterns = {"^(?!^0)\\d{4}$", "^[1-9]|([1][0-2])$",
"^[1-9]|([12][0-9])|([3][01])$",

                                "^[0-9]|([1][0-9])|([2][0-4])$", "^[0-9]|([1-5][0-
9])|([6][0])$", "^(?!^0)\\d{1,9}$",

                                "^[([A-Z][a-z]+)|([A-Z][a-z]*)([\\s][A-Z][a-
z]*)$"};

            while((str = reader.readLine()) != null) {

                data = str.split("\\s*(;|\\s*");

                for(i = 2; i < 9; i++) {

                    pattern = Pattern.compile(patterns[i-2]);

                    matcher = pattern.matcher(data[i]);

                    if(!matcher.matches()) {

                        System.out.println("Wrong data in line.
Moving to next line.");

                        i = 10;

                    }

                }

            }

            if(i == 11) {

                continue;

```

```

    }

    people.clear();
    pattern = Pattern.compile(patterns[6]);
    i--;
    for (; i < data.length; i++) {
        matcher = pattern.matcher(data[i]);

        if(!matcher.matches()) {
            System.out.println("Wrong name " + data[i]
+ " in line. It wont be added.");
        }
        else {
            people.add(data[i]);
        }
    }

    arr.add(new Event(new
GregorianCalendar(Integer.parseInt(data[2]),Integer.parseInt(data[3]),Integer.parse
Int(data[4]),Integer.parseInt(data[5]),Integer.parseInt(data[6]),0),
Integer.parseInt(data[7]),data[0],people,data[1]));
    }
}
catch(IOException ex) {
    System.out.println(ex.getMessage());
}

System.out.println("\nOutputing data with toArray:");
tempArr = arr.toArray();
for (int i = 0; i < tempArr.length; i++) {

```

```
        System.out.println(i+1 + " ");
        ((Event)tempArr[i]).outputData();
        System.out.println( );
    }
```

```
    Pattern pattYear = Pattern.compile("^(2019)|(2018)|(2020)$");
    Pattern pattCity = Pattern.compile("Харьков(.*)");
    Pattern pattDuration = Pattern.compile("^[2][5-9]+|([3-9][0-9]+)|([1-9][0-9]{2,})$");
    Matcher matcher1, matcher2, matcher3;

    System.out.println("Outputting array with regex...\n");

    for(var i : arr) {
        matcher1 =
pattYear.matcher(Integer.toString(i.getStartTime().get(Calendar.YEAR)));
        matcher2 = pattCity.matcher(i.getAddress());
        matcher3 =
pattDuration.matcher(Integer.toString(i.getDuration()));

        System.out.println( );

        if(matcher1.matches() && matcher2.matches() &&
matcher3.matches()) {
            i.outputData();
        }
    }

    return arr;
}
```

```

private static MyContainer<Event> menu(MyContainer<Event> arr) {
    Scanner scan = new Scanner(System.in);
    boolean stop = false;
    int choice, choice2;

    ArrayList<String> people = new ArrayList<String>();
    people.add("John");
    people.add("Bill");
    people.add("Івасик");

    Event evToCompare = new Event(new
GregorianCalendar(2002,3,28), 120, "ул. Революции",
        people, "Pest party ever");
    arr.add(evToCompare);

    do {
        System.out.println("What to do?");
        System.out.println("1. Output data");
        System.out.println("2. Add element");
        System.out.println("3. Delete element");
        System.out.println("4. Is empty?");
        System.out.println("5. Serialization");
        System.out.println("6. Deserialization");
        System.out.println("7. Sort data");
        System.out.println("8. Terminate program");
        System.out.println("=====");
        System.out.print("Your choice: ");
        choice = scan.nextInt();

        switch(choice) {

```

case 1:

```
System.out.println("\nChoose the output method");
System.out.println("1. Using foreach");
System.out.println("2. Using toArray");
System.out.println("3. Find element by criteria");
System.out.println("4. Return");
System.out.println("=====");
System.out.print("Your choice: ");
choise2 = scan.nextInt();
System.out.println( );
```

```
switch(choise2) {
```

case 1:

```
    if(arr.getSize() > 0){
        for(var i : arr) {
            i.outputData();
            System.out.println( );
        }
        System.out.println( );
    }
    else {
        System.out.println("Array is empty.\n");
    }
    break;
```

case 2:

```
    if(arr.getSize() > 0) {
        Object[] tempArr = arr.toArray();
        for (int i = 0; i < tempArr.length; i++) {
```

```

        System.out.println(i + " ");
        ((Event)tempArr[i]).outputData();
    }
}
else {
    System.out.println("Array is empty.");
}
break;
case 3:
    if(arr.getSize() == 0) {
        System.out.println("Array is empty.\n");
        break;
    }

    Pattern pattYear;
    Pattern pattCity;
    Pattern pattDuration = Pattern.compile("^([2][5-9]+)|([3-9][0-9]+)|([1-9][0-9]{2,})$");
    Matcher matcher1, matcher2, matcher3;

    String regex = "^(?)(?)(?)$";

    System.out.println("Task: Знайти всі конференції,
що пройшли\n"
        + "-за останні три роки "
        + "\n-в Харкові та області "
        + "\n-з тривалістю не менше доби.");

    System.out.println("Передбачити можливість
незначної зміни умов пошуку.");

```



```

        System.out.print("\nEnter the year: ");
        int year = scan.nextInt();
        for (int i = 0; i < 3; i++) {
            regex = regex.substring(0,regex.indexOf('?'))
+ Integer.toString(year - i) + regex.substring(regex.indexOf('?') + 1,
regex.length());
        }
        pattYear = Pattern.compile(regex);

        System.out.print("Enter the city: ");
        scan.nextLine();
        String city = scan.nextLine();
        city = city.concat("(.*)");
        pattCity = Pattern.compile(city);

        for(var i : arr) {
            matcher1 =
pattYear.matcher(Integer.toString(i.getStartTime().get(Calendar.YEAR)));
            matcher2 = pattCity.matcher(i.getAddress());
            matcher3 =
pattDuration.matcher(Integer.toString(i.getDuration()));

            System.out.println( );
            if(matcher1.matches() &&
matcher2.matches() && matcher3.matches()) {
                i.outputData();
            }
        }

        break;

```

```

        case 4:
            System.out.println("\nReturning\n");
            break;

        default:
            System.out.println("You've entered the wrong
number");
            break;
    }
    break;

    case 2:
        Event newEvent = inputNewEvent();
        arr.add(newEvent);

        break;

    case 3:
        if(arr.getSize() > 0) {
            System.out.print("Enter the index of element: ");
            chose = scan.nextInt();

            arr.delete(chose);
        } else {
            System.out.println("Array is empty.");
        }
        break;

```

case 4:

```
if(arr.isEmpty()) {  
    System.out.println("Array is empty.");  
} else {  
    System.out.println("Array isn't empty.");  
}  
break;
```

case 5:

```
System.out.println("\nChoose the method");  
System.out.println("1. Standard serialization");  
System.out.println("2. XML serialization");  
System.out.println("3. Return");  
System.out.println("=====");  
System.out.print("Your choice: ");  
choise2 = scan.nextInt();
```

```
switch(choise2) {
```

case 1:

```
    scan.nextLine();  
    System.out.print("Enter the name of file: ");  
    String filename = scan.nextLine();
```

```
    if (filename.indexOf(".ser") == -1) {  
        filename += ".ser";  
    }
```

```

        try(ObjectOutputStream oos = new
ObjectOutputStream(new BufferedOutputStream(new
FileOutputStream(filename)))){

            oos.writeObject(arr);

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

        }catch(Exception ex){

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

        }

        break;

    case 2:

        scan.nextLine();

        System.out.print("Enter the name of file: ");
        filename = scan.nextLine();

        if (filename.indexOf(".xml") == -1) {
            filename += ".xml";
        }

        try(XMLEncoder encoder = new
XMLEncoder(new BufferedOutputStream(new FileOutputStream(filename)))){

            encoder.writeObject(arr);

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

        }

        catch(Exception ex){

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

```

```

        }
        break;

case 3:
    break;

default:
    System.out.println("You've entered the wrong
command.");

    break;
}

break;

case 6:
    System.out.println("\nChoose the method");
    System.out.println("1. Standard deserialization");
    System.out.println("2. XML deserialization");
    System.out.println("3. Return");
    System.out.println("=====");
    System.out.print("Your choice: ");
    choise2 = scan.nextInt();

    switch(choise2) {
case 1:
        scan.nextLine();
        System.out.print("Enter the name of file: ");
        String filename = scan.nextLine();

```

```

        if (filename.indexOf(".ser") == -1) {
            filename += ".ser";
        }

        try(ObjectInputStream oos = new
ObjectInputStream(new BufferedInputStream(new FileInputStream(filename)))){
            arr.clear();
            arr = (MyContainer<Event>)
oos.readObject();

            System.out.println("\nSerialization
successful.");

        }catch(Exception ex){
            System.out.println(ex.getMessage());
        }

        break;

    case 2:
        scan.nextLine();
        System.out.print("Enter the name of file: ");
        filename = scan.nextLine();

        if (filename.indexOf(".xml") == -1) {
            filename += ".xml";
        }

        try(XMLDecoder decoder = new
XMLDecoder(new BufferedInputStream(new FileInputStream(filename)))){
            arr.clear();

```

```

arr = (MyContainer<Event>)
decoder.readObject();

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

} catch(IOException ex){
    System.out.println( );
}
break;

case 3:
    break;

default:
    System.out.println("You've entered the wrong
command.");

    break;
}

break;

case 7:
    System.out.println("\nChoose sorting field:");
    System.out.println("1. Sort by event date");
    System.out.println("2. Sort by event length");
    System.out.println("3. Sort by number of people");
    System.out.println("4. Return");

System.out.println("=====");
System.out.print("Your choice: ");
choise2 = scan.nextInt();

```

```

switch(choise2) {
case 1:
    arr.sort(new EventDateComparator());
    System.out.println("\nData sorted\n");
    break;

case 2:
    arr.sort(new EventLengthComparator());
    System.out.println("\nData sorted\n");
    break;

case 3:
    arr.sort(new EventPeopleNumberComparator());
    System.out.println("\nData sorted\n");
    break;

case 4:
    System.out.println("\nReturning\n");
    break;

default:
    System.out.println("\nYou have entered the wrong
number.\n");
    break;
}
break;

case 8:
    System.out.println("Terminating the program.");

```



```
        stop = true;
        break;

    default:
        System.out.println("You have entered the wrong
number.");
        break;
    }
}while(!stop);

scan.close();
return arr;
}
```

Результат роботи програми

```
What to do?
1. Output data
2. Add element
3. Delete element
4. Is empty?
5. Serialization
6. Deserialization
7. Sort data
8. Terminate program
=====
Your choice: 1

Choose the output method
1. Using foreach
2. Using toArray
3. Find element by criteria
4. Return
=====
Your choice: 3

Task: Знайти всі конференції, що пройшли
-за останні три роки
-в Харкові та області
-з тривалістю не менше доби.
Передбачити можливість незначної зміни умов пошуку.

Enter the year: 2020
Enter the city: Kharkiv
|

Event start time: Sun Apr 28 19:30:00 EEST 2019
Duration of the event (in minutes): 200
Event address: Kharkiv
Event description: 200
List of participants:
1. Dmitry
2. Ivan
```

Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з регулярними виразами у тексті.

Програма протестована, виконується без помилок.