Ministry of Science and Higher Education of the Russian Federation
Federal State Autonomous Educational Institution of Higher Education "South Ural State University"
(National Research University)
Institute of Natural Sciences and Mathematics
Faculty of Mathematics, Mechanics and Computer Technologies
Department of Applied Mathematics and Programming

Lab Report 4
in the discipline "Programming in the Java language"

Author of the work
Sychev Artemy ET-212
_____/NAME.
"____"_____2024

Supervisor of the work,
_____/Olenchikova T.Y.
"____"_____2024

Chelyabinsk 2024

Purpose of the work

Regular expressions. Lambda

Learn how to work with strings. Learn how to use labda expressions to work with functional interfaces.

Task

1. Write a regular expression that will check a simple e-mail address (for example, test@test.test).

2. Write a program that uses regular expressions to replace the word "function" in the text with "<mark>function</mark>".

3. Write a function that uses regular expressions to break the text into individual words and find the frequency of occurrence of individual words.

4. Write a functional interface with a method that takes a number and returns a Boolean value. Write an implementation of such an interface in the form of a lambda expression that returns true if the passed number is evenly divisible by 13.

5. Write a functional interface with a method that takes two strings and returns the same string. Write an implementation of such an interface in the form of a lambda that returns the longer string.

6. Write a functional interface with a method that takes three fractional numbers: a, b, c and returns a fractional number as well. Write an implementation of such an interface as a lambda expression that returns a discriminant. Who forgot, D = b^2 – 4ac.

7. Using the functional interface from Task 6, write a lambda expression that returns the result of the operation a * b^c

Program code:

```
import java.util.HashMap;
import java.util.Map;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class ReComparator {

    public static boolean Email(String text){
        String mask = "([a-zA-Z0-9]+[-_.]? [a-zA-Z0-9]+)+@[a-zA-Z]+. [a-zA-Z]+";
        boolean result = text.matches(mask);
        return result;
    }

    public static String Replacer(String text){
        String replacement = "\\b(function)\\b";

        Pattern pattern = Pattern.compile(replacement);
        Matcher matcher = pattern.matcher(text);

        return matcher.replaceAll("<mark>$1<mark>");
    }

    public static Map<String, Integer> Splitter(String text) {
        Map<String, Integer> wordFrequency = new HashMap<>();
        Pattern pattern = Pattern.compile("\\b\\w+\\b");
        Matcher matcher = pattern.matcher(text);

        while (matcher.find()) {
            String word = matcher.group();
            wordFrequency.put(word, wordFrequency.getOrDefault(word, 0) + 1);
        }

        return wordFrequency;
    }

}

public interface checkNumber {
    boolean check(int num);
}

public interface longerString {
    String biggerString(String a, String b);
}

public interface IFunction {
    double calculate(double a, double b, double c);
}
```

Example of the program execution:

```
import java.util.Map;

public class Main {
    public static void main(String[] args) {
        /*regular expressions

         */
        System.out.println(ReComparator.Email("artemsm174@gmail.com"));
        System.out.println(ReComparator.Email("artemsm..174@gmail.com"));
        System.out.println(ReComparator.Replacer("function is good"));
        Map<String, Integer> words = ReComparator.Splitter("My, crew, is, on, fired, up,
I, loveu, u, Ich, Liebe, Dich");
        for (Map.Entry<String, Integer> entry : words.entrySet()) {
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }
        /*
         lambdas
         */
        checkNumber checker = (number) -> number % 13 == 0;
        longerString compare = (a, b) -> a.length() > b.length() ? a : b;
        IFunction discriminant = (a, b, c) -> b * b - 4 * a * c;
        IFunction function1 = (a, b, c) -> a * Math.pow(b, c);

        System.out.println(checker.check(13));
        System.out.println(compare.biggerString("123", "12345"));
        System.out.println(discriminant.calculate(1, -4, 4));
        System.out.println(function1.calculate(2, 2, 2));

    }
}
```

```
true
false
<mark>function<mark> is good
fired: 1
loveu: 1
u: 1
I: 1
is: 1
Dich: 1
up: 1
Liebe: 1
My: 1
crew: 1
on: 1
Ich: 1
true
12345
0.0
8.0

Process finished with exit code 0
```

Conclusion: In this lab, I showed how to work with regular expressions in the Java language. In addition, I showed how to work with interfaces and the implementation of functional interfaces through lambda expressions