**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа №5

По дисциплине «Информационные технологии и программирование»

Выполнил: Студент группы

БВТ2203

Попов Александр

Москва

2023

**Цель работы:** Знакомство с регулярными выражениями в ЯП Java.

**Задание 1: Поиск всех чисел в тексте**

Необходимо написать программу, которая будет искать все числа в заданном тексте и выводить их на экран. При этом программа должна использовать регулярные выражения для поиска чисел и обрабатывать возможные ошибки.

**Задание 2: Проверка корректности ввода пароля**

Необходимо написать программу, которая будет проверять корректность ввода пароля. Пароль должен состоять из латинских букв и цифр, быть длиной от 8 до 16 символов и содержать хотя бы одну заглавную букву и одну цифру. При этом программа должна использовать регулярные выражения для проверки пароля и обрабатывать возможные ошибки.

**Задание 3: Замена всех ссылок на гиперссылки**

Необходимо написать программу, которая будет заменять все ссылки в заданном тексте на гиперссылки. При этом программа должна использовать регулярные выражения для поиска ссылок и замены и обрабатывать возможные ошибки.

**Задание 4: Проверка корректности ввода IP-адреса**

Необходимо написать программу, которая будет проверять корректность ввода IP-адреса. IP-адрес должен состоять из 4 чисел, разделенных точками, и каждое число должно быть в диапазоне от 0 до 255. При этом программа должна использовать регулярные выражения для проверки IP-адреса и обрабатывать возможные ошибки.

**Задание 5: Поиск всех слов, начинающихся с заданной буквы**

Необходимо написать программу, которая будет искать все слова в заданном тексте, начинающиеся с заданной буквы, и выводить их на экран. При этом программа должна использовать регулярные выражения для поиска слов и обрабатывать возможные ошибки.

**Ход работы**

import java.util.regex.Matcher;

import java.util.regex.Pattern;

*public* *class* Main {

*public* *static* void main(String[] *args*) {

        numberFinder("19.99abc55 19,99");

        System.out.println(correctPassword("12345@@DDDddd"));

        System.out.println(replaceLinks("https://www.youtube.com/ hi"));

        System.out.println(isValidIPAddress("255.128.0.1"));

        myStartsWith("Aaa aaa bb abd", 'A');

    }

*public* *static* void numberFinder(String *text*) {

        Pattern pattern = Pattern.compile("-?\\d+([\\.\\,]\\d+)?");

        Matcher matcher = pattern.matcher(*text*);

*while* (matcher.find()) {

            System.out.println(matcher.group());

        }

    }

*public* *static* boolean correctPassword(String *pass*) {

        Pattern pattern = Pattern.compile("^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\\d)[a-zA-Z\\d]{8,16}$");

        Matcher matcher = pattern.matcher(*pass*);

*return* matcher.matches();

    }

*public* *static* String replaceLinks(String *link*) {

        Pattern pattern = Pattern.compile("\\bhttp[s]?://\\S+\\b"); *// \\S непробельный символ*

        Matcher matcher = pattern.matcher(*link*);

        StringBuffer result = *new* StringBuffer();

*while* (matcher.find()) {

            String newlink = matcher.group();

            matcher.appendReplacement(result, "<a href='" + newlink + "'>" + newlink + "</a>");

        }

        matcher.appendTail(result);

*return* result.toString();

    }

*public* *static* boolean isValidIPAddress(String *ip*) {

        String str =

        "^([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\." +

        "([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\." +

        "([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\." +

        "([01]?\\d\\d?|2[0-4]\\d|25[0-5])$";

        Pattern p = Pattern.compile(str);

        Matcher m = p.matcher(*ip*);

*return* m.matches();

    }

*public* *static* void myStartsWith(String *text*, char *letter*) {

        Pattern pattern = Pattern.compile("\\b" + *letter* + "\\w\*\\b");

        Matcher matcher = pattern.matcher(*text*);

*while* (matcher.find()) {

            System.out.println(matcher.group());

        }

    }

}

Файл 1. Main.java

В этой файле метод NumberFinder ищет числа в строке и выводит их на экран. Метод correctPassword проверяет корректность пароля и возвращает true или false. Метод replaceLinks меняет ссылки на гиперссылки, добавляя html-тег <a/> заместо ссылки. Метод IsValidIPAddress проверяет валидность введенного ip-адреса. Метод myStartsWith находит слова, начинающиеся на переданную букву, и выводит их на экран.

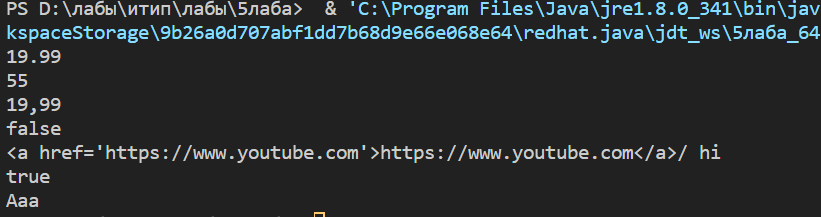
****

Рис. 1. Результат выполнения программы.

**Выводы**

1. Я научился работать с регулярными выражениями в ЯП Java.