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

Выполнил: Студент группы

БВТ2203

Георгян Сергей

Москва

2023

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

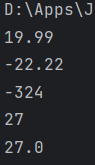
Необходимо написать программу, которая будет искать все слова в заданном тексте, начинающиеся с заданной буквы, и выводить их на экран. При этом программа должна использовать регулярные выражения для поиска слов и обрабатывать возможные ошибки.

Ход работы:

**Задание 1**

import java.util.regex.\*;  
  
public class NumberFinder {  
 public static void main(String[] args) {  
 String text = "The price of the pro22duct is $19.99 -22.22 43.02.2 aaa22 -324 27 27.0 27a";  
  
 Pattern digPattern = Pattern.*compile*("(?<=(\\s|\\$|^))-?\\d+(\\.\\d+)?(?=\\s|$)");  
 Matcher digMatcher = digPattern.matcher(text);  
 while (digMatcher.find()) {  
 System.*out*.println(digMatcher.group());  
 }  
 }  
}

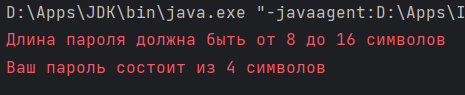
Файл 1 – программа для поиска всех чисел в строке

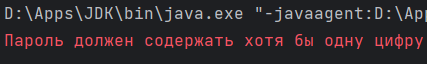
 Рис.1 – вывод программы

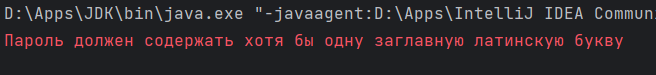
**Задание 2**

import java.util.ArrayList;  
import java.util.regex.\*;  
  
class PassError extends Exception {  
 public PassError(String message) {  
 super(message);  
 }  
}  
  
public class PasswordCheck {  
 public static void main(String[] args) {  
 String pass = "@eregacooler2212";  
  
 if (args.length != 0) {  
 pass = String.*join*(" ", args);  
 }  
  
 try {  
 *passCheck*(pass);  
 System.*out*.println("Пароль введен верно");  
 } catch (PassError e) {  
 System.*err*.println(e.getMessage());  
 }  
 }  
  
 public static void passCheck(String pass) throws PassError {  
 ArrayList<String> symbols = new ArrayList<>();  
  
 Pattern symbolsPatern = Pattern.*compile*("\\W");  
 Matcher symbolsMatcher = symbolsPatern.matcher(pass);  
  
 Pattern digitPatern = Pattern.*compile*("\\d");  
 Matcher digitMatcher = digitPatern.matcher(pass);  
  
 Pattern letterPatern = Pattern.*compile*("[A-Z]");  
 Matcher letterMatcher = letterPatern.matcher(pass);  
  
 while (symbolsMatcher.find()) {  
 symbols.add(symbolsMatcher.group());  
 }  
  
 if (pass.length() < 8 || pass.length() > 16) {  
 throw new PassError("Длина пароля должна быть от 8 до 16 символов\n" +  
 "Ваш пароль состоит из " + pass.length() + " символов");  
 }  
  
 if (!symbols.isEmpty()) {  
 throw new PassError("Пароль не должен содержать любые знаки и пробел, кроме латинских букв и цифр\n" +  
 "Ваш пароль содержит такие символы как: " + String.*join*(", ", symbols));  
 }  
  
 if (!digitMatcher.find()) {  
 throw new PassError("Пароль должен содержать хотя бы одну цифру");  
 }  
  
 if (!letterMatcher.find()) {  
 throw new PassError("Пароль должен содержать хотя бы одну заглавную латинскую букву");  
 }  
 }  
}

Файл 2 – программа для проверки верности пароля







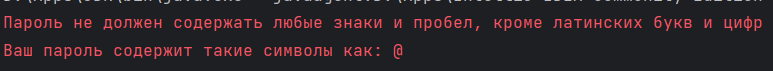


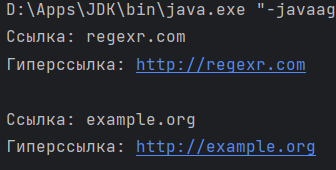


Рис.(2-6) – варианты вывода программы

**Задание 3**

import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class HyperlinkReplacer {  
 public static void main(String[] args) {  
 String text = "Это текст с ссылкой: regexr.com и еще одной example.org.";  
  
 String regex = "((\\b\\S+\\.org\\b)|(\\b\\S+\\.com\\b))";  
  
 Pattern pattern = Pattern.*compile*(regex);  
 Matcher matcher = pattern.matcher(text);  
  
 while (matcher.find()) {  
 System.*out*.println("Ссылка: " + matcher.group());  
 System.*out*.println("Гиперссылка: http://" + matcher.group() + "\n");  
 }  
 }  
}

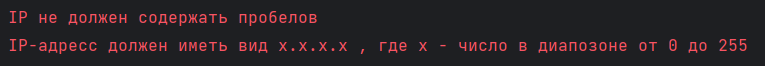
Файл 3 – программа для замены ссылок на гиперссылки

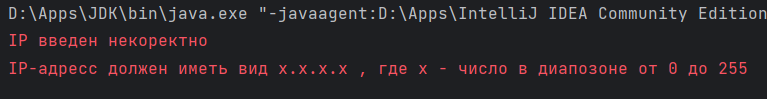
 Рис.7 – вывод программы

**Задание 4**

import java.util.regex.\*;  
  
class IPError extends Exception {  
 public IPError(String message) {  
 super(message);  
 }  
}  
  
public class IPCheck {  
 public static void main(String[] args) {  
 String ip = "110.12.12.234";  
  
 if (args.length != 0) {  
 ip = String.*join*(" ", args);  
 }  
  
 try {  
 *ipCheck*(ip);  
 System.*out*.println("IP введен верно");  
 } catch (IPError e) {  
 System.*err*.println(e.getMessage());  
 }  
 }  
  
 public static void ipCheck(String ip) throws IPError {  
  
 Pattern spacePatern = Pattern.*compile*("\\s");  
 Matcher spaceMatcher = spacePatern.matcher(ip);  
  
 Pattern correctPatern = Pattern.*compile*("\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}");  
 Matcher correctMatcher = correctPatern.matcher(ip);  
  
 Pattern numPatern = Pattern.*compile*("\\d+");  
 Matcher numMatcher = numPatern.matcher(ip);  
  
  
  
  
 if (spaceMatcher.find()) {  
 throw new IPError("IP не должен содержать пробелов\n" +  
 "IP-адресс должен иметь вид x.x.x.x , где x - число в диапозоне от 0 до 255");  
 }  
  
 if (!correctMatcher.find()) {  
 throw new IPError("IP введен некоректно\n" +  
 "IP-адресс должен иметь вид x.x.x.x , где x - число в диапозоне от 0 до 255");  
 }  
  
 if (correctMatcher.group().length() != ip.length()) {  
 throw new IPError("IP введен некоректно\n" +  
 "IP-адресс должен иметь вид x.x.x.x , где x - число в диапозоне от 0 до 255");  
 }  
  
 while (numMatcher.find()) {  
 if (Integer.*parseInt*(numMatcher.group()) >= 255) {  
 throw new IPError("Отдельное число IP-адресса не должно превышать 254\n" +  
 "В вашем варианте присутсвует число: " + numMatcher.group());  
 }  
 }  
  
  
  
 }  
}

Файл 4 – программа для проверки верности IP-адресса





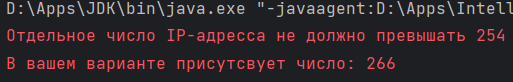


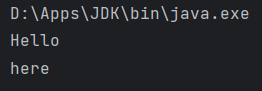


Рис.(8-11) – варианты вывода программы

**Задание 5**

import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class WordFinder {  
 public static void main(String[] args) {  
 String text = "Hello world, I am igor so proud to be behind here!";  
 String targetLetter = "h";  
 String regex = "(?<=(\\s|^))(" + targetLetter.toUpperCase() + "|" + targetLetter.toLowerCase() + ")[A-z]+\\b" +  
 "|(?<=(\\s|^))(" + targetLetter.toUpperCase() + "|" + targetLetter.toLowerCase() + ")\\b";  
  
 Pattern wordPatern = Pattern.*compile*(regex);  
 Matcher wordMatcher = wordPatern.matcher(text);  
  
 while (wordMatcher.find()) {  
 System.*out*.println(wordMatcher.group());  
 }  
 }  
}

Файл 5 – программа для вывода слов начинающихся на определенную букву

 Рис.12 – вывод программы

**Вывод:**

Я научился пользоваться регулярными выражениями java