|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №17-18**

Выполнил студент группы ИНБО-02-20 Маркарянц Р.Д.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

1. Написать регулярное выражение, определяющее является ли данная строка строкой "abcdefghijklmnopqrstuv18340" или нет.

– пример правильных выражений: abcdefghijklmnopqrstuv18340.

– пример неправильных выражений: abcdefghijklmnoasdfasdpqrstuv18340.

1. Написать регулярное выражение, определяющее является ли заданная строка правильным MAC-адресом.

– пример правильных выражений: aE:dC:cA:56:76:54.

– пример неправильных выражений: 01:23:45:67:89:Az.

1. Есть текст со списками цен. Извлечь из него цены в USD, RUR, EU.

– пример правильных выражений: 23.78 USD.

– пример неправильных выражений: 22 UDD, 0.002 USD.

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

package ru.mirea.inbo0220.markaryants.pr17;  
import java.util.Scanner;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Task1 {  
 static void CheckString(String s) {  
 Pattern p = Pattern.*compile*("qwertyuiop");  
 Matcher m =p.matcher(s);  
 boolean b = m.matches();  
 if (b){  
 System.*out*.println("Данная сторка является строкой qwertyuiop");  
 } else {  
 System.*out*.println("Данная сторка не является строкой qwertyuiop");  
 }  
 }  
  
 public static void main(String[] args) {  
 String s1 = "qwertyuiop";  
 String s2 = "qwertyuiop";  
  
 *CheckString*(s1);  
 *CheckString*(s2);  
  
 Scanner sc = new Scanner(System.*in*);  
 String s3 = sc.nextLine();  
  
 *CheckString*(s3);  
 }  
}

package ru.mirea.inbo0220.markaryants.pr17;  
import java.util.Scanner;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Task2 {  
 static void CheckMac(String s) {  
 //Pattern p = Pattern.compile("([a-f]+[A-F]+:){3}+([0-9]{2}+:){2}+[0-9]{2}");  
 Pattern p = Pattern.*compile*("([0-9a-fA-F]{2}[:-]){5}[0-9a-fA-F]{2}");  
 Matcher m =p.matcher(s);  
 boolean b = m.matches();  
 if (b){  
 System.*out*.println("Данная строка является МАС-адресом");  
 } else {  
 System.*out*.println("Данная строка не является МАС-адресом");  
 }  
 }  
  
 public static void main(String[] args) {  
 String s1 = "aE:dC:cA:56:76:54";  
 String s2 = "01:23:45:67:89:Az";  
  
 *CheckMac*(s1);  
 *CheckMac*(s2);  
  
 Scanner sc = new Scanner(System.*in*);  
 String s3 = sc.nextLine();  
  
 *CheckMac*(s3);  
 }  
}

package ru.mirea.inbo0220.markaryants.pr17;  
import java.util.StringTokenizer;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Task3 {  
 static void FindUsdRurEu(String text) {  
 String[] t;  
 StringTokenizer st = new StringTokenizer(text, "\n");  
 t = new String[st.countTokens()];  
 int i = 0;  
 while(st.hasMoreTokens()) t[i++] = st.nextToken();  
 for (int j = 0;j<i;j++){  
 if (*Find*(t[j]))  
 System.*out*.println(t[j]);  
 }  
 }  
  
 static boolean Find(String t) {  
 Pattern p = Pattern.*compile*("[0-9]+[.]{0,1}[0-9]{0,2}\\s((USD)|(RUR)|(EU))");  
 Matcher m = p.matcher(t);  
 boolean b = m.matches();  
 if (b)  
 return true;  
 else  
 return false;  
 }  
  
 public static void main(String[] args) {  
 String text = "23.78 USD\n22 UDD\n0.002 USD";  
 *FindUsdRurEu*(text);  
 String text2 = "10 RUB\n11.11 RUR\n10 RUR\n11.11 RUB\n0.001 RUR\n137 EU\n144 BC";  
 *FindUsdRurEu*(text2);  
 }  
}

## **Вывод**

Выполнив практическую работу, я изучил работу с регулярными выражениями на языке Java.