|  |
| --- |
|  |
| 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 ПРАКТИЧЕСКИМ ЗАНЯТИЯМ №17, №18**

Выполнил студент группы ИНБО-02-20 Чернов Я. А.

Принял Степанов П.В.

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

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

Отметка о выполнении

**Москва 2021 г.**

**Практические занятия №17-18**

**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.

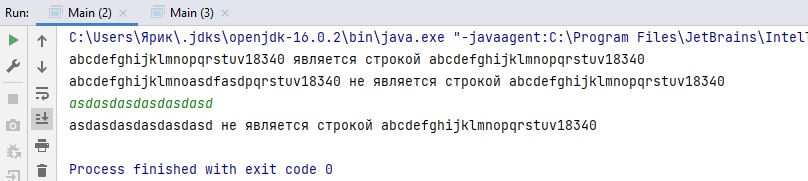
– пример неправильных выражений: 22 UDD, 0.002 USD.

**Код программы**

**Задание 1**

**package** ru.mirea.Practice17\_18.Task1;  
  
**import** java.util.Scanner;  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
  
**public class** Main {  
 **static void** Check(String s) {  
 Pattern p = Pattern.*compile*(**"abcdefghijklmnopqrstuv18340"**);  
 Matcher m = p.matcher(s);  
 **boolean** a = m.matches();  
 **if** (a){  
 System.***out***.println(s + **" является строкой abcdefghijklmnopqrstuv18340"**);  
 } **else** {  
 System.***out***.println(s + **" не является строкой abcdefghijklmnopqrstuv18340"**);  
 }  
 }  
  
 **public static void** main(String[] args) {  
 String s1 = **"abcdefghijklmnopqrstuv18340"**;  
 String s2 = **"abcdefghijklmnoasdfasdpqrstuv18340"**;  
  
 *Check*(s1);  
 *Check*(s2);  
  
 Scanner sc = **new** Scanner(System.***in***);  
 String s3 = sc.nextLine();  
  
 *Check*(s3);  
 }  
}

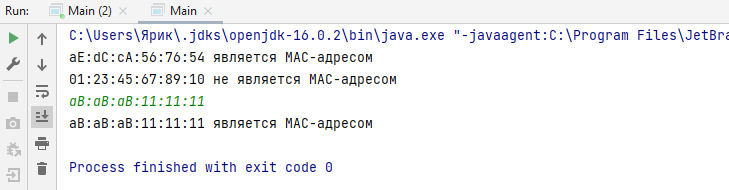
**Тестирование**



**Задание 2**

**package** ru.mirea.Practice17\_18.Task2;  
  
**import** java.util.Scanner;  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
  
**public class** Main {  
 **static void** Check(String s) {  
 Pattern p = Pattern.*compile*(**"([a-f]+[A-F]+:){3}+([0-9]{2}+:){2}+[0-9]{2}"**);  
 Matcher m = p.matcher(s);  
 **boolean** b = m.matches();  
 **if** (b){  
 System.***out***.println(s + **" является МАС-адресом"**);  
 } **else** {  
 System.***out***.println(s + **" не является МАС-адресом"**);  
 }  
 }  
  
 **public static void** main(String[] args) {  
 String s1 = **"aE:dC:cA:56:76:54"**;  
 String s2 = **"01:23:45:67:89:10"**;  
  
 *Check*(s1);  
 *Check*(s2);  
  
 Scanner sc = **new** Scanner(System.***in***);  
 String s3 = sc.nextLine();  
  
 *Check*(s3);  
 }  
}

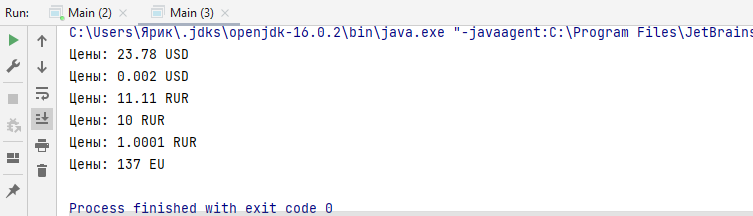
**Тестирование**



**Задание 3**

**package** ru.mirea.Practice17\_18.Task3;  
  
**import** java.util.StringTokenizer;  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
  
**public class** Main {  
 **static void** Check(String s) {  
 Pattern p = Pattern.*compile*(**"(\\d+)([.]?)(\\d\*)(\\s)(USD|RUR|EU)"**);  
 Matcher m = p.matcher(s);  
 **while** (m.find()) {  
 System.***out***.println(**"Цены: "** + m.group());  
 }  
 }  
 **public static void** main(String[] args) {  
 String s1 = **"23.78 USD\n22 UDD\n0.002 USD"**;  
 *Check*(s1);  
  
 String s2 = **"10 RUB\n11.11 RUR\n10 RUR\n11.11 RUB\n1.0001 RUR\n137 EU\n144 BC"**;  
 *Check*(s2);  
 }  
}

**Тестирование**



**Выводы**

По ходу работы было изучено создание регулярных выражений и сравнивание с помощью них входящих данных из консоли.

Git: <https://github.com/YaroslavChernov0220/PracticeMIREA>