МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Математическое обеспечение и применение ЭВМ»

ОТЧЕТ

по лабораторной работе №2

по курсу «Основы Web-программирования» на тему

«Использование регулярных выражений»

Выполнили: студенты группы 21ВП2

Лакеева Софья

Малькова Анастасия

Принял: к.т.н. Такташкин Д.В.

Пенза 2022

# Название

Использование регулярных выражений.

# Цель работы

Во второй лабораторной работе необходимо реализовать консольное приложение, позволяющее манипулировать строкой, разбив ее на элементы путем использования регулярных выражений.

# Лабораторное задание

Составить регулярное выражение, определяющее является ли заданная строка IP адресом, записанным в десятичном виде.

* пример правильных выражений: 127.0.0.1, 255.255.255.0.
* пример неправильных выражений: 1300.6.7.8, abc.def.gha.bcd.

# Листинг

Файл laba2.java

*//21ВП2 Малькова, Лакеева  
//8 вариант  
//Составить регулярное выражение, определяющее является ли заданная строка IP адресом, записанным в десятичном виде.  
// ♣ пример правильных выражений: 127.0.0.1, 255.255.255.0.  
// ♣ пример неправильных выражений: 1300.6.7.8, abc.def.gha.bcd.***import** java.util.Scanner;  
**import** java.util.regex.Pattern;  
**import** java.util.regex.Matcher;  
  
**public class** laba2 {  
 **public static void** main(String[] args){  
 System.***out***.println(**"Введите IP-адрес"**);  
 Scanner in = **new** Scanner(System.***in***);  
 String IP = in.nextLine();  
 **boolean** check = Pattern.*matches*(**"^[^, ]+(, [^, \\n]+)\*$"**, IP);  
 **while** (!check){  
 System.***out***.println(**"Введите IP-адреса через одну запятую и один пробел"**);  
 IP = in.nextLine();  
 check = Pattern.*matches*(**"^[^, ]+(, [^, \\n]+)\*$"**, IP);  
 }  
  
 Pattern p1 = Pattern.*compile*(**", "**);  
 String[] words = p1.split(IP);  
 **for** (**int** i = 0; i < words.**length**; i++){  
 Pattern p = Pattern.*compile*( **"^([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])$"**);  
 Matcher m = p.matcher(words[i]);  
 **boolean** f = m.matches();  
 **if**(!f)  
 System.***out***.println(words[i] + **" Не является IP-адресом. Примеры корректного IP-адреса: "** +  
 **"127.0.0.1, 255.255.255.0"**);  
 **else** System.***out***.println(words[i] + **" Это IP-адрес"**);  
 }  
 }  
}

# Пояснительный текст к программе

Переменная p — объект класса Pattern, соответствующий нужному регулярному выражению.

Переменная m – объект класса Matcher, с помощью которого находятся соответствия в строке.

Переменная f – логическая переменная, хранящая результат соответствия строки регулярному выражению.

Переменная check – логическая переменная, хранящая результат соответствия строки регулярному выражению.

# Результаты работы программы

Результаты работы программы показаны на рисунках 1, 2, 3 и 4.

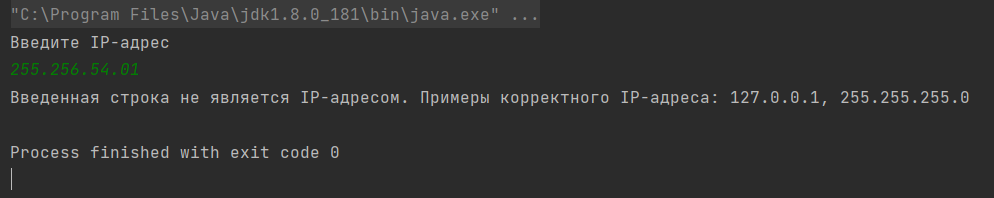


Рисунок 1 — Результаты работы программы

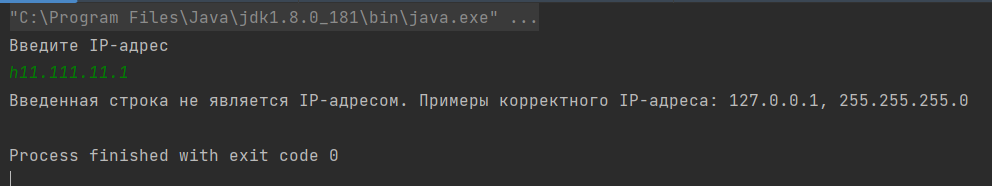


Рисунок 2 — Результаты работы программы

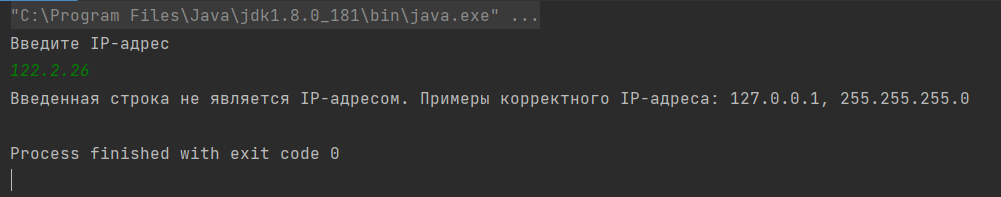


Рисунок 3 — Результаты работы программы

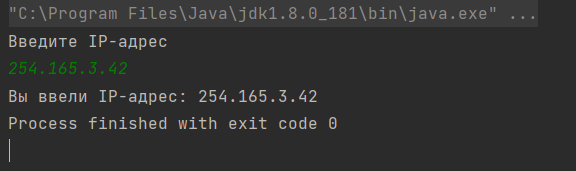


Рисунок 4 — Результаты работы программы

# 

# Выводы

В ходе выполнения лабораторной работы было разработано консольное приложение, определяющее является ли введенная строка IP адресом, записанным в десятичном виде.