федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

Институт математики, естественных и компьютерных наук

Кафедра прикладной математики

Отчет по лабораторной работе №14

Дисциплина: «Программирование»

Название: «Регулярные выражения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \_\_\_\_01.03.02\_\_.  код направления  подготовки/  специальности | \_\_\_\_.  код выпускающей  кафедры | \_\_\_\_\_\_1\_\_\_\_\_\_  регистрационный номер по журналу | \_\_\_\_\_1\_\_\_\_\_  код формы  обучения | \_\_\_2024\_\_\_  год |

|  |  |
| --- | --- |
| Руководитель | доцент Ганичева Е.М. |
|  |  |
| Выполнил (а) студент | Шлыкова Э.Э. |
|  | |
| Группа, курс | 4Б01 ПМ-11, 1 курс |
| Дата сдачи | |
| Дата защиты |  |
| Оценка по защите |  |
| *(подпись преподавателя)* | |

Вологда

2024 г.

**Практическая часть №1**

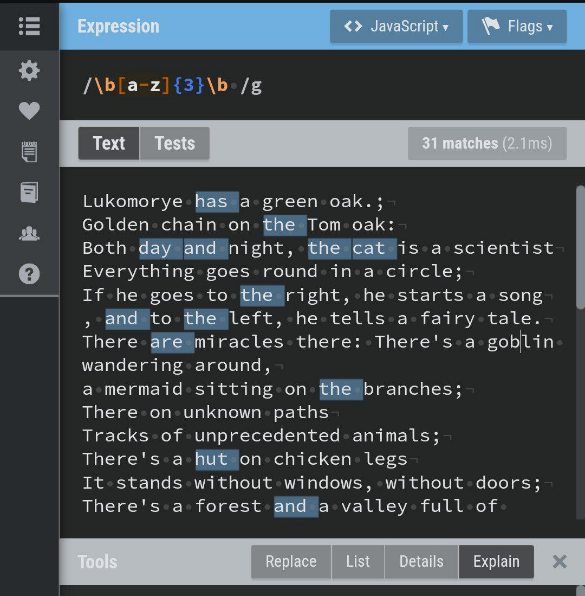
**Задание 1.** Отработать нахождение определенных символов и их соединений в https://regexr.com/ (провести не менее 10 различных по сложности регулярных выражений). Нельзя использовать регулярные выражения, подходящие для решения задач из второго задания.

**Ответ на задание:**

1. Находит слова, состоящие ровно из 3 букв в нижнем регистре.

Выражение: \b[a-z]{3}\b

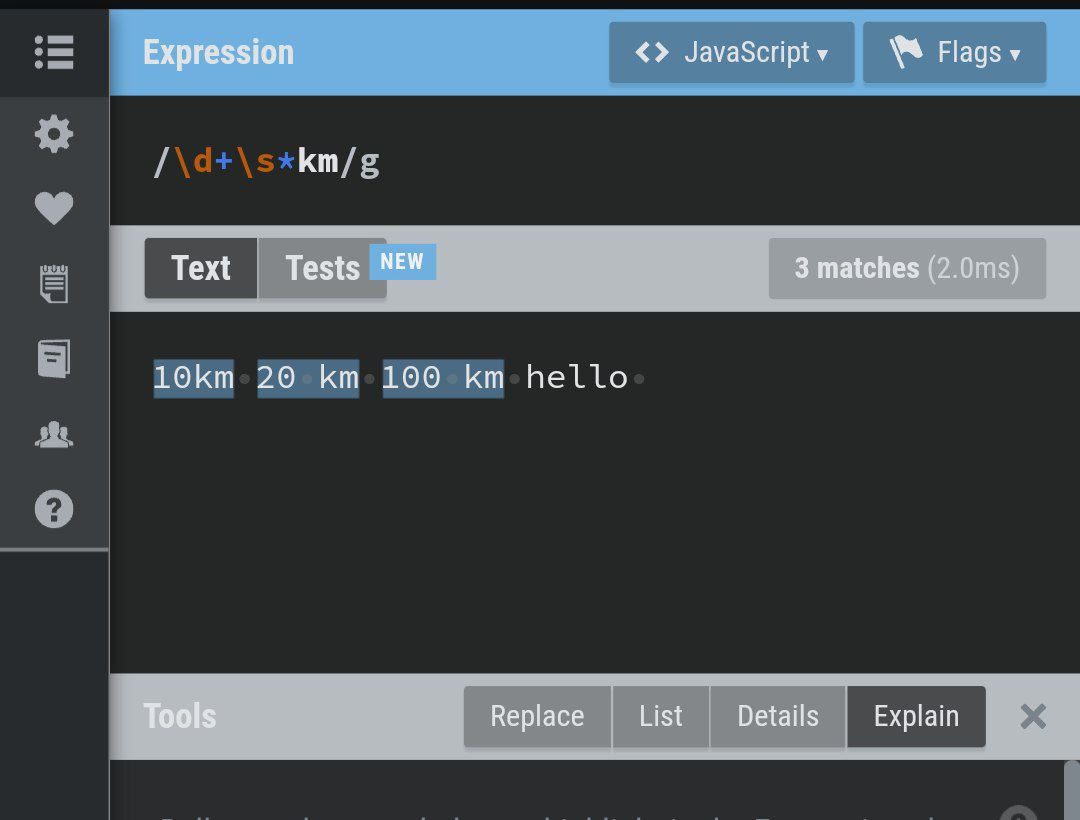
Результат:



1. Находит одно или несколько чисел, за которыми следует ноль или более пробелов, а затем “km”

Выражение: \d+\s\*km

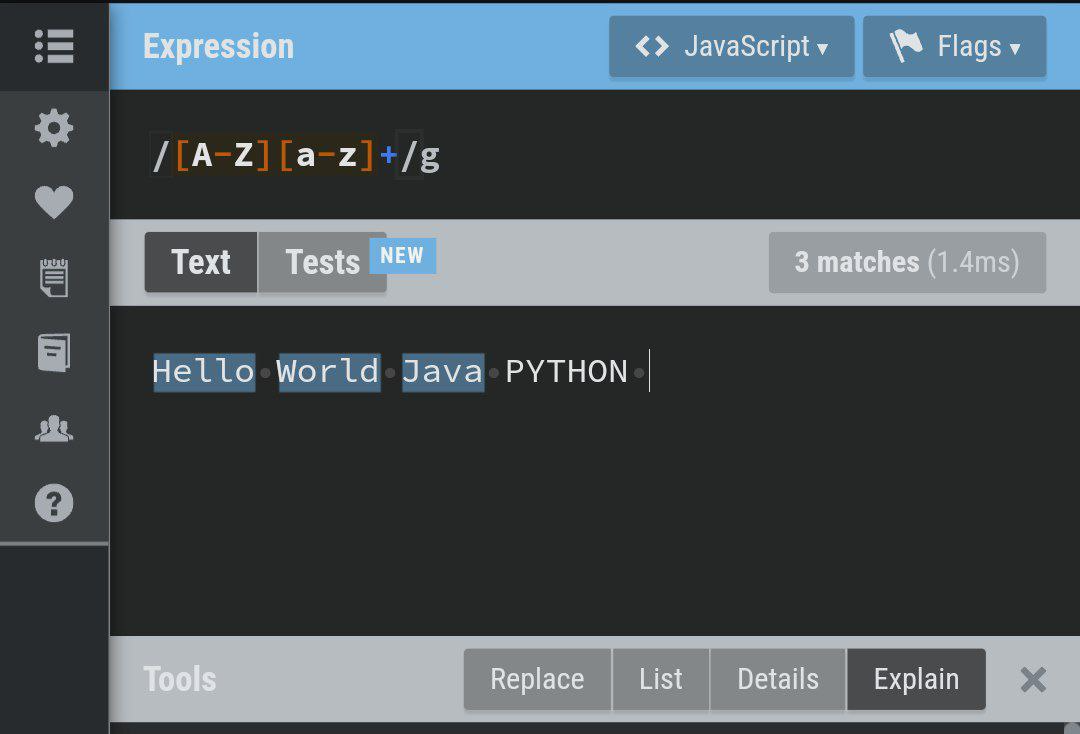
Результат:



1. Находит слова, начинающиеся с большой буквы, за которой следует одна или более маленьких букв.

Выражение: [A-Z][a-z]+

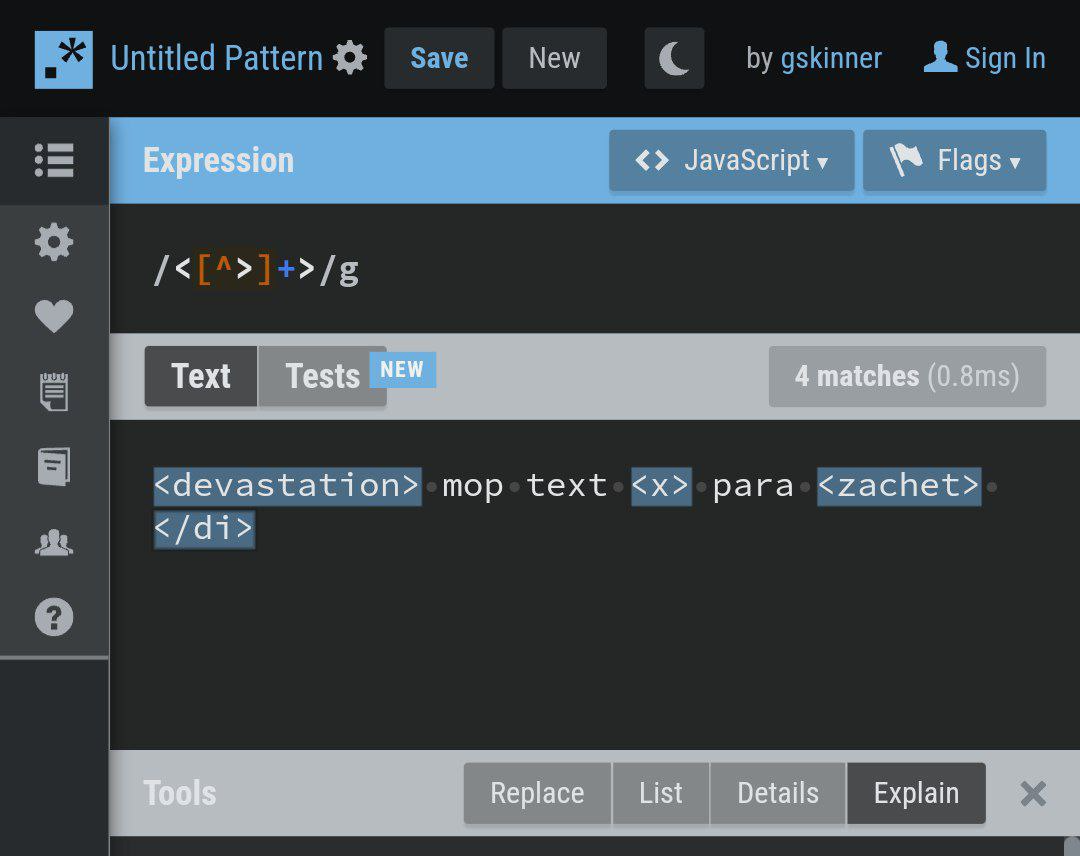
Результат:



1. Находит HTML/XML-подобные теги

Выражение: <[^>]+>

Результат:



1. Находит слова, заканчивающиеся на “ing”

Выражение: \b\w\*ing\b

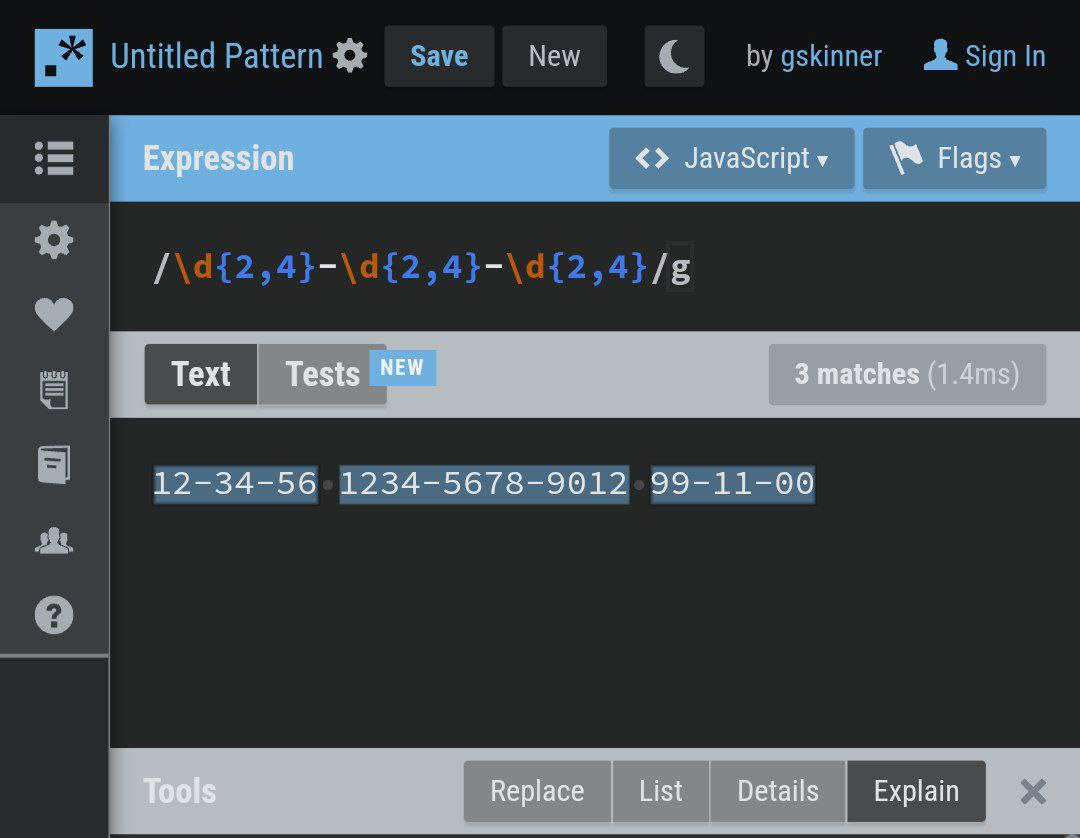
Результат:



1. Находит последовательности цифр в формате “XX-XX-XX” или “XXXX-XXXX-XXXX”

Выражение: \d{2,4}-\d{2,4}-\d{2,4}

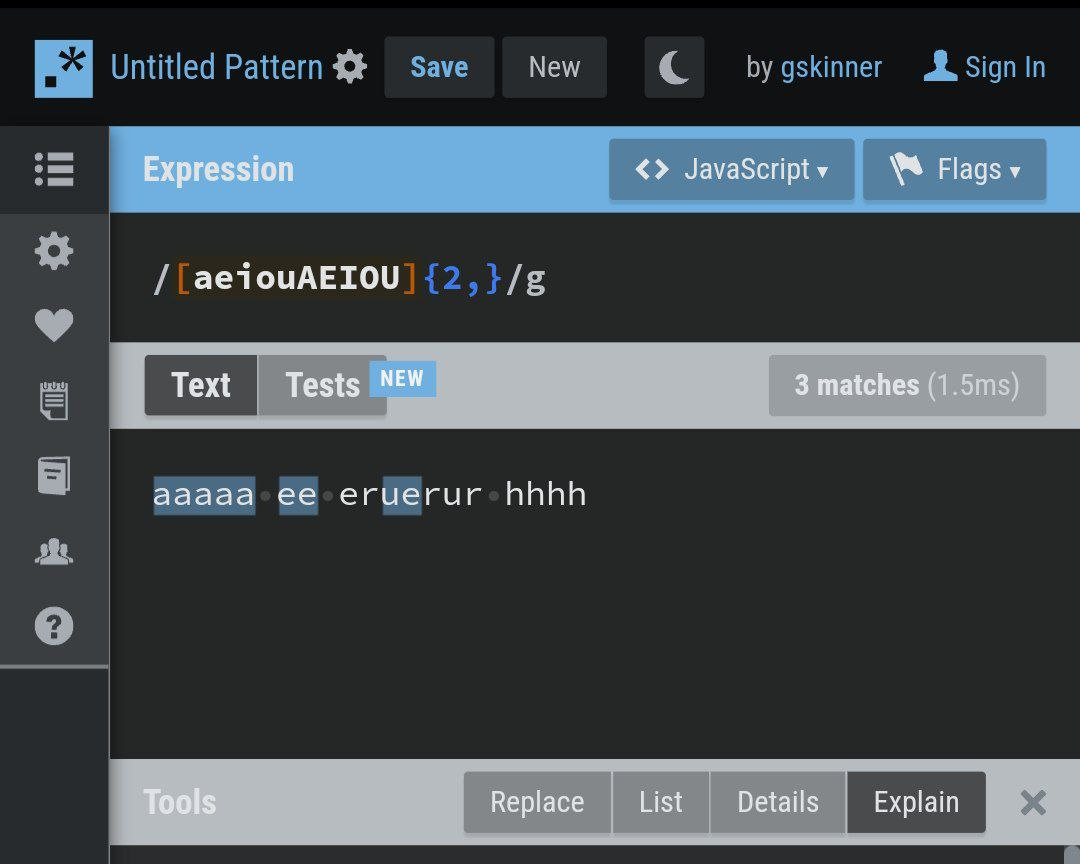
Результат:



1. Находит 2 и более гласных буквы подряд

Выражение: [aeiouAEIOU]{2,}

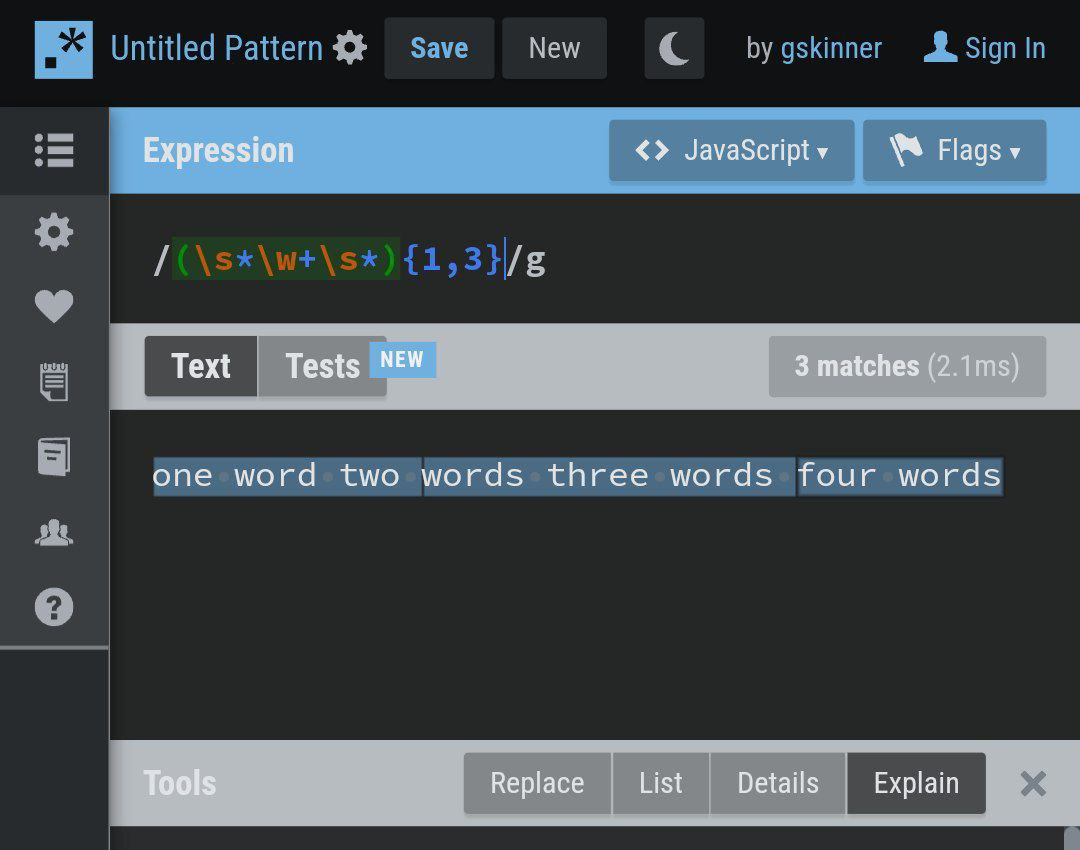
Результат:



1. Находит строки, содержащие от 1 до 3 слов, разделенных пробелами, в начале и конце строк могут быть пробелы.

Выражение: ^(\s\*\w+\s\*){1,3}$

Результат:



1. Находит 6-значные шестнадцатеричные коды цветов (с #).

Выражение: #[\da-fA-F]{6}

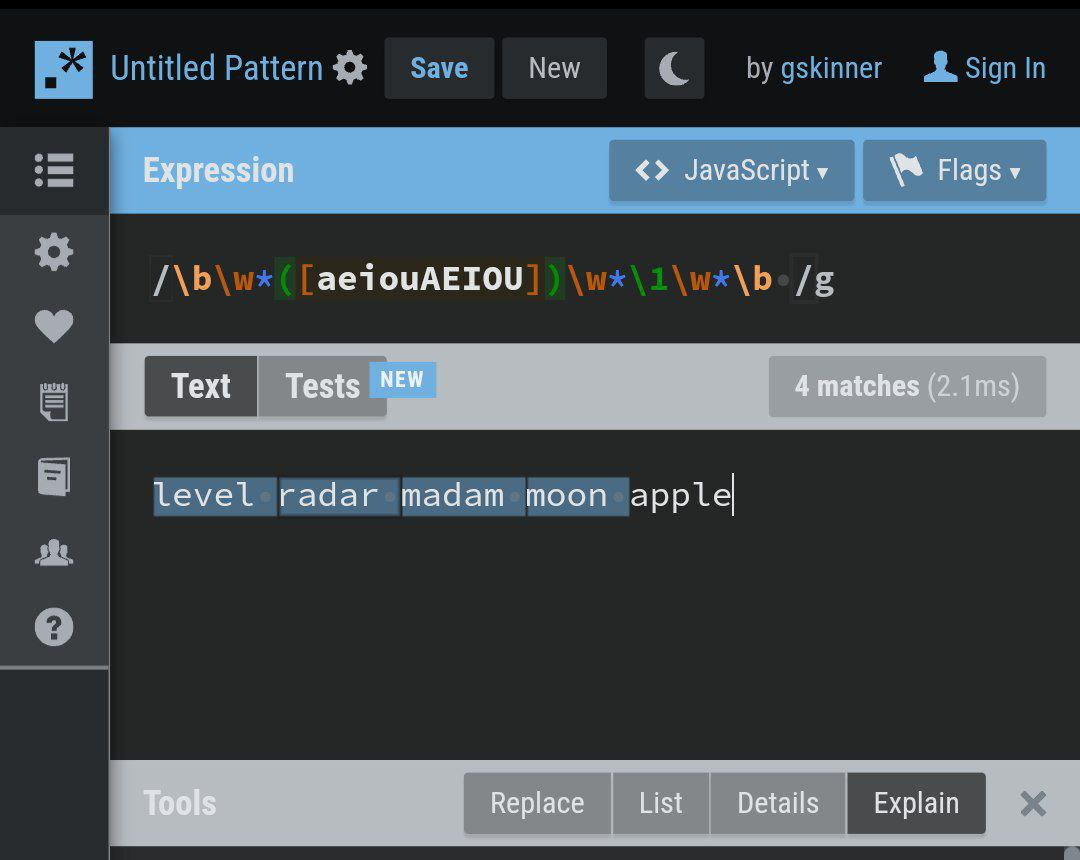
Результат:



1. Находит слова, содержащие повторяющуюся гласную.

Выражение: \b\w\*([aeiouAEIOU])\w\*\1\w\*\b

Результат:

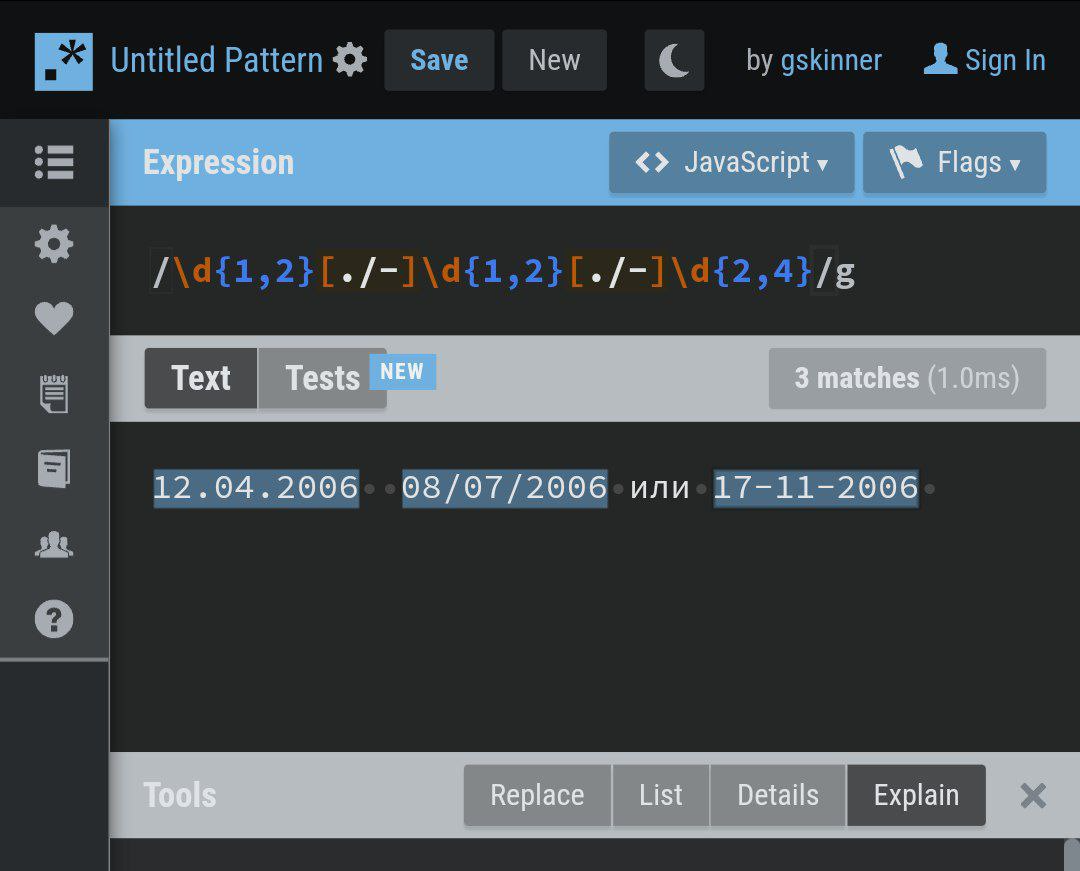


**Задание 2.** Написать и протестировать в https://regexr.com/ следующие регулярные выражения:

1. Поиск даты рождения.

Выражение: \d{1,2}[./-]\d{1,2}[./-]\d{2,4}

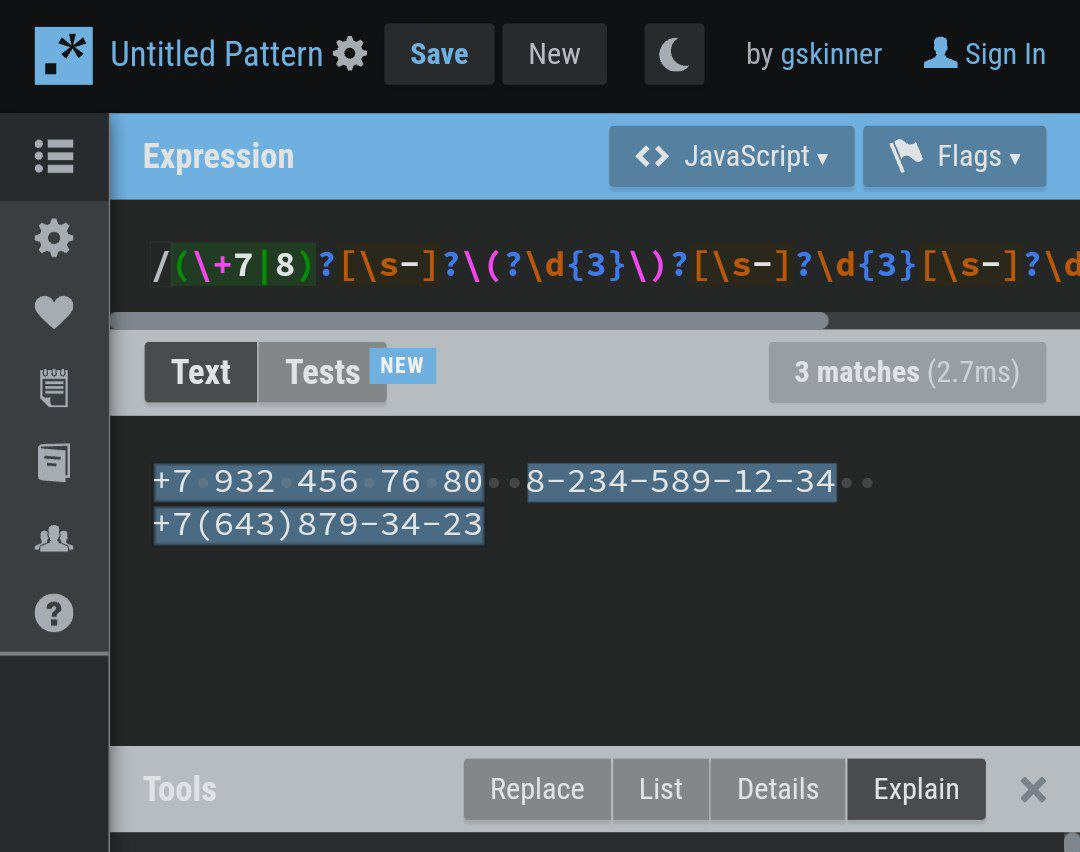
Результат:



1. Нахождение номера сотового телефона.

Выражение: (\+7|8)?[\s-]?\(?\d{3}\)?[\s-]?\d{3}[\s-]?\d{2}[\s-]?\d{2}

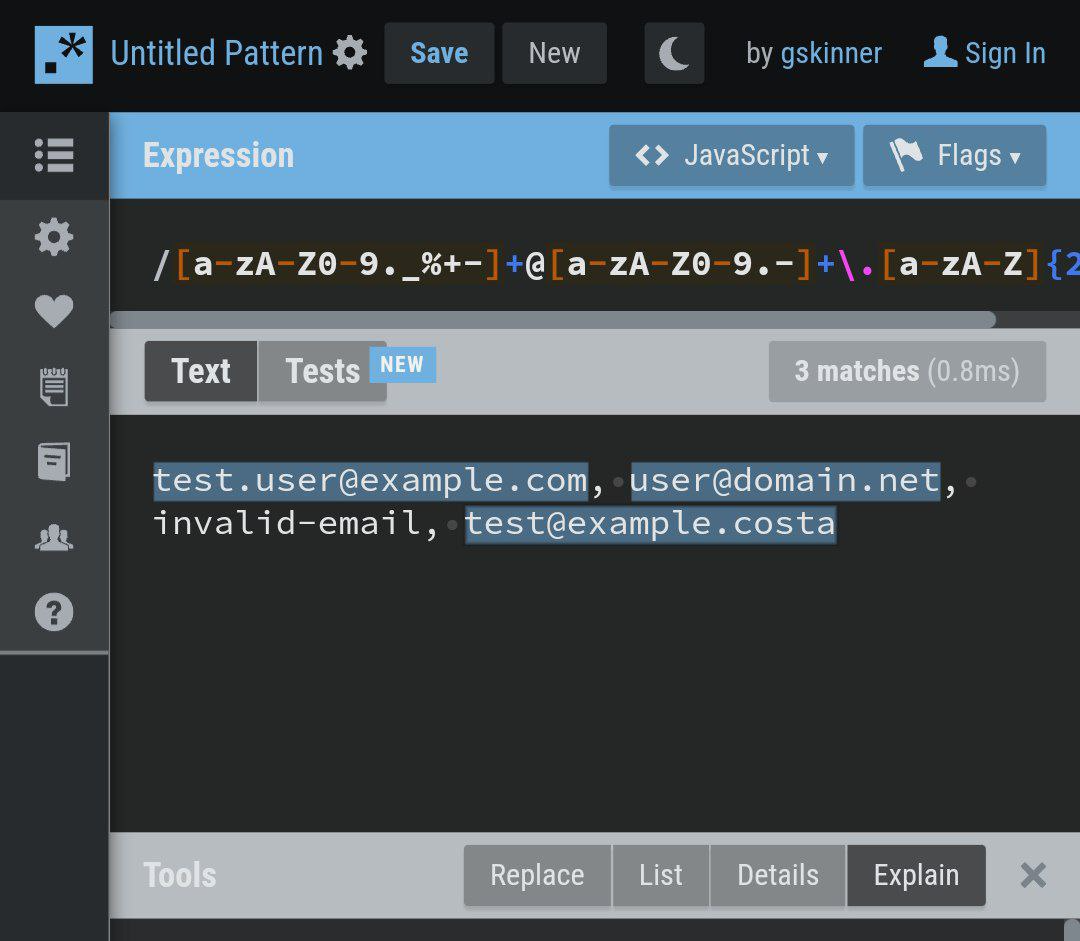
Результат:



1. Проверка корректного ввода email.

Выражение: [a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}

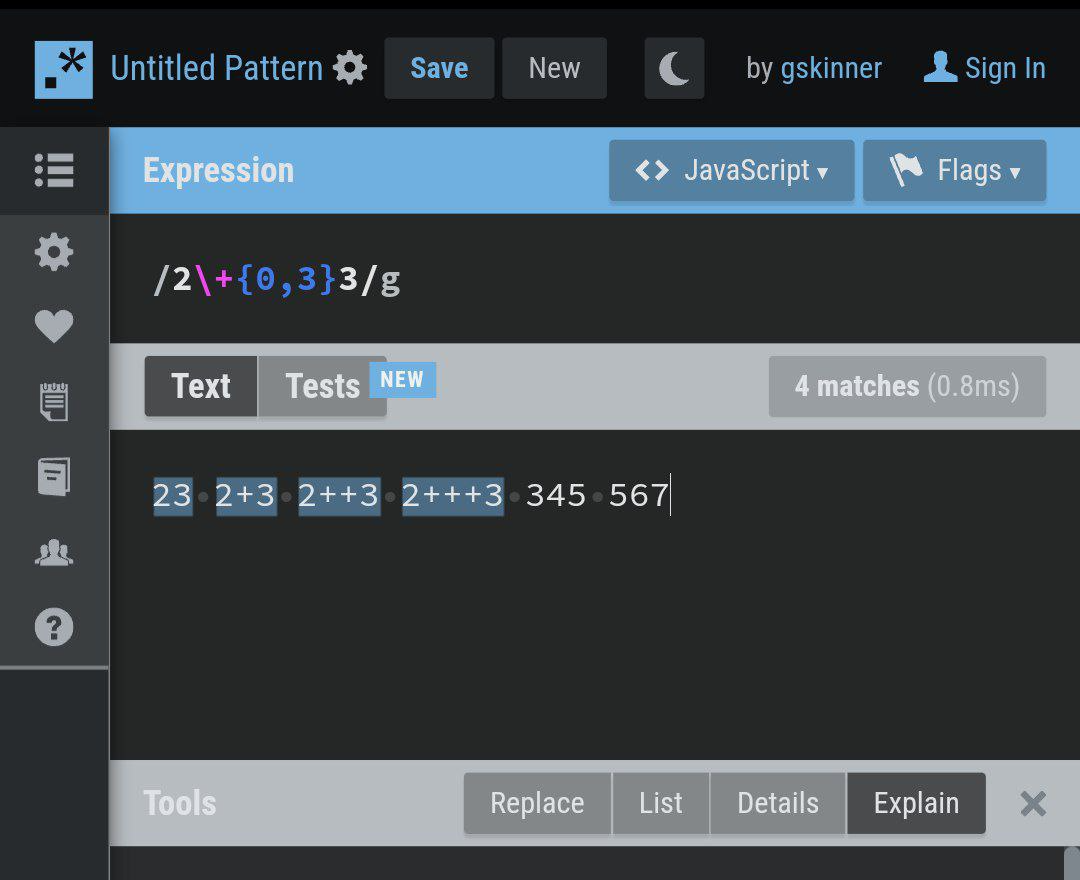
Результат:



1. Дана строка '23 2+3 2++3 2+++3 345 567'. Напишите регулярное выражение, которое найдет строки 2+3, 2++3, 2+++3, не захватив остальные («+» может быть любое количество).

Выражение: 2\+{0,3}3

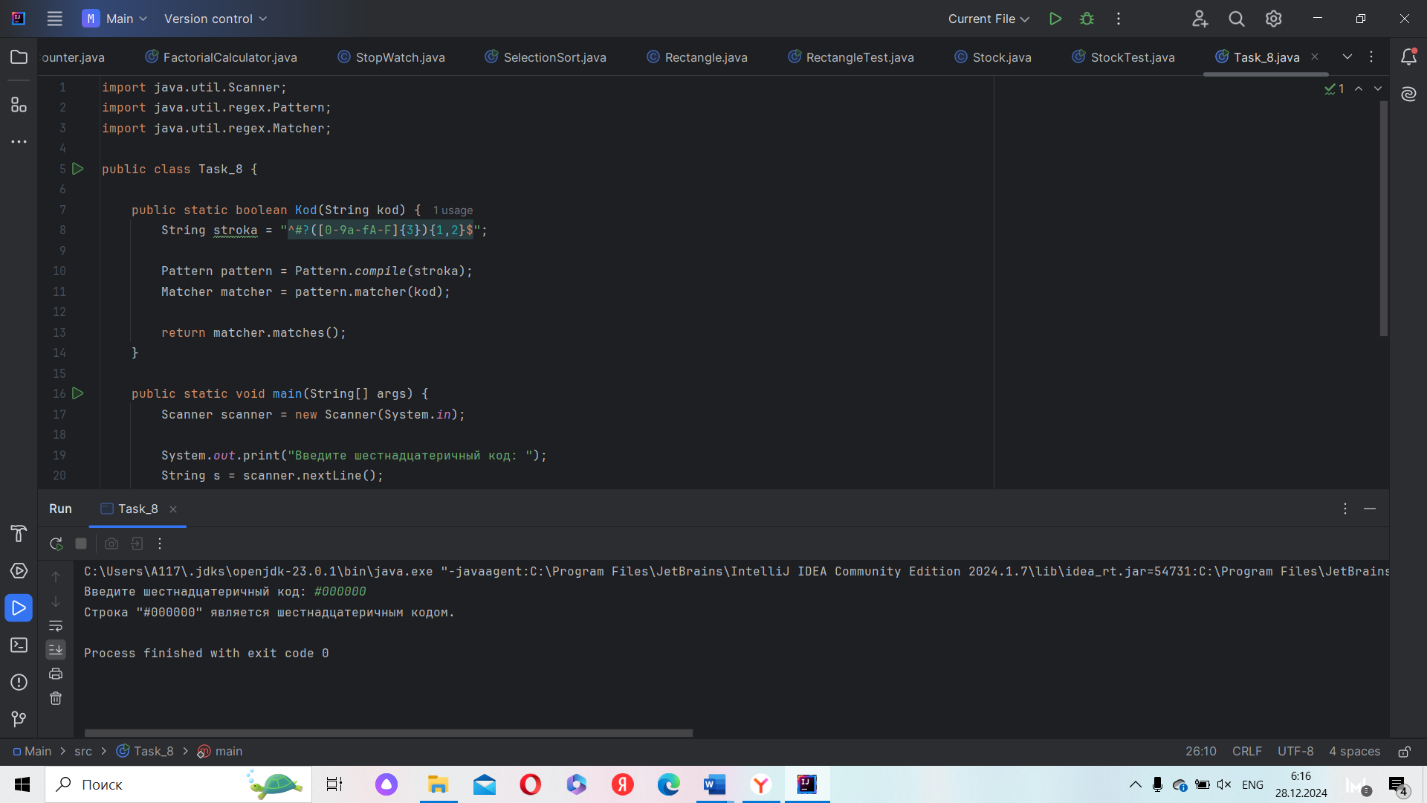
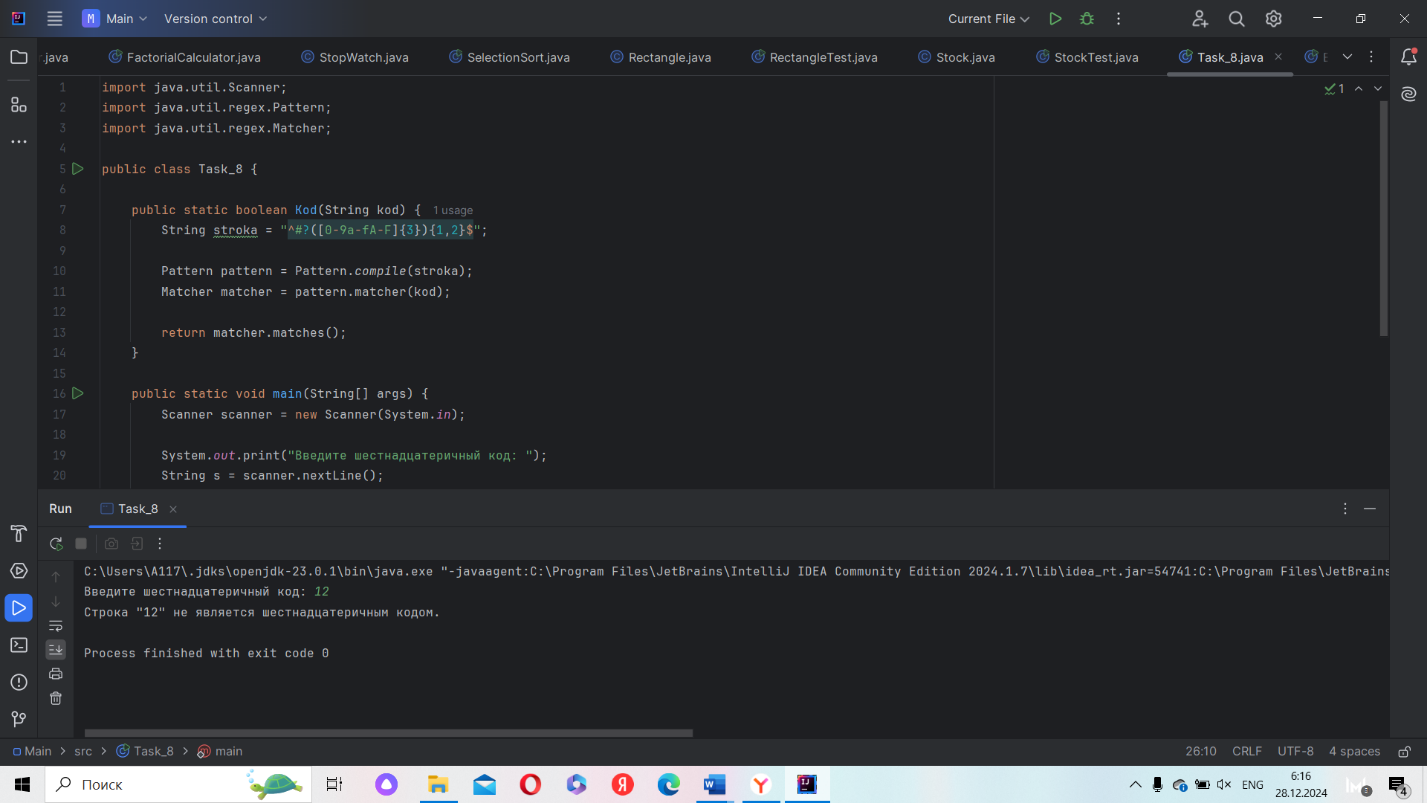
Результат:



**Практическая часть №2**

**Задание 4.** Напишите программу, чтобы проверить, является ли данная строка шестнадцатеричным кодом или нет.

import java.util.Scanner;  
import java.util.regex.Pattern;  
import java.util.regex.Matcher;  
  
public class Task\_4 {  
  
 public static boolean Kod(String kod) {  
 String stroka = "^#?([0-9a-fA-F]{3}){1,2}$";  
  
 Pattern pattern = Pattern.*compile*(stroka);  
 Matcher matcher = pattern.matcher(kod);  
  
 return matcher.matches();  
 }  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите шестнадцатеричный код: ");  
 String s = scanner.nextLine();  
  
 if (*Kod*(s)) {  
 System.*out*.println("Строка \"" + s + "\" является шестнадцатеричным кодом.");  
 } else {  
 System.*out*.println("Строка \"" + s + "\" не является шестнадцатеричным кодом.");  
 }  
 }  
}

**Результат:**