Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Дисциплина: «Информатика»

**Лабораторная работа по информатике №3**

**«Регулярные выражения»**

­

Выполнил: Третьяков Илья Антонович

Группа: P3108

Вариант: 235

Преподаватель: Малышева Татьяна Алексеевна

Санкт-Петербург

2021

# Оглавление

[Оглавление 2](#_Toc86746924)

[Задание 1 2](#_Toc86746925)

[Задание 2 (вариант 5) 3](#_Toc86746926)

[Задание 3 (вариант 3) 4](#_Toc86746927)

[Вывод 5](#_Toc86746928)

[Список литературы 5](#_Toc86746929)

# Задание 1

1. Программа для определения варианта и рисования смайлика на языке Python:

**isu = 338987  
  
eyes = [":", ";", "X", "="]  
nose = ["-", "<", "-{", "<{"]  
mouth = ["(", ")", "O", "|", "\\", "/", "P"]  
  
print(eyes[isu % 5] + nose[isu % 4] + mouth[isu % 7])**

Результат работы программы:

**X<{/**

1. Тесты:

X<{/:<{(;-{O=<{):<P;-{\=-{)X<{O=<{)X-{O

=<\:-\=-)X-{/X<{\;-{/X<{/;<\:-{)X<{/

X<{/;<{OX<{/X<{/X-|=<(=-{(;<{OX-{\;-{P

;-(X<{/:<{/;<{):-P=-{|X<{/X<{/X<{/X<{/

X-{\=-{)X-{\X<{/=-PX<P;-{(;-{\X<{)=<{|

1. Найдем вручную количество смайликов предложенного типа (X<{/):

|  |  |
| --- | --- |
| **Тест** | **Количество смайликов** |
| **X<{**/:<{(;-{O=<{):<P;-{\=-{)X<{O=<{)X-{O | 1 |
| =<\:-\=-)X-{/**X<{\**;-{/X<{/;<\:-{)**X<{/** | 2 |
| **X<{/**;<{O**X<{/X<{/**X-|=<(=-{(;<{OX-{\;-{P | 3 |
| ;-(**X<{/**:<{/;<{):-P=-{|**X<{/X<{/X<{/X<{/** | 5 |
| X-{\=-{)X-{\**X<{/**=-PX<P;-{(;-{\X<{)=<{| | 1 |

1. Найдем количество смайликов с помощью программы:

**import re  
  
tests = [r'X<{/:<{(;-{O=<{):<P;-{\=-{)X<{O=<{)X-{O',  
 r'=<\:-\=-)X-{/X<{\;-{/X<{/;<\:-{)X<{/',  
 r'X<{/;<{OX<{/X<{/X-|=<(=-{(;<{OX-{\;-{P',  
 r';-(X<{/:<{/;<{):-P=-{|X<{/X<{/X<{/X<{/',  
 r'X-{\=-{)X-{\X<{/=-PX<P;-{(;-{\X<{)=<{|']  
  
smile = re.compile('X<{/')  
  
for i in tests:  
 print(len(re.findall(smile, i)))**

Результат работы программы:

**1**

**2**

**3**

**5**

**1**

# Задание 2 (вариант 5)

С помощью регулярного выражения найти в тексте все слова, в которых две гласные стоят подряд, а после этого слова идёт слово, в котором не больше 3 согласных.

Пример:

|  |  |
| --- | --- |
| Ввод | Вывод |
| Кривошеее существо гуляет по парку | гуляет |

1. Тесты:

Прямошея абстракция перемащается по неевклидову пространству

Регулярные выражения не очень и регулярные на самом деле

Задание на 100 баллов

Системное и прикладное программное что

Приказ об отчислении от 31.02.22

1. Найдем вручную правильный ответ:

|  |  |
| --- | --- |
| Тест | Ответ |
| Прямошея абстракция **перемащается** по неевклидову пространству | перемещается |
| Регулярные **выражения** не очень и **регулярные** на самом деле | выражения регулярные |
| **Задание** на 100 баллов | Задание |
| **Системное** и прикладное **программное** что | Системное программное |
| Приказ об **отчислении** от 31.02.22 | отчислении |

1. Найдем ответ с помощью программы:

**import re  
  
tests = ['Прямошея абстракция перемащается по неевклидову пространству',  
'Регулярные выражения не очень и регулярные на самом деле',  
'Задание на 100 баллов',  
'Системное и прикладное программное что',  
'Приказ об отчислении от 31.02.22']  
  
regex = re.compile(r"([А-Яа-я]\*[уеыаоэёяию]{2}[А-Яа-я]\*)(?=\s[уеыаоэёяию]\*[^уеыаоэёяию]?[уеыаоэёяию]\*[^уеыаоэёяию]?[уеыаоэёяию]\*[^уеыаоэёяию]?[уеыаоэёяию]\*(?:\s|$))")  
for i in tests:  
 print(' '.join(re.findall(regex,i)))**

Результат работы программы:

**перемащается  
выражения регулярные  
Задание  
Системное программное  
отчислении**

# Задание 3 (вариант 3)

Вывесили списки стипендиатов текущего семестра, которые представляют из себя список людей ФИО и номер группы этого человека. Вы решили подшутить над некоторыми из своих одногруппников и удалить их из списка.

С помощью регулярного выражения найдите всех студентов своей группы, у которых инициалы начинаются на одну и туже букву и исключите их из списка.

Пример (группа P000):

|  |  |
| --- | --- |
| Ввод | Вывод |
| Петров П.П. P000  Анищенко А.А. P33113  Примеров Е.В. P000  Иванов И.И. P000 | Анищенко А.А. P33113  Примеров Е.В. P000 |

1. Тесты:

Аталян А.А. P3108

Приходько М.А. P3112

Короткова К.К. P3108

Акимов А.А. P3111

Петров П.П. P3108

Третьяков И.А. P3108

1. Программа:

**import re  
  
tests = ["Аталян А.А. P3108",  
 "Приходько М.А. P3112",  
 "Короткова К.К. P3108",  
 "Акимов А.А. P3111",  
 "Петров П.П. P3108",  
 "Третьяков И.А. P3108"]  
  
regex = re.compile(r"([А-Я])[а-я]+ \1\.\1\. P3108")  
for i in tests:  
 res = re.findall(regex, i)  
 if res:  
 continue  
 print(i)**

1. Вывод программы:

**Приходько М.А. P3112  
Акимов А.А. P3111  
Третьяков И.А. P3108**

# Вывод

Проделав лабораторную работу, я научился работать с регулярными выражениями, а именно: научился синтаксису регулярных выражений, научился составлять их для поиска совпадений в каком-либо тексте, научился использовать их с помощью встроенной библиотеки *re* языка Python, а также освежил свои знания этого языка.

# Список литературы

1. Регулярные выражения [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Регулярные_выражения>.
2. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. – 592 с.