НИУ ИТМО

Факультет программной инженерии и компьютерной техники

Отчет по Лабораторной работе №3    
по курсу “Информатика”

“Регулярные выражения”  
   
Вариант №115

Выполнил:

Студент группы P3118

Шульга Артём Игоревич

Преподаватель:

Малышева Татьяна Алексеевна

Санкт-Петербург

2021

Оглавление

[Задание для Варианта 101 2](#_Toc85077405)

[Дополнительное задание №1 (5 вариант) 4](#_Toc85077408)

[Дополнительное задание №2 (1 вариант) 6](#_Toc85077410)

[Вывод 7](#_Toc85077412)

[Список литературы 8](#_Toc85077413)

# Изображение выглядит как текст, мебель, белый, шкафчик Автоматически созданное описаниеЗадание для Варианта 101

По данной таблице и табельному номеру мой смайлик - ;</

# Исходный код

import re

# Мой смайлик: 338861 -> ;</

print(str(338861%5) + " " + str(338861%4) + " " + str(338861%7))

print("My smile: ;</")

def calculate(*string*):

    print("Current string: "+string)

#подсчёт с помощью регулярных выражений

    print("Regular: "+str(len(re.findall(";</",string))))

#подсчёт с помощью средств python

    print("Simple: "+str(string.count(";</")))

my\_string = input()

# строки для тестирования

test = [";<\\",

" ; < / ;< / ; </ ;</ /<;",

";<</ ;;</ ;<// ;<?",

"",

";</;</;</;</ ;</ "]

if my\_string != "":

    test.append(my\_string)

for s in test:

    calculate(s)

# Результат выполнения программы

Тестовые строки:

;<\ (ответ - 0)

; < / ;< / ; </ ;</ /<; (ответ - 1)

;<</ ;;</ ;<// ;<? (ответ - 2)

(пустая строка) (ответ – 0)

;</;</;</;</ ;</ (ответ – 5)

- (ответ - 0)

Изображение выглядит как текст

Автоматически созданное описание

# Дополнительное задание №1 (5 вариант)

Изображение выглядит как текст

Автоматически созданное описание

# Исходный код

import re

print(338861%6)

regex = r"\b\w\*[аюуяыэёеио]{2}\w\*\b(?=\W\*(\b([аюуяыэёеио]\*[^аюуяыэёеио\s][аюуяыэёеио]\*){1,3}\b)|(\b[аюуяыэёеио]+\b))"

test\_str = ("Кривошеее существо гуляет по парку\n"

    "Здравствууйте, амиго\n"

    "Аа регулааааа\n"

    "Проживающий в общежитии купил себе что-то\n"

    "\n"

    "Опр ааа рппп аааааа реуууасн\n")

my\_string = input()

test\_str += my\_string + '\n'

# находит все вхождения в строке и возвращает итератор на коллекцию подходящих элементов

matches = re.finditer(regex, test\_str, re.MULTILINE | re.IGNORECASE)

for match in matches:

    print (match.group())

**Результат выполнения программы**

Тестовые строки:

Кривошее существо гуляет по парку (подходит гуляет)

Здравствууйте, амиго (подходит Здравствууйте)

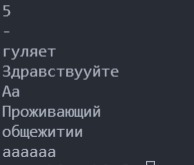
Аа регулаааа (подходит Аа)

Проживающий в общежитии купил себе что-то (подходит Проживающий и общежитии)

(пустая строка) (ничего не подходит)

Опр ааа рппп аааааа реуууасн (подходит аааааа)

- (ничего не подходит)



# Дополнительное задание №2 (1 вариант)

Изображение выглядит как стол

Автоматически созданное описание

**Исходный код**

import re

print(338861%4)

regex = r"\b[^аиеёоуыэюя\s]\*([аиеёоуыэюя])([^аиеёоуыэюя\s]\*\1)\*[^аиеёоуыэюя\s,\-\!\?\.]\*\b"

def search\_strings(*string*):

    result = re.finditer(regex,string,re.IGNORECASE)

    matches = []

    for i in result:

        matches.append(i.group(0))

    matches.sort(*key*=len)

    for i in matches:

        print(i)

test\_str = ["Классное слово – обороноспособность,которое должно идти после слов: трава и молоко.\n",

    "Олово оловянный\n",

    "\n",

    "Колокол течет речь\n",

    "Висит\n",

    "Ерись несёт ценизм ва-банк\n"]

for i in test\_str:

    search\_strings(i)

# Результат выполнения программы

Тестовые строки:

Классное слово - обороноспособность, которое должно идти после слов: трава и молоко. (ответ: и идти слов слово трава должно молоко обороноспособность)

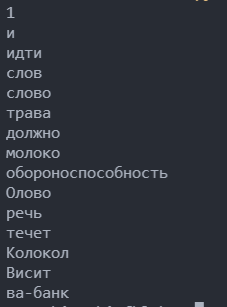
Олово оловянный (ответ: олово)

(пустая строка) (ответ: ничего)

Колокол течет речь (ответ: речь течет колокол)

Висит (ответ: висит)

Ерись несёт ценизм ва-банк (ответ: ва-банк)



# Вывод

В ходе выполнения данной лабораторной работы я познакомился с регулярными выражениями и освоил их использование на языке программирования Python. Я изучил основные понятия по теме “Регулярные выражения” и применил их на практике. Регулярные выражения очень полезны для работы со строками, поэтому знание регулярных выражений – весьма полезный навык.

# Список литературы

# “Регулярные выражения в Python от простого к сложному. Подробности, примеры, картинки, упражнения”. – Текст электронный // https://habr.com – URL https://habr.com/ru/post/349860/

# “Python RegEx: практическое применение регулярок”. – Текст электронный // https://tproger.ru – URL https://tproger.ru/translations/regular-expression-python/