Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

*Факультет Программной инженерии и компьютерной техники*

**Лабораторная работа №3**

Регулярные выражения

Вариант №368308

Группа: P3112

Выполнил: Кобелев Р.П.

Проверил:

к.т.н. преподаватель Белозубов А.В.



г.Санкт—Петербург

2022г

Оглавление

[Введение 3](#_Toc117085678)

[Задания 4](#_Toc117085679)

[Смайлики 4](#_Toc117085680)

[Тесты 4](#_Toc117085681)

[Программа 6](#_Toc117085682)

[Вывод программы 7](#_Toc117085683)

[Необязательное задания (+18) 7](#_Toc117085684)

[Тесты 8](#_Toc117085685)

[Программа 9](#_Toc117085686)

[Вывод программы 10](#_Toc117085687)

[Необязательное задание (+22) 11](#_Toc117085688)

[Тесты 12](#_Toc117085689)

[Программа 13](#_Toc117085690)

[Вывод программы 14](#_Toc117085691)

[Заключение 16](#_Toc117085692)

[Литература 17](#_Toc117085693)

# Введение

В данной лабораторной работе я буду писать алгоритмы для форматирования текста с использованием регулярных выражений. Алгоритмы буду писать с помощью языка программирования python.

# Задания

## Смайлики

1) Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице 1.

2) Для своей программы придумайте минимум 5 тестов. Каждый тест является отдельной сущностью, передаваемой регулярному выражению для обработки. Для каждого теста необходимо самостоятельно (без использования регулярных выражений) найти правильный ответ. После чего сравнить ответ, выданный программой, и полученный самостоятельно.

3) Программа должна считать количество смайликов определённого вида (вид смайлика описан в таблице вариантов) в предложенном тексте. Все смайлики имеют такую структуру:

[глаза][нос][рот].

Вариантом является различные наборы глаз, носов и ртов.

368308%6 = 4

368308%4 = 0

368308%7 = 3

Смайлик – { = - | }

Изображение выглядит как текст, белый, снимок экрана

Автоматически созданное описание

**Таблица 1 Смайлики**

### Тесты

* Тест 1 (12 вхождений смайлика)

=-|Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod =-|tempor invidunt ut labore et dol=-|ore magna aliquyam erat, sed diam voluptua.

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit=-| amet.

Lorem ipsum dolor sit =-|=-|amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

At v=-|ero eos et accusam et justo duo dolo=-|-=res et ea rebum. Stet clita kasd gube=-|=-|=-|rgren, no sea takimata sanctus est Lorem ipsum dolor sit amet.=-|

* Тест 2 (30 вхождений смайлика)

=-|=--|=-|=-|=-56|=-|=-|5=-|=-|13=-|5=-|=-|=-|=-905|=-|=-|=-|

=-|=5===--=-1|-=-53||=-|=-|=357-|=-|=-5|-=-|

=-|=55-|=-|

=-|3=-||||||-==-=-=-||=-|=75-|=-|=-|

=-11|=-|-=-|=-55|=-|

* Тест 3 (20 вхождений смайлика)

Я помню чудное =-|:

Передо мной явилась ты,

Как мимолетное =-|,

Как гений чистой красоты

В томленьях грусти безнадежной,

В тревогах шумной суеты,

Звучал мне долго голос нежный

И снились милые =-|.

Шли =-|. Бурь =-| мятежный

Рассеял прежние мечты,

И я забыл твой =-| нежный,

Твои небесные =-|.

В глуши, во мраке заточенья

Тянулись тихо =-| мои

Без =-|, без =-|,

Без =-|, без =-|, без =-|.

Душе настало пробужденье:

И вот опять явилась =-|,

Как мимолетное =-|,

Как гений чистой красоты.

И сердце бьется в упоенье,

И для него воскресли вновь

И =-|, и =-|,

И =-|, и =-|, и =-|.

* Тест 4 (4 вхождения смайлика)

=-$|=-^|='-|[=-|]{=-|}=\-/|=-|$=-|

* Тест 5 (0 вхождений смайлика)

=================================---------------------------------------------||||||||||||||||||||||||||||||||

### Программа

**import** **re**

**def** **readfile**(s):

tmp = open(s, encoding='utf8')

tmp1 = ''

**for** i **in** tmp.readlines():

tmp1 += i

tmp.close()

**return** tmp1

smile = r"=-\|"

t1 = re.findall(smile, readfile("Smile\_tests/Smile\_test1.txt"))

t2 = re.findall(smile, readfile("Smile\_tests/Smile\_test2.txt"))

t3 = re.findall(smile, readfile("Smile\_tests/Smile\_test3.txt"))

t4 = re.findall(smile, readfile("Smile\_tests/Smile\_test4.txt"))

t5 = re.findall(smile, readfile("Smile\_tests/Smile\_test5.txt"))

print("Количество вхождений смайлика в тексте 1 = ", len(t1))

print("Количество вхождений смайлика в тексте 2 = ", len(t2))

print("Количество вхождений смайлика в тексте 3 = ", len(t3))

print("Количество вхождений смайлика в тексте 4 = ", len(t4))

print("Количество вхождений смайлика в тексте 5 = ", len(t5))

### Вывод программы

Количество вхождений смайлика в тексте 1 = 12

Количество вхождений смайлика в тексте 2 = 30

Количество вхождений смайлика в тексте 3 = 20

Количество вхождений смайлика в тексте 4 = 4

Количество вхождений смайлика в тексте 5 = 0

**Как мы видим, результат подсчёта без регулярных выражений и с ними совпадает.**

## Необязательное задания (+18)

1) Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице.

2) Для своей программы придумайте минимум 5 тестов. Каждый тест является отдельной сущностью, передаваемой регулярному выражению для обработки. Для каждого теста необходимо самостоятельно (без использования регулярных выражений) найти правильный ответ. После чего сравнить ответ, выданный программой, и полученный самостоятельно.

Пример тестов приведён в фото Тест 1.

3) Можно использовать циклы и условия, но основной частью решения должны быть регулярные выражения.

368308%6 = 4

Изображение выглядит как стол

Автоматически созданное описание

**Тест 1**

### Тесты

* Тест из варианта

Уважаемые студенты! В эту субботу в 15:00 планируется доп. занятие на 2 часа. То есть в 17:00:01 оно уже точно кончится.

Уважаемые студенты! В эту субботу в (TMD) планируется доп. занятие на 2 часа. То есть в (TMD) оно уже точно кончится.

* Тест 1

Привет, брат. Приеду я к 22:30, можете начинать без меня. А завтра в 19:58:31 откроется запись на курс, не пропусти его.

Привет, брат. Приеду я к (TMD), можете начинать без меня. А завтра в (TMD) откроется запись на курс, не пропусти его.

* Тест 2

ВНИМАНИЕ!!!

ЗАВТРА В 23:59:59 ВАСЯ НАЧНЁТ ИГРАТЬ В ДОТУ!

ПОЭТОМУ ДО 05:00 ОН БУДЕТ ОЧЕНЬ СИЛЬНО ОРАТЬ И КРУШИТЬ СВОЮ КВАРТИРУ!

БЕРЕГИТЕ СЕБЯ И СВОИХ БЛИЗКИХ!

ВНИМАНИЕ!!!

ЗАВТРА В (TMD) ВАСЯ НАЧНЁТ ИГРАТЬ В ДОТУ!

ПОЭТОМУ ДО (TMD) ОН БУДЕТ ОЧЕНЬ СИЛЬНО ОРАТЬ И КРУШИТЬ СВОЮ КВАРТИРУ!

БЕРЕГИТЕ СЕБЯ И СВОИХ БЛИЗКИХ!

* Тест 3

Объявление для учащихся!

16 октября в 00:00 откроется тест, который вы должны пройти до 23:59:59. На следующий день в 05:03:02 на сайте будут доступны результаты теста!

Объявление для учащихся!

16 октября в (TMD) откроется тест, который вы должны пройти до (TMD). На следующий день в (TMD) на сайте будут доступны результаты теста!

* Тест 4

РОБОКОП 25:24:23 - новейшая модель нашего робота, вершащего справедливость!

Утилизация старых версий робота (05:70, 05:71, 00:00:60 и 00:00:61) начнётся в 13:00:05 в нашем Центре Робототехники.

РОБОКОП 25:24:23 - новейшая модель нашего робота, вершащего справедливость!

Утилизация старых версий робота (05:70, 05:71, 00:00:60 и 00:00:61) начнётся в (TMD) в нашем Центре Робототехники.

* Тест 5

Приветствую всех зрителей моего канала!

Я провожу раздачу промокодов на бесплатные обеды в местной столовке! Поэтому, с 12:00 до 13:55:00 в своём телеграм канале я буду переодически выкладывать промокоды. Не пропустите!

Шаблон промокода - 00:00:00:00:00:00:00

Приветствую всех зрителей моего канала!

Я провожу раздачу промокодов на бесплатные обеды в местной столовке! Поэтому, с (TMD) до (TMD) в своём телеграм канале я буду переодически выкладывать промокоды. Не пропустите!

Шаблон промокода - 00:00:00:00:00:00:00

### Программа

**import** **re**

**def** **readfile**(s):

tmp = open(s, encoding='utf8')

tmp1 = ''

**for** i **in** tmp.readlines():

tmp1 += i

tmp.close()

**return** tmp1

**def** **subtime**(s):

t1 = r'[0-1][0-9]:[0-5][0-9]:[0-5][0-9],'

tmp = re.sub(t1,"(TMD),", s)

t1 = r'2[0-3]:[0-5][0-9]:[0-5][0-9],'

tmp = re.sub(t1,"(TMD),", tmp)

t1 = r'[0-1][0-9]:[0-5][0:9],'

tmp = re.sub(t1,"(TMD),", tmp)

t1 = r'2[0-3]:[0-5][0:9],'

tmp = re.sub(t1,"(TMD),", tmp)

*# ---------------------------------------------*

t1 = r'[0-1][0-9]:[0-5][0-9]:[0-5][0-9]\.'

tmp = re.sub(t1,"(TMD).", tmp)

t1 = r'2[0-3]:[0-5][0-9]:[0-5][0-9]\.'

tmp = re.sub(t1,"(TMD).", tmp)

t1 = r'[0-1][0-9]:[0-5][0:9]\.'

tmp = re.sub(t1,"(TMD).", tmp)

t1 = r'2[0-3]:[0-5][0:9]\.'

tmp = re.sub(t1,"(TMD).", tmp)

*# ---------------------------------------------*

t1 = r'[0-1][0-9]:[0-5][0-9]:[0-5][0-9]\s'

tmp = re.sub(t1,"(TMD) ", tmp)

t1 = r'2[0-3]:[0-5][0-9]:[0-5][0-9]\s'

tmp = re.sub(t1,"(TMD) ", tmp)

t1 = r'[0-1][0-9]:[0-5][0:9]\s'

tmp = re.sub(t1,"(TMD) ", tmp)

t1 = r'2[0-3]:[0-5][0:9]\s'

tmp = re.sub(t1,"(TMD) ", tmp)

**return** tmp

print(subtime(readfile("dop1\_tests/dop1\_test.txt")))

print('')

print(subtime(readfile("dop1\_tests/dop1\_test1.txt")))

print('')

print(subtime(readfile("dop1\_tests/dop1\_test2.txt")))

print('')

print(subtime(readfile("dop1\_tests/dop1\_test3.txt")))

print('')

print(subtime(readfile("dop1\_tests/dop1\_test4.txt")))

print('')

print(subtime(readfile("dop1\_tests/dop1\_test5.txt")))

### Вывод программы

Уважаемые студенты! В эту субботу в (TMD) планируется доп. занятие на 2 часа. То есть в (TMD) оно уже точно кончится.

Привет, брат. Приеду я к (TMD), можете начинать без меня.

А завтра в (TMD) откроется запись на курс, не пропусти его.

ВНИМАНИЕ!!!

ЗАВТРА В (TMD) ВАСЯ НАЧНЁТ ИГРАТЬ В ДОТУ!

ПОЭТОМУ ДО (TMD) ОН БУДЕТ ОЧЕНЬ СИЛЬНО ОРАТЬ И КРУШИТЬ СВОЮ КВАРТИРУ!

БЕРЕГИТЕ СЕБЯ И СВОИХ БЛИЗКИХ!

Объявление для учащихся!

16 октября в (TMD) откроется тест, который вы должны пройти до (TMD).

На следующий день в (TMD) на сайте будут доступны результаты теста!

РОБОКОП 25:24:23 - новейшая модель нашего робота, вершащего справедливость!

Утилизация старых версий робота (05:70, 05:71, 00:00:60 и 00:00:61) начнётся в (TMD) в нашем Центре Робототехники.

Приветствую всех зрителей моего канала!

Я провожу раздачу промокодов на бесплатные обеды в местной столовке!

Поэтому, с (TMD) до (TMD) в своём телеграм канале я буду переодически выкладывать промокоды. Не пропустите!

Шаблон промокода - 00:00:00:00:00:00:00

**Как мы видим, окончательный текст без регулярных выражений и с ними совпадает.**

## Необязательное задание (+22)

1) Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице.

2) Для своей программы придумайте минимум 5 тестов.

3) Протестируйте свою программу на Тесте 2.

4) Можно использовать циклы и условия, но основной частью решения должны быть регулярные выражения.

368308%5 = 3

Изображение выглядит как текст

Автоматически созданное описание

**Тест 2**

### Тесты

* Тест из варианта

Петров П.П. P0000

Анищенко А.А. P33113

Примеров Е.В. P0000

Иванов И.И. P0000

* Тест 1

Петров-Васичкин П.П. M5144

Анищенко-Кабачок А.В. M5144

Примеров-Экземпляров В.В. M5144

Иванов-Романов И.Л. M51441

* Тест 2

Платочник П.П. N1211

Люберов-Виниц А.А. K757

Урик В.В. F1121

Шабилов И.И. L52

* Тест 3

Петров С.С. J5555

Анищенко А.А. J5555

Примеров У.У. J5555

Иванов Р.Р. J5555

* Тест 4

Курагин К.К. K918273

Ларин А.Р. P33113

Римский-Корсаков Е.В. P731

Паганини И.И. M4141

* Тест 5

Мусоргский М.П. P0001

Глинка М.И. P3112

Прокофьев С.С. P3112

Шостакович Д.Д. P0004

### Программа

**import** **re**

**def** **readfile**(s):

pattern1 = r'[ЙЦУКЕНГШЩЗФЫВАПРБОЛДЯЧСМИТЬ]+[йцукенгшщзфывапрболдячсмитью]+-[ЙЦУКЕНГШЩЗФЫВАПРБОЛДЯЧСМИТЬ]+[йцукенгшщзфывапрболдячсмитью]+'

pattern2 = r'[ЙЦУКЕНГШЩЗФЫВАПРБОЛДЯЧСМИТЬ]+[йцукенгшщзфывапрболдячсмитью]+'

pattern3 = r'[ЙЦУКЕНГШЩЗФЫВАПРБОЛДЯЧСМИТЬ]\.[ЙЦУКЕНГШЩЗФЫВАПРБОЛДЯЧСМИТЬ]\.'

pattern4 = r'[A-Z][0-9]+'

file = open(s, encoding='utf8')

s = file.readlines()

k = []

**for** i **in** s:

tmp = i.rstrip()

k.append(tmp)

**return** k

group = r'P0000'

groups = [r'M5144', r'N121', r'J5555', r'K918273', r'P3112']

s = readfile("dop2\_tests/dop2\_test.txt")

**for** i **in** s:

tmp = re.findall(r'(?<=[а-я]\s)[А-Я](?<!\.)', i)

tmp = r''+tmp[**0**]+'\.'+tmp[**0**]+'\.'

**if** **not** bool(re.search(tmp,i)) **or** **not** bool(re.search(group,i)):

print(i)

print('')

**for** j **in** range(**1**,**6**):

print(str(j), 'тест')

s = readfile("dop2\_tests/dop2\_test"+str(j)+".txt")

**for** i **in** s:

tmp = re.findall(r'(?<=[а-я]\s)[А-Я](?<!\.)', i)

tmp = r'' + tmp[**0**] + '\.' + tmp[**0**] + '\.'

**if** **not** bool(re.search(tmp, i)) **or** **not** bool(re.search(groups[j-**1**], i)):

print(i)

print('')

### Вывод программы

Анищенко А.А. P33113

Примеров Е.В. P0000

1 тест

Петров-Васичкин П.П. M5144

Анищенко-Кабачок А.В. M5144

Примеров В.В. M5144

Иванов-Романов И.Л. M51441

2 тест

Платочник П.П. N1211

Люберов-Виниц А.А. K757

Урик В.В. F1121

Шабилов И.И. L52

3 тест

Петров С.С. J5555

Анищенко А.А. J5555

Примеров У.У. J5555

Иванов Р.Р. J5555

4 тест

Курагин К.К. K918273

Паганини И.И. M4141

5 тест

Глинка М.И. P3112

Прокофьев С.С. P3112

Шостакович Д.Д. P0004

# Заключение

В результате выполненной работы я познакомился с python библиотекой re, позволяющей использовать регулярные выражения для форматирования и сортировки текста. Ознакомился с синтаксисом регулярных выражений.

# Литература

1. <https://regex101.com/>
2. [https://docs.python.org/3/library/re.html#](https://docs.python.org/3/library/re.html)