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»

Группа: P3131

Выполнил: Воронин И.А.

Проверил:

к.т.н. преподаватель Белозубов А.В.

Санкт-Петербург

2021г

Оглавление

[Задание 3](#_Toc85987900)

[Доп. Задание №1 4](#_Toc85987901)

[Доп. Задание 2 5](#_Toc85987902)

[Заключение 6](#_Toc85987903)

# Задание

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.

3) Программа должна считать количество смайликов определённого вида в предложенном тексте. Все смайлики имеют такую структуру:

[глаза][нос][рот].

Вариантом является различные наборы глаз, носов и ртов.

Вариант для номера ИСУ 333580: “ :-O”

**Код программы:**

**import** **re**

**print**("Количество смайликов ':-O' : " + str(len(re.findall ( ":-O" ,input()))))

**Примеры вывода:**

**Ввод:** :-O:-O:-Oasdasdassdads:-O1231312edw:-O123q2w

Количество смайликов ':-O' : 5

**Ввод:** 1:-O2:-O3:-O4:-O5:-O6:-O

Количество смайликов ':-O' : 6

**Ввод:** :-O

Количество смайликов ':-O' : 1

**Ввод:** hello

Количество смайликов ':-O' : 0

**Ввод:** :-o

Количество смайликов ':-O' : 0

# Доп. Задание №1

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.

3) (Вариант 4) Анатолий выложил пост с расписанием доп. занятий по информатике, но везде перепутал время. Поэтому нужно заменить все вхождения времени на строку (TBD).

Время – это строка вида HH:MM:SS или HH:MM, в которой HH – число от 00 до 23, а MM и SS – число от 00 до 59.

**Код программы:**

**import** **re**

**print**(re.sub("\d{2}[:]\d{,2}[:]{,1}\d{,2}",

'(TBD)',input()))

**Примеры вывода:**

**Ввод:** Hello, do you want to meet today at 15:12:10?

Hello, do you want to meet today at (TBD)?

**Ввод:** Time: 20:10

Time: (TBD)

**Ввод:** 15:51:12:00:41:43:59:11:04:23asd:15

(TBD):(TBD):(TBD):23asd:15

**Ввод:** 17:51 12:12:51:42

(TBD) (TBD):42

**Ввод:** 05 12 13 1:31 03:45

05 12 13 1:31 (TBD)

# Доп. Задание 2

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

2) Для своей программы придумайте минимум 5 тестов.

3) (Вариант 0) Написать регулярное выражение, которое проверяет корректность email и в качестве ответа выдаёт почтовый сервер (почтовый сервер – часть email идущая после «@»).

Для простоты будем считать, что почтовый адрес может содержать в себе буквы, цифры, «.» и «\_», а почтовый сервер только буквы и «.». При этом почтовый сервер, обязательно должен содержать верхний уровень домена («.ru», «.com», etc.)

**Код программы:**

**import** **re**

a = re.findall('[^@]+[@]\[^@,\D]{1,50}[.]\D{1,10}',

input())

**if** len(a) != **0**:

**print**(a[**0**])

**else**:

**print**('Fail!')

**Примеры вывода:**

**Ввод:** belozubov@itmo.ru

@itmo.ru

**Ввод:** belozubovitmo.ru

Fail!

**Ввод:** belozubov@itmo

Fail!

**Ввод:** @itmo.ru

Fail!

**Ввод:** asd@asd@asd.ru

Fail!

# Заключение

В ходе выполнения данной лабораторной работы, я научился пользоваться регулярными выражениями на языке программирования Python и создавать для них тесты.