**“Университет ИТМО”**

Факультет программной инженерии

И компьютерных технологий.

Лабораторная работа №3

Регулярные выражения

Выполнил

Студент группы Р31112

Шаматульский Роман Константинович

Преподаватель

Рыбаков Степан Дмитриевич

Санкт-Петербург, 2023

Оглавление

[Задания 3](#_Toc148272519)

[Основные этапы вычисления 4](#_Toc148272520)

[Вывод 6](#_Toc148272521)

[Источники 7](#_Toc148272522)

# Задания

1. Программа должна считать число смайликов определённого вида “*=-{P”* (вариант №426) в предложенном тексте. Все смайлики имеют такую структуру: [*глаза*][*нос*][*рот*].
2. 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 (вариант №4).
3. 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.) (вариант №0).

# Основные этапы вычисления

1. Решение с помощью python:

# smile: 426 - =-{P

# some other smiles: :-( :-) :-O X<| X-{P =-O ;<{/ 8<{)

import re

test\_exps = ['', '=-{P=-{P8<{):-O=-{PX<|=-{P', ':-(:-(:-)X-{P=-O', '8<{)=-{PX<|X<|=-O', 'X<|X-{P=-O;<{/8<{)=-{P:-(:-):-OX<|']

for i in test\_exps:

print(len(re.findall('=-{P', i)))

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| *Пустая строка* | 0 |
| =-{P=-{P8<{):-O=-{PX<|=-{P | 4 |
| :-(:-(:-)X-{P=-O | 0 |
| 8<{)=-{PX<|X<|=-O | 1 |
| X<|X-{P=-O;<{/8<{)=-{P:-(:-):-OX<| | 1 |

1. Решение с помощью python:

# isu % 6 -> 4

import re

test\_exps = ['Уважаемые студенты! В эту субботу в 15:00 планируется доп. занятие на 2 часа. То есть в 17:00:01 оно уже точно кончится.',

'Cамолёт вылетит ровно в 23:59, в 00:00:00 уже будет поздно',

'Робин заказал кофе в комнату 17:52:566, но официант ошибся и пришел в комнату 17:52:60, что была напротив 17:52:60:',

'Старый счётчик загадочно показывал 09:14:, хотя на обратной стороне мы увидели 09:14 без двоеточия в конце',

'На часах было уже 15:38:09, - это значило, что мы опоздали на встречу с Мартином в 15:00.']

for i in test\_exps:

print(re.sub(r'\b([01][0-9]|2[0-3])(:[0-5]\d){1,2}(?![:\d])\b', '(TBD)', i))

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 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 (TBD) планируется доп. занятие на 2 часа. То есть в (TBD) оно уже точно кончится. |
| Cамолёт вылетит ровно в 23:59, в 00:00:00 уже будет поздно | Cамолёт вылетит ровно в (TBD), в (TBD) уже будет поздно |
| Робин заказал кофе в комнату 17:52:566, но официант ошибся и пришел в комнату 17:52:60, что была напротив 17:52:60: | Робин заказал кофе в комнату 17:52:566, но официант ошибся и пришел в комнату 17:52:60, что была напротив 17:52:60: |
| Старый счётчик загадочно показывал 09:14:, хотя на обратной стороне мы увидели 09:14 без двоеточия в конце | Старый счётчик загадочно показывал 09:14:, хотя на обратной стороне мы увидели (TBD) без двоеточия в конце |
| На часах было уже 15:38:09, - это значило, что мы опоздали на встречу с Мартином в 15:00. | На часах было уже (TBD), - это значило, что мы опоздали на встречу с Мартином в (TBD). |

1. Решение с помощью python:

# isu % 5 -> 0

import re

s = input()

correct = re.fullmatch(r'([\w.\_])+@[a-z]+[.][a-z]+', s) # проверка на соответсвие шаблону

if bool(correct):

print(re.split('@', s)[1]) # если соот-вие подтвердилось, выводим почтовый сервер

else:

print('Fail!')

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| aboba@yandex.ru | yandex.ru |
| \_\_\_..@gmail.com | gmail.com |
| tony\_h@yahoo | Fail! |
| robin\_a@.com | Fail! |
| jack\_brown\_work.acc@bing.us | bing.us |

# Вывод

В ходе выполнения данной лабораторной работы я научился составлять регулярные выражения с использованием синтаксиса python. Полученные мной в ходе выполнения данной работы знания лягут в основу моего дальнейшего изучения информатики.

# Источники

1. Регулярные выражения в python: от простого к сложному. [Электронный ресурс]. – URL: <https://habr.com/ru/articles/349860/> (дата обращения: 15.10.2023).
2. Re – regular expression operations. [Электронный ресурс]. – URL: https://docs.python.org/3/library/re.html (дата обращения: 15.10.2023).
3. Регулярные выражения в python. [Электронный ресурс]. – URL: <https://skillbox.ru/media/code/regulyarnye-vyrazheniya-v-python-sintaksis-poleznye-funktsii-i-zadachi/> (дата обращения 15.10.2023).