ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №3**

**«Регулярные выражения»**

по дисциплине «Информатика»

Вариант № 367468

**Группа:** P3112

**Выполнил**: Пархоменко К. А.

**Проверил:** Белозубов А.В.

г. Санкт-Петербург

2022 г.

Оглавление

[Задания 3](#_Toc118839147)

[Задание на 60 баллов (Смайлики) 3](#_Toc118839148)

[Необязательное задания для получения оценки «4» или «5» (позволяет набрать +18 баллов от максимального числа баллов БаРС за данную лабораторную) 4](#_Toc118839149)

[Необязательное задания для получения оценки «4» или «5» (позволяет набрать +22 балла от максимального числа баллов БаРС за данную лабораторную) 4](#_Toc118839150)

[Заключение 6](#_Toc118839151)

[Список литературы 6](#_Toc118839152)

# Задания

# Задание на 60 баллов (Смайлики)

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 (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.

Так как мой номер в ИСУ – 367468, то смайликом для моего варианта является следующая последовательность символов: **=-|** (403).

from re import compile

import configparser

from TestGenerator import TestGenerator

"""

4: =

0: -

3: |

"""

cfg = configparser.ConfigParser()

generator = TestGenerator()

cfg.read('TESTS.ini')

# SMILE = input()

SMILE = "=-|"

SMILE\_PATTERN = compile(r"=-\|")

for i in range(1, 6):

with open(f'./TestsT1/test{i}', 'r', encoding='utf-8') as TEST:

TEST\_DATA = TEST.read()

print(f"TEST {i}\nТестовая строка: \"{TEST\_DATA}\"")

print(f"Верный ответ: {cfg['TUSK1'][f'TEST{i}']}")

print(f"Ответ полученный через RegExp: {len(SMILE\_PATTERN.findall(TEST\_DATA))}\n")

print("АВТОТЕСТЫ\n")

for i in range(1, 10):

TEST\_DATA, TEST\_ANSWER = generator.smile\_test(SMILE)

print(f"AUTO-TEST {i}\nТестовая строка: \"{TEST\_DATA}\"")

print(f"Верный ответ: {TEST\_ANSWER}")

print(f"Ответ полученный через RegExp: {len(SMILE\_PATTERN.findall(TEST\_DATA))}\n")

Программа на задания номер 1

# Необязательное задания для получения оценки «4» или «5» (позволяет набрать +18 баллов от максимального числа баллов БаРС за данную лабораторную)

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

Так как 367468 даёт в остатке 4 при делении на 6, то моему варианту соответствует задание номер 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»

from re import compile, sub

TIME\_PATTERN1 = compile(r'(([0-1][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9])')

TIME\_PATTERN2 = compile(r'(([0-1][0-9]|2[0-3]):[0-5][0-9])')

for i in range(1, 6):

with open(f'./TestsT2/test{i}', 'r', encoding='utf-8') as TEST:

TEST\_DATA = TEST.read()

print(f"Тест {i}:\n\"{TEST\_DATA}\"")

print(f"Ответ:\n\"{sub(TIME\_PATTERN2, '(TBD)', sub(TIME\_PATTERN1, '(TBD)', TEST\_DATA))}\"\n")

Программа на задания номер 2

# Необязательное задания для получения оценки «4» или «5» (позволяет набрать +22 балла от максимального числа баллов БаРС за данную лабораторную)

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

Так как 367468 даёт в остатке 3 при делении на 5, то моему варианту соответствует задание номер 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. Могут существовать двойные фамилии, которые тоже нужно учитывать (студенты с такими фамилиями тоже должны иметь право быть удаленными из списка стипендиатов текущего семестра).»

from re import compile, findall

import configparser

from TestGenerator import TestGenerator

cfg = configparser.ConfigParser()

generator = TestGenerator()

cfg.read("./TESTS.ini")

NAME\_PATTERN = compile(r'(([A-Z]|[А-Я])\.)\1')

def test(test\_data, group\_pattern):

global NAME\_PATTERN

c = 0

for row in test\_data:

print(row)

print("Ответ:")

for row in test\_data:

if len(findall(group\_pattern, row)) and len(findall(NAME\_PATTERN, row)):

continue

print(row)

print()

for i in range(1, 6):

with open(f'./TestsT3/test{i}', 'r', encoding='utf-8') as TEST:

TEST\_DATA = [i.replace("\n", "") for i in TEST]

GROUP\_PATTERN = compile(rf'{cfg["TUSK3"][f"TEST{i}"]}$')

print(f"Тест {i}:")

test(TEST\_DATA, GROUP\_PATTERN)

print("АВТО-ТЕСТЫ")

for i in range(1, 6):

data, group = generator.studs\_test()

print(f"AUTO-TEST {i}")

test(data, group)

Программа на задания номер 3

# Заключение

В ходе выполнения лабораторной работы номер 3 «Регулярные выражения» было написано 3 программы на языке Python с использованием регулярных выражений, для решения различных задач.

# Список литературы

[В Интернете].

**Python RegEx: практическое применение регулярок** [В Интернете] // tproger. - 19 Январь 2021 г.. - 8 Ноябрь 2022 г.. - https://tproger.ru/translations/regular-expression-python/.

**Регулярные выражения в Python от простого к сложному. Подробности, примеры, картинки, упражнения** [В Интернете] / авт. ShashkovS // habr. - 26 Февраль 2018 г.. - 8 Ноябрь 2022 г.. - https://habr.com/ru/post/349860/.