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 №367786

Выполнил:

Агаев Хамза Рустам оглы

Группа: P3234

Проверил:

Балакшин Павел Валерьевич

г. Санкт-Петербург, 2023

**СОДЕРЖАНИЕ**

[Задание 3](#_Toc149743490)

[Основные этапы вычисления 5](#_Toc149743491)

[Задание 1. 5](#_Toc149743492)

[Дополнительное задание №1. 7](#_Toc149743493)

[Заключение 10](#_Toc149743494)

[Список использованных источников 11](#_Toc149743495)

Задание

Обязательное задание.

Исходя из номера варианта, получаем следующий смайлик: «=-{P».

Для выполнения обязательного задания, необходимо:

1. Реализовать программный продукт на языке Python, используя регулярные выражения по варианту.
2. Для своей программы придумать 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.  
   Вариант смайлика: «=-{P».

Дополнительное задание №1.

1. Реализовать программный продукт на языке Python, используя регулярные выражения по заданию.
2. Для своей программы придумать 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 1.
3. Можно использовать циклы и условия, но основной частью решения должны быть регулярные выражения.

Текст задания:

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.

Таблица - Пример для дополнительного задания №1

|  |  |
| --- | --- |
| 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. |

Основные этапы вычисления

Задание 1.

Для этого задания было придумано 5 тестов, содержащих смайлик «=-{P». Для начала посчитаем вручную количество смайликов в каждом из тестов.

Тест №1.

#!OCiH%hl+WP8@rn\_a=-{PA=-{P

Как видно из теста, в тексте содержится 2 смайлика.

Тест №2.

N;+)mZ\*ZR610TU'BzNIH?IA(hAR^eOTyi}:l&0Vl~=-{PMG=-{P<4xY+z)3x=-{PX=^@O(L[I|^{pEV5N')@ee`9}:Dqj[l=-{Pd

Как видно из теста, в тексте содержится 4 смайлика.

Тест №3.

%./u8n@TYR<Pgei8h/AQUWL8OFU!5pI>]q)^=nCQ:toDf`U(p&\_Xqn{^q=yqg)x%~#>hB'mt\*\_w3^w|?D:rFuZFK=hrU.NFo=-{PCQJV\*f0LF>dA9/vPAs:)x)jy[{fFMeZ2'K`|KPEzbJ!-iDlP\_;Z\*`B,wI<L-:EYlG/di<Wm$nAHv9B|'\!<{9jiy;?1%;=-{PY\_#PB!U=-{PppQBS-,7C=-{P:zQ[

Как видно из теста, в тексте содержится 4 смайлика

Тест №4.

h2X4}eD7G(X\_-anI)+\*xo76Skl9Xj=-{PR\_A'^<LO`/osLc\*]r"BT^V=8gX%71lP5We.KmTGO3mk%Z%h"EIA?Bu-nzL&"sw"QLI2NT->S"tG66y|s$#Qc}mI'=ki',q73rqjTu7\_;uJ8(mHXun;Z:7\*CV-i4u{IksP+|\*`w5b+Zb~PeC\_qmC+qqs^g2G\;7z+m7$X;{}\*LR<IE\*le")Y13gpX2m/)Lntu!W8O[C?)\_+\*e\-{PJ(`L]z1dMfP\*})+WPSc!.KKSIW|"yGU;]]UjJ`E6\_S/>&>ji{B=M8BBw?"KL!1>8kBP;u~mGp=-{PEa;o)`ey@ky`&4a.9lrHT,a@Ud![L[r<<C9l,ajndmbKp-Rs5o4nCt;]SnUrEr+=oK^n)51"=-{P!HD2b@

Как видно из теста, в тексте содержится 3 смайлика.

Тест №5.

mgz7gN7Kd0[vm>I,S9WKpMyak=8JlC.hCY"bu56i7z[,SwVu\*OlqTuNBkR~J,vB!NHU{];NT?tC2zp!i';!KVa|Nt\*|WI\_"D&0\_=;jM;QReBv}\_iMetE:jz?tmym[anK~BE-G`:k.0J\_"{6+"A%.XHBFujz)!<>!+%MK-GIPn6s}@equj\*$)`q{a(X\v`C\|QI^3X/m1Be%#&zd+}\*ZI:FdOZYM~;SP|\_bjF9Hcw~uzqEi'9YJD35"{!'WwyxWH@.u]fouxaPQ.+UeuqS4dj@GOe!/=yq>cdiO)q,wwWyT@p:r[>=-{PQd>OcE!(4,4~IUEDH8~3(~2x0\'{\npgZ.nnn:;wH(pWbWT&7H"WM<SD~wY\_|mk=Bqh!\_mMY]gL]E][7"I+1nS\k]1CU}Z=-{PU5^p!x`USO\_O]=W7aO=-{PE3TJlGyjac\*PK%5R-nQi`~W]~TMRs3mmSnLe8=(|'Lnq<iT0yrbJH<k|d&>t]MZlvnKg8.@va4}?fll.Dlq\*.L3`o9[F?'"3U4DxJ1L\_P4-AJKt#%(Akb>,CI.96Q4A&rHy=-{Pkefpe}ki=@x!VAf:A1grq)=-{PpA5z.D+QVz-=!MWY/E-Q20E6xM;=-{P=-{Pr

Как видно из теста, в тексте содержится 7 смайликов.

Чтобы посчитать количество смайликов в текстах, была написана следующая программа на Python с использованием библиотеки re. Она находит непересекающиеся строки “=-{P” в каждом тесте, и считает их количество.

import re

smile\_pattern = re.compile(r"=-{P")

print("Смайлик: =-{P")

with open("tests1.txt", "r") as read\_file:

    all\_tests\_str = read\_file.read()

all\_tests\_list = all\_tests\_str.split("\n")

for i in range(len(all\_tests\_list)):

    test\_str = all\_tests\_list[i]

    match\_test = re.findall(smile\_pattern, test\_str)

    print(f"Количество смайликов в {i+1} строке: {len(match\_test)}")

Поместим все тесты в текстовый файл и передадим его программе. В результате получим следующий вывод (рисунок 1).

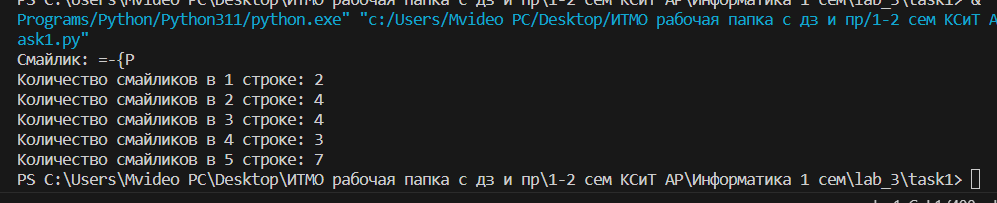


Рисунок - Результат работы программы для задания №1

Получается, что наша программа работает правильно, потому что подсчитанное вручную количество смайликов в каждом тесте совпало с количеством смайликов, подсчитанными программой.

Дополнительное задание №1.

Для этого задания было придумано 5 тестов, содержащих временные метки. Для начала найдем вручную все временные метки в каждом из тестов и заменим их на строку «(TBD)».

Тест №1.

Уважаемые студенты! Консультация 9-го числа с 14:00 переносится на 2 часа, ровно в 16:00:00 всем быть в аудитории 3316!

Найдено 2 временные метки. Заменим на «(TBD)».

Уважаемые студенты! Консультация 9-го числа с (TBD) переносится на 2 часа, ровно в (TBD) всем быть в аудитории 3316!

Тест №2.

18:00 - занятие заканчивается, затем перерыв 1 час!

Найдено 1 временная метка. Заменим на «(TBD)».

(TBD) - занятие заканчивается, затем перерыв 1 час!

Тест №3.

2001:2004 - студенты этих годов рождения могут прийти на следующий день в 16:30, не опаздывайте!

Найдено 1 временная метка. Заменим на «(TBD)».

2001:2004 - студенты этих годов рождения могут прийти на следующий день в (TBD), не опаздывайте!

Тест №4.

Время с миллисекундами - 22:23:24:25, время без миллисекунд - 22:23:24.

Найдено 1 временная метка. Заменим на «(TBD)».

Время с миллисекундами - 22:23:24:25, время без миллисекунд – (TBD).

Тест №5.

Плохо указывать время без пробела -22:34, хорошо с пробелом - 22:34.

Найдено 1 временная метка. Заменим на «(TBD)».

Плохо указывать время без пробела -22:34, хорошо с пробелом – (TBD).

Тест №6.

В 22:20 или в 32:30, 16:50

Найдено 2 временные метки. Заменим на «(TBD)».

В (TBD) или в 32:30, (TBD)

Чтобы заменить все временные метки в тестах на строку «(TBD)», была также написана программа на Python с использованием библиотеки re. Она выбирает такие подстроки формата “HH:MM”, “HH:MM:SS”, перед которыми стоит либо пробел, либо начало строки и после которых не стоит буква или цифра и/или стоит пробел или конец строки. Так можно точно определить, что данная строка является временной меткой.

import re

def editTime(text: str) -> str:

    before\_time = r"(?P<before>^|\s)"           # символы, которые могут находиться перед временем

    time = r"([0-1]\d|2[0-3])(:[0-5]\d)?(:[0-5]\d)"

    after\_time = r"(?P<after>(\W$|\W\s|\s|$))"    # символы, которые могут находиться после времени

    time\_pattern = re.compile(before\_time + time + after\_time)

    match\_test = re.search(time\_pattern, text)

    if match\_test != None:

        result = re.sub(time\_pattern, match\_test.group("before") + r"(TBD)" + match\_test.group("after"), text)

    else:

        result = text

    return result

with open("tests2.txt", "r", encoding="utf-8") as read\_file:

    all\_tests\_list = read\_file.read().split("\n")

    for i in range(len(all\_tests\_list)):

        test\_str = all\_tests\_list[i]

        res = editTime(test\_str)

        print(res)

Результат работы данной программы представлен ниже на рисунке 2.

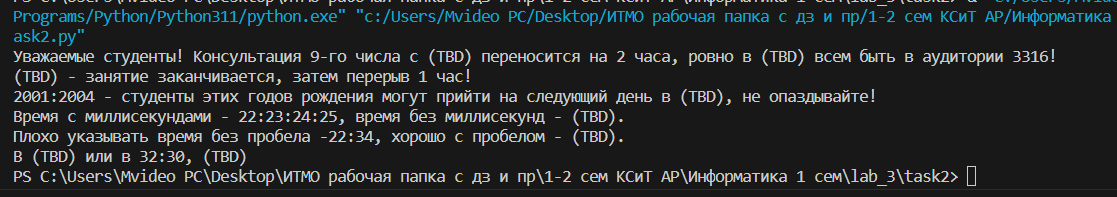


Рисунок - Результат работы программы для задания №2

Как можно заметить, программа работает правильно и выдает тот же результат, что и рассчитанный вручную.

Заключение

В ходе лабораторной работы была проведена работа с регулярными выражении при использовании модуля re в языке программирования Python. Было проведено сравнение результатов, выданных программой, использующей регулярные выражения, и расчётов, сделанных вручную.

Список использованных источников

1. Балакшин Е.А., Соснин П.В., Машина В.В. Информатика. – СПб: Университет ИТМО, 2020.