

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3

Регулярные выражения

Вариант №512 / 5 / 4

Выполнил: студент группы Р3108
Васильев Никита

Проверил: Балакшин Павел
Валерьевич, доцент факультета
ПИиКТ, кандидат технических наук

Санкт-Петербург 2023

Содержание

Задание	3
Основные этапы вычисления	4
Задание №1	4
Реализация на языке программирования Python	5
Дополнительное задание №1	5
Реализация на языке программирования Python	6
Дополнительное задание №2	7
Реализация на языке программирования Python	8
Заключение.....	9
Список литературы	10

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. Все 5 тестов необходимо показать при защите.
3. Программа должна считать число смайликов определённого вида (вид смайлика описан в таблице вариантов) в предложенном тексте. Все смайлики имеют такую структуру: [глаза][нос][рот].

Номер в ИСУ % 6	Глаза	Номер в ИСУ % 4	Нос	Номер в ИСУ % 7	Рот
0	:	0	-	0	(
1	;	1	<	1)
2	X	2	-{	2	O
3	8	3	<{	3	
4	=			4	\
5	[5	/
				6	P

Дополнительное 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. Все 5 тестов необходимо показать при защите. Пример тестов приведён в таблице.
3. Можно использовать циклы и условия, но основной частью решения должны быть регулярные выражения.

Дополнительное 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. Все 5 тестов необходимо показать при защите.
3. Протестируйте свою программу на этих тестах.
4. Можно использовать циклы и условия, но основной частью решения должны быть регулярные выражения.

Основные этапы вычисления

Задание №1

ИСУ ID = 366389

1. $366389 \% 6 = 5$
2. $366389 \% 4 = 1$
3. $366389 \% 7 = 2$

Вариант: 512

Смайлик: [<O

Тест 1: 88\(<)POP=<{([<O;:- (8<{/// #1 совпадение

Тест 2: X-|<<{\|/[<O[<O[<O=<OO[<O #4 совпадения

Тест 3: ;<{---\P[</*-P=-{|:<OXXXO[<O=-P #1 совпадение

Тест 4: X[<<{\O[<O[<O<O<O[[8-O #2 совпадения

Тест 5: =<{):<P;-{[<O8<{O8<{OX-P=-) #1 совпадение

Реализация на языке программирования Python

```
import re

ISU_ID = 366389

var_1 = ISU_ID % 6
var_2 = ISU_ID % 4
var_3 = ISU_ID % 7

var = str(var_1) + str(var_2) + str(var_3)
# var: 512
print("Вариант:", var)

smile = "\[<0"
print("Смайлик:", smile[1:])

def count(test):
    match = re.findall(smile, test)
    print(len(match))

count("88[<)POP=<{ ( [<0; :- ( 8<{ ")
count("X- |<<{ \ | / [<0 [<0 [<0=<00 [<0 ")
count("; <{ ---P [ </ *-P = - { | : <0XXX0 [ 0 = -P ")
count("X [ <<{ 0 [ <0 [ <0 <0 <0 [ [ 8-0 ")
count("=<{ ) : <P; - { [ <08 <{ 08 <{ 0X-P = - ")
```

Рисунок 1 – Код для задания №1

Дополнительное задание №1

ИСУ ID = 366389

$366389 \% 6 = 5$

Вариант: 5

Задание: с помощью регулярного выражения найти в тексте все слова, в которых две гласные стоят подряд, а после этого слова идёт слово, в котором не больше 3 согласных.

Тест 1: Солнечное лето наступило, и каждый день становится все ярче, как будто вся природа улыбается. #1 совпадение

Тест 2: Маленький кот мяукал на улице, пес радостно лаял без остановки, а птицы пели звонкие песни. #2 совпадения

Тест 3: Красочные осенние листья шуршат под ногами, напоминая о приближении холодов.

#3 совпадения

Тест 4: Мелодия ветра в горах унесла мои заботы, оставив лишь спокойствие и вдохновение.

#2 совпадения

Тест 5: Сияние полной луны озаряло ночное небо, создавая неповторимую атмосферу магии и загадочности. *#1 совпадение*

Реализация на языке программирования Python

```
import re

ISU_ID = 366389

print("Вариант:", ISU_ID % 6)

vowels = 'АаИиОоУуЫыЭэЕеЁёЮюЯя'
consonants = 'БбВвГгДдЖжЗзЙйКкЛлМмНнПпРрСсТтФфХхЦцЧчШшЩщ'

def text(test):
    pattern_first = fr"\b\w*[{vowels}]+[{vowels}]\w*\b"
    pattern_second = fr"\b^(?![{consonants}]).*[{consonants}].*[{consonants}].*[{consonants}]).*$\b"
    words = test.split()
    for i in range(len(words) - 1):
        current_word = words[i]
        next_word = words[i + 1]
        if re.match(pattern_first, current_word, re.IGNORECASE):
            if re.match(pattern_second, next_word, re.IGNORECASE):
                print(current_word)
    print("---")

text("Солнечное лето наступило, и каждый день становится все ярче, как будто вся природа улыбается.")
text("Маленький кот мяукал на улице, пес радостно лаял без остановки, а птицы пели звонкие песни.")
text("Красочные осенние листья шуршат под ногами, напоминая о приближении холодов.")
text("Мелодия ветра в горах унесла мои заботы, оставив лишь спокойствие и вдохновение.")
text("Сияние полной луны озаряло ночное небо, создавая неповторимую атмосферу магии и загадочности.")
```

Рисунок 2 - Код для дополнительного задания №1

Дополнительное задание №2

ИСУ ID = 366389

$366389 \% 5 = 4$

Вариант: 4

Задание: необходимо выбрать три любых буквы и расстояние между ними. С помощью регулярного выражения нужно найти все слова (последовательность символов ограниченная пробелами), в которых встречаются эти буквы в заданной последовательности и расстоянии (например, через один друг от друга).

Маска: буквы А, Е, Б через две строго в такой последовательности.

Тест 1: иконопечатание мурАвьЕлюБ баскетбол *#1 совпадение*

Тест 2: саблезуб бакенбарды ораниенбаумец *#0 совпадений*

Тест 3: АкнЕфоБия камнедобыча кАмнЕотБор *#2 совпадения*

Тест 4: гАрдЕроБ кАндЕляБр лось *#2 совпадение*

Тест 5: амеба усадьба чаесборщик *#0 совпадений*

Реализация на языке программирования Python

```
import re

ISU_ID = 366389

print("Вариант:", ISU_ID % 5)
print("---")

def words(test):
    pattern = r'\b([аеё]*a[аеё]{2}e[аеё]{2}6[аеё]*)\b'
    matches = re.findall(pattern, test, re.IGNORECASE)
    for match in matches:
        words = match.split()
        for word in words:
            if re.match(pattern, word, re.IGNORECASE):
                print(word)
    if matches != []:
        print("---")

words("иконопечатание муравьялюб баскетбол")
words("саблезуб бакенбарды ораниенбаумец")
words("акнефобия камнедобыча камнеотбор")
words("гардероб канделябр лось")
words("амеба усадьба чаесборщик")
```

Рисунок 3 - Код для дополнительного задания №2

Заключение

В ходе выполнения данной лабораторной работы я научился применять регулярные выражения на языке программирования Python, познакомился со специальными методами и метасимволами.

Список литературы

Джеффри Фридл Регулярные выражения [Книга] / ред. А. Галунов / перев. Матвеева Е. Киселева А.. - Санкт-Петербург - Москва : СИМВОЛ, 2008. - 3-е : стр. 598. - 9785932861219.

Лопес Феликс, Ромеро Виктор Освоение регулярных выражений Python [Книга]. - Бирмингем : Packt Publishing, 2014. - стр. 110. - 9781783283156.