

Санкт-Петербургский Национальный Исследовательский
Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3

“Регулярные выражения”

По дисциплине

“Информатика”

Вариант 373750

Выполнил:

Студент группы Р3117

Пономарёв М. И.

Преподаватель:

Машина Е. А.



Оглавление

Задание	3
Основные этапы вычисления	4
Вывод	6

Задание

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.

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

Пример смайлика: 8<{P

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.

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.	
	Пример:	
	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.

Необязательные 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.

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.)	
	Пример:	
	Ввод	Вывод
	students.spam@yandex.ru	yandex.ru

example@example	Fail!
example@example.com	example.com

Основные этапы вычисления

- Задание №1

```
import re
```

```
# isu_number = 373750
regexp = r":-{P"
```

```
def testRegexp(value):
    match = re.findall(regexp, value)
    print(len(match))
```

```
testRegexp(":-{p :-{P :-{"")
testRegexp(":::---{{P")
testRegexp(":-{p :-{P test")
testRegexp(":-{P :-{P :-{P")
testRegexp(":-uwupP%<{{P\/"")
```

- Задание №2

```
import re
```

```
# isu_number = 373750
```

```
def testRegexp(value):
```

```
    print(re.sub("([0-1][0-9]|2[0-3]):[0-5][0-9](?:[0-5][0-9])?",
                  '(TBD) ', value))
```

```
testRegexp("""Важные действия:
```

```
1. До конца месяца (30.09.22 пятница 23:59 мск) заполнить форму:
```

```
https://forms.yandex.ru/u/6331fd47de885a5baaf254c9/
```

```
Таким образом вам будет зачтено посещение лекции 13.09.22. Время заполнения  
занимает 5-7 минут.
```

```
2. Для набора баллов за 1-ю аннотацию и за набор баллов за посещение лекции  
27.09.22 есть 2 варианта:
```

```
2.1. 27.09.22 с 00:00 до 23:59 мск прислать аннотацию на почту pybalakshin AT itmo.ru
```

```
2.2. Сдать очно на занятии 11.10.22. При этом дату сдачи по-прежнему писать  
27.09.22.
```

```
3. По итогам лекции №2 от 27.09.22 вам надо будет сделать аннотацию №2 и  
сдать её на лекции №3 11.10.22.
```

```
4. Всего в рамках дисциплины надо будет сделать 6 лабораторных работ.  
Варианты заданий описываются в самих работах. Ждать выдачи варианта не надо.  
Лабораторные надо именно защищать на занятиях, а выполнять и готовиться к  
защите – дома.
```

```
5. Проверьте, что ваши одногруппники подписаны на канал. Не хватает 35  
человек.""")
```

```
testRegexp("60:00:00 10:70:99 10:59:60 10:15 15:00:00")
```

```
testRegexp("00:00:00:00:00:10:20:30:40:50:60:59:58")
```

```
testRegexp("24:24:24 24:24:24 23:23:23")
```

```
testRegexp("")
```

- Задание 3

```
import re

zero_to_255 = "([01]?[0-9][0-9]?|2[0-4][0-9]|25[0-5])"

# isu_number = 373750
def testRegexp(value):
    result = re.search(r"^[a-zA-Z0-9._]+@(?P<domain>([a-z]+\.)+[a-z]+)|((%s\.){3,%s})$" % (
        zero_to_255, zero_to_255), value)
    try:
        print(result.group("domain"))
    except AttributeError:
        print("Fail!")

testRegexp("students.spam@yandex.ru")
testRegexp("373750@niu.itmo.ru")
testRegexp("yandex@work.yandex.ru.something")
testRegexp("1234@gmail.com")
testRegexp("yandex.ru@192.168.0.9")
testRegexp("yandex.ru@255.25.100.99")
testRegexp("yandex.ru@192.168.0.9")

testRegexp("example@example")
testRegexp("this_is_illegal_$_@gmail.com")
testRegexp("yandex.ru@@test.com")
testRegexp("yandex.ru@192.168.0.0000")
testRegexp("yandex.ru@a.168.0.0")
testRegexp("yandex.ru@256.168.0.9")
```

Вывод

В процессе выполнения задания я использовал язык программирования Python и его встроенную библиотеку "re", а также использовал регулярные выражение для нахождения шаблонов в тексте.