

Практическая работа №14

Тема: составление программ с использованием регулярных выражений в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием с использованием регулярных выражений в IDE PyCharm Community.

Постановка задачи

Из исходного текстового файла (ip_address.txt) из раздела «Частоупотребимые маски» перенести в первый файл строки с нулевым четвертым октетом, а во второй – все остальные. Посчитать количество полученных строк в каждом файле.

Текст программы:

```
# Вариант - 5

# Из исходного текстового файла (ip_address.txt) из раздела «Частоупотребимые
# маски» перенести в первый файл строки с нулевым четвертым октетом, а во второй
# - все остальные. Посчитать количество полученных строк в каждом файле.

import re

ip_pattern_with_zero = re.compile(r'^\d+\.\d+\.\d+\.0$', re.S)
ip_pattern_other = re.compile(r'^\d+\.\d+\.\d+\.\d+', re.S)

result_table = []
second_result_table = []
with open('ip_address.txt', 'r', encoding='utf-8') as source:
    b_find = False
    for i in source:
        str_find = i.find('Частоупотребимые маски')
        if str_find != -1 and b_find == False:
            b_find = True

        str_find_1 = i.find('Количество адресов подсети не равно количеству
возможных узлов. Нулевой IP-адрес резервируется для идентификации подсети,
последний – в качестве широковещательного адреса. Таким образом, в реально
действующих сетях возможно количество узлов на два меньше количества адресов.')

        if str_find_1 != -1:
            break

        if b_find:
            res_ip_pattern_with_zero = ip_pattern_with_zero.findall(i)
            if res_ip_pattern_with_zero:
                result_table.append(res_ip_pattern_with_zero)
            else:
                res_ip_pattern_other = ip_pattern_other.findall(i)

                if res_ip_pattern_other:
                    second_result_table.append(res_ip_pattern_other)

with open('result_zero.txt', 'w') as location:
    count = 0
    for i in result_table:
        location.writelines('{}\n'.format(i))
        count += 1

print('В первом файле', count, 'строк')
```

```
with open('result.txt', 'w') as location:
    count = 0
    for i in second_result_table:
        location.writelines('{}\n'.format(i))
        count += 1

    print('Во втором файле', count, 'строк')
```

Файл:

Маски и размеры подсетей

Частоупотребимые маски

255.255.128.0
255.255.192.0
255.255.224.0
255.255.240.0
255.255.248.0
255.255.252.0
255.255.254.0
255.255.255.0
255.255.255.128
255.255.255.192
255.255.255.224
255.255.255.240
255.255.255.248
255.255.255.252

Протокол работы программы:

В первом файле 8 строк

Во втором файле 6 строк

Process finished with exit code 0

1.

```
['255.255.128.0']  
['255.255.192.0']  
['255.255.224.0']  
['255.255.240.0']  
['255.255.248.0']  
['255.255.252.0']  
['255.255.254.0']  
['255.255.255.0']
```

2.

```
['255.255.255.128']  
['255.255.255.192']  
['255.255.255.224']  
['255.255.255.240']  
['255.255.255.248']  
['255.255.255.252']
```

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с использованием регулярных выражений в IDE PyCharm Community. Были использованы языковые конструкции `print`, `with open`, `import`. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.