

## Практическое занятие № 10

**Тема:** Составление программ для работы с текстовыми файлами.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с текстовыми файлами в IDE PyCharm Community.

### Постановка задачи 1.

Разработать программу, которая формирует два текстовых файла, в одном из которых содержится последовательность целых положительных и отрицательных чисел, а во втором их обработка.

### Текст программы 1:

```
# Вариант 2.
# Средствами языка Python сформировать текстовый файл (.txt), содержащий
последовательность из целых
# положительных и отрицательных чисел. Сформировать новый текстовый файл
(.txt) следующего вида,
# предварительно выполнив требуемую обработку элементов.

n = ['17 -6 -12 36 24 -45 92 -15'] # Последовательность чисел

# Формирование первого файла с числами
f1 = open('data_1.txt', 'w')
f1.writelines(n)
f1.close()

# Формирование второго файла с числами
f2 = open('data_2.txt', 'w')
f2.write('Исходные данные: ')
f2.writelines(n)
f2.close()

# Разбивание строки на элементы и преобразование их в числа
f1 = open('data_1.txt')
k = f1.read().split()
for i in range(len(k)):
    k[i] = int(k[i])
f1.close()

# Ищем максимальный элемент в файле data_1.txt
f1 = open('data_1.txt')
max = 0
p = 1
for i in range(len(k)):
    max = max if max > k[i] else k[i]
for i in range(round(len(k)/2)):
    if k[i] < 0:
        p *= k[i]
f1.close()

# Записываем в файл data_2.txt количество элементов и максимальный
из них
f2 = open('data_2.txt', 'a')
f2.write(f'\nКоличество элементов: {len(k)}\nМаксимальный элемент: {max}')
```

```
f' \nПроизведение элементов меньших 0 в первой половине: {p}')
```

```
f2.close()
```

## Протокол работы программы 1:

Process finished with exit code 0

## Постановка задачи 2.

Разработать программу, которая выведет содержимое текстового файла на экран и сформирует новый с обратным порядком строк.

## Текст программы 2:

```
# Вариант 2.
# Из предложенного текстового файла (text18-2.txt) вывести на экран его
# содержимое, количество знаков препинания.
# Сформировать новый файл, в который поместить текст в стихотворной форме
# выведя строки в обратном порядке.
z = 0
f1 = open('text18-1.txt', encoding='UTF-8')
print(f1.read())

for i in open('text18-1.txt', encoding='UTF-8'):
    for j in i:
        if j == ',':
            z += 1
        if j == '.':
            z += 1
        if j == '!':
            z += 1
        if j == ':':
            z += 1
        if j == '-':
            z += 1
        if j == '...':
            z += 1

print('\nКоличество знаков препинания: ', z)

# Обратный порядок строк
f1 = open('text18-1.txt')
N = f1.readlines()
N = N[::-1]
f1.close()

# Создание нового файла со строками в обратном порядке
f2 = open('text18-2.txt', 'w')
f2.writelines(N)
f2.close()
```

## Протокол работы программы 2:

— Да, были люди в наше время,  
Не то, что нынешнее племя:  
Богатыри — не вы!

Плохая им досталась доля:  
Немногие вернулись с поля...  
Не будь на то господня воля,  
Не отдали б Москвы!

Количество знаков препинания: 11

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия выработал(а) навыки составления программ с текстовыми файлами в IDE PyCharm Community. Были использованы языковые конструкции `for`, `if`.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.  
Готовые программные коды выложены на GitHub.