

## 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 PyCharmCommunity.

**Постановка задачи №1** 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:  
Исходные данные:  
Количество элементов:  
Максимальный элемент:  
Среднее арифметическое элементов первой трети:

**Тип алгоритма:** смешанный

**Текст программы:**

```
# 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:
# Исходные данные:
# Количество элементов:
# Максимальный элемент:
# Среднее арифметическое элементов первой трети:

lst = [10, 9, 2, 4, -3, 33, -32, 1, -6] # последовательность из целых
положительных и отрицательных чисел.

t = open('xml.txt', 'w', encoding='utf-8') # открываю файл для записи и вношу
последовательность
print(*lst, file=t)
t.close()

t1 = open('xml2.txt', 'w', encoding='utf-8') # открываю файл для записи
t1.write(f'Исходные данные: {str(lst)}\n')
t1.write(f'Количество элементов: {len(lst)}\n')
t1.write(f'Максимальный элемент: {max(lst)}\n')
t1.write(f'Среднее арифметическое элементов первой трети: {sum(lst[:3]) /
len(lst[:3])}\n')
t1.close()
```

### Протокол работы программы:

Process finished with exit code 0

#### Содержимое первого файла:

```
10 9 2 4 -3 33 -32 1 -6
```

#### Содержимое второго файла:

```
Исходные данные: [10, 9, 2, 4, -3, 33, -32, 1, -6]
Количество элементов: 9
Максимальный элемент: 33
Среднее арифметическое элементов первой трети: 7.0
```

**Постановка задачи №2:** Из предложенного текстового файла (text18-12.txt) вывести на экран его содержимое,

количество пробельных символов. Сформировать новый файл, в который поместить текст в стихотворной форме предварительно вставив после каждой строки строку из символов «\*».

#### Текст программы:

```
# Из предложенного текстового файла (text18-12.txt) вывести на экран его
содержимое,
# количество пробельных символов. Сформировать новый файл, в который поместить
текст
# в стихотворной форме предварительно вставив после каждой строки строку из
символов
# «*».
f = open('text18-12.txt', 'r', encoding='utf-8') # открываю файлы через
кодировку ютф-8, чтобы избежать проблем
# вывожу содержимое файла
print(*f)
f = open('text18-12.txt', 'r', encoding='utf-8')
# находим пробелы
count = f.read().count(' ')
print('')
print('Количество пробелов в файле:', count)
f.close()
# открываю один для чтения, другой для записи(файлы)
f = open('text18-12.txt', 'r', encoding='utf-8')
f1 = open('my_file.txt', 'w', encoding='utf-8')
f1.write(f.read()) # переношу текст в другой файл

f = open('text18-12.txt', 'r', encoding='utf-8')
f1 = open('my_file.txt', 'w', encoding='utf-8')
z = '***'
# в конце каждой строки вставляю ***
for name in f:
    f1.write(name.rstrip() + z + '\n')
f1.close()
```

### Протокол работы программы:

### Первый текстовый файл:

```
Изведал враг в тот день немало,  
Что значит русский бой удалый,  
Наш рукопашный бой!..  
Земля тряслась — как наши груди,  
Смешались в кучу кони, люди,  
И залпы тысячи орудий  
Слились в протяжный вой...
```

Process finished with exit code 0

### Второй Текстовый файл:

```
Изведал враг в тот день немало,***  
Что значит русский бой удалый,***  
Наш рукопашный бой!..***  
Земля тряслась — как наши груди,***  
Смешались в кучу кони, люди,***  
И залпы тысячи орудий***  
Слились в протяжный вой...***
```

**Вывод:** в процессе выполнении практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ для работы с текстовыми файлами. Выполнена разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.