

## Практическое занятие №11

**Тема:** Составление программ с текстовыми файлами в IDE PyCharm Community.

**Цель:** Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием списковых включений, итераторов, генераторов в IDE PyCharm Community.

### Задача №1

**Постановка задачи.**

**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:**

**Минимальный элемент:**

**Количество положительных элементов в первой половине:**

**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:
# Минимальный элемент:
# Количество положительных элементов в первой половине:

numbers = [i for i in range(-10, 11)]
with open('input.txt', 'w') as file: # Открывает файл с именем 'input.txt' в режиме
записи
    file.write(' '.join(map(str, numbers)))

with open('input.txt', 'r') as file: # Открывает в режиме чтения
    numbers = list(map(int, file.read().split())) # Эта строка читает содержимое
```

```
файла разбивает его на список строк ( используя пробелы) преобразует каждую строку в целое число и присваивает результат переменной numbers
```

```
count = len(numbers)
min_element = min(numbers)
positive_count = sum(1 for num in numbers if num > 0) # Это строка подсчитывает количество положительных элементов в списке
first_half_positive_count = positive_count // 2
```

```
with open('output.txt', 'w') as file: # Открывает файл с именем 'output.txt' в режиме записи
    file.write(f'Исходные данные: {numbers}\n')
    file.write(f'Количество элементов: {count}\n')
    file.write(f'Минимальный элемент: {min_element}\n')
    file.write(f'Количество положительных элементов в первой половине: {first_half_positive_count}')
```

### Протокол работы программы:

```
Исходные данные: [-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Количество элементов: 21
Минимальный элемент: -10
Количество положительных элементов в первой половине: 5
```

## Задача №2

### Постановка задачи.

Из предложенного текстового файла (text18-11.txt) вывести на экран его содержимое, количество знаков препинания. Сформировать новый файл, в который поместить строку наименьшей длины

### Текст программы:

```
# Из предложенного текстового файла (text18-11.txt) вывести на экран его содержимое,
# количество знаков препинания. Сформировать новый файл, в который поместить строку наименьшей длины.
```

```
with open('text18-11.txt', 'r', encoding='utf-8') as file: # Строка открывает файл с именем 'text18-11.txt' в режиме чтения с кодировкой 'utf-8'
    content = file.read() # Эта строка читает содержимое файла и сохраняет его в переменной content
    punctuation_count = sum([1 for symbol in content if symbol in ',,:;!?!'])
```

```
print(content)
```

```
print('Количество знаков препинания:', punctuation_count)
```

```
lines = content.split('\n') # Эта строка разбивает содержимое файла на список строк используя символ новой строки в качестве разделителя
```

```
shortest_line = min(lines, key=len) # Эта строка находит самую короткую строку в списке строк.
```

```
with open('shortest_line.txt', 'w', encoding='utf-8') as file:  
    file.write(shortest_line)
```

### Протокол работы программы:

Вам не видать таких сражений!..  
Носились знамена, как тени,  
В дыму огонь блестел,  
Звучал булат, картечь визжала,  
Рука бойцов колоть устала,  
И ядрам пролетать мешала  
Гора кровавых тел.

Количество знаков препинания: 10

**Вывод:** В процессе выполнения практического занятия выработал(а) навыки составления программ с текстовыми файлами в IDE PyCharm Community. Были использованы языковые конструкции `for`, `in`, `encoding`, `with`, `key`, `as`, `\n`, `if`, `in`.

.Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub.