

**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.**

Вариант 32.

1. Средствами языка Python сформировать два текстовых файла (.txt), содержащих по одной последовательности из целых положительных и отрицательных чисел. Сформировать новый текстовый файл (.txt) следующего вида, предварительно выполнив требуемую обработку элементов:

Элементы первого и второго файлов:

Количество элементов первого и второго файлов:

Элементы последней трети:

Индекс максимального элемента последней трети:

2. Из предложенного текстового файла (text18-32.txt) вывести на экран его содержимое, количество знаков пунктуации в первых четырёх строках. Сформировать новый файл, в который поместить текст в стихотворной форме предварительно вставив после каждой строки строку из символов «\*».

**Тип алгоритма:** линейный

**Текст программы №1:**

```
# создание файлов
with open('file1.txt', 'w') as file1:
    file1.write('1 2 3 4 5 6 7')

with open('file2.txt', 'w') as file2:
    file2.write('-1 -2 -3 -4 -5')

# чтение файлов
with open('file1.txt', 'r') as file1:
    content1 = file1.read().split()

with open('file2.txt', 'r') as file2:
    content2 = file2.read().split()

# Объединение содержимого файлов
combined_content = content1 + content2

# Запись объединенного содержимого в новый файл
with open('output.txt', 'w') as output_file:
    # Запись элементов первого и второго файла
    output_file.write("Элементы первого файла: {}\n".format(content1))
    output_file.write("Элементы второго файла: {}\n".format(content2))

    # Запись количества элементов в первом и втором файле
    output_file.write("Количество элементов первого файла:
{}\n".format(len(content1)))
    output_file.write("Количество элементов второго файла:
{}\n".format(len(content2)))

    # Выборка последней трети элементов
    last_third = combined_content[-(len(combined_content)//3):]

    # Запись элементов последней трети
    output_file.write("Элементы последней трети: {}\n".format(last_third))
```

```
# Поиск индекса максимального элемента в последней трети
max_index = last_third.index(max(last_third))

# Запись индекса максимального элемента
output_file.write("Индекс максимального элемента последней трети:
{}\n".format(max_index))
```

### Протокол работы программы №1:

Элементы первого файла: ['1',

'2', '3', '4', '5', '6', '7']

Элементы второго файла: ['-1',

'-2', '-3', '-4', '-5']

Количество элементов первого

файла: 7

Количество элементов второго

файла: 5

Элементы последней трети: ['-

2', '-3', '-4', '-5']

Индекс максимального

элемента последней трети: 3

### Текст программы №2:

```
import string

# Чтение содержимого файла
with open('text18-32.txt', 'r', encoding='utf-8') as file:
    content = file.readlines()

# Вывод содержимого файла
for line in content:
    print(line.strip())

# Расчет количества знаков пунктуации в первых четырех строках
punctuation_count = 0
for i in range(4):
    line = content[i]
    punctuation_count += sum(1 for char in line if char in string.punctuation)

# Вывод количества знаков пунктуации
print("Количество знаков пунктуации в первых четырех строках:",
      punctuation_count)

# Создание нового файла с текстом в стихотворной форме
with open('output1.txt', 'w', encoding='utf-8') as output_file:
    for line in content:
        output_file.write(line.strip() + "\n")
    output_file.write("*****\n")
```

### Протокол работы программы №2:

Скажи-ка, дядя, ведь не даром

\*\*\*\*\*

Москва, спаленная пожаром,

\*\*\*\*\*

Французу отдана?

\*\*\*\*\*

Ведь были ж схватки боевые,

\*\*\*\*\*

Да, говорят, еще какие!

\*\*\*\*\*

Недаром помнит вся Россия

\*\*\*\*\*

Про день Бородина!

\*\*\*\*\*

**Вывод:**

закрепил усвоенные знания, понятия,

алгоритмы,

основные принципы составления

программ, приобрел навыки

составление

программ с текстовыми файлами в IDE PyCharm Community.