

Практическое занятие № 8

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 текстовыми файлами в IDE PyCharm Community.

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:
Минимальный элемент:
Количество положительных элементов в первой половине:
2. Из предложенного текстового файла (text18-11.txt) вывести на экран его содержимое, количество знаков препинания.
Сформировать новый файл, в который поместить строку наименьшей длины.
Вывод содержимого файла

Текст программы:

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:  
# Минимальный элемент:  
# Количество положительных элементов в первой половине:  
  
# Запишем в файл data_1.txt структуру данных – список.  
a = ['1 -21 65 -111 90 -5 0']  
f1 = open('data_1.txt', 'w')
```

```

f1.writelines(a)
f1.close()

# Дубликация списка в новый текстовый файл data_2.txt
f2 = open('data_2.txt', 'w')
f2.write("Исходные данные: ")
f2.write('\n')
f2.writelines(a)
f2.close()

# Разбиваем строку и ее значение преобразуем в элементы
f3 = open('data_1.txt')
k = f3.read()
k = k.split()
for i in range(len(k)):
    k[i] = int(k[i])
f3.close()

# Находим минимальный элемент и количество положительных элементов в
первой половине
# в файле data_1.txt и записываем в файл data_2.txt.
f3 = open('data_1.txt')
min_el, t = 0, 0
for i in range(len(k)):
    min_el = min_el if min_el < k[i] else k[i]

number = len(k)
one_half = int(number / 2)
k1 = k[: one_half] # Определяем первую половину списка
for a in range(len(k1)):
    k1[a] = int(k1[a])
    if k1[a] > 0:
        t += 1

f4 = open('data_2.txt', 'a')
f4.write('\n')
print("Количество элементов:", len(k), '\n'
      "Минимальный элемент:", min_el, '\n'
      "Количество положительных элементов в первой половине:', t,
      file=f4)
f4.close()
nums = []

```

```
f4 = open("data_2.txt", 'r')
print(f4.read())
```

2.

Из предложенного текстового файла (text18-11.txt) вывести на экран его содержимое,

количество знаков препинания.

Сформировать новый файл, в который поместить строку наименьшей длины.

Вывод содержимого файла

```
f1 = open('text18-11.txt', 'r', encoding='utf-8')
print(f1.read())
f1.close()
```

Подсчёт знаков препинания

sym = 0

for i in open('text18-11.txt', encoding='utf-8'):

a = list(i)

for x in a:

if '!' == x:

sym += 1

if ',' == x:

sym += 1

if '.' == x:

sym += 1

print("Количество знаков препинания: ", sym)

Определяем длину наименьшей строки.

a = 0

s = 0

for i in open('text18-11.txt', encoding='utf-8'):

m = len(list(i))

a = min(m, s)

s = m

Поиск наименьшей строки и отправка её в файл.

f1 = open('string.txt', 'w')

for i in open('text18-11.txt', encoding='utf-8'):

if a == len(list(i)):

f1.writelines(i)

f1.close()

```
f1 = open('string.txt', 'r')
print("Строка наименьшей длины -", f1.read())
f1.close()
```

Протокол работы программы:

1.

Исходные данные:

1 -21 65 -111 90 -5 0

Количество элементов: 7

Минимальный элемент: -111

Количество положительных элементов в первой половине: 2

Process finished with exit code 0

2.

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

Строка наименьшей длины - Гора кровавых тел.

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

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

Готовые программные коды выложены на GitHub.

