ФГБОУ ВО Чувашский государственный

университет имени И.Н. Ульянова

**Лабораторная работа № 6**

**"Файлы"**Вариант № 11

Выполнил студент ЭЭ-21-21

Михайлов Николай Алексеевич

Чебоксары 2022

**Цели работы:**

Дать студентам практический навык в написании программ, в которых выполняются операции с текстовыми файлами - чтение, запись.

**Краткие теоретические сведения:**

При выполнении работы было использовано:

* Конструкция

**with** open() **as:**

**Рабочее задание:**

Выполнить корректировку программ, написанных для лабораторных работ №1, №4 и №5, с таким условием, что бы ввод данных и вывод результатов работы осуществлялся с использованием файлов.

1) Вид файла, для считывания данных (в конце пустая строка):

*a =*

*-19*

*0*

*0.1*

*2.5*

*4*

Используем написанные функции

def lab1():

    input\_data = []

    with open('Лаб6\_1\_in.txt', encoding='utf-8') as r\_file:

        for row in r\_file:

            input\_data.append(row[:-1])

    a = [float(data) for data in input\_data[2:]]

    with open('Лаб6\_1\_out.txt', mode='w+', encoding='utf-8') as w\_file:

        for a in a:

            w\_file.write(f'a = {a}: z\_1 = {z\_1(a)}, z\_2 = {z\_2(a)}\n')

lab1()

2) Вид файла, для считывания данных (в конце пустая строка):

*n = 5*

*a = -2*

*b = 2*

Изменим метод **\_\_init\_\_**(), чтобы считывал данные из файла.

def \_\_init\_\_(self, w\_file):

    """

    Заполняет массив указанным количеством элементов

    """

    self.w\_file = w\_file

    input\_data = []

    with open('Лаб6\_4\_in.txt', encoding='utf-8') as r\_file:

        for row in r\_file:

            input\_data.append(int(row.split(' = ')[-1][:-1]))

    self.n, self.a, self.b = input\_data

    self.array = []

    for \_ in range(self.n):

        self.array.append(uniform(-5, 5))

    self.w\_file.write(f'Количество элементов {self.n}\nНачальное состояние:\n')

    self.w\_file.write(f'{self.array}\n')

def lab4():

    with open('Лаб6\_4\_out.txt', mode='w+', encoding='utf-8') as w\_file:

        lab4 = Lab4(w\_file)

        w\_file.write(f'Минимальный по модулю элемент = {lab4.sort\_for\_task()}\n')

        w\_file.write(f'Минимальный по модулю элемент = {lab4.comb\_sort()}\n')

        w\_file.write(f'Сумма модулей элементов, после первого отрицательного =  {lab4.sum\_abs()}\n')

        lab4.compress()

        w\_file.write(f'Сжатый массив\n{lab4.array}')

lab4()

3) Вид файла, для считывания данных (в конце пустая строка):

*n = 5*

*m = 4*

*limit = 2*

Изменим метод **show\_array**(), чтобы он записывал в файл данные, а не выводил консоль.

def show\_array(self, w\_file):

    """

    Выводит на экран массив в привычном виде.

    """

    for row in self.array:

        w\_file.write('[')

        for elem in row:

            if elem % 1 == 0 or int(elem\*1e3) == 0:

*# если число целое или пренебрежимо мало(соласно точности выввода), то выводим его целую часть*

                w\_file.write(f'{int(elem): ^8},')

            else:

                w\_file.write(f'{elem: ^8.3F},')

        w\_file.write(']\n')

def lab5():

    input\_data = []

    with open('Лаб6\_5\_in.txt') as r\_file:

        for row in r\_file:

            input\_data.append(int(row.split(' = ')[-1][:-1]))

    n, m, limit = input\_data

    qwe = array(n, m)

    with open('Лаб6\_5\_out.txt', mode='w+', encoding='utf-8') as w\_file:

        w\_file.write(f'Размерность массива {n} × {m}:\n\n')

        qwe.show\_array(w\_file)

        number\_limit\_row = qwe.number\_that\_row(limit=limit)

        w\_file.write(f'Количество строк, среднее арифметическое элементов которых меньше порогового значения: {number\_limit\_row}\n')

        qwe.form\_step\_array()

        w\_file.write('\nТреугольный вид:\n')

        qwe.show\_array(w\_file)

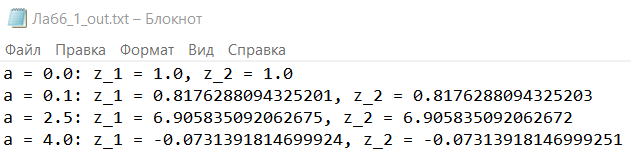
        number\_limit\_row = qwe.number\_that\_row(limit=limit)

        w\_file.write(f'Количество строк, среднее арифметическое элементов которых меньше порогового значения: {number\_limit\_row}\n')

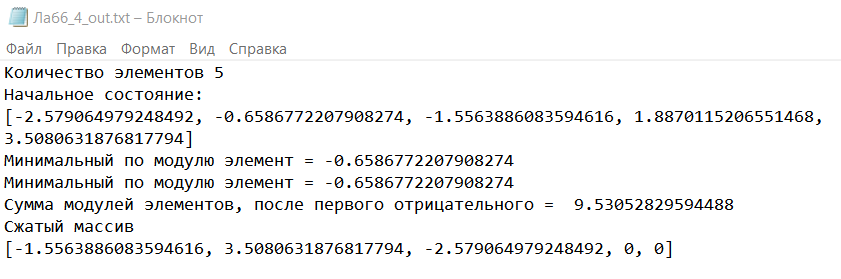
lab5()

**Результат:**

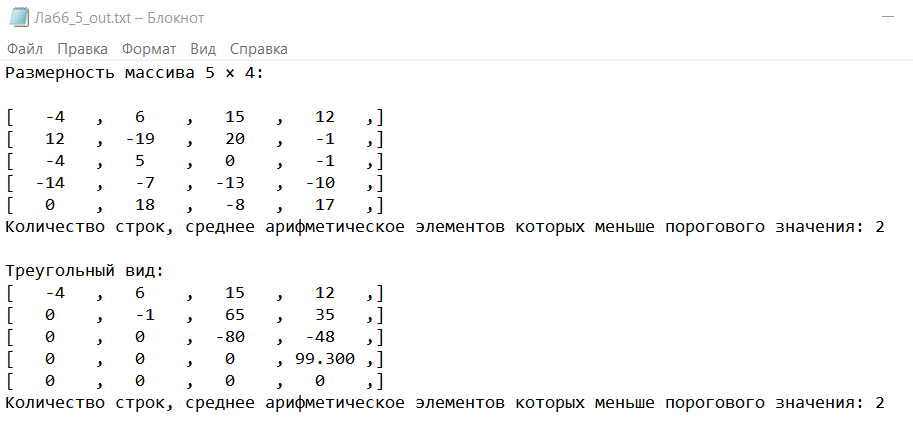
1)



2)



3)



**Выводы:**

При выполнении лабораторной работы был глубже изучена функция **open()**