РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Компьютерный практикум по моделированию

Студент:

Воробьев Александр Олегович

1032193998

Группа:

НФИбд-01-19

МОСКВА

2020 г.

Цель работы

1. Изучение методов работы с файлами в Python

2. Изучение списка с размером mxn (двумерный массив) в Python

3. Знакомство с библиотекой для визуализации данных Matplotlib

Задание 1.

Создать модуль music_serialize.py. В этом модуле определить словарь для вашей любимой музыкальной группы. С помощью модулей json и pickle сериализовать данный словарь в json и в байты, вывести результаты в терминал. Записать результаты в файлы group.json, group.pickle соответственно. В файле group.json указать кодировку utf-8.

Листинг программы на языке Python:

```
import json
import pickle

my_favourite_singer = {
    'name': 'Troye Sivan',
    'tracks': ['Seventeen', 'Lucky Strike'],
    'albums': [{'name': 'Bloom', 'year': 2018}, {'name': 'Blue Neighbourhood', 'year': 2015}]
}

with open('singer.pickle', 'wb') as f:
    pickle.dump(my_favourite_singer, f)


with open('singer.json', 'w', encoding='utf-8') as f:
    json_m_f_s = json.dump(my_favourite_singer, f)

print(pickle.dumps(my_favourite_singer))
print(json.dumps(my_favourite_singer))
```

Результат выполнения программы:

```python
import json
import pickle

my_favourite_singer = {
    'name': 'Troye Sivan',
    'tracks': ['Seventeen', 'Lucky Strike'],
    'albums': [{'name': 'Bloom', 'year': 2018}, {'name': 'Blue Neighbourhood', 'year': 2015}]
}

with open('singer.pickle', 'wb') as f:
    pickle.dump(my_favourite_singer, f)


with open('singer.json', 'w', encoding='utf-8') as f:
    json_m_f_s = json.dump(my_favourite_singer, f)

print(pickle.dumps(my_favourite_singer))
print(json.dumps(my_favourite_singer))
```

```
music_serialize
/Users/sandwor/PycharmProjects/lab5/venv/bin/python /Users/sandwor/PycharmProjects/lab5/venv/m
b'\x80\x03}q\x00(X\x04\x00\x00\x00nameq\x01X\x0b\x00\x00\x00Troye Sivanq\x02X\x06\x00\x00\x00t
{"name": "Troye Sivan", "tracks": ["Seventeen", "Lucky Strike"], "albums": [{"name": "Bloom",

Process finished with exit code 0
```

music_serialize.py | singer.pickle | singer.json

music_serialize.py | singer.pickle | singer.json

File was loaded in the wrong encoding: 'UTF-8' — Reload in another encoding

```
1  }q(X nameqX Troye Sivanq X tracksq ]q (X Seventeenq X Lucky Strikeq eX albumsq ]q(}q (hX Bloomq
2  X earq M u}q (hX Blue Neighbourhoodq
3  h M ueu.
```

music_serialize.py | singer.pickle | singer.json

```
1  {"name": "Troye Sivan", "tracks": ["Seventeen", "Lucky Strike"], "albums": [{"name": "Bloom",
```

Задание 2.

1) Найти сумму сходящегося ряда: $\frac{1}{1\cdot2}+\frac{1}{2\cdot3}+\frac{1}{3\cdot4}+\cdots+\frac{1}{n(n+1)}+\cdots$

Входные данные:
целое число n – номер частичной суммы.

Выходные данные:
частичная сумма при заданном n

2) Вычислите с заданной точностью ε сумму ряда: $\sum_{i=1}^{\infty}\frac{\sqrt{i+1}}{ie^{i}}$

Пример:

№	Точность ε	Сумма ряда
1	0.1	0.637464
2	0.001	0.685288
3	0.0001	0.685782
4	0.000001	0.685848

Листинг программы на языке Python:

1)

```
n = int(input('Enter N: '))
i = 1; r = 0

while i <= n:
   r = r + 1 / (i*(i+1))
   i = i + 1

print(r)
```

2)

```
import math

eps = float(input('Enter Epsylon: '))

i = 1; p = 0; r = 1

while r >= eps:
   r = (math.sqrt(i + 1)) / (i * (math.e ** i))
   p += r
   i += 1

print(p)
```

Результат выполнения программы:

1)

2)

```python
import math

eps = float(input('Enter Epsylon: '))

i = 1; p = 0; r = 1

while r >= eps:
    r = (math.sqrt(i + 1)) / (i * (math.e ** i))
    p += r
    i += 1

print(p)
```

```
ex2_2
/Users/sandwor/PycharmProjects/lab5/venv/bin/python /Users/sandwor/PycharmProjects
Enter Epsylon: 0.000001
0.6858483840545726

Process finished with exit code 0
```

Задание 3.

Дана функция: $f(x) = \frac{x^n + p^n}{x^n}$, при различных значениях *n* и *p*

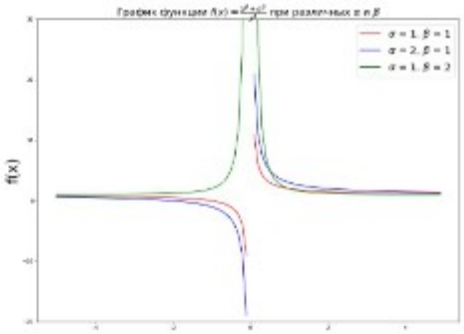
Необходимо:

- Построить график (размер графика должен быть достаточным, чтобы визуально увидеть особенности изучаемых функций), график каждой функции должен быть одного цвета для одного значения *n* и *p*
- Подписать оси и заголовок
- Создать легенду
- Сохранить изображение в svg файл
- Код не должен вызывать ошибки исполнения (например, из-за деления на 0 или корня из отрицательной величины)

Построить в общих осях графики для:

- $p = 1$, $n = 1$
- $p = 2$, $n = 1$
- $p = 1$, $n = 2$

Результат будет похож на следующий график (только не забудьте, цвет и тип линий сделать разными для каждой кривой):

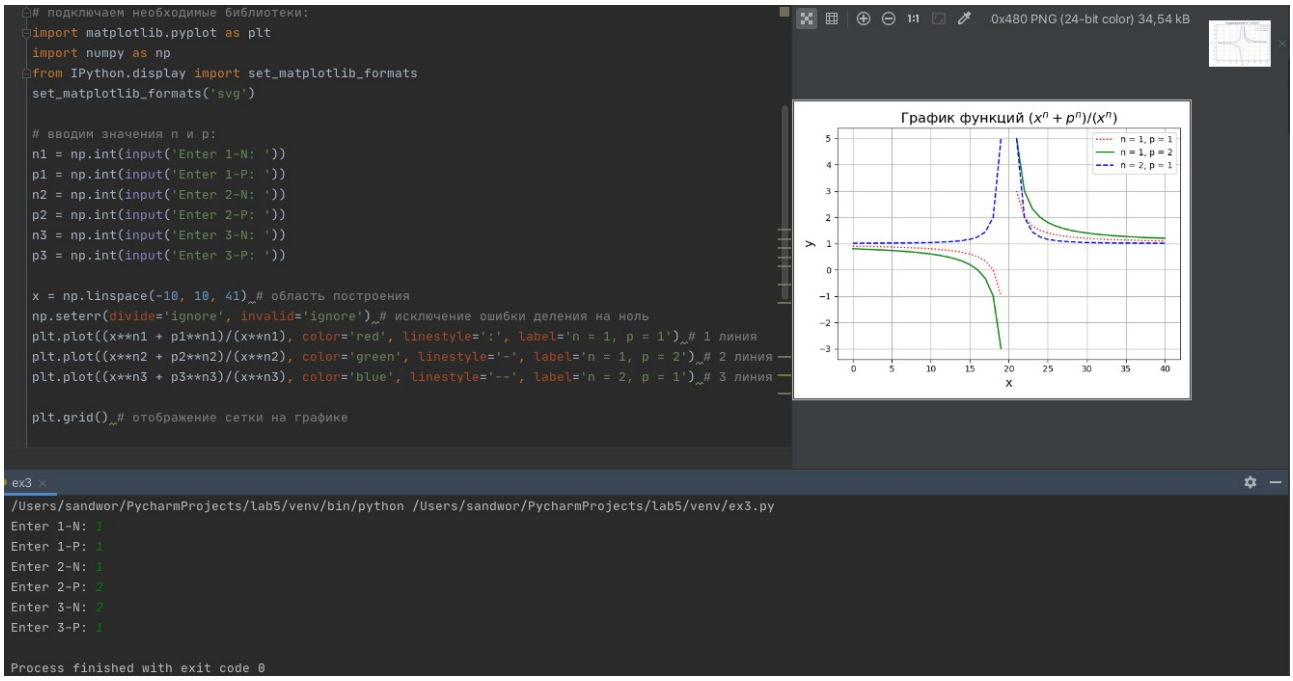


Листинг программы на языке Python:

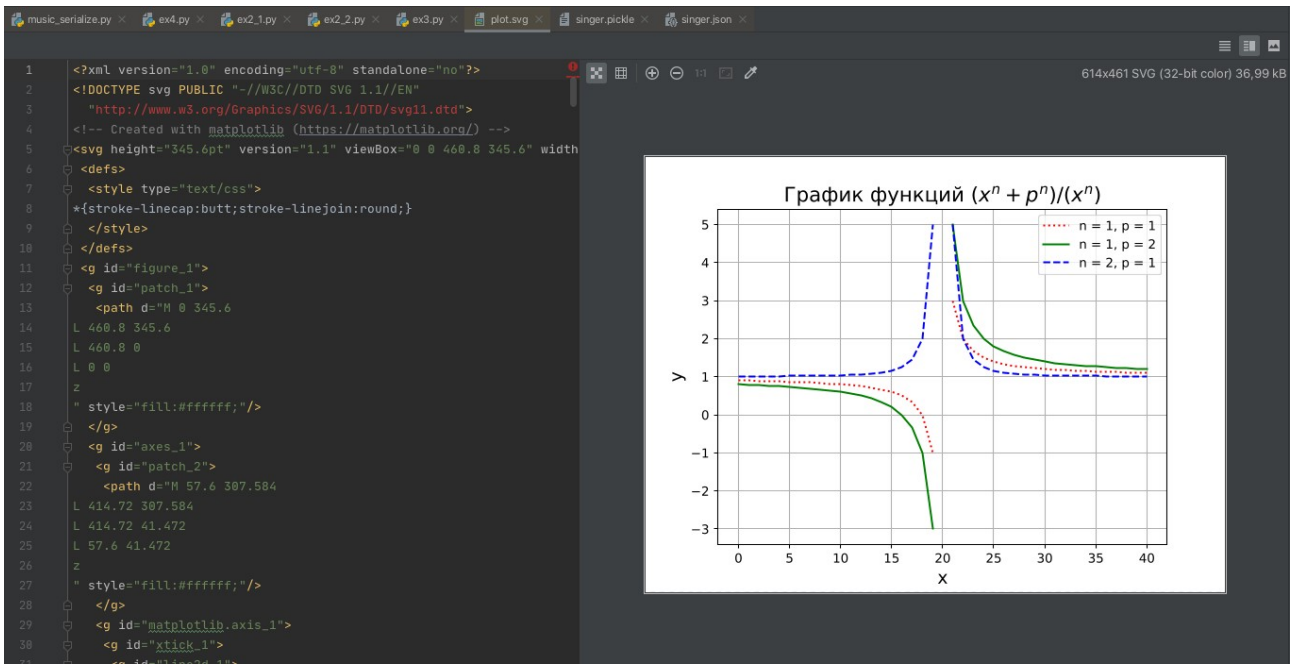
```python
# подключаем необходимые библиотеки:
import matplotlib.pyplot as plt
```

```python
import numpy as np
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('svg')

# вводим значения n и p:
n1 = np.int(input('Enter 1-N: '))
p1 = np.int(input('Enter 1-P: '))
n2 = np.int(input('Enter 2-N: '))
p2 = np.int(input('Enter 2-P: '))
n3 = np.int(input('Enter 3-N: '))
p3 = np.int(input('Enter 3-P: '))

x = np.linspace(-10, 10, 41) # область построения
np.seterr(divide='ignore', invalid='ignore') # исключение ошибки деления на ноль
plt.plot((x**n1 + p1**n1)/(x**n1), color='red', linestyle=':', label='n = 1, p = 1') # 1 линия
plt.plot((x**n2 + p2**n2)/(x**n2), color='green', linestyle='-', label='n = 1, p = 2') # 2 линия
plt.plot((x**n3 + p3**n3)/(x**n3), color='blue', linestyle='--', label='n = 2, p = 1') # 3 линия

plt.grid() # отображение сетки на графике

plt.legend(loc='best') # создание легенды в удобном месте

# обозначем оси координат
plt.xlabel('x', fontsize=14)
plt.ylabel('y', fontsize=14)

plt.title(r'График функций $(x^n + p^n)/(x^n)$', fontsize=16) # Пишем заголовок

plt.savefig('plot.svg') # сохраняем график в svg

plt.show() # выводим график
```

Результат выполнения программы:



Файл plot.svg, содержащий график:



Задание 4.

Вариант 7

1) Квадратная матрица, симметричная относительно главной диагонали, задана верхним треугольником в виде одномерного массива. Восстановить исходную матрицу и напечатать по строкам.

2) Для заданной квадратной матрицы сформировать одномерный массив из ее диагональных элементов. Найти след матрицы, просуммировав элементы одномерного массива. Преобразовать исходную матрицу по правилу: четные строки разделить на полученное значение, нечетные оставить без изменения.

Листинг программы на языке Python:

1)

```
# подключаем библиотеку
import random

n = int(input('Enter size of matrix: ')) # Пользователь вводит размер кв. матрицы
c = 0 # объявляем счетчик

m_up = [random.randint(-10,10) for i in range((n*(n+1)) // 2)] # заполняем массив с элементами верхнего треугольника

print("Elements of the upper triangular matrix: ", m_up) # выведем элементы верхнего треугольника
# создадим макет матрицы под заполнение
m = []
for i in range(n):
    m.append([0]*n)
# заполняем верхний треугольник матрицы имеющимися элементами
for i in range(n):
    for j in range(n):
        if i == j or i < j:
```

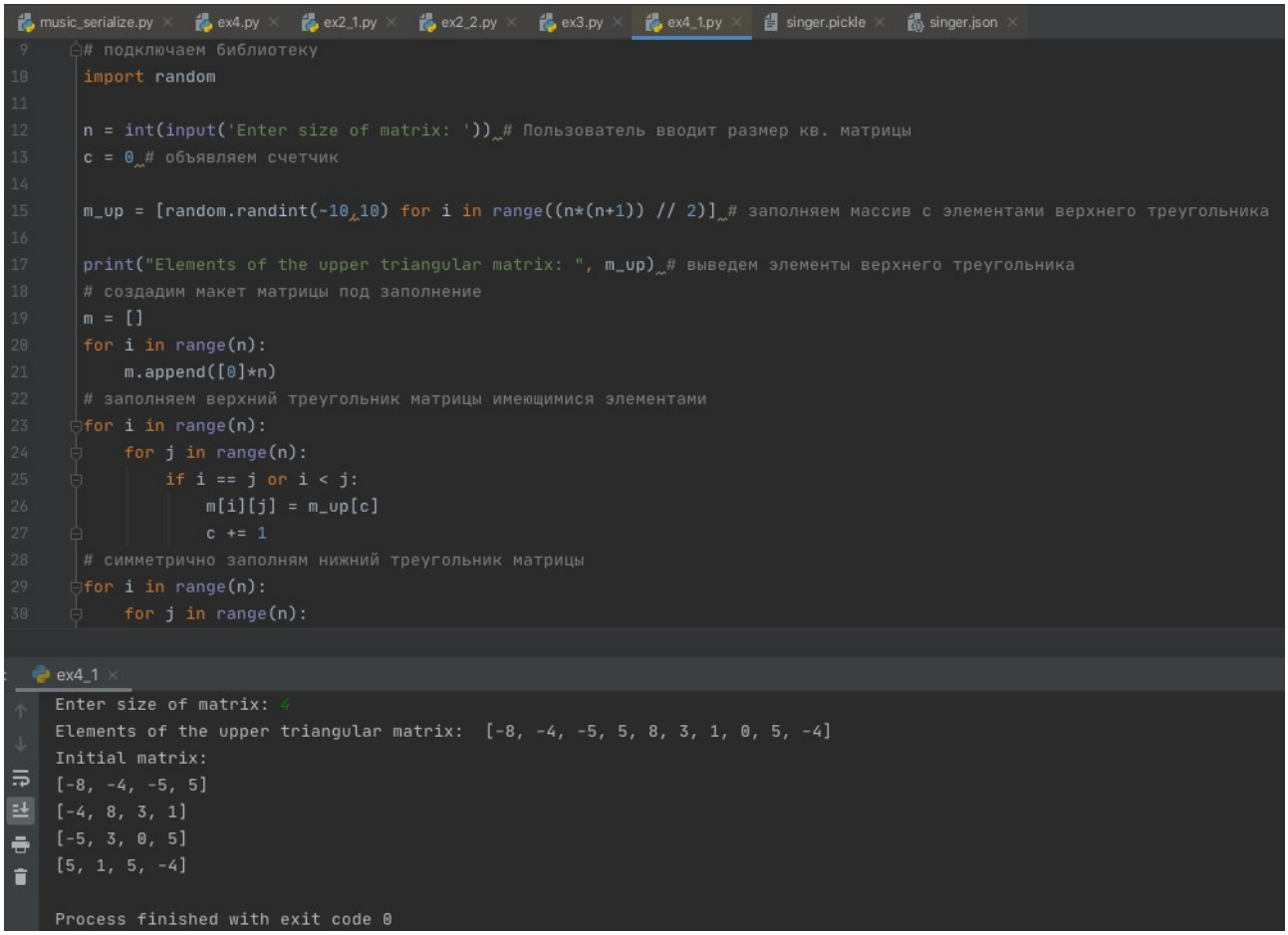
```
            m[i][j] = m_up[c]
            c += 1
# симметрично заполням нижний треугольник матрицы
for i in range(n):
    for j in range(n):
        if i > j:
            m[i][j] = m[j][i]
# выводим заполненную исходную матрицу
print('Initial matrix: ')
for i in m:
    print(i)
```

2)

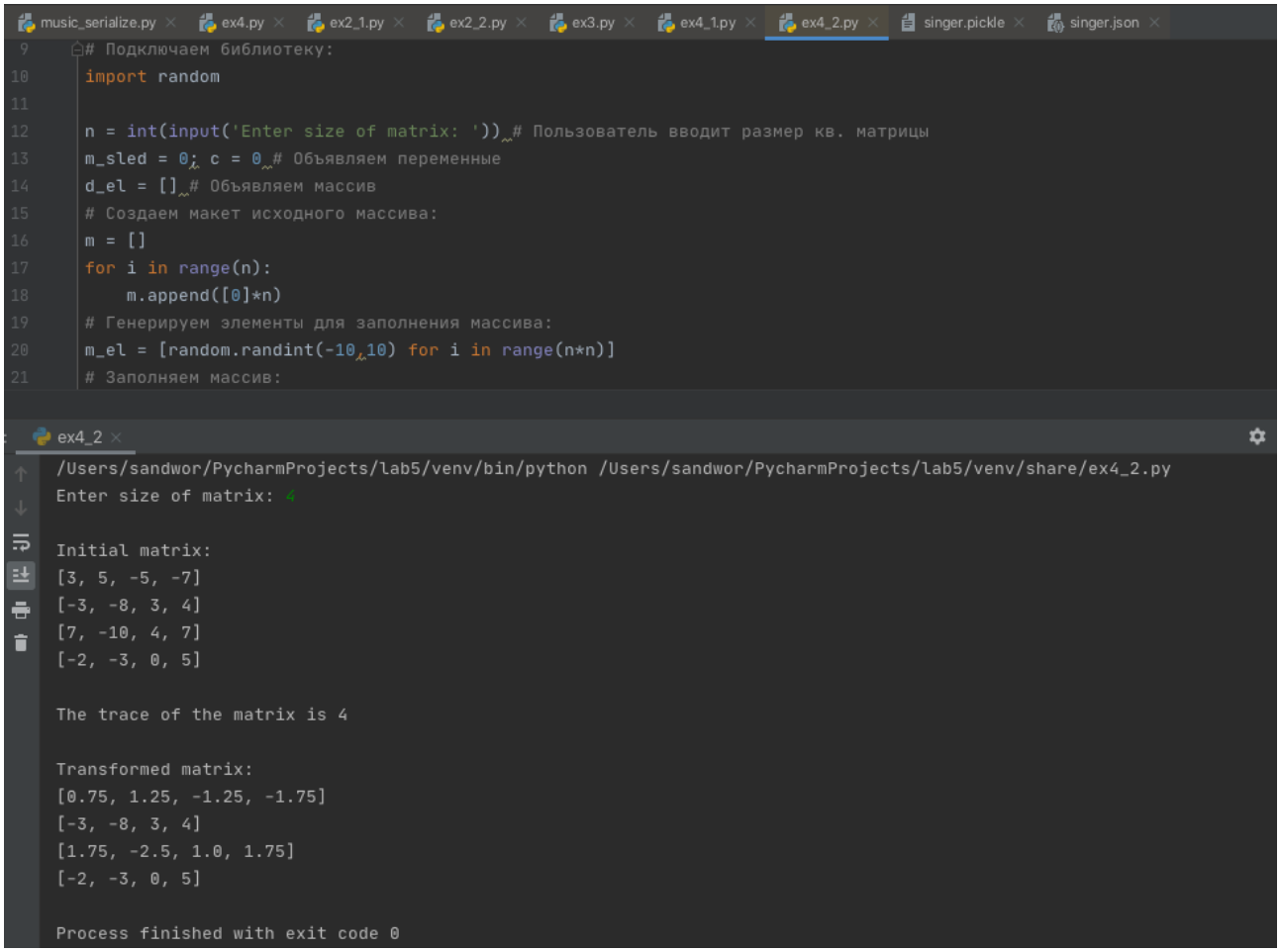
```
# Подключаем библиотеку:
import random

n = int(input('Enter size of matrix: ')) # Пользователь вводит размер кв. матрицы
m_sled = 0; c = 0 # Объявляем переменные
d_el = [] # Объявляем массив
# Создаем макет исходного массива:
m = []
for i in range(n):
    m.append([0]*n)
# Генерируем элементы для заполнения массива:
m_el = [random.randint(-10,10) for i in range(n*n)]
# Заполняем массив:
for i in range(n):
    for j in range(n):
        m[i][j] = m_el[c]
        c += 1
# Находим элементы главной диагонали и заносим их в отдельный массив:
for i in range(n):
    for j in range(n):
        if i == j:
            d_el.append(m[i][j])
# Проходимся по элементам массива и суммируем их для вычисления следа:
for i in range(len(d_el)):
    m_sled += d_el[i]
# Выводим исходную матрицу:
print("\nInitial matrix: ")
for i in m:
    print(i)
# Выводим след матрицы:
print("\nThe trace of the matrix is", m_sled)
# Преобразовываем матрицу - делим элементы четных строк(отсчет от нуля) на "след":
for i in range(n):
    for j in range(n):
        if (i) % 2 == 0:
            m[i][j] /= m_sled
# Выводим преобразованную матрицу:
print("\nTransformed matrix: ")
for i in m:
    print(i)
```

Результат выполнения программы:

1)


music_serialize.py
ex4.py
ex2_1.py
ex2_2.py
ex3.py
ex4_1.py
singer.pickle
singer.json
# подключаем библиотеку
import random
n = int(input('Enter size of matrix: '))# Пользователь вводит размер кв. матрицы
c = 0# объявляем счетчик
m_up = [random.randint(-10,10) for i in range((n*(n+1)) // 2)]# заполняем массив с элементами верхнего треугольника
print("Elements of the upper triangular matrix: ", m_up)# выведем элементы верхнего треугольника
# создадим макет матрицы под заполнение
m = []
for i in range(n):
m.append([0]*n)
# заполняем верхний треугольник матрицы имеющимися элементами
for i in range(n):
for j in range(n):
if i == j or i < j:
m[i][j] = m_up[c]
c += 1
# симметрично заполням нижний треугольник матрицы
for i in range(n):
for j in range(n):
ex4_1
Enter size of matrix: 4
Elements of the upper triangular matrix: [-8, -4, -5, 5, 8, 3, 1, 0, 5, -4]
Initial matrix:
[-8, -4, -5, 5]
[-4, 8, 3, 1]
[-5, 3, 0, 5]
[5, 1, 5, -4]
Process finished with exit code 0

2)



Вывод

В ходе выполнения работы изучил методы работы с файлами в Python, изучил список с размером mxn (двумерный массив) в Python, познакомился с библиотекой для визуализации данных Matplotlib.

Ответы на контрольные вопросы

1. Сколько способов создания списка размером nxm (двумерный массив) вы знаете? Приведите примеры.

Объявление готовой матрицы по столбцам:

```
m = [[0, 0,0],
   [0, 0, 0],
   [0, 0, 0],
   [0, 0, 0]]
```

или в строчку:

```
m = [[0, 0,0], [0, 0, 0], [0, 0, 0], [0, 0, 0]]
```

Создание пустого массива и заполнение через цикл:

```
mas = []
for i in range(a):
   mas.append([])
   for j in range(b):
      mas[i].append(r)
      r += 1
```

Создание массива с одинаковыми элементами:

```
m = 3; n = 2
a = [[0] * m] * n
```

2. Как удалять файлы в Python (фрагмент вашего кода)?

Для удаления файла можно подключить библиотеку os и использовать команду os.remove():

```
import os
os.remove("file.txt")
```

3. Какие методы (список) вам необходимы для построения простого графика, применив библиотеку Matplotlib?

Вывод линии на график с параметрами

```
plt.plot()
```

Отображение сетки

```
plt.grid()
```

Создание легенды с параметрами

```
plt.legend()
```

Обозначение осей координат с параметрами

```
plt.xlabel()
```

Вывод заголовка графика с параметрами

```
plt.title()
```

Сохранение графика в файл с параметрами

```
plt.savefig()
```

Вывод графика

```
plt.show()
```