

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: Журавлев Георгий Иванович

Группа: НФИбД 02-20

МОСКВА

2021__ 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

Список сокращений: -

Список обозначений: -

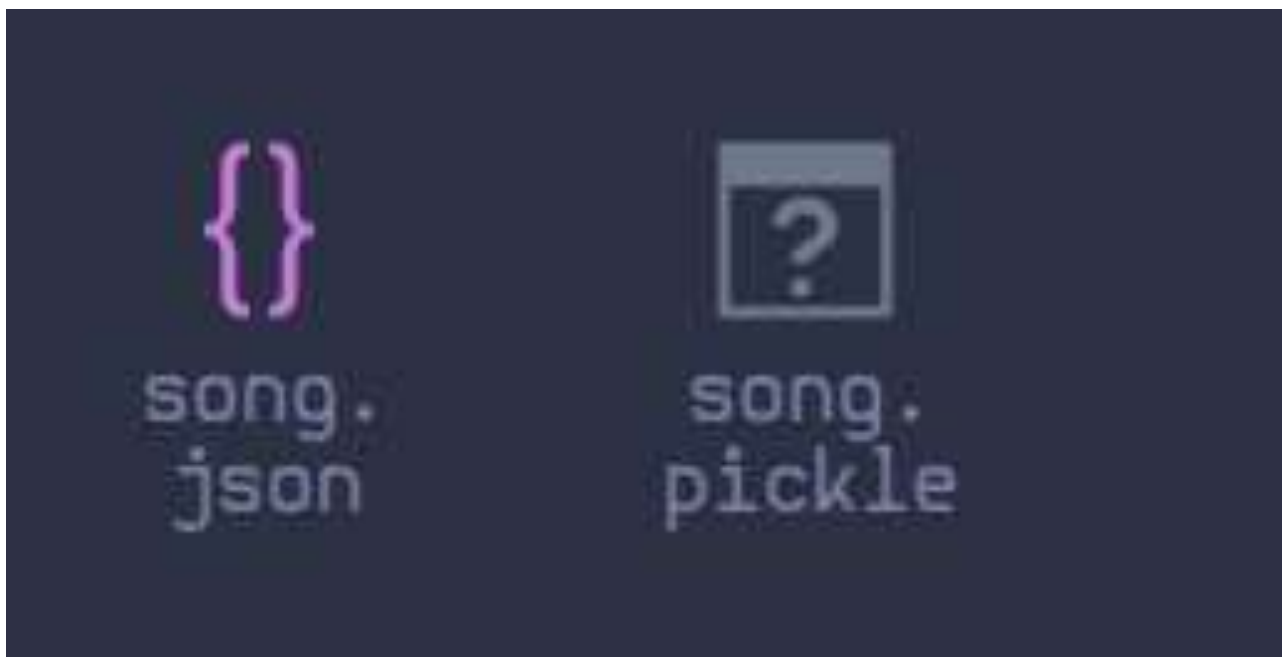
Выполнение работы

Задание 1.

```
1  import pickle
2  import json
3  my_favourite_group={
4      'name': 'Baroque composers',
5      'tracks': ['Bach, partita no5 prelude',
6      *      'BWV829', 'Bach, toccata in e-moll BWV 914'],
7      'Albums': [{'name': 'BWV829', 'year': '1730'} {'name': 'BW
8      *      V914', 'year': '1710'}]
9  }
10
11 my_f_g=pickle.dumps(my_favourite_group)
12 print(type(my_f_g))
13
14 with open('song.json', 'w', encoding='utf-8') as f:
15     json.dump(my_favourite_group, f)
16
17 with open('song.pickle', 'wb') as f:
18     pickle.dump(my_favourite_group, f)
```

1. Создадим «my_favourite_group»
2. Воспользуемся модулями pickle и json для создания файла

пример работы программы:



Задание 2. 1.

```
1  num=int(input('n= '))
2  sum_r= 0
3  for i in range (1, num):
4      sum_r+=1.0/(i*(i+1))
5      i+=1
6  print('Sum= ', sum_r)
7  |
```

1. Ввод данных

2. Создаем функцию:

Выполняем условие последовательности

3. Выводим значение

пример работы программы: (последовательность сходится к 1)

```
/usr/bin/python3.8 /home/g_zhuravlev/python-projects/1st.project/venv/1st.file.py
n= 100000000
Sum= 0.99999989999998053

Process finished with exit code 0
```

Задание 2.2.

```
1  import math
2  sum_r = 0
3  i = 1
4  e = math.e
5  E = float(input('\nAccuracy: '))
6  formula = math.sqrt(i+1)/(i*e)
7  while (formula >=E):
8      e *= math.e
9      sum_r += formula
10     i += 1
11     formula = math.sqrt(i+1)/(i*e)
12  print('Сумма ряда: ', sum_r)
```

1. Просим ввести точность

2. Создаем алгоритм

пример работы программы:

```
/usr/bin/python3.8 /home/g_zhuravlev/python-projects/1st.project/venv/1st.file.py

Accuracy: 0.00000000001
Сумма ряда: 0.6858487406440874

Process finished with exit code 0
```

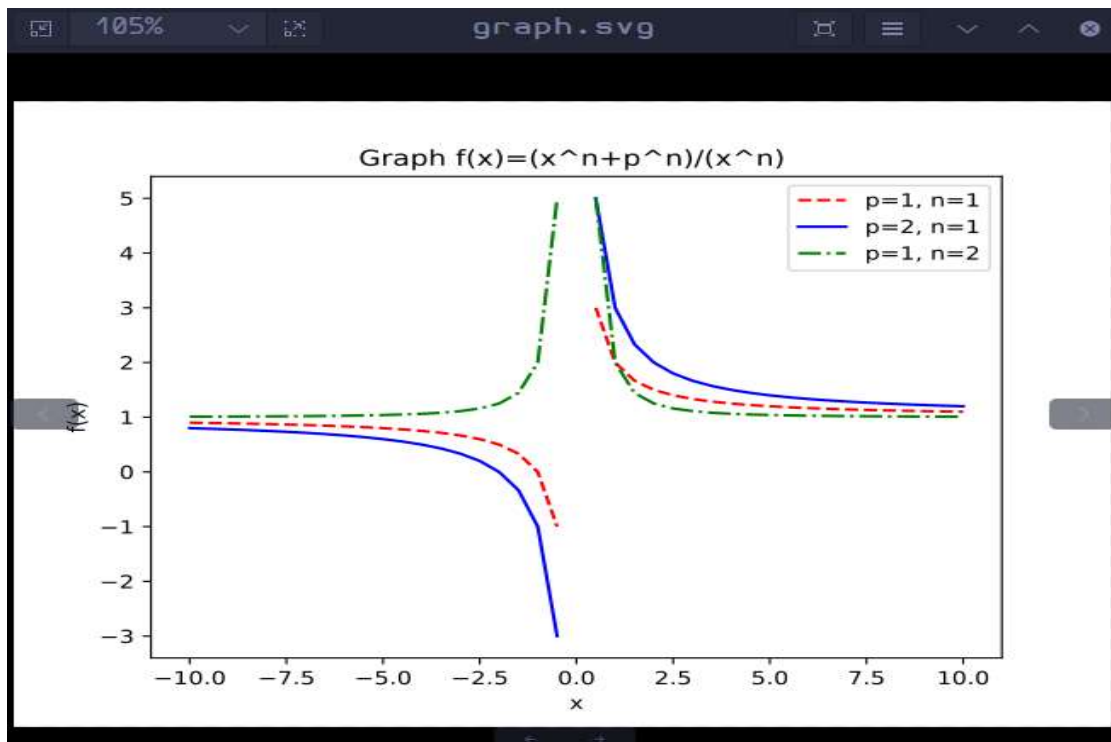
Задание 3.

```

1  import matplotlib.pyplot as plt
2  import numpy as np
3  x = np.linspace(-10, 10, 41)
4  p1=1; n1=1; p2=2; n2=1; p3=1; n3=2
5  y = (x**n1+p1**n1)/x**n1
6  z = (x**n2+p2**n2)/x**n2
7  v = (x**n3+p3**n3)/x**n3
8  fig, ax = plt.subplots()
9  ax.plot(x, y, color='red', linestyle='--', label='p=1,
•  n=1')
10 ax.plot(x, z, color='blue', linestyle='-', label='p=2,
•  n=1')
11 ax.plot(x, v, color='green', linestyle='-.',
•  label='p=1, n=2')
12 plt.legend(loc='upper right')
13 plt.xlabel('x')
14 plt.ylabel('f(x)')
15 plt.title(r'Graph f(x)=(x^n+p^n)/(x^n) ')
16 plt.show()
17 fig.savefig('graph.svg')
18

```

1. Воспользуемся модулями, определим 3 графика, с параметрами p,n 123
 2. Зададим параметры для графиков(цвет,линия, значения)
 3. Определим легенду в правом верхнем углу
 4. Создадим название графика и сохраним его
- пример работы программы:



Задание 4.1.

Вычислить сумму и число положительных элементов матрицы $A[N, N]$, находящихся над главной диагональю.

```

1  import random
2  higher = 0
3  s = 0
4  N = int(input('Size of a matrix: '))
5  Matrix = [[random.randrange(10) for i in range(N)] for
6             j in range(N)]
7  for i in range(N):
8      for j in range(i+1, N):
9          if Matrix[i][j] <= 0:
10             continue
11          if Matrix[i][j] > 0:
12             higher += 1
13             s += Matrix[i][j]
14
15  for i in Matrix:
16      for j in i:
17          print(j, end=" ")
18      print()
19  print('Sum:', s)
20  print('num_of_el:', higher)

```

1. Создаем матрицу с рандомными числами и заданным размером.
2. Для всех эл-ов над диагональю отбираем положительные, складываем их в сумму и считаем их кол-во
3. Печатаем матрицу

пример работы программы:

```
/usr/bin/python3.8 /home/g_zhuravlev/python-projects/1st.project/venv/1st.file.py
Size of a matrix: 3
3 6 6
0 9 3
8 2 0
Sum: 15
num_of_el: 3

Process finished with exit code 0
```

Задание 4.2.

Дана матрица $B[N, M]$. Найти в каждой строке матрицы максимальный и минимальный элементы и поменять их с первым и последним элементами строки соответственно.

```

1  import random
2  N = int(input('Size of N: '))
3  M = int(input('Size of M: '))
4  Matrix = [[random.randrange(10) for i in range(M)] for
5  *   j in range(N)]
6  print('before: ')
7  for i in Matrix:
8      for j in i:
9          print(j, end=" ")
10     print()
11
12  for i, row in enumerate(Matrix):
13      max = min = 0
14      for j, elem in enumerate(row):
15          if elem > row[max]:
16              max = j
17          if elem < row[min]:
18              min = j
19      temp = row[0]
20      row[0] = row[max]
21      row[max] = temp
22      temp1 = row[-1]
23      row[-1] = row[min]
24      row[min] = temp1
25
26  print('after: ')
27  for i in Matrix:
28      for j in i:
29          print(j, end=" ")
30     print()

```

1. Вводим данные с клавиатуры.
2. Пересчитываем номера всех строк. Для каждой строки проходимся и собираем наибольшие/наименьшие эл-ты.
3. Меняем местами эти эл-ты

4. Вывод массива

пример работы программы:

```
Size of N: 5
Size of M: 5
before:
6 6 7 4 9
6 4 8 5 9
6 4 1 6 3
6 7 6 6 5
6 5 1 9 8
after:
9 6 7 6 4
9 6 8 5 4
6 4 3 6 1
7 6 6 6 5
9 5 8 6 1
Process finished with exit code 0
```

Заключение.

Благодаря этой лабораторной работе, я узнал: как работать с файлами в Python; Познакомился с библиотекой Matplotlib; Научился работать с матрицами(двумерными массивами).

Контрольные вопросы:

1. 1 способ — вручную

```
for i in range(num_of_elements_arr_1):
    k = int(input(f'element { i } of an arr_1 = '))
    arr_1.append(k)
```

2 способ - `Matrix = [[random.randrange(10) for i in range(M)] for j in range(N)]`

2. Удалять файлы можно с помощью функций `os.remove()` и `os.unlink()`.

3. Чтобы построить график, применив библиотеку Matplotlib, нужно:

- задать значения переменных, которые будем использовать, или функции через которые их нужно посчитать
- `plt.plot(x, y)` чтобы построить график переменных x, y
- `plt.show()` чтобы график появился в окне