

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: Коне Сирики

Группа: НФИбд-01-20

МОСКВА

2021 г.

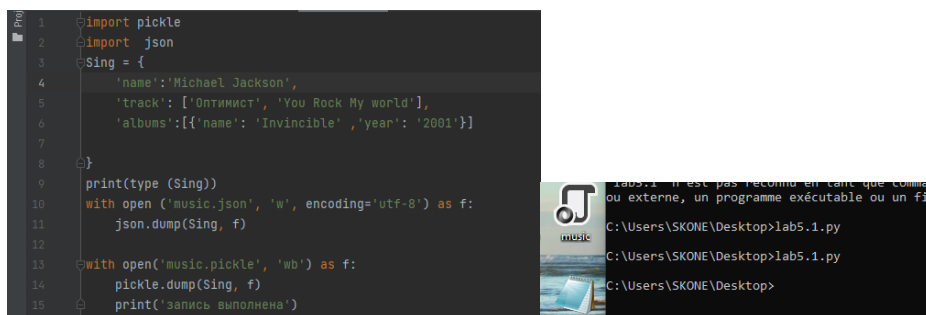
Цель:

1. Изучение методов работы с файлами в Python
2. Изучение списка с размером `m x n` (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, например:

`my_favourite_group = { 'name': 'Г.М.О.', 'tracks': ['Последний месяц осени', 'Шапито'], 'Albums': [{ 'name': 'Делать панк-рок', 'year': 2016}, { 'name': 'Шапито', 'year': 2014}]}` 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`.



The image shows a Python script in a code editor and its execution output in a terminal. The script defines a dictionary `Sing` for the group 'Г.М.О.' and serializes it to `music.json` and `music.pickle`. The terminal output shows the type of `Sing` as `dict` and the successful completion of the file operations.

```
1 import pickle
2 import json
3 Sing = {
4     'name': 'Michael Jackson',
5     'track': ['Оптимист', 'You Rock My world'],
6     'albums': [{ 'name': 'Invincible', 'year': '2001'}]
7 }
8
9 print(type(Sing))
10 with open('music.json', 'w', encoding='utf-8') as f:
11     json.dump(Sing, f)
12
13 with open('music.pickle', 'wb') as f:
14     pickle.dump(Sing, f)
15 print('запись выполнена')
```

lab5.1: if est pas reconnu en tant que commande interne, un programme exécutable ou un fichier de commandes.
C:\Users\SKONE\Desktop>lab5.1.py
C:\Users\SKONE\Desktop>lab5.1.py
C:\Users\SKONE\Desktop>

Задание 2.

(тема из курса «Математический анализ») 1) Найти сумму сходящегося ряда:
$$1/1 \cdot 2 + 1/2 \cdot 3 + 1/3 \cdot 4 + \dots + 1/(n \cdot (n+1)) + \dots$$
 Входные данные: целое число n — номер частичной суммы. Выходные данные: частичная сумма при заданном



```
n=int(input("Введите номер частичной суммы:"))
sum=0
while(n>=1):
    x=1/(n*(n+1))
    sum=sum+x
    n=n-1
print(sum)
```

lab5.2.1 x

C:\Users\SKONE\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/SKONE/PycharmProjects/pythonProject2/lab5.2.1.py

Введите номер частичной суммы: 1

0.75

Process finished with exit code 0

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

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

```
import math
sum = 0.0
i=1
c=float(input('Введите точность:'))
sum1 = 1
while (sum1>c):
    sum1 = (math.sqrt(i + 1)) / (i* math.exp(i))
    sum+=(math.sqrt(i+1))/(i*math.exp(i))
    i+=1
print(sum)
```

lab5.2.2 x

C:\Users\SKONE\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/SKONE/PycharmProjects/pythonProject2/lab5.2.2.py

Введите точность: 0.00000001

0.6858487404747285

Process finished with exit code 0

Задание 3. (библиотека Matplotlib)

Дана функция: $f(x)$, при различных значениях n и p

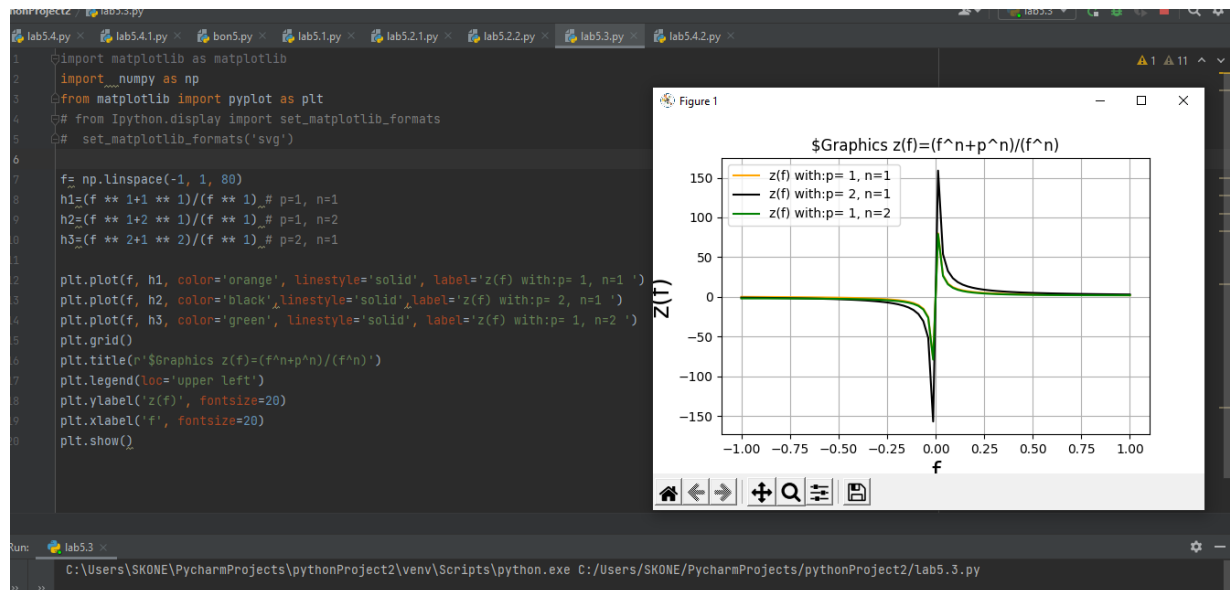
Необходимо:

- 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:

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



Задание 4

1. Для заданной квадратной матрицы найти такие k , что k -я строка матрицы совпадает с k -м столбцом.

```

n=int(input("l:"))
m=int(input("c:"))
matrix= []
list_diagonal=[]
maximum_value=0
for i in range(n):
    matrix.append([])
    for j in range(m):
        print("input [", i, "] [", j, "]")
        new = int(input())
        matrix[i].append(new)
        if new > maximum_value:
            maximum_value= new
            x = i
            y = j
    print("Matrix:")
for i in range(m):
    for j in range(n):
        if j==n-1:
            print(matrix[i][j])
        else:
            print(matrix[i][j], end=' ')

```

```

        else:
            print(matrix[i][j], end=' ')

print("-----Main Diagonal Matrix-----")
for i in range(m):
    for j in range(n):
        if i==j:
            list_diagonal.append(matrix[i][j])
    print(list_diagonal)

for i in range(n) > for j in range(m)
lab54.1 (1) <
input [ 2 ] [ 1 ]
input [ 2 ] [ 2 ]

Matrix:
1 -1 0
4 5 6
6 7 8

-----Main Diagonal Matrix-----
[1]
[1, 5]
[1, 5, 8]

Process finished with exit code 0

```

2. Дана действительная матрица размером $n \times m$. Требуется преобразовать матрицу: поэлементно вычесть последнюю строку из всех строк, кроме последней.

```

import random
def create_matrix(rows, columns):
    matrix = []
    for i in range(0, rows):
        inner_list = []
        for j in range(0, columns):
            inner_list.append(round(random.uniform(-10, 10), 2))
        matrix.append(inner_list)
    return matrix
def spec_subtract(matrix):
    for i in range(0, len(matrix) - 1):
        for j in range(0, len(matrix[i])):
            matrix[i][j] = matrix[i][j] - matrix[len(matrix) - 1][j]
    return matrix
def print_matrix(matrix):
    print('\n'.join([' '.join(['{:2f}'.format(item) for item in row]) for row in matrix]))

n = int(input("Input the number of rows: "))
m = int(input("Input the number of columns: "))
matrix_ = create_matrix(n, m)
print_matrix(matrix_)
print()
matrix_ = spec_subtract(matrix_=matrix_)
print_matrix(matrix_)

```

```
print('\n'.join([''.join([' {:.2f}'.format(item) for item in row]) for row in matrix]))

n = int(input("Input the number of rows: "))
m = int(input("Input the number of columns: "))
matrix_ = create_matrix(n, m)
print_matrix(matrix_)
print()
matrix_ = spec_subtract(matrix=matrix_)
print_matrix(matrix_)

spec_subtract()  > for i in range(0, len(matrix)-1):
                  > for j in range(0, len(matrix[i])-1):
```

lab5.4.2

C:\Users\SKONE\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/SKONE/PycharmProjects/pythonProject2/lab5.4.2.py

Input the number of rows: 3
Input the number of columns: 4

-2.35 5.70 7.21 6.05
-0.73 -0.76 -5.84 -0.07
-1.88 -6.51 -2.74 -4.14

-0.47 12.21 9.95 10.19
1.15 5.75 -3.10 4.07
-1.88 -6.51 -2.74 -4.14

Process finished with exit code 0

Выход:

1. Изучил методов работы с файлами в Python
2. Изучил списка с размером $m \times n$ (двумерный массив) в Python
- 3.Познакомился с библиотекой для визуализации данных Matplotlib

Контрольный Вопрос:

1. Сколько способов создания списка размером $p \times m$ (двумерный массив) вы знаете? Приведите примеры.
 - a. `a= [[0] *m for in range(n)]`
 - b.

```
for i in range(n):
    b=[]
    for j in range(m):
        b.append(int(input()))
    a.append(b)
```
 - c.

```
for i in range(n):
    a.append([0]*m)
```
2. Как удалять файлы в Python (фрагмент вашего кода)?

Можно подключить модуль `os` и командой `os.remove("имя_файла")` или командой `os.unlink("имя_файла")` удалить файл.

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

Методы: plot, linspace, show, title, xlabel/ylabel, legend