

Обработка и визуализация данных в Python ЛАБОРАТОРНАЯ РАБОТА

<http://www.kimrt.ru>

Онлайн-курс

[Суперкомпьютерные технологии в задачах моделирования](#)

Для статистической обработки данных в языке Python есть как стандартная библиотека **statistics**, так и сторонние библиотеки, например **Numpy**. Для построения графиков удобно использовать библиотеку **matplotlib**, а пользовательский интерфейс в данной лабораторной работе предлагается реализовать стандартным средством языка Python – **tkinter**.

Для проверки результатов текущей работы можно скачать архив с данными [1], использовавшийся в одной из самостоятельных работ [2].

В самостоятельной работе [3] описывается способ создания оконного приложения с выводом графика некоторой функции. В данной работе график будет построен не для задаваемой функции, а для некоторого входного набора данных.

Для начала нужно получить список входных файлов.

```
fileNames = filedialog.askopenfilenames(initialdir=os.path.abspath(os.getcwd()))
```

В интерфейсе программы можно отобразить количество выбранных файлов. Это можно сделать, привязав виджет Label к глобальной переменной типа StringVar. Данные типы реализованы в tkinter. Этой переменной придется присваивать новое значение каждый раз, когда мы выбираем файлы.

```
def getFileNames():  
    global fileNames  
    fileNames = filedialog.askopenfilenames(initialdir=os.path.abspath(os.getcwd()))  
    fileNamesCountString.set(f'Выбрано файлов: {len(fileNames)}')
```

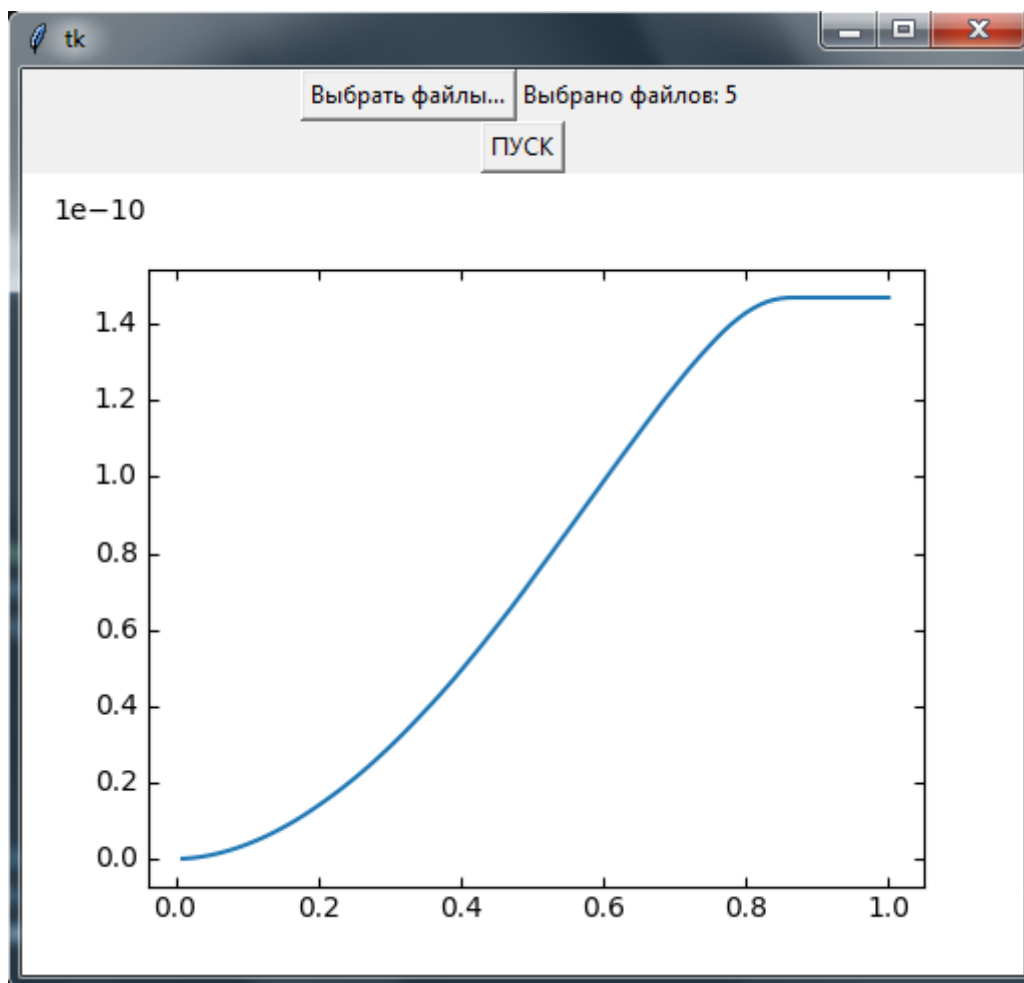
После того как мы получим все имена файлов, мы можем их считать. В результате получаем список матриц. Можно провести усреднение по всем матрицам с помощью функции average, реализованной в библиотеке Numpy.

```
def start():  
    inputTables = []  
    for name in fileNames:  
        with open(name, 'r') as f:  
            inputTables.append([[float(x) for x in line.split()] for line in  
f.readlines()])  
  
    temp = np.average(inputTables, axis=0)
```

<http://www.kimrt.ru>

```
x = [l[0] for l in temp]
y = [l[1] for l in temp]
ax.clear() # отчистка графика
ax.plot(x, y) # построение графика
canvas.draw() # прорисовка графика
```

Подобно тому, как был реализован интерфейс в самостоятельной работе [3], можно реализовать интерфейс, представленный на рисунке.



Виджеты кнопок, текста и т.д. удобно располагать на виджетах Frame. Привязывать команды к кнопкам удобно в конструкторе при инициализации объектов.

```
getFilesFrame = tk.Frame(window)
openFilesButton = tk.Button(getFilesFrame, text='Выбрать файлы...',
command=getFileNames)
fileNamesCountString = tk.StringVar(master=getFilesFrame, value=f'Выбрано файлов:
{len(fileNames)}')
fileNamesCountView = tk.Label(getFilesFrame, textvariable=fileNamesCountString)
openFilesButton.pack(side=tk.LEFT)
fileNamesCountView.pack(side=tk.RIGHT)
getFilesFrame.pack()
```

Задание. Реализуйте программу на языке python, позволяющую считывать текстовые файлы, содержащие по 2 столбца чисел. Программа должна по нажатию кнопки проводить усреднение данных и построение графика. Добавьте в программу переключатель (Checkbutton), позволяющий включать и отключать вычисление и отображение погрешности усреднения [2]. Функция для вычисления погрешностей есть в библиотеке Numpy (вычисляет среднеквадратичное отклонение).

http://www.kimrt.ru/index/course_stm/0-24

Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Источники

1. http://www.kimrt.ru/courses/stm/self_work/msdaverage.zip
2. http://www.kimrt.ru/courses/stm/self_work/Plot_with_error_bar_in_Excel.pdf
3. http://www.kimrt.ru/courses/stm/self_work/Tkinter_and_Matplotlib.pdf