

Санкт-Петербургский Национальный Исследовательский Университет  
Информационных Технологий, Механики и Оптики

Кафедра Систем Управления и Информатики

**Лабораторная работа №6**

**Вариант №1**

Выполнили:

Волгин Л.А.

Шляхов Д.О.

Проверил:

Мусаев А.А.

Санкт-Петербург,

2022

!!!\*Код всех программ можно посмотреть, перейдя по приложенной ссылке на гитхаб.!!!  
[https://github.com/Puroblast/Programming\\_hoomework/tree/main/Dz\\_6](https://github.com/Puroblast/Programming_hoomework/tree/main/Dz_6)

## Задание №1

### Описание задания:

Имеется документ в google-таблицах, содержащий информацию о курсе любой валюты, выбранной студентом. Оценить, завышен или занижен её курс, используя авторегрессионную модель. Допускается использовать любые библиотеки.

### Листинг (Решение):

Программа берёт данные и применяет к ним алгоритм регрессии, с коэффициентами, рассчитанными с помощью мнк.

```
def cov(X, Y):
    average_x = sum(X) / len(X)
    average_y = sum(Y) / len(Y)
    numerator = 0
    for i in range(len(X)):
        numerator += (Y[i] - average_y) * (X[i] - average_x)

    return numerator / (len(X) - 1)

def var(X):
    average_x = sum(X) / len(X)
    numerator = 0
    for i in range(len(X)):
        numerator += math.pow((X[i] - average_x), 2)

    return numerator / len(X)

def findB(X, Y):
    numerator = cov(X, Y)
    denominator = var(X)
    return numerator / denominator

def findA(X, Y, b):
    a = 0
    for i in range(len(X)):
        a += Y[i] - b * X[i]
    return a / len(X)

def regression(data):
    X = data[:-1]
    Y = data[1:]
    b = findB(X, Y)
    a = findA(X, Y, b)
    print(a, b)
    result = []
```

```

for i in data:
    result.append(a + b * i + random.uniform(-0.01 * i, 0.01 * i))

fig, ax = plt.subplots()
ax.plot(data)
ax.plot(result)
plt.show()

gc = gspread.service_account(filename="Token.json")
gsheet = gc.open("MySheet").worksheet("1")
data_all = gsheet.get_all_records()
series = []
for i in range(len(data_all)):
    series.append(data_all[i].get('price'))
regression(series)

```

Рисунки 1,2,3. Собственно сама программа.

### Результат выполнения программы:



Рисунок 4. Результат выполнения программы.

### Вывод:

Изучили алгоритм регрессионного анализа, так же заметили в нём применение МНК.

## Задание №2

### Описание задания:

Имеется оформленный шаблон в таблицах google, содержащий описание и пустое поле для некоторых переменных ( $x$ ,  $y$ ,  $z$ , ...) и функции ( $f(x, y, z, \dots)$ )

Пользователь задает созданном интерфейсе название PDF-файла и переменные, после чего происходит расчет функций и все данные сохраняются в заранее подготовленной форме в гугл-таблицах. Аналогичная таблица сохраняется в виде PDF-файла, название которого содержит дату создания файла (гггг-мм-дд) и название, заданное в интерфейсе (пример: «2022-02-02 – кинетическая энергия первый расчет», «2022-02-03 – кинетическая энергия второй расчет»)

### Листинг (Решение):

Программа просто берёт данные, введенные пользователем, и переносит их в гугл таблицу.

```
def alert(title, message, type_of_alert='info'):
    show_method = getattr(messagebox, 'show{}'.format(type_of_alert))
    show_method(title, message)

def finish(event):
    global sin_mass, sin_radius, filename
    mass = str(int(sin_mass.get()))
    radius = str(int(sin_radius.get()))
    gsheets.update('B1', str(datetime.date.today()))
    gsheets.update('B3', mass)
    gsheets.update('B4', radius)
    name = str(datetime.date.today()) + '-' + filename.get()
    fig, ax = plt.subplots()
    ax.axis('off')
    df = pd.DataFrame(gsheets.get_all_records())
    ax.table(cellText=df.values, colLabels=df.columns, loc='center')
    plt.savefig(f'{name}.pdf')
    alert('Success!', 'Data successfully saved!')
```

Рисунок 5. Отправление данных в таблицу.

```

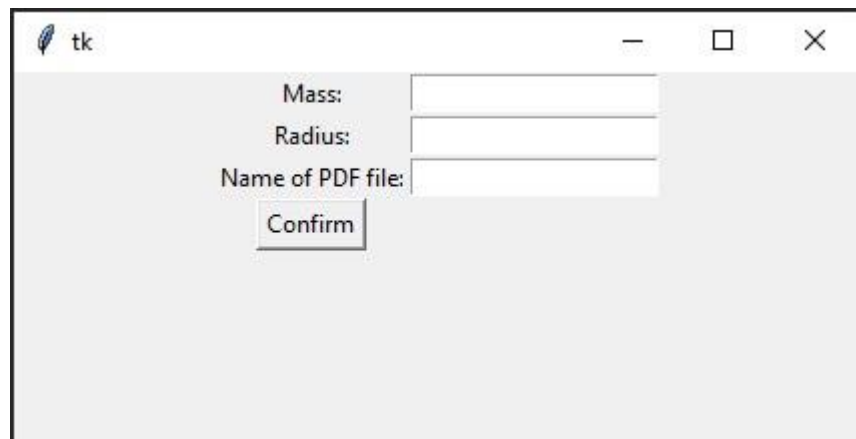
gc = gspread.service_account(filename="Token.json")
gsheet = gc.open("MySheet").worksheet("3")

window = Tk()
frame = Frame(master=window)
label_mass = Label(master=frame, text='Mass:')
label_mass.grid(row=0, column=0)
sin_mass = Entry(master=frame)
sin_mass.grid(row=0, column=1)
label_radius = Label(master=frame, text='Radius:')
label_radius.grid(row=1, column=0)
sin_radius = Entry(master=frame)
sin_radius.grid(row=1, column=1)

```

Рисунок 6. Пример оформления интерфейса.

**Результат выполнения программы:**



The screenshot shows a standard Tkinter window with a title bar containing a feather icon and the text 'tk'. The window has three standard control buttons: minimize, maximize, and close. The main content area is light gray and contains three labels stacked vertically: 'Mass:', 'Radius:', and 'Name of PDF file:'. Each label is followed by a white rectangular input field with a thin gray border. Below these input fields is a single button labeled 'Confirm'.

	2022-06-20
Инерция Шара	3
Масса	2
Радиус	2

	A	B	C
1		2022-06-20	
2	i	11	Инерция Шара
3	m	3	Масса
4	r	3	Радиус

Рисунок 7,8,9. Результат выполнения программы.

**Вывод:**

Если честно, то в реализации выполнения программы не было ничего интересного(.

### Задание №3

#### Описание задания:

Создать интерфейс, который предлагает выбрать и открыть ранее созданные PDF-файлы (директория задается заранее в программе). Пользователь видит название файла без даты (пример: «кинетическая энергия первый расчет», «кинетическая энергия второй расчет»).

#### Листинг (Решение):

Программа ищет pdf-файлы в текущей директории и отправляет их список в radiobutton, после чего открывается выбранный пользователем файл.

```
if __name__ == '__main__':
    regex = r"\d{4}-\d{2}-\d{2}-"
    for x in os.listdir():
        if x.endswith(".pdf"):
            files.append(x)

    row = 0
    column = 0

    window = Tk()
    frame = Frame(master=window)
    label = Label(master=frame, text='Which PDF file you want to open?')
    label.grid(row=row, column=0)
    row += 1
```

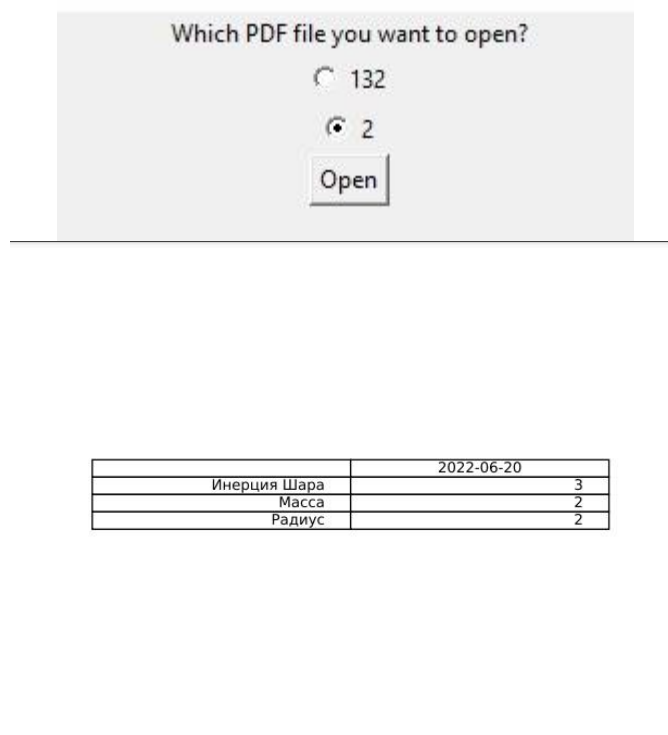
Рисунок 10. Основной код программы.

```
def end():
    global file, var, regex
    d = os.path.dirname(os.path.abspath(var.get()))
    os.system(f'start chrome {d}\\{var.get()}')

def radiobuttons():
    global var, row, column, regex
    for i in range(len(files)):
        f = files[i]
        text = re.sub(regex, '', f.replace('.pdf', ''), 0)
        rb = Radiobutton(master=frame, text=text, value=files[i], variable=var)
        rb.select()
        rb.grid(row=row, column=column)
        row += 1
```

Рисунок 11. Функции программы.

### Результат выполнения программы:



The screenshot shows a window with a dialog box at the top and a table below it. The dialog box asks "Which PDF file you want to open?" and has two radio buttons: "132" (unselected) and "2" (selected). Below the radio buttons is an "Open" button. The table below has three rows and two columns. The first row has a date "2022-06-20" in the second column. The second row has "Инерция Шара" in the first column and "3" in the second column. The third row has "Масса" in the first column and "2" in the second column. The fourth row has "Радиус" in the first column and "2" in the second column.

	2022-06-20
Инерция Шара	3
Масса	2
Радиус	2

Рисунок 12. Результат выполнения программы.

### Вывод:

В реализации этой программы мы познакомились с radio-button.



