МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ТЭВН

Лабораторная работа № 6

“Графический интерфейс пользователя в Python”

Вариант № 11

Факультет: ФЭН

Группа: ЭН2-31

Студент: Кутбиддинов Т.Р.

Преподаватель: Петрищев А.В.

Новосибирск 2024

# 1. Цель работы

Изучение более сложных виджетов библиотеки tkinter: флажки, радиокнопки, фреймы и т. д. Генерация автономного приложения python (выполняемого .exe файла).

# 2. Задание

Написать программу, строящую функции

* предусмотреть флажок: удалять или нет предыдущий график;
* тип линии графика должен выбираться при помощи радиокнопок (три варианта: сплошная, пунктирная и штрих-пунктирная);
* выполнить упаковку программы в автономный выполняемый (.exe) файл.

# 3. Пояснения к заданию

## Листинг программы

### Проведение расчётов

import numpy as np

import sympy as sy

# Задание функции

def f(x):

    global A, alpha, m

    return A \* np.e\*\*(-alpha \* (x - m))

# Поиск определённого интеграла

def solve():

    global x\_min, x\_max, A, alpha, m

    x = sy.Symbol("x")

    solve = sy.integrate(f(x), (x, x\_min, x\_max))

    return solve

### Настройки графика

import matplotlib.pyplot as plt

import locale

def plotting\_settings():

    global x\_min, x\_max, A, alpha, m, fig, ax

    # Настройки графика

    fig, ax = plt.subplots()

    locale.setlocale(locale.LC\_NUMERIC, "de\_RU")

    font = {'family': font\_root,

            'size': fontsize\_root}

    plt.rc('font', \*\*font)

    ax.ticklabel\_format(useLocale=True)

    ax.grid(linewidth = 0.5, color='#7b7b7b')

    ax.spines['left'].set\_position('zero')

    ax.spines['right'].set\_color('none')

    ax.yaxis.tick\_left()

    ax.spines['bottom'].set\_position('zero')

    ax.spines['top'].set\_color('none')

    ax.xaxis.tick\_bottom()

    # Настройка офомления графика

    ax\_color = foreground\_root

    ax.set\_facecolor(background\_root)

    fig.set\_facecolor(background\_root)

    ax.spines['bottom'].set\_color(ax\_color)

    ax.spines['left'].set\_color(ax\_color)

    ax.tick\_params(axis='x', colors=ax\_color)

    ax.tick\_params(axis='y', colors=ax\_color)

### Построение интеграла

def plotting\_integral():

    global x\_min, x\_max, A, alpha, m, fig, ax

    # Откладывание интеграла в виде пунктирной прямой

    x = np.linspace(x\_min, x\_max, 1000)

    y = solve() + x\*0

    ax.plot(x, y, '--', color=bluecolor)

    plt.ylim(0)

    # Сохранение файла

    fig.set\_size\_inches(7.2, 7.2)

    fig.savefig('graph.png', dpi=100)

### Построение нового графика поверх старого

def plotting\_new():

    global x\_min, x\_max, A, alpha, m, fig, ax

    # Постройка графика

    x = np.linspace(x\_min, x\_max, 1000)

    ax.plot(x, f(x), color=bluecolor)

    plt.ylim(0)

    # Сохранение файла

    fig.set\_size\_inches(7.2, 7.2)

    fig.savefig('graph.png', dpi=100)

### Создание интерфейса

from tkinter import \*

from tkinter import ttk

import re

# Костыль для того, чтобы интерфейс не был размытым

from ctypes import windll

windll.shcore.SetProcessDpiAwareness(1)

# Создание окна программы

root = Tk()

root.title("Калькулятор")

root.geometry("1080x720")

root.resizable(False, False)

# Настройка оформления окна

background\_root = '#181818'

foreground\_root = '#cccccc'

bluecolor = '#4a9cd6'

fontsize\_root = 16

font\_root = 'Arial'

root.configure(bg=background\_root)

# Настройка веса столбцов, т. е. какую часть окна занимает каждый столбец

root.columnconfigure(index=0, weight=4)

for column in range(1, 2+1): root.columnconfigure(index=column, weight=1)

# Вывод графика

initial\_graph = PhotoImage(file="initial\_graph.png")

canvas\_graph = Canvas(width=720, height=720, background=background\_root, highlightthickness=0)

canvas\_graph.create\_image(360, 360, image=initial\_graph)

canvas\_graph.grid(row=0, column=0, rowspan=10)

# Вывод интеграла

image\_integral = PhotoImage(file="integral.png")

label\_integral = ttk.Label(image=image\_integral, background=background\_root)

label\_integral.grid(row=0, column=1, columnspan=2)

## Ввод параметров

# Задание валидации

def is\_valid(newval):

    return re.match("^[-]?\d\*[.,]?\d\*$", newval) is not None

check = (root.register(is\_valid), "%P")

# Настройка оформления полей ввода

font\_entry = 'Arial'

fontsize\_entry = fontsize\_root

background\_entry = '#1f1f1f'

foreground\_entry = foreground\_root

insertbackground\_entry = foreground\_entry

highlightthickness\_entry = 0

border\_entry = 0

width\_entry = 15

# A

image\_A = PhotoImage(file="A.png")

label\_A = ttk.Label(image=image\_A, background=background\_root)

label\_A.grid(row=1, column=1)

entry\_A = Entry(validate="key",

                validatecommand=check,

                background=background\_entry,

                insertbackground=insertbackground\_entry,

                foreground=foreground\_entry,

                highlightthickness=highlightthickness\_entry,

                border=border\_entry,

                font=(font\_entry, fontsize\_entry, 'bold'),

                width=width\_entry

                )

entry\_A.insert(0, "1")

entry\_A.grid(row=1, column=2)

# alpha

image\_alpha = PhotoImage(file="alpha.png")

label\_alpha = ttk.Label(image=image\_alpha, background=background\_root)

label\_alpha.grid(row=2, column=1)

entry\_alpha = Entry(validate="key",

                    validatecommand=check,

                    background=background\_entry,

                    insertbackground=insertbackground\_entry,

                    foreground=foreground\_entry,

                    highlightthickness=highlightthickness\_entry,

                    border=border\_entry,

                    font=(font\_entry, fontsize\_entry, 'bold'),

                    width=width\_entry

                    )

entry\_alpha.insert(0, "-1")

entry\_alpha.grid(row=2, column=2)

# m

image\_m = PhotoImage(file="m.png")

label\_m = ttk.Label(image=image\_m, background=background\_root)

label\_m.grid(row=3, column=1)

entry\_m = Entry(validate="key",

                validatecommand=check,

                background=background\_entry,

                insertbackground=insertbackground\_entry,

                foreground=foreground\_entry,

                highlightthickness=highlightthickness\_entry,

                border=border\_entry,

                font=(font\_entry, fontsize\_entry, 'bold'),

                width=width\_entry

                )

entry\_m.insert(0, "0")

entry\_m.grid(row=3, column=2)

# x\_min

image\_x\_min = PhotoImage(file="x\_min.png")

label\_x\_min = ttk.Label(image=image\_x\_min, background=background\_root)

label\_x\_min.grid(row=4, column=1)

entry\_x\_min = Entry(validate="key",

                    validatecommand=check,

                    background=background\_entry,

                    insertbackground=insertbackground\_entry,

                    foreground=foreground\_entry,

                    highlightthickness=highlightthickness\_entry,

                    border=border\_entry,

                    font=(font\_entry, fontsize\_entry, 'bold'),

                    width=width\_entry

                    )

entry\_x\_min.insert(0, "0")

entry\_x\_min.grid(row=4, column=2)

# x\_max

image\_x\_max = PhotoImage(file="x\_max.png")

label\_x\_max = ttk.Label(image=image\_x\_max, background=background\_root)

label\_x\_max.grid(row=5, column=1)

entry\_x\_max = Entry(validate="key",

                    validatecommand=check,

                    background=background\_entry,

                    insertbackground=insertbackground\_entry,

                    foreground=foreground\_entry,

                    highlightthickness=highlightthickness\_entry,

                    border=border\_entry,

                    font=(font\_entry, fontsize\_entry, 'bold'),

                    width=width\_entry

                    )

entry\_x\_max.insert(0, "1")

entry\_x\_max.grid(row=5, column=2)

## Вывод флажков

# Для построения значения интеграла

enabled\_checkbutton\_integral = IntVar()

checkbutton\_integral = Checkbutton(text='Отложить значение интеграла',

                                   variable=enabled\_checkbutton\_integral,

                                   background=background\_root,

                                   foreground=foreground\_root,

                                   activebackground=background\_root,

                                   activeforeground=foreground\_root,

                                   font=('Arial', 12, 'bold'),

                                   selectcolor=background\_entry,

                                   )

checkbutton\_integral.grid(row=6, column=1, columnspan=2, sticky='w')

# Для построения нового графика поверх старого

enabled\_checkbutton\_new = IntVar()

checkbutton\_new = Checkbutton(text='Не удалять предыдущий график',

                              variable=enabled\_checkbutton\_new,

                              background=background\_root,

                              foreground=foreground\_root,

                              activebackground=background\_root,

                              activeforeground=foreground\_root,

                              font=('Arial', 12, 'bold'),

                              selectcolor=background\_entry

                              )

checkbutton\_new.grid(row=7, column=1, columnspan=2, sticky='w')

## Вывод кнопок

# Для очищения полей ввода

def click\_clear():

    entry\_A.delete(0, END)

    entry\_alpha.delete(0, END)

    entry\_m.delete(0, END)

    entry\_x\_min.delete(0, END)

    entry\_x\_max.delete(0, END)

btn\_clear = Button(text='C',

                   command=click\_clear,

                   background=bluecolor,

                   foreground=background\_root,

                   activebackground=background\_root,

                   activeforeground=bluecolor,

                   highlightthickness=0,

                   width=3,

                   height=1,

                   border=0,

                   cursor='hand2',

                   font=('Arial', 16, 'bold'),

                   relief=SUNKEN

                   )

btn\_clear.grid(row=8, column=1)

# Для расчёта

def click\_solve():

    global x\_min, x\_max, A, alpha, m, graph

    A = float(entry\_A.get())

    alpha = float(entry\_alpha.get())

    m = float(entry\_m.get())

    x\_min = float(entry\_x\_min.get())

    x\_max = float(entry\_x\_max.get())

    label\_solve["text"] = solve()

    if enabled\_checkbutton\_new.get() == 0:

        plotting\_settings()

    plotting\_new()

    if enabled\_checkbutton\_integral.get() == 1:

        plotting\_integral()

    graph = PhotoImage(file="graph.png")

    canvas\_graph.create\_image(360, 360, image=graph)

btn\_solve = Button(text="Результат",

                   command=click\_solve,

                   background=bluecolor,

                   foreground=background\_root,

                   activebackground=background\_root,

                   activeforeground=bluecolor,

                   highlightthickness=0,

                   width=13,

                   height=1,

                   border=0,

                   cursor='hand2',

                   font=('Arial', 16, 'bold'),

                   relief=SUNKEN

                   )

btn\_solve.grid(row=8, column=2)

# Вывод результата

label\_solve = Label(

    text="Ответ",

    background=background\_root,

    foreground=bluecolor,

    highlightbackground=bluecolor,

    highlightthickness=2,

    height=2,

    width=20,

    font=(font\_root, fontsize\_root, 'bold')

    )

label\_solve.grid(row=9, column=1, columnspan=2)

# Вывод окна

plotting\_settings()

root.mainloop()

Для создания автономного приложения в консоль была импортирована библиотека pyinstaller и в консоль введена команда:

**pyinstaller --onefile --noconsole InfLab6.py**

Интерфейс программы представлен на рис. № 1.

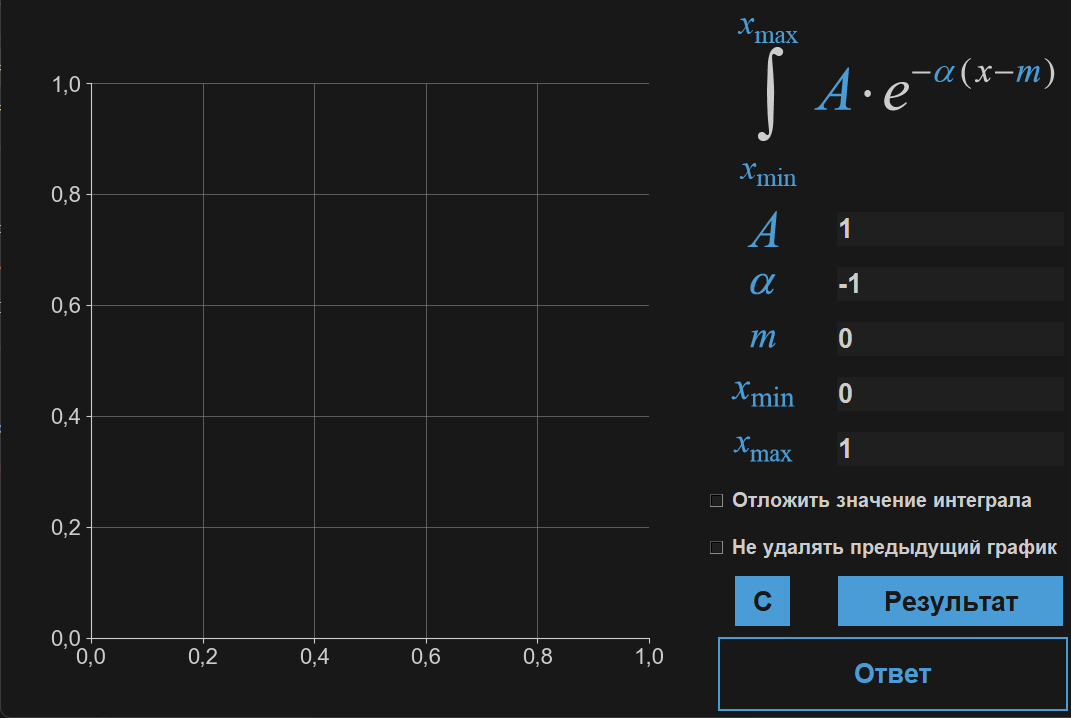


Рис. № 1 – Интерфейс программы

# 4. Выводы

Были изучены более сложные виджеты библиотеки tkinter и создано автономное приложение.