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

# Задание функции

def f(x):

    global A1, A2, T1, T2

    return A1 \* np.exp(-x / T1) + A2 \* np.exp(-x / T2)

### Настройки графика

import matplotlib.pyplot as plt

import locale

def plotting\_settings():

    global x\_min, x\_max, A1, A2, T1, T2, fig, ax, linestyle

    # 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 нового графика поверх старого

def plotting\_new():

    global x\_min, x\_max, A1, A2, T1, T2, fig, ax

    # Постройка графика

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

    ax.plot(x, f(x), color=bluecolor, linestyle=linestyle.get())

    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 = '#66cc33'

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='white', highlightthickness=0)

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

canvas\_graph.grid(row=0, column=0, rowspan=12)

# Вывод функции

image\_function = PhotoImage(file="function.png")

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

label\_function.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

# A1

image\_A1 = PhotoImage(file="A1.png")

label\_A1 = ttk.Label(image=image\_A1, background=background\_root)

label\_A1.grid(row=1, column=1)

entry\_A1 = 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\_A1.insert(0, "1")

entry\_A1.grid(row=1, column=2)

# A2

image\_A2 = PhotoImage(file="A2.png")

label\_A2 = ttk.Label(image=image\_A2, background=background\_root)

label\_A2.grid(row=2, column=1)

entry\_A2 = 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\_A2.insert(0, "1")

entry\_A2.grid(row=2, column=2)

# T1

image\_T1 = PhotoImage(file="T1.png")

label\_T1 = ttk.Label(image=image\_T1, background=background\_root)

label\_T1.grid(row=3, column=1)

entry\_T1 = 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\_T1.insert(0, "1")

entry\_T1.grid(row=3, column=2)

# T2

image\_T2 = PhotoImage(file="T2.png")

label\_T2 = ttk.Label(image=image\_T2, background=background\_root)

label\_T2.grid(row=4, column=1)

entry\_T2 = 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\_T2.insert(0, "1")

entry\_T2.grid(row=4, 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=5, 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=5, 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=6, 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=6, column=2)

## Вывод флажков

# 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 радиокнопок

# Сплошная линия

solid = '-'

linestyle = StringVar(value=solid)

solid\_radiobutton = Radiobutton(text='Сплошная линия',

                                value=solid,

                                variable=linestyle,

                                background=background\_root,

                                foreground=foreground\_root,

                                activebackground=background\_root,

                                activeforeground=foreground\_root,

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

                                selectcolor=background\_entry

                                )

solid\_radiobutton.grid(row=8, column=1, columnspan=2, sticky='w')

# Пунктирная линия

dashed\_radiobutton = Radiobutton(text='Пунктирная линия',

                                value='--',

                                variable=linestyle,

                                background=background\_root,

                                foreground=foreground\_root,

                                activebackground=background\_root,

                                activeforeground=foreground\_root,

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

                                selectcolor=background\_entry

                                )

dashed\_radiobutton.grid(row=9, column=1, columnspan=2, sticky='w')

# Штрих-пунктирная линия

dasheddotted\_radiobutton = Radiobutton(text='Штрих-пунктирная линия',

                                       value='-.',

                                       variable=linestyle,

                                       background=background\_root,

                                       foreground=foreground\_root,

                                       activebackground=background\_root,

                                       activeforeground=foreground\_root,

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

                                       selectcolor=background\_entry

                                       )

dasheddotted\_radiobutton.grid(row=10, column=1, columnspan=2, sticky='w')

## Вывод кнопок

# Для очищения полей ввода

def click\_clear():

    entry\_A1.delete(0, END)

    entry\_A2.delete(0, END)

    entry\_T1.delete(0, END)

    entry\_T2.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=11, column=1)

# Для расчёта

def click\_solve():

    global x\_min, x\_max, A1, A2, T1, T2, graph

    A1 = float(entry\_A1.get())

    A2 = float(entry\_A2.get())

    T1 = float(entry\_T1.get())

    T2 = float(entry\_T2.get())

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

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

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

        plotting\_settings()

    plotting\_new()

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

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

btn\_solve = Button(text="Построить",

                   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=11, column=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.

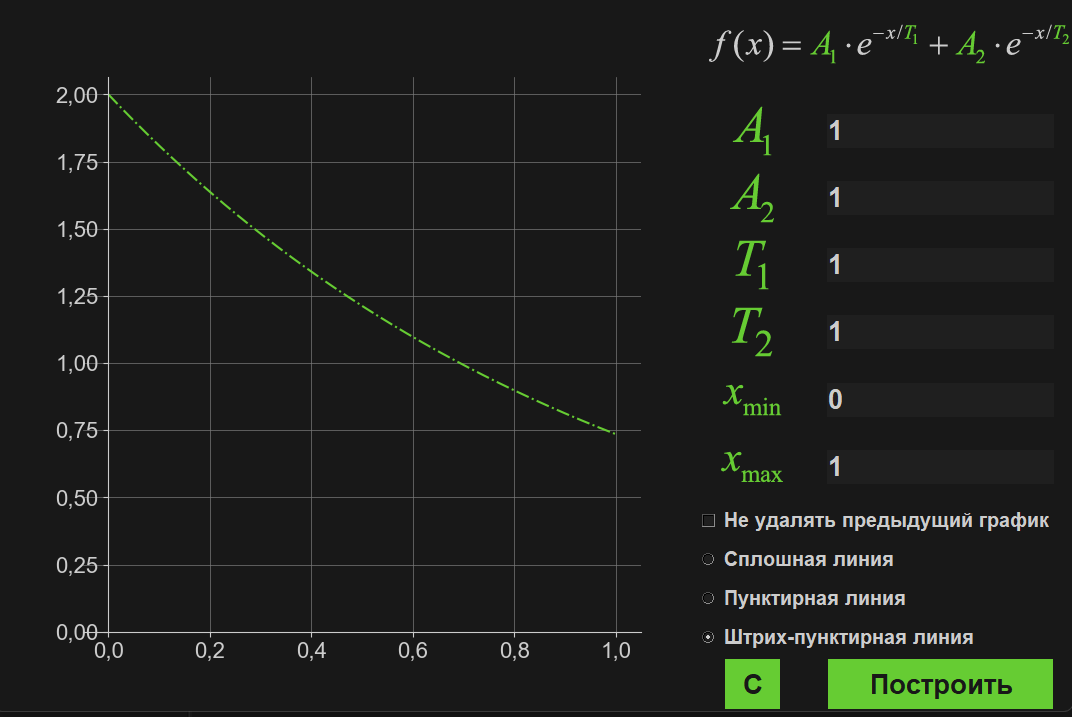


Рис. № 1 – Интерфейс программы

# 4. Выводы

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