

## Лабораторная работа 5

### Графический интерфейс

#### Задание.

С использованием библиотеки Tkinter реализовать графический редактор. Использовать классы Menu, Button, Edit и др. В редакторе предусмотреть кроме основных функций Создать, Сохранить, Сохранить как. (выбор формата), выбор цветовой палитры для пера и кисти, простую анимацию созданных объектов. Отдельным пунктом меню реализовать графический построитель, причем функцию пользователь вводит сам, а не выбирает из списка. Диапазон построения задается через соответствующие компоненты.

Код:

```
from tkinter import *
from tkinter import ttk
from tkinter.colorchooser import askcolor
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt

class Paint(Frame):

    def __init__(self, parent):
        Frame.__init__(self, parent)

        self.parent = parent
        self.color = "black"
        self.brush_size = 2
        self.setUI()

    def draw(self, event):
        self.canv.create_oval(event.x - self.brush_size / 2, event.y -
self.brush_size / 2,
                                event.x + self.brush_size / 2, event.y +
self.brush_size / 2,
                                fill=self.color, outline=self.color)

    def setUI(self):
        self.pack()
        self.canv = Canvas(self, width=1000, height=500, bg="white")
        self.canv.place(relx=0.5, rely=0.5)
        self.columnconfigure(0, weight=1)
        self.rowconfigure(0, weight=1)
        self.canv.grid(padx=10, pady=50)
        self.canv.bind("<B1-Motion>", self.draw)
        self.canv.bind("<Button-1>", self.draw)

def hide_bth(widget):
    widget.place_forget()

def show_btn(widget, lx, ly):
    widget.place(x = lx, y = ly)
```

```

def change_color():
    colors = askcolor(title="Tkinter Color Chooser")
    app.color = colors[1]
def change_size():
    res = int(brs_t.get())
    app.brush_size = res
def brush(color, brush_size):
    app.color = color
    app.brush_size = brush_size
    er_br.configure(text="Eraser", command=eraser)
    brs_l.configure(text="Brush Size")
    brs_b.configure(text="Select Brush Size")
    show_btn(clr_b, 10, 20)
def eraser():
    cl_tmp = app.color
    brs_tmp = app.brush_size
    app.color = "white"
    app.brush_size = 20
    er_br.configure(text="Brush", command=lambda: brush(cl_tmp, brs_tmp))
    brs_l.configure(text="Eraser Size")
    brs_b.configure(text="Select Eraser Size")
    hide_bth(clr_b)
def save_pic():
    fileName = (ent_n.get())
    fileType = (cb_t.get())
    app.canv.postscript(file = fileName + '.eps', colormode='color')
    img = Image.open(fileName + '.eps')
    if fileType == "":
        img.save(fileName + '.jpg')
        hide_bth(ent_n)
        hide_bth(ent_nl)
        hide_bth(sav_b)
    if fileType != "":
        img.save(fileName + fileType)
        cb_t.current(0)
        hide_bth(ent_n)
        hide_bth(ent_nl)
        hide_bth(cb_t)
        hide_bth(cb_tl)
        hide_bth(sav_b)
def func():
    ax = plt.subplot()
    x = np.linspace(float(fn_a.get()), float(fn_b.get()), 1000)
    y = eval(fn_f.get())
    ax.plot(x, y)
    plt.show()

window = Tk()
window.geometry("1000x700")

app = Paint(window)

clr_b = Button(window, text='Select Color', command=change_color)
clr_b.place(x=10, y=20)
brs_l = Label(text="Brush size")
brs_l.place(x=110, y=2)
brs_t = Entry(window, width=10)
brs_t.place(x=110, y=24)
brs_b = Button(window, text='Select Brush Size', command=change_size)
brs_b.place(x=180, y=20)
er_br = Button(window, text='Eraser', command=eraser)
er_br.place(x=290, y=20)

```

```

ent_n1 = Label(text="Img Name")
ent_n = Entry(window, width=10)
cb_t1 = Label(text="Img Type")
cb_t = ttk.Combobox(window, values=[".jpg", ".png", ".jpeg"])
cb_t.current(0)
sav_b = Button(window, text='Save', command=save_pic)

menubar = Menu(window)
savemenu = Menu(menubar, tearoff=0)
savemenu.add_command(label="Save", command=lambda:[show_btn(ent_n, 380, 24),
show_btn(ent_n1, 380, 2), show_btn(sav_b, 520, 20)])
savemenu.add_command(label="Save As", command=lambda:[show_btn(ent_n, 380,
24), show_btn(ent_n1, 380, 2), show_btn(cb_t, 450, 24),
show_btn(cb_t1, 450,
2), show_btn(sav_b, 600, 20)])
menubar.add_cascade(label="Save", menu=savemenu)
window.config(menu=menubar)

fn_a = Entry(window, width=10)
fn_a.place(x=660, y=24)
fn_al = Label(text="From")
fn_al.place(x=660, y=2)
fn_b = Entry(window, width=10)
fn_b.place(x=730, y=24)
fn_bl = Label(text="To")
fn_bl.place(x=730, y=2)
fn_f = Entry(window, width=10)
fn_f.place(x=800, y=24)
fn_fl = Label(text="Func")
fn_fl.place(x=800, y=2)
fn_but = Button(window, text='Draw', command=func)
fn_but.place(x=870, y=20)

window.mainloop()

```

Результат:

