

# GUI

May 27, 2023

```
[ ]: Lab11: GUI
```

```
[ ]: 1. Aplikacja do wczytywania obrazków
```

```
# -*- coding: utf-8 -*-

from tkinter import BOTH, Tk, W, E, N, S, Canvas, NW, messagebox
from tkinter.ttk import Frame, Style, Label, Entry, Button, Combobox

# script 6
from PIL import Image, ImageTk, ImageFilter

# script 5
max_h = 500
max_w = 900

class Okno(Frame):
    def __init__(self, parent):
        super().__init__(parent)
        self.parent = parent
        self.inicializuj()

    # script 6
    def wczytaj_ponownie(self):
        # script 12
        szer, wys = self.im.size
        factor_w = 1
        factor_h = 1
        if szer > max_w:
            factor_w = max_w / szer
        if wys > max_h:
            factor_h = max_h / wys
        if factor_h >= factor_w:
            factor = factor_w
        else:
            factor = factor_h
```

```

size = int(szer * factor), int(wys * factor)
self.image = ImageTk.PhotoImage(self.im.resize(size)) # script 12 self.
→ im

self.base.create_image(0, 0, image=self.image, anchor=NW)

def wczytaj_obraz(self):
    sciezka = self.o.get()
    # script 11
    try:
        self.im = Image.open(sciezka)
        self.fbtn.config(state='normal')
        self.zbtn.config(state='normal')
        self.sbtn.config(state='normal')
        self.pbtn.config(state='normal')
        self.obraz_oryg = self.im
        self.wczytaj_ponownie()
    except FileNotFoundError:
        messagebox.showerror('Błąd!', 'Plik nie istnieje!')
    except OSError:
        messagebox.showerror('Błąd!', 'Podaj plik graficzny!')

# script 9
def skaluj(self):
    w, h = self.im.size
    mnoznik = float(self.scbox.get())
    size = int(w * mnoznik), int(h * mnoznik)
    self.im = self.im.resize(size)
    self.wczytaj_ponownie()

def przywroc_obraz(self):
    self.im = self.obraz_oryg
    self.wczytaj_ponownie()

# script 8
def zapisz(self):
    sciezka = self.z.get()
    if sciezka == '':
        sciezka = self.o.get()
    self.im.save(sciezka)

# script 10
def zastosuj_filtr(self):
    filtr = self.fcbox.get()
    if filtr == 'BLUR':
        self.im = self.im.filter(ImageFilter.BLUR)
    elif filtr == 'CONTOUR':
        self.im = self.im.filter(ImageFilter.CONTOUR)

```

```

else:
    self.im = self.im.filter(ImageFilter.EMBOSS)
self.wczytaj_ponownie()

def inicializuj(self):
    self.parent.title("PiSoft")
    self.style = Style()
    self.style.theme_use("winnative") # default
    self.pack(fill=BOTH, expand=1)
    # script 3
    self.columnconfigure(1, weight=1)

    # script2
    lbl = Label(self, text="Ścieżka do pliku:")
    lbl.grid(sticky=W, pady=4, padx=5)

    # script3
    self.o = Entry(self)
    self.o.grid(row=1, column=0, columnspan=2, rowspan=1, padx=5, pady=4,
→sticky=E + W + S + N)

    self.z = Entry(self)
    self.z.grid(row=2, column=0, columnspan=2, rowspan=1, padx=5, pady=4,
→sticky=E + W + S + N)

    otbtn = Button(self, text="Otwórz", command=self.wczytaj_obraz) #
→command - script6
    otbtn.grid(row=1, column=3)

    self.zbtn = Button(self, text="Zapisz", command=self.zapisz) # script
→8 - command
    self.zbtn.grid(row=2, column=3)
    self.zbtn.config(state='disabled')

    # script 4
    self.scbox = Combobox(self, values='0.1 0.2 0.3 0.4')
    # script 11
    self.scbox.current(0)
    self.scbox.grid(row=3, column=0, padx=5, pady=4, sticky=W + N)

    self.fcbox = Combobox(self, values='BLUR CONTOUR EMBOSS')
    # script 11
    self.fcbox.current(0)
    self.fcbox.grid(row=4, column=0, padx=5, pady=4, sticky=W + N)

    self.sbtn = Button(self, text="Skaluj", command=self.skaluj) # script
→9 - command

```

```

        self.sbtn.grid(row=3, column=1, padx=5, pady=4, sticky=W + N)
        self.sbtn.config(state='disabled')

        self.fbtn = Button(self, text="Filtruj", command=self.zastosuj_filtr)
→ # script 10 - command
        self.fbtn.grid(row=4, column=1, padx=5, pady=4, sticky=W + N)
        self.fbtn.config(state='disabled')

        # script 5
        self.base = Canvas(self, width=max_w, height=max_h)
        self.base.grid(row=5, column=0, padx=5, pady=4, sticky=E + W + S + N,
→ columnspan=3)

        self.pbtn = Button(self, text="Przywróć", command=self.przywroc_obraz)
→ # script 9 - command
        self.pbtn.grid(row=5, column=3, padx=5, pady=4, sticky=W + N)
        self.pbtn.config(state='disabled')

def main():
    gui = Tk()
    gui.geometry("1000x700")
    app = Okno(gui)
    gui.mainloop()

if __name__ == '__main__':
    main()

```

## [ ]: 2. Zadanie

Do każdego z podanych niżej zadań dołącz skrypt prezentujący działanie kodu

Zadanie 1 (5 pkt)

-----

Zmodyfikuj zadanie dotyczące prostej bazy produktów spożywczych.  
Aplikacja ma posiadać interfejs graficzny. Użyj biblioteki TkInter