



ЗВО: Національний університет
«Львівська політехніка»
Навчальний рік: 2024/2025
Семестр: весняний

**Кафедра систем автоматизованого
проектування**
Викладач:
Юрій ПАТЕРЕГА

Навчальна дисципліна: Розробка
кросплатформених додатків
Лабораторна робота № 5:
Розроблення додатку для налаштування
параметрів віджета у діалоговому вікні

Група: ПП-26
Студент: Ігор Якіб'юк

Мета роботи

Розробити додаток на основі бібліотеки tkinter для налаштування параметрів заданого віджета у діалоговому вікні.

Додаток повинен містити:

- головне вікно, в якому розміщені
 - віджет, параметри якого необхідно налаштувати (мітка, повзунок, полотно і т.д.);
 - кнопка виклику діалогового вікна «Adjust»;
- діалогове вікно, в якому розміщені
 - група віджетів для налаштування параметрів (текстове поле, прапорці і т.д.);
 - кнопка для затвердження змін «Ok» та кнопка для скасування змін «Cancel».

Інструкція до виконання роботи

1. Вибрати віджет та його параметри для налаштування згідно із вашим номером варіанту:

1	2	3	4	5	6	7	8
tk.Label text width= height bg fg padx=pady	tk.Label text width=height anchor justify wraplength	tk.Label text width= height bg bd relief	tk. Label bitmap width= height bg fg cursor	tk.Button text width=height overrelief activeforeground activebackground	tk.Entry* show width highlightbackground highlightcolor highlightthickness	tk.Entry* state width disabledforeground disabledbackground readonlybackground	tk.Checkbutton text width indicatoron selectcolor offrelief
9	10	11	12	13	14	15	16
tk.Label text width= height bg fg padx=pady	tk.Label text width=height anchor justify wraplength	tk.Label text width= height bg bd relief	tk. Label bitmap width= height bg fg cursor	tk.Button text width=height overrelief activeforeground activebackground	tk.Entry* show width highlightbackground highlightcolor highlightthickness	tk.Entry* state width disabledforeground disabledbackground readonlybackground	tk.Checkbutton text width indicatoron selectcolor offrelief
17	18	19	20	21	22	23	24
tk.Label text width= height bg fg padx=pady	tk.Label text width=height anchor justify wraplength	tk.Label text width= height bg bd relief	tk. Label bitmap width= height bg fg cursor	tk.Button text width=height overrelief activeforeground activebackground	tk.Entry* show width highlightbackground highlightcolor highlightthickness	tk.Entry* state width disabledforeground disabledbackground readonlybackground	tk.Checkbutton text width indicatoron selectcolor offrelief
25	26	27	28	29	30	31	32
tk.Label text width= height bg fg padx=pady	tk.Label text width=height anchor justify wraplength	tk.Label text width= height bg bd relief	tk. Label bitmap width= height bg fg cursor	tk.Button text width=height overrelief activeforeground activebackground	tk.Entry* show width highlightbackground highlightcolor highlightthickness	tk.Entry* state width disabledforeground disabledbackground readonlybackground	tk.Checkbutton text width indicatoron selectcolor offrelief

*Примітка. Задати попередньо текст в текстовому полі tk.Entry.

2. Створити головне вікно та налаштувати його параметри.

2.1. Додати віджет, параметри якого необхідно налаштувати.

2.2. Додати кнопку «Adjust» та обробник для виклику діалогового вікна.

3. Додати діалогове вікно та налаштувати його параметри.

3.1. Додати усі необхідні віджети для налаштування заданих параметрів.

3.2. Додати кнопку «Ok» та обробник для затвердження змін.

3.3. Додати кнопку «Cancel» та обробник для скасування змін.

4. Забезпечити передавання даних між головним та діалоговим вікном.
5. Заблокувати можливість відкриття кількох діалогових вікон одночасно та забезпечити коректне закриття діалогового вікна.
6. Запустити головний цикл обробки подій.

1. Теоретичні відомості

2. Лістинг коду програми

```
import tkinter as tk
import tkinter.ttk as ttk
import os
import csv
import numpy as np
import pandas as pd

root = tk.Tk()
root.title("Розумний список")
root.geometry("570x400+350+0")

toplevel = None

entry_varW = tk.StringVar(value="50")
entry_varH = tk.StringVar(value="30")
bitmap_var = tk.StringVar(value="info")
bg_var = tk.StringVar(value="white")
fg_var = tk.StringVar(value="black")
cursor_var = tk.StringVar(value="arrow")

labelBitmap = tk.Label(
    root,
    bg=bg_var.get(),
    fg=fg_var.get(),
    bitmap=bitmap_var.get(),
    width=int(entry_varW.get()),
    height=int(entry_varH.get()),
    cursor=cursor_var.get(),
    anchor="center"
)
labelBitmap.pack(pady=10)

def close_toplevel():
    global toplevel
    if toplevel is not None:
        toplevel.grab_release()
        toplevel.destroy()
        toplevel = None

def handle_ok():
    labelBitmap.config(
        width=int(entry_varW.get()),
        height=int(entry_varH.get()),
        bg=bg_var.get(),
        fg=fg_var.get(),
        bitmap=bitmap_var.get(),
        cursor=cursor_var.get()
    )
    save_data()
    close_toplevel()

def handle_cancel():
    close_toplevel()
```

```

def save_data():
    data = [bitmap_var.get(), bg_var.get(), fg_var.get(), cursor_var.get()]
    file_exists = os.path.isfile("dataset.csv")

    with open("dataset.csv", "a", newline="") as f:
        writer = csv.writer(f)
        if not file_exists:
            writer.writerow(["bitmap", "bg", "fg", "cursor"])
        writer.writerow(data)

def read_data():
    line = pd.read_csv("dataset.csv")
    intermediateData = line[["bitmap", "bg", "fg", "cursor"]].values
    result = [[item[0], list(item[1:len(item)])] for item in intermediateData]
    return result

def suggestBestChoice(event):
    arrayData = read_data()
    def changeData(bg, fg, cursor):
        bg_var.set(bg)
        fg_var.set(fg)
        cursor_var.set(cursor)
    list(map(lambda item: changeData(*item[1]) if bitmap_var.get() in item else -1,
arrayData))

def handle_adjust():
    global toplevel
    if toplevel is not None:
        return

    toplevel = tk.Toplevel()
    toplevel.title("Налаштування Label")
    toplevel.resizable(False, False)
    toplevel.protocol("WM_DELETE_WINDOW", close_toplevel)

    ttk.Label(toplevel, text="Width:").pack(anchor="w", padx=5)
    ttk.Entry(toplevel, textvariable=entry_varW).pack(fill="x", padx=5, pady=2)

    ttk.Label(toplevel, text="Height:").pack(anchor="w", padx=5)
    ttk.Entry(toplevel, textvariable=entry_varH).pack(fill="x", padx=5, pady=2)

    ttk.Label(toplevel, text="Bitmap:").pack(anchor="w", padx=5)
    bitmap_combo = ttk.Combobox(toplevel, textvariable=bitmap_var, values=[
        "error", "gray75", "gray50", "gray25", "gray12",
        "hourglass", "info", "questhead", "question", "warning"
    ])
    bitmap_combo.bind("<<ComboboxSelected>>", suggestBestChoice)
    bitmap_combo.pack(fill="x", padx=5, pady=2)

    ttk.Label(toplevel, text="Background color:").pack(anchor="w", padx=5)
    ttk.Entry(toplevel, textvariable=bg_var).pack(fill="x", padx=5, pady=2)

    ttk.Label(toplevel, text="Foreground color:").pack(anchor="w", padx=5)
    ttk.Entry(toplevel, textvariable=fg_var).pack(fill="x", padx=5, pady=2)

    ttk.Label(toplevel, text="Cursor:").pack(anchor="w", padx=5)
    cursor_combo = ttk.Combobox(toplevel, textvariable=cursor_var, values=[
        "arrow", "circle", "cross", "hand2", "heart", "plus", "watch", "xterm"
    ])
    cursor_combo.pack(fill="x", padx=5, pady=2)

    tk.Button(toplevel, text="OK", command=handle_ok, width=10).pack(pady=5)
    tk.Button(toplevel, text="Cancel", command=handle_cancel, width=10).pack()

```

```

    toplevel.grab_set()

tk.Button(root, text="Adjust", command=handle_adjust).pack(pady=10)

root.mainloop()

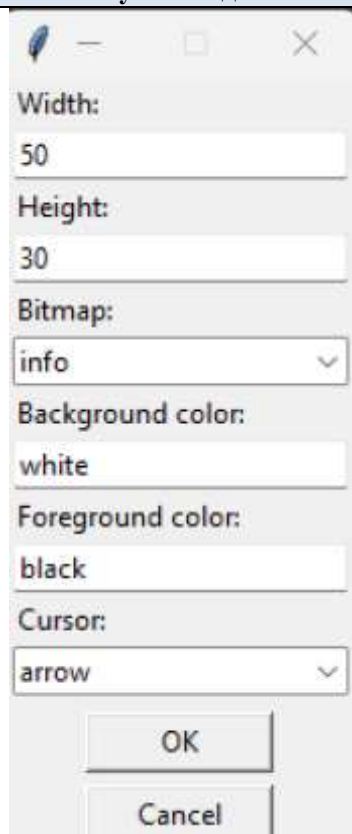
```

3. Перевірка та тестування програми

3.1. Тестування головного вікна



3.2. Тестування діалогового вікна



Висновок

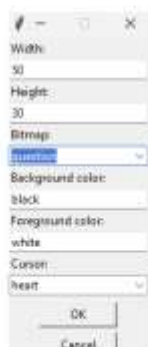
Висновок має відповісти на запитання «Що зроблено?», «Як зроблено?», «Що це дало?».

На даній лабораторній, я навчився працювати з елементом TopLevel. Створив діалогове вікно, яке надає можливість редагувати bitmap іконку. Також програма запам'ятовує наш вибір параметрів та пропонує дані, які ми раніше вводили для певного типу іконки. Виконана робота продемонстрована

вище.

Презентація фішки:

Діалогове вікно, що запам'ятовує твій вибір



Що
зроблено?

Можливість обрати іконку

Можливість налаштувати параметри
іконки через діалогове вікно

Програма запам'ятовує наш останній
вибір по кожній іконці і пропонує його

Зберігання даних у файлі

