

TRƯỜNG ĐẠI HỌC HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



Báo Cáo

Bài Tập: Python Gui

SVTH: Lâm Quang Tùng
MSSV: 2274802010973

GVHD: Huỳnh Thái Học

2024

1. Giao diện

The screenshot shows a window titled "Tính toán" (Calculate) with standard Windows window controls. Inside, there are two tabs: "tab1" (selected) and "tab2". The "tab1" view is divided into two main sections. The left section, titled "Nhập dữ liệu" (Enter data) in blue, contains two input fields labeled "a" and "b" in red. The right section, titled "Chọn" (Choose) in blue, contains two buttons: "Solve" and "Reset". Below these sections, the text "KẾT QUẢ" (Result) is displayed in blue, followed by a large yellow rectangular box intended for the output.

The screenshot shows the same "Tính toán" (Calculate) window, but with the "tab2" view selected. This view features a dark gray rectangular display area at the top. Below the display is a numeric keypad with buttons for digits 0-9, a decimal point, and arithmetic operators: "+", "*", "/", and "-". A "C" button (clear) is also present. The "tab1" tab is visible but not selected.

2. Chức năng

- Tab1: Giải phương trình $ax+b=0$
 - Người dùng nhập vào 2 giá trị a và b trên giao diện
 - Nếu người dùng nhập sai yêu cầu sẽ xuất hiện messagebox báo lỗi
- Tab2: Một chiếc máy tính
 - Thực hiện các tác vụ như tính cộng, trừ nhân chia.
 - Sẽ hiện thông báo “lỗi” nếu nhập phép tính sai
 - Người dùng có thể tự gõ phép tính vào
 - Có thể ấn vào button trên giao diện để nhập phép tính

3. Mã nguồn

```
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
```

```
class MathApp:
    def __init__(self):
        self.win = tk.Tk()
        # self.reset_count = 0
        self.create_tab()
        self.win.title("Tính toán")
        self.win.mainloop()

    def create_tab(self):
        tabControl = ttk.Notebook(self.win)

        # Tab 1 - Solve Linear Equation
        tab1 = ttk.Frame(tabControl)
        tabControl.add(tab1, text='tab1')
        self.create_tab1(tab1)

        # Tab 2 - Calculator
        tab2 = ttk.Frame(tabControl)
        tabControl.add(tab2, text='tab2')
        self.create_tab2(tab2)

        tabControl.grid(column=0, row=0, padx=10, pady=10)
```

```

def create_tab1(self, tab1):

    # LabelFrame
    frbnhap = tk.LabelFrame(tab1, text="Nhập dữ liệu", fg="blue")
    frbnhap.grid(column=0, row=0, sticky="nsew")

    frbxuat = tk.LabelFrame(tab1, text="Chọn", fg="blue")
    frbxuat.grid(column=1, row=0, padx=10, sticky="nsew")

    # Label
    lba = tk.Label(frbnhap, text="a", fg="red")
    lba.grid(column=0, row=0, padx=5, pady=5)

    lbb = tk.Label(frbnhap, text="b", fg="red")
    lbb.grid(column=0, row=1, padx=5, pady=5)

    kq = tk.Label(tab1, text="KẾT QUẢ", width=10, fg="blue")
    kq.grid(column=0, row= 2, sticky="w", padx=5, pady=5)

    # entry
    self.dla = tk.StringVar()
    entra = tk.Entry(frbnhap, textvariable=self.dla, width="20")
    entra.grid(column=1, row=0, padx=5, pady=5)

    self.dlb = tk.StringVar()
    entrb = tk.Entry(frbnhap, textvariable=self.dlb, width="20")
    entrb.grid(column=1, row=1, pady=5, padx=5)

    self.kqq = tk.StringVar()
    entrkqq = tk.Entry(tab1, textvariable=self.kqq, width=25, background="yellow")
    entrkqq.grid(column=1, row=2, sticky="w", padx=5, pady=5)

    solve = ttk.Button(frbxuat, text="Solve", command=self.solve_output)
    solve.grid(column=0, row=0, padx=5, pady=5)

    reset = ttk.Button(frbxuat, text="Reset", command=self.reset_)
    reset.grid(column=0, row=3, padx=5, pady=5)

    entra.focus()

```

```
def solve_output(self):
    try:
        a = float(self.dla.get())
        b = float(self.dlb.get())
        if a != 0:
            result = -b/a
            self.kqq.set(f"x = {result}")
        elif a == 0 and b != 0:
            self.kqq.set("vô nghiệm")
        elif a == 0 and b == 0:
            self.kqq.set("Nghiệm đúng với mọi x")
    except ValueError:
        messagebox.showerror("lỗi nhập liệu", "\n vui lòng nhập lại")

def reset_(self):
    # if self.reset_count == 0:
    #     self.dlb.set("")
    #     self.reset_count += 1
    # elif self.reset_count == 1:
    #     self.dla.set("")
    #     self.reset_count = 0
    self.dla.set('')
    self.dlb.set('')
    self.kqq.set('')
```

```

def create_tab2(self, tab2):
    # Hiển thị ô nhập liệu
    self.display = tk.StringVar()
    displaya = tk.Entry(tab2, textvariable=self.display, width=20, background="grey", justify='right')
    displaya.grid(column=0, row=0, columnspan=4, padx=5, pady=5)

    # Nút số và phép tính
    btt_7 = tk.Button(tab2, text="7", width=5, command=lambda: self.update_display("7")).grid(column=0, row=1, padx=5, pady=5)
    btt_8 = tk.Button(tab2, text="8", width=5, command=lambda: self.update_display("8")).grid(column=1, row=1, padx=5, pady=5)
    btt_9 = tk.Button(tab2, text="9", width=5, command=lambda: self.update_display("9")).grid(column=2, row=1, padx=5, pady=5)
    btt_cong = tk.Button(tab2, text="+", width=5, command=lambda: self.update_display("+")).grid(column=3, row=1, padx=5, pady=5)

    btt_4 = tk.Button(tab2, text="4", width=5, command=lambda: self.update_display("4")).grid(column=0, row=2, padx=5, pady=5)
    btt_5 = tk.Button(tab2, text="5", width=5, command=lambda: self.update_display("5")).grid(column=1, row=2, padx=5, pady=5)
    btt_6 = tk.Button(tab2, text="6", width=5, command=lambda: self.update_display("6")).grid(column=2, row=2, padx=5, pady=5)
    btt_nhan = tk.Button(tab2, text="*", width=5, command=lambda: self.update_display("*")).grid(column=3, row=2, padx=5, pady=5)

    btt_1 = tk.Button(tab2, text="1", width=5, command=lambda: self.update_display("1")).grid(column=0, row=3, padx=5, pady=5)
    btt_2 = tk.Button(tab2, text="2", width=5, command=lambda: self.update_display("2")).grid(column=1, row=3, padx=5, pady=5)
    btt_3 = tk.Button(tab2, text="3", width=5, command=lambda: self.update_display("3")).grid(column=2, row=3, padx=5, pady=5)
    btt_chia = tk.Button(tab2, text="/", width=5, command=lambda: self.update_display("/")).grid(column=3, row=3, padx=5, pady=5)

    btt_0 = tk.Button(tab2, text="0", width=5, command=lambda: self.update_display("0")).grid(column=0, row=4, padx=5, pady=5)
    btt_reset = tk.Button(tab2, text="C", width=5, command=self.reset_calculator, background="grey").grid(column=1, row=4, padx=5, pady=5)

    btt_tru = tk.Button(tab2, text="-", width=5, command=lambda: self.update_display("-")).grid(column=2, row=4, padx=5, pady=5)
    btt_bang = tk.Button(tab2, text="=", width=5, command=self.calculate).grid(column=3, row=4, padx=5, pady=5)

def update_display(self, value):
    current_text = self.display.get()
    self.display.set(current_text + value)

def reset_calculator(self):
    self.display.set('')

def calculate(self):
    try:
        result = eval(self.display.get())
        self.display.set(result)
    except Exception:
        self.display.set("Lỗi")

if __name__ == "__main__":
    math = MathApp()

```

4. Github