## TRƯỜNG ĐẠI HỌC HỌC VĂN LANG KHOA CÔNG NGHỆ THÔNG TIN



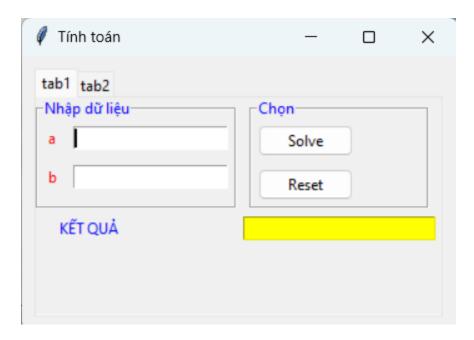


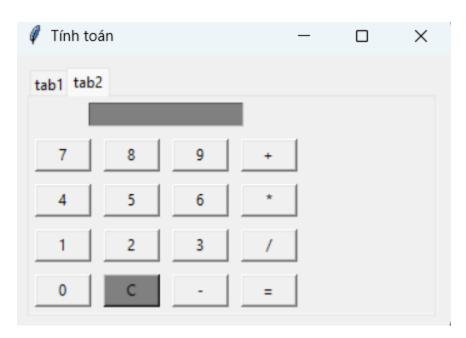
# Báo Cáo Bài Tập: Python Gui

**SVTH:** Lâm Quang Tùng **MSSV:** 2274802010973

GVHD: Huỳnh Thái Học

### 1. Giao diện





#### 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
```

```
MathApp:
 ef init (self):
    self.win = tk.Tk()
    self.create tab()
   self.win.title("Tinh 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)
```

```
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="Chon", fg="blue")
frbxuat.grid(column=1, row=0, padx=10, sticky="nsew")
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="KET QUA", width=10, fg="blue")
kq.grid(column=0, row= 2, sticky="w", padx=5, pady=5)
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()
```

```
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):
   # self.reset count += 1
   # elif self.reset_count == 1:
   self.dla.set('')
    self.dlb.set('')
    self.kqq.set('')
```

```
create_tab2(self, tab2):
      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)
      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)
      update_display(self, value):
current_text = self.display.get()
      self.display.set(current_text + value)
      reset calculator(self):
      self.display.set('')
      calculate(self):
          result = eval(self.display.get())
           self.display.set(result)
       except Exception:
          self.display.set("Lỗi")
 __name__ == "__main__":
math = MathApp()
```

#### 4. Github