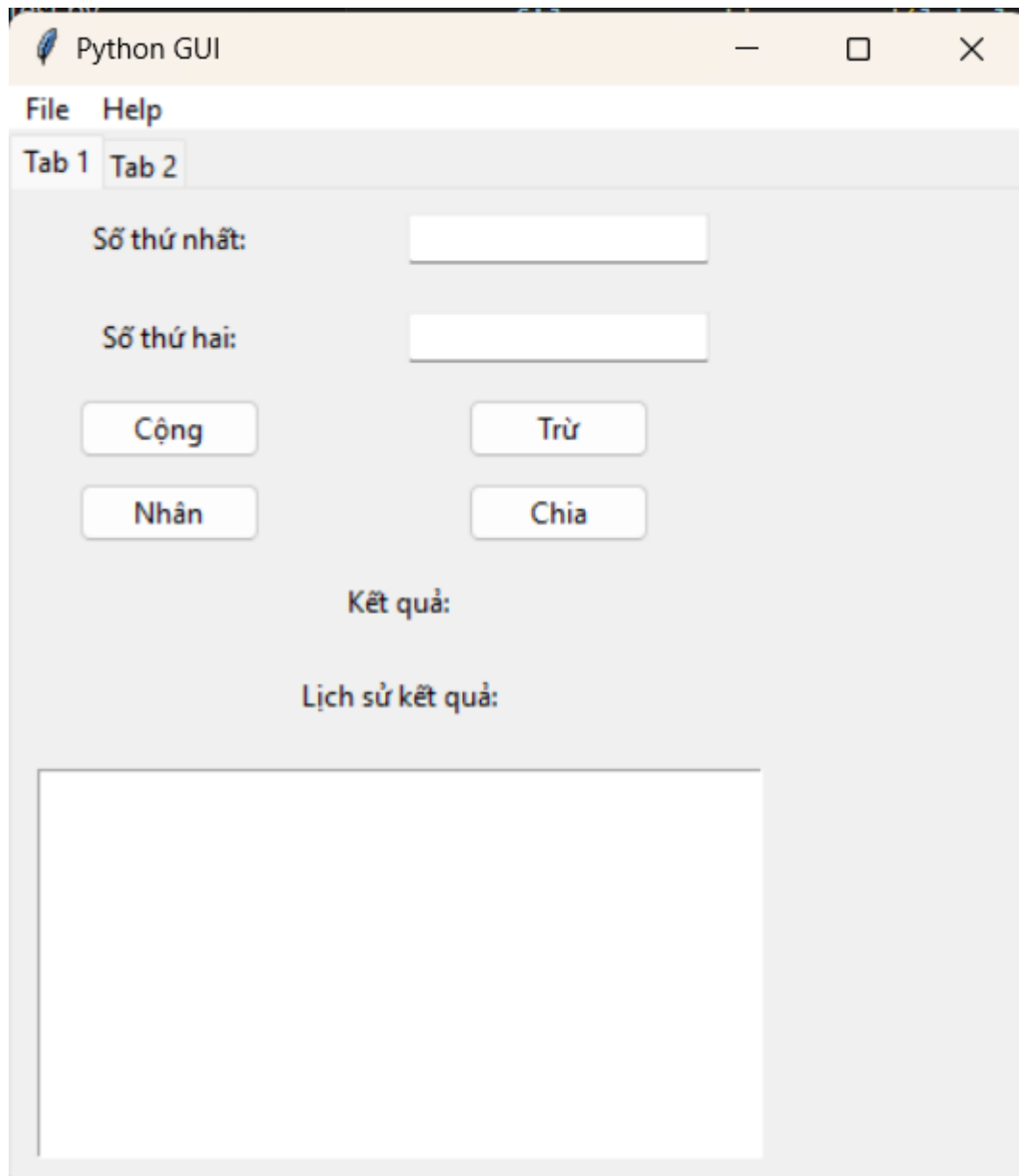


## Project1

2274802010863 Nguyễn Minh Thuận

Nội dung:

### 1. Giao diện



Python GUI

File Help

Tab 1 Tab 2

Tên nhân viên:

Năm sinh:

Chức vụ:

Thêm Nhân viên

## 2.Chức năng

Ở giao diện đầu tên có những chức năng như sau:

- Chức năng ‘cộng’ ‘trừ’ ‘nhân’ ‘chia’ dùng để tính toán theo nhu cầu của người dùng
- Input số thứ nhất và số thứ 2
- Hiển thị kết quả ở ô bên dưới

Số thứ nhất:

Số thứ hai:

Cộng

Trừ

Nhân

Chia

Kết quả:

Lịch sử kết quả:

Ở giao diện thứ 2 có các chức năng như sau:

- Input ‘Tên thành viên’, ‘Năm sinh’, ‘Chức vụ’
- Nút Thêm Nhân viên

Tên nhân viên:

Năm sinh:

Chức vụ:

Thêm Nhân viên

### 3. Mã chương trình:

```
import tkinter as tk
from tkinter import ttk
from tkinter import scrolledtext
from tkinter import Menu
```

```
# Create instance
```

```
win = tk.Tk()
```

```
# Add a title
```

```
win.title("Python GUI")
```

```
tabControl = ttk.Notebook(win)
```

```
tab1 = tk.Frame(tabControl)
```

```

tabControl.add(tab1, text='Tab 1')
tabControl.pack(expand=1, fill="both")

tab2 = tk.Frame(tabControl)
tabControl.add(tab2, text='Tab 2')
tabControl.pack(expand=1, fill="both")
# We are creating a container frame to hold all other widgets
mighty = ttk.LabelFrame(tab1, text=' Mighty Python ')
mighty.grid(column=0, row=0, padx=8, pady=4)
def calculate(operation):
    try:
        num1 = float(entry1.get())
        num2 = float(entry2.get())
        if operation == "add":
            result = num1 + num2
            result_text = f"{num1} + {num2} = {result}"
        elif operation == "subtract":
            result = num1 - num2
            result_text = f"{num1} - {num2} = {result}"
        elif operation == "multiply":
            result = num1 * num2
            result_text = f"{num1} * {num2} = {result}"
        elif operation == "divide":
            if num2 == 0:
                result_text = "Không thể chia cho 0."
            else:
                result = num1 / num2
                result_text = f"{num1} / {num2} = {result}"

        result_label.config(text=result_text)
        history_listbox.insert(tk.END, result_text)
    except ValueError:
        result_label.config(text="Vui lòng nhập số hợp lệ.")

def reload_history():
    history_listbox.delete(0, tk.END) # Xóa tất cả các mục trong danh sách lịch sử

# Tạo các thành phần giao diện
label1 = ttk.Label(tab1, text="Số thứ nhất:")
label1.grid(column=0, row=0, padx=10, pady=10)

entry1 = ttk.Entry(tab1)
entry1.grid(column=1, row=0, padx=10, pady=10)

```

```

label2 = ttk.Label(tab1, text="Số thứ hai:")
label2.grid(column=0, row=1, padx=10, pady=10)

entry2 = ttk.Entry(tab1)
entry2.grid(column=1, row=1, padx=10, pady=10)

# Nút tính toán
add_button = ttk.Button(tab1, text="Cộng", command=lambda: calculate("add"))
add_button.grid(column=0, row=2, pady=5)

subtract_button = ttk.Button(tab1, text="Trừ", command=lambda: calculate("subtract"))
subtract_button.grid(column=1, row=2, pady=5)

multiply_button = ttk.Button(tab1, text="Nhân", command=lambda:
calculate("multiply"))
multiply_button.grid(column=0, row=3, pady=5)

divide_button = ttk.Button(tab1, text="Chia", command=lambda: calculate("divide"))
divide_button.grid(column=1, row=3, pady=5)

result_label = ttk.Label(tab1, text="Kết quả:")
result_label.grid(column=0, row=4, columnspan=2, pady=10)

# Khung cho lịch sử kết quả
history_label = ttk.Label(tab1, text="Lịch sử kết quả:")
history_label.grid(column=0, row=5, columnspan=2, pady=10)

history_listbox = tk.Listbox(tab1, width=50, height=10)
history_listbox.grid(column=0, row=6, columnspan=2, padx=10, pady=10)
# Tạo các nhãn và ô nhập liệu
ttk.Label(tab2, text="Tên nhân viên:").pack(pady=5)
name_entry = ttk.Entry(tab2)
name_entry.pack(pady=5)

ttk.Label(tab2, text="Năm sinh:").pack(pady=5)
year_entry = ttk.Entry(tab2)
year_entry.pack(pady=5)

ttk.Label(tab2, text="Chức vụ:").pack(pady=5)
position_entry = ttk.Entry(tab2)
position_entry.pack(pady=5)

```

```

# Tạo ScrolledText để hiển thị danh sách nhân viên
employee_list = scrolledtext.ScrolledText(tab2, width=50, height=10)
employee_list.pack(pady=10)

# Hàm để thêm nhân viên
def add_employee():
    name = name_entry.get()
    year = year_entry.get()
    position = position_entry.get()

    if name and year and position:
        employee_info = f"Tên: {name}, Năm sinh: {year}, Chức vụ: {position}\n"
        employee_list.insert(tk.END, employee_info)

        # Xóa các ô nhập sau khi thêm
        name_entry.delete(0, tk.END)
        year_entry.delete(0, tk.END)
        position_entry.delete(0, tk.END)
    else:
        print("Vui lòng nhập đầy đủ thông tin.")

# Tạo nút để thêm nhân viên
add_button = tk.Button(tab2, text="Thêm Nhân viên", command=add_employee)
add_button.pack(pady=10)
# Creating a Menu Bar
menu_bar = Menu(win)
win.config(menu=menu_bar)

# Create menu and add menu items
file_menu = Menu(menu_bar, tearoff=0) # create File menu
file_menu.add_command(label="New") # add File menu item
menu_bar.add_cascade(label="File", menu=file_menu) # add File menu to menu bar and
give it a label
file_menu.add_separator()
file_menu.add_command(label="Exit")

help_menu = Menu(menu_bar, tearoff=0) # create File menu
menu_bar.add_cascade(label="Help", menu=help_menu) # add File menu item
help_menu.add_command(label="About")

year_entry.focus()
# Chạy vòng lặp chính
win.mainloop()

```

4.Link github