

ỨNG DỤNG TÍNH TOÁN

Họ tên: Lê Thúy Hiền

MSSV: 2274802010236

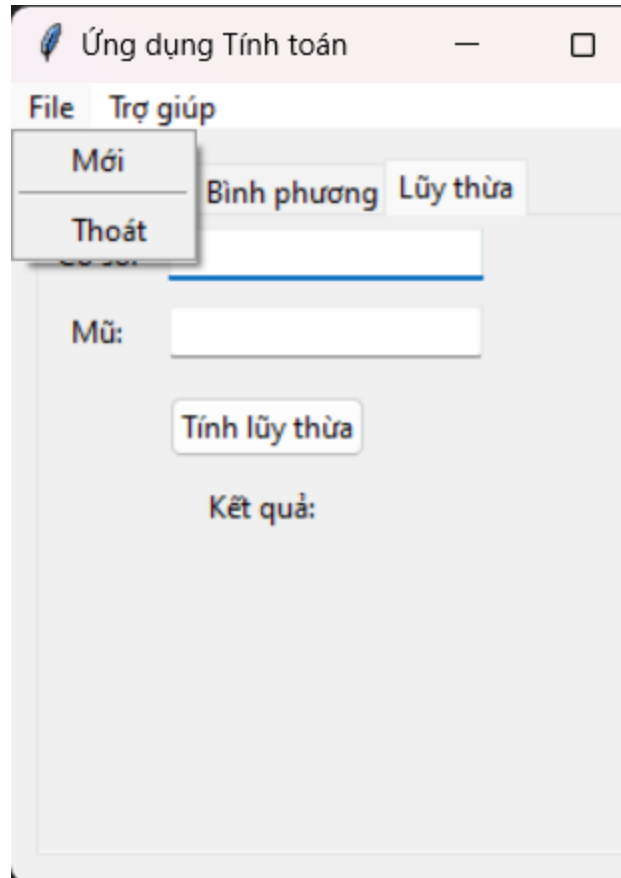
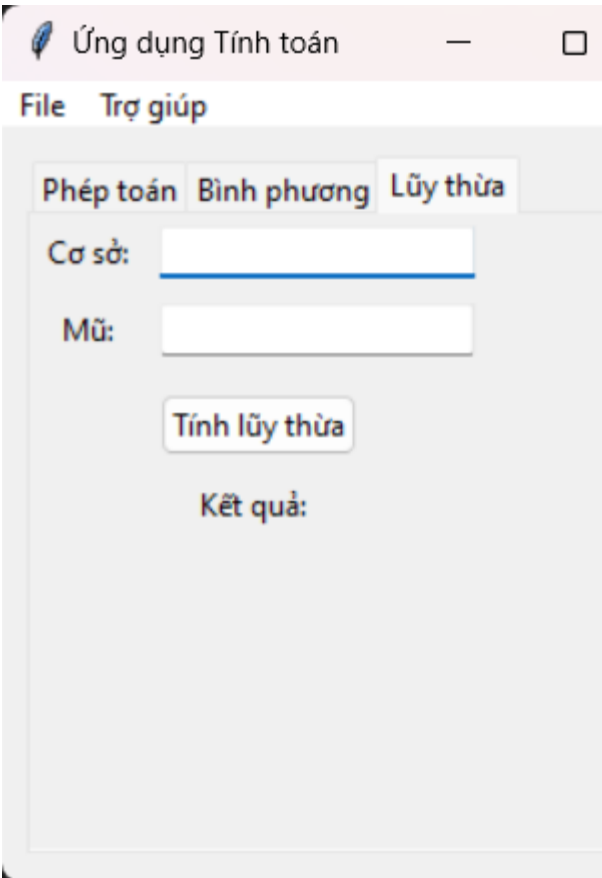
1. Giao diện

The image displays two screenshots of a web-based calculator application titled "Ứng dụng Tính toán" (Calculation Application). Both screenshots show a menu bar with "File" and "Trợ giúp" (Help) options, and three tabs: "Phép toán" (Arithmetic), "Bình phương" (Square), and "Lũy thừa" (Power).

The left screenshot shows the "Phép toán" tab selected. It features two input fields labeled "Số thứ nhất:" (First number) and "Số thứ hai:" (Second number). Below these is a section titled "Các phép toán" (Arithmetic operations) containing four rows, each with a calculation button and a "Kết quả:" (Result) label:

- Tính tổng (Sum)
- Tính hiệu (Difference)
- Tính tích (Product)
- Tính thương (Quotient)

The right screenshot shows the "Bình phương" (Square) tab selected. It features a single input field labeled "Số:" (Number) and a button labeled "Tính bình phương" (Calculate square), followed by a "Kết quả:" (Result) label.



2. Chức năng

Ứng dụng có các chức năng chính như sau:

- Cộng: Tính tổng hai số.
- Trừ: Tính hiệu hai số.
- Nhân: Tính tích hai số.
- Chia: Tính thương hai số, có xử lý trường hợp chia cho 0.
- Bình phương: Tính bình phương của một số.
- Lũy thừa: Tính lũy thừa với cơ sở và mũ do người dùng nhập.

Ngoài ra, ứng dụng còn bao gồm các thông báo lỗi nếu người dùng nhập sai định dạng (không phải số).

3. Mã nguồn

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

```

class CalculatorApp(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title("Ứng dụng Tính toán")
        self.geometry("300x300")

        self.create_menu()
        self.create_notebook()

    def create_menu(self):
        menubar = tk.Menu(self)
        self.config(menu=menubar)

        file_menu = tk.Menu(menubar, tearoff=0)
        menubar.add_cascade(label="File", menu=file_menu)
        file_menu.add_command(label="Mới", command=self.clear_entries)
        file_menu.add_separator()
        file_menu.add_command(label="Thoát", command=self.quit)

        help_menu = tk.Menu(menubar, tearoff=0)
        menubar.add_cascade(label="Trợ giúp", menu=help_menu)
        help_menu.add_command(label="Về ứng dụng",
command=self.show_about)

    def create_notebook(self):
        notebook = ttk.Notebook(self)
        notebook.pack(expand=True, fill="both", padx=10, pady=10)

        self.add_arithmetic_tab(notebook)
        self.add_square_tab(notebook)
        self.add_power_tab(notebook)

    def add_arithmetic_tab(self, notebook):
        arithmetic_frame = ttk.Frame(notebook)
        notebook.add(arithmetic_frame, text="Phép toán")

        ttk.Label(arithmetic_frame, text="Số thứ nhất:").grid(row=0,
column=0, padx=5, pady=5)
        self.add_num1 = ttk.Entry(arithmetic_frame)
        self.add_num1.grid(row=0, column=1, padx=5, pady=5)
        self.add_num1.focus()

```

```

        ttk.Label(arithmetic_frame, text="Số thứ hai:").grid(row=1,
column=0, padx=5, pady=5)
        self.add_num2 = ttk.Entry(arithmetic_frame)
        self.add_num2.grid(row=1, column=1, padx=5, pady=5)

        operation_frame = ttk.LabelFrame(arithmetic_frame, text="Các phép
toán", padding=(10, 10))
        operation_frame.grid(row=2, column=0, columnspan=2, padx=10,
pady=10, sticky="nsew")

        ttk.Button(operation_frame, text="Tính tổng",
command=self.calculate_sum).grid(row=0, column=0, pady=5)
        self.add_result = ttk.Label(operation_frame, text="Kết quả: ")
        self.add_result.grid(row=0, column=1, padx=5, pady=5)

        ttk.Button(operation_frame, text="Tính hiệu",
command=self.calculate_difference).grid(row=1, column=0, pady=5)
        self.sub_result = ttk.Label(operation_frame, text="Kết quả: ")
        self.sub_result.grid(row=1, column=1, padx=5, pady=5)

        ttk.Button(operation_frame, text="Tính tích",
command=self.calculate_product).grid(row=2, column=0, pady=5)
        self.mul_result = ttk.Label(operation_frame, text="Kết quả: ")
        self.mul_result.grid(row=2, column=1, padx=5, pady=5)

        ttk.Button(operation_frame, text="Tính thương",
command=self.calculate_quotient).grid(row=3, column=0, pady=5)
        self.div_result = ttk.Label(operation_frame, text="Kết quả: ")
        self.div_result.grid(row=3, column=1, padx=5, pady=5)

    def add_square_tab(self, notebook):
        square_frame = ttk.Frame(notebook)
        notebook.add(square_frame, text="Bình phương")

        ttk.Label(square_frame, text="Số:").grid(row=0, column=0, padx=5,
pady=5)
        self.square_num = ttk.Entry(square_frame)
        self.square_num.grid(row=0, column=1, padx=5, pady=5)
        self.square_num.focus()

```

```

        ttk.Button(square_frame, text="Tính bình phương",
command=self.calculate_square).grid(row=1, column=0, columnspan=2,
pady=10)

        self.square_result = ttk.Label(square_frame, text="Kết quả: ")
        self.square_result.grid(row=2, column=0, columnspan=2)

    def add_power_tab(self, notebook):
        power_frame = ttk.Frame(notebook)
        notebook.add(power_frame, text="Lũy thừa")

        ttk.Label(power_frame, text="Cơ sở:").grid(row=0, column=0,
padx=5, pady=5)
        self.base_num = ttk.Entry(power_frame)
        self.base_num.grid(row=0, column=1, padx=5, pady=5)
        self.base_num.focus()

        ttk.Label(power_frame, text="Mũ:").grid(row=1, column=0, padx=5,
pady=5)
        self.exponent_num = ttk.Entry(power_frame)
        self.exponent_num.grid(row=1, column=1, padx=5, pady=5)

        ttk.Button(power_frame, text="Tính lũy thừa",
command=self.calculate_power).grid(row=2, column=0, columnspan=2, pady=10)

        self.power_result = ttk.Label(power_frame, text="Kết quả: ")
        self.power_result.grid(row=3, column=0, columnspan=2)

    def calculate_sum(self):
        try:
            num1 = float(self.add_num1.get())
            num2 = float(self.add_num2.get())
            result = num1 + num2
            self.add_result.config(text=f"Kết quả: {result}")
            self.add_num1.focus()
        except ValueError:
            messagebox.showerror("Lỗi", "Vui lòng nhập số hợp lệ")

    def calculate_difference(self):
        try:
            num1 = float(self.add_num1.get())
            num2 = float(self.add_num2.get())
            result = num1 - num2

```

```

        self.sub_result.config(text=f"Kết quả: {result}")
        self.add_num1.focus()
    except ValueError:
        messagebox.showerror("Lỗi", "Vui lòng nhập số hợp lệ")

def calculate_product(self):
    try:
        num1 = float(self.add_num1.get())
        num2 = float(self.add_num2.get())
        result = num1 * num2
        self.mul_result.config(text=f"Kết quả: {result}")
        self.add_num1.focus()
    except ValueError:
        messagebox.showerror("Lỗi", "Vui lòng nhập số hợp lệ")

def calculate_quotient(self):
    try:
        num1 = float(self.add_num1.get())
        num2 = float(self.add_num2.get())
        if num2 == 0:
            raise ZeroDivisionError
        result = num1 / num2
        self.div_result.config(text=f"Kết quả: {result}")
        self.add_num1.focus()
    except ValueError:
        messagebox.showerror("Lỗi", "Vui lòng nhập số hợp lệ")
    except ZeroDivisionError:
        messagebox.showerror("Lỗi", "Không thể chia cho 0")

def calculate_square(self):
    try:
        num = float(self.square_num.get())
        result = num ** 2
        self.square_result.config(text=f"Kết quả: {result}")
        self.square_num.focus()
    except ValueError:
        messagebox.showerror("Lỗi", "Vui lòng nhập số hợp lệ")

def calculate_power(self):
    try:
        base = float(self.base_num.get())
        exponent = float(self.exponent_num.get())
        result = base ** exponent

```

```

        self.power_result.config(text=f"Kết quả: {result}")
        self.base_num.focus()
    except ValueError:
        messagebox.showerror("Lỗi", "Vui lòng nhập số hợp lệ")

def clear_entries(self):
    self.add_num1.delete(0, tk.END)
    self.add_num2.delete(0, tk.END)
    self.square_num.delete(0, tk.END)
    self.base_num.delete(0, tk.END)
    self.exponent_num.delete(0, tk.END)

    self.add_result.config(text="Kết quả: ")
    self.sub_result.config(text="Kết quả: ")
    self.mul_result.config(text="Kết quả: ")
    self.div_result.config(text="Kết quả: ")
    self.square_result.config(text="Kết quả: ")
    self.power_result.config(text="Kết quả: ")

    self.add_num1.focus()

def show_about(self):
    messagebox.showinfo("Về ứng dụng", "Ứng dụng Tính toán\nPhiên bản
1.0\n© 2024 LeeHyeon")

if __name__ == "__main__":
    app = CalculatorApp()
    app.mainloop()

```

4. Github

[CalculatorApp](#)