**ỨNG DỤNG TÍNH TOÁN**

Họ tên: Lê Thúy Hiền

MSSV: 2274802010236

1. Giao diện

|  |  |
| --- | --- |
|  |  |
|  |  |

1. 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).

1. 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()

1. Github