

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ - BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÀI TẬP KẾT THÚC MÔN HỌC
MÔN: LẬP TRÌNH PYTHON

NGÀNH : KỸ THUẬT MÁY TÍNH

HỆ : ĐẠI HỌC CHÍNH QUY

THÁI NGUYỄN – 2025

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ - BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÀI TẬP KẾT THÚC MÔN HỌC
MÔN: LẬP TRÌNH PYTHON

GIÁO VIÊN HƯỚNG DẪN	:	TS.Nguyễn Văn Huy
HỌ TÊN SINH VIÊN	:	Lý Văn Cường
MSSV	:	K225480106100
LỚP	:	K58KTP

BÀI TẬP KẾT THÚC MÔN HỌC
MÔN HỌC: LẬP TRÌNH PYTHON
BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Sinh viên: Lý Văn Cường

MSSV: K225480106100

Lớp: K58KTP

Ngành: Kỹ Thuật Máy Tính

Giáo viên hướng dẫn: TS. Nguyễn Văn Huy

Ngày giao nhiệm vụ: 19/05/2025

Ngày hoàn thành nhiệm vụ: 07/06/2025

Tên đề tài: *Máy tính đơn giản (Simple Calculator GUI)*

Yêu cầu: Viết chương trình máy tính có giao diện GUI cho phép người dùng nhập hai số và chọn phép toán $+$, $-$, \times , \div để tính toán.

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

PHIẾU GHI ĐIỂM

Sinh viên: Lý Văn Cường

Lớp: K58KTP

GVHD: TS. Nguyễn Văn Huy

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

Xếp loại : Điểm :

Thái Nguyên, ngày tháng năm 2025.

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan rằng bài tập kết thúc môn Lập trình Python do em tự thực hiện và là kết quả của riêng em dưới sự hướng dẫn của thầy Nguyễn Văn Huy. Các nội dung, số liệu và kết quả trình bày trong bài là trung thực, không sao chép từ bất kỳ nguồn nào một cách trái phép và không được trích dẫn rõ ràng.

Em cam kết mọi thông tin, số liệu và kết quả trong báo cáo là chính xác và trung thực. Nếu có bất kỳ sai phạm nào trong quá trình thực hiện, em xin hoàn toàn chịu trách nhiệm trước nhà trường và giảng viên phụ trách.

Em xin chân thành cảm ơn!

Họ và tên

Lý Văn Cường

MỤC LỤC

LỜI CAM ĐOAN	4
DANH MỤC BẢNG VẼ VÀ ĐỒ THỊ	6
LỜI MỞ ĐẦU	7
CHƯƠNG I: GIỚI THIỆU ĐẦU BÀI	8
1.1. Tên đề tài	8
1.2. Mô tả bài toán	8
1.3. Mục tiêu của chương trình	8
1.4. Thách thức và giải pháp	8
CHƯƠNG II: CƠ SỞ LÝ THUYẾT	9
2.1. Lập trình Python	9
2.2. Thư viện Tkinter	9
2.3. Biến liên kết và xử lý sự kiện	10
2.4. Bắt lỗi bằng try – except	10
CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH	12
3.1. Sơ đồ khối hệ thống	12
3.1.1. Các module chính của chương trình	12
3.2. Sơ đồ khối các thuật toán chính	13
3.2.1. Thuật toán hàm calculate()	13
3.2.2. Thuật toán hàm reset	14
3.3. Cấu trúc dữ liệu	14
3.4. Chương trình	15
CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN	16
4.1. Thực nghiệm	16
4.1.1. Ghi lại kết quả các tính năng	16
4.1.2. Mô tả các tính năng	18
4.2. Kết luận	20
4.2.1. Những điều học được trong quá trình thực hiện	20
4.2.2. Hướng cải tiến trong tương lai	20
TÀI LIỆU THAM KHẢO	21

DANH MỤC BẢNG VẼ VÀ ĐỒ THỊ

Hình 3.2.1: Sơ đồ khối thuật toán hàm calculate()

Hình 3.1.2: Biểu đồ phân cấp chức năng

Hình 3.2.2: Sơ đồ khối thuật toán hàm reset()

Hình 4.1.1: Kết quả nhập đúng

Hình 4.1.2: Kết quả nhập sai định dạng số

Hình 4.1.3: Kết quả nhập chia cho 0

Hình 4.1.4: code giao diện đồ họa GUI

Hình 4.1.5: code nhập dữ liệu cho các số

Hình 4.1.6: code chọn phép tính và tính toán

LỜI MỞ ĐẦU

Ngày nay, việc xây dựng các ứng dụng có giao diện đồ họa (GUI) ngày càng trở nên phổ biến, giúp người dùng tương tác dễ dàng và trực quan hơn với phần mềm. Python là một trong những ngôn ngữ lập trình linh hoạt, hỗ trợ mạnh mẽ việc phát triển các ứng dụng GUI thông qua nhiều thư viện như Tkinter, PyQt, hay Kivy.

Trong khuôn khổ môn học Lập trình Python, em đã chọn đề tài “Xây dựng máy tính đơn giản bằng giao diện đồ họa (GUI)” nhằm áp dụng các kiến thức đã học vào một ứng dụng thực tiễn. Đề tài này không chỉ giúp em rèn luyện kỹ năng lập trình, tư duy logic mà còn làm quen với cách thiết kế giao diện người dùng cơ bản trong Python. Ứng dụng máy tính được xây dựng cho phép thực hiện các phép toán cơ bản như cộng, trừ, nhân, chia với giao diện thân thiện và dễ sử dụng.

Bài báo cáo này trình bày toàn bộ quá trình xây dựng ứng dụng, từ phân tích yêu cầu, thiết kế giao diện đến cài đặt và kiểm thử chương trình. Trong quá trình thực hiện, em đã học hỏi và tích lũy được nhiều kinh nghiệm quý báu.

Em xin chân thành cảm ơn thầy Nguyễn Văn Huy đã tận tình hướng dẫn, tạo điều kiện để em hoàn thành đề tài này. Em rất mong nhận được sự góp ý của thầy để có thể hoàn thiện hơn trong các dự án sau.

CHƯƠNG I: GIỚI THIỆU ĐẦU BÀI

1.1. Tên đề tài

- Máy tính đơn giản (Simple Calculator GUI)

1.2. Mô tả bài toán

- Bài toán yêu cầu xây dựng một chương trình máy tính đơn giản có giao diện người dùng (GUI). Người dùng có thể nhập hai số thực, chọn phép toán (+, -, ×, ÷), nhấn nút để tính kết quả và xem kết quả hiển thị trực tiếp. Chương trình cần xử lý các lỗi như nhập sai định dạng hoặc chia cho 0.

1.3. Mục tiêu của chương trình

- Tạo giao diện người dùng thân thiện cho phép nhập dữ liệu và hiển thị kết quả.
- Hỗ trợ bốn phép toán cơ bản: cộng, trừ, nhân, chia.
- Xử lý và thông báo lỗi khi người dùng nhập không hợp lệ.
- Hỗ trợ chức năng “Reset” để xóa dữ liệu đã nhập.

1.4. Thách thức và giải pháp

- Việc xử lý nhập liệu trong giao diện đồ họa thường gặp các lỗi như nhập sai định dạng (nhập chữ thay vì số), chia cho 0, hoặc người dùng quên chọn phép toán. Do đó, một trong những thách thức chính là đảm bảo chương trình không bị dừng khi xảy ra lỗi mà vẫn hướng dẫn người dùng nhập lại đúng cách.

- Hướng tiếp cận được sử dụng là: kiểm tra đầu vào bằng cách chuyển đổi dữ liệu sang float, và nếu có lỗi thì bắt ngoại lệ bằng *try-except*, sau đó hiển thị thông báo lỗi bằng hộp thoại *messagebox*.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

2.1. Lập trình Python

- Python là một ngôn ngữ lập trình bậc cao, có cú pháp đơn giản, dễ học và rất phù hợp với người mới bắt đầu. Python hỗ trợ lập trình hướng đối tượng, xử lý ngoại lệ, và đặc biệt có sẵn nhiều thư viện mạnh mẽ để xây dựng các ứng dụng thực tế, bao gồm cả ứng dụng có giao diện người dùng (GUI).

Các kiến thức Python được vận dụng trong chương trình bao gồm:

- Biến và kiểu dữ liệu: float, str, bool
- Hàm: sử dụng để chia nhỏ chương trình thành các phần xử lý riêng biệt (ví dụ: calculate(), reset())
- Cấu trúc điều kiện: if, elif, else
- Xử lý ngoại lệ: try – except

2.2. Thư viện Tkinter

- Tkinter là thư viện tiêu chuẩn của Python dùng để xây dựng giao diện đồ họa (GUI). Đây là công cụ giúp người lập trình tạo ra các cửa sổ tương tác, nhập liệu, nút bấm và các thành phần đồ họa khác.

- Một số widget chính của tkinter được sử dụng:

- Entry: ô nhập liệu cho phép người dùng nhập số.
- Label: dùng để hiển thị kết quả hoặc thông báo.
- Button: nút dùng để kích hoạt hành động như “Tính” hoặc “Reset”.
- Radiobutton: lựa chọn một trong nhiều phép toán.
- messagebox: hiển thị thông báo lỗi cho người dùng.

- Tkinter cho phép bố trí các thành phần theo layout grid(), giúp dễ dàng tổ chức giao diện theo hàng – cột.

2.3. Biến liên kết và xử lý sự kiện

- Trong tkinter, để lấy hoặc lưu trữ dữ liệu từ các thành phần giao diện (Entry, Radiobutton, v.v...), người ta sử dụng các biến liên kết như:

- StringVar() – dùng cho chuỗi (dữ liệu từ Entry).
- IntVar() – dùng cho số nguyên (dữ liệu từ Radiobutton).

- Các biến này giúp liên kết dữ liệu hai chiều giữa giao diện và chương trình. Khi người dùng nhập số hoặc chọn phép toán, biến liên kết sẽ được cập nhật tự động và có thể truy xuất trong mã nguồn.

- Xử lý sự kiện trong tkinter là việc gán hành động (function) cho các nút. Khi người dùng nhấn nút “Tính”, hàm calculate() sẽ được gọi để xử lý phép toán.

- Hàm calculate() thực hiện các bước:

- Đọc giá trị từ Entry.
- Chuyển kiểu dữ liệu sang float.
- Thực hiện phép toán.
- Hiển thị kết quả hoặc báo lỗi.

2.4. Bắt lỗi bằng try – except

- Để đảm bảo chương trình không bị dừng khi người dùng nhập sai (ví dụ nhập chữ hoặc chia cho 0), cần sử dụng khối try – except để bắt và xử lý ngoại lệ.

- Hai lỗi phổ biến cần xử lý:

- `ValueError`: khi người dùng nhập ký tự không phải số (ví dụ: “a”, “abc”).
- `ZeroDivisionError`: khi thực hiện phép chia mà số chia là 0.

- Việc xử lý lỗi bằng hộp thoại giúp người dùng dễ hiểu và sử dụng chương trình thuận tiện hơn mà không làm gián đoạn ứng dụng

CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. Sơ đồ khối hệ thống

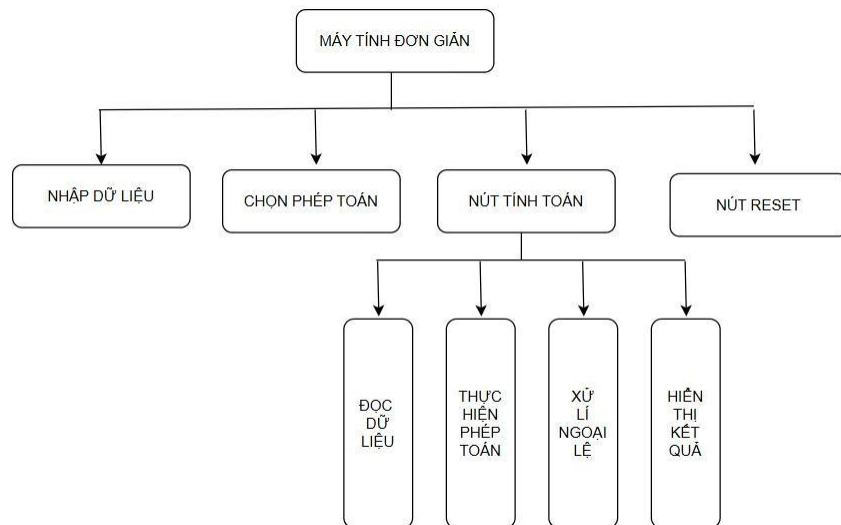
3.1.1. Các module chính của chương trình

- Chương trình bao gồm 3 phần chính:

- Giao diện người dùng (GUI): thiết kế với thư viện tkinter, bao gồm các thành phần Label, Entry, Button, Radiobutton, Frame.
- Xử lý phép toán: thực hiện phép tính theo lựa chọn của người dùng (+, -, ×, ÷).
- Kiểm tra và bắt lỗi: sử dụng try – except để đảm bảo chương trình không bị lỗi khi nhập sai hoặc chia cho 0.

3.1.2. Biểu đồ phân cấp chức năng

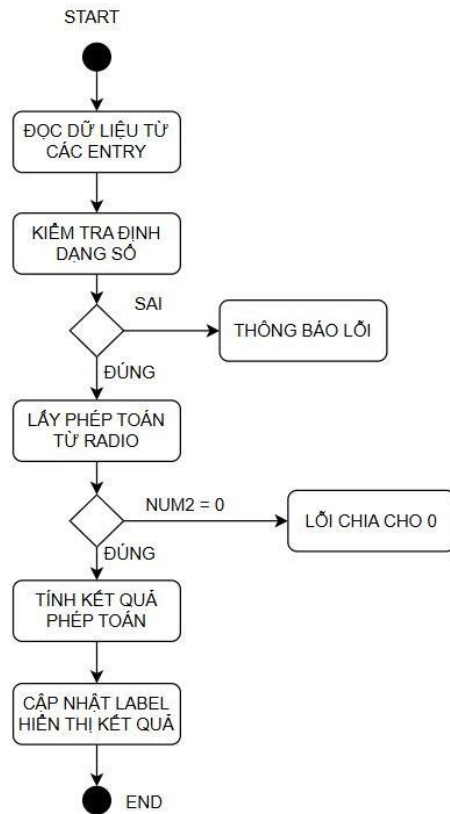
- Dựa trên phân tích yêu cầu của bài toán, ta có biểu đồ phân cấp chức năng sau:



Hình 3.1.2: Biểu đồ phân cấp chức năng

3.2. Sơ đồ khối các thuật toán chính

3.2.1. Thuật toán hàm calculate()



Hình 3.2.1: Sơ đồ khối thuật toán hàm `calculate()`

3.2.2. Thuật toán hàm reset



Hình 3.2.2: Sơ đồ khối thuật toán hàm reset()

3.3. Cấu trúc dữ liệu

- Trong chương trình "Máy tính đơn giản", tuy không sử dụng các cấu trúc dữ liệu phức tạp như danh sách hay từ điển, nhưng có áp dụng một số kiểu dữ liệu cơ bản và biến liên kết để xử lý thông tin từ giao diện.
- Dữ liệu đầu vào từ người dùng được nhập qua hai ô Entry, ban đầu ở dạng chuỗi (string), sau đó được chuyển thành số thực (float) để tính toán. Biến liên kết StringVar được sử dụng để kết nối giữa nhóm nút chọn phép toán (các Radiobutton) và chương trình, giúp kiểm soát và lấy giá trị lựa chọn một cách dễ dàng.
- Kết quả tính toán được lưu vào một biến và hiển thị lên giao diện qua một nhãn (Label). Trong quá trình xử lý, chương trình còn sử dụng thêm các biến tạm (ví dụ như num1, num2) để lưu trữ giá trị nhập vào sau khi chuyển đổi kiểu dữ liệu.

- Việc sử dụng biến liên kết và cấu trúc dữ liệu đơn giản như StringVar và float cho phép chương trình vừa dễ viết, vừa dễ mở rộng về sau.

3.4. Chương trình

- Chương trình được tổ chức thành hai hàm chính: calculate() và reset(), mỗi hàm đảm nhiệm một vai trò riêng biệt, giúp tách biệt logic xử lý và tương tác giao diện.
- Hàm calculate() là trung tâm xử lý của chương trình. Khi người dùng nhấn nút “Tính”, hàm này sẽ được gọi để đọc dữ liệu từ các ô nhập, kiểm tra định dạng đầu vào, lấy phép toán đã chọn và thực hiện phép tính tương ứng. Nếu người dùng nhập sai kiểu dữ liệu (ví dụ nhập chữ thay vì số), hoặc thực hiện chia cho 0, hàm sẽ bắt các lỗi này và hiển thị thông báo bằng hộp thoại, giúp đảm bảo chương trình không bị gián đoạn và thân thiện hơn với người dùng. Kết quả sau khi tính toán sẽ được định dạng và hiển thị trực tiếp trên giao diện.
- Hàm reset() dùng để xóa toàn bộ dữ liệu đã nhập, đặt lại phép toán về mặc định là cộng (“+”) và xóa kết quả cũ đang hiển thị. Hàm này giúp người dùng thao tác lại nhanh chóng mà không cần đóng mở lại chương trình. Mặc dù chức năng đơn giản, nhưng đây là một phần quan trọng giúp tăng trải nghiệm người dùng.
- Cấu trúc tổ chức mã chương trình rõ ràng, dễ hiểu, chia nhỏ thành từng chức năng độc lập, giúp thuận tiện trong việc kiểm thử, sửa lỗi hoặc mở rộng thêm tính năng trong tương lai.

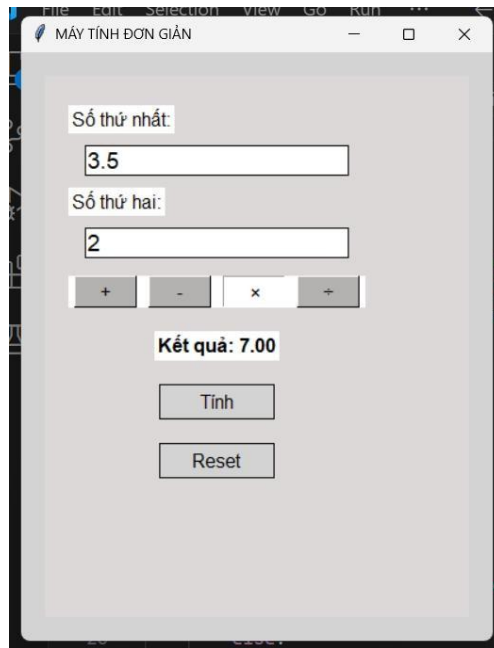
CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN

4.1. Thực nghiệm

4.1.1. Ghi lại kết quả các tính năng

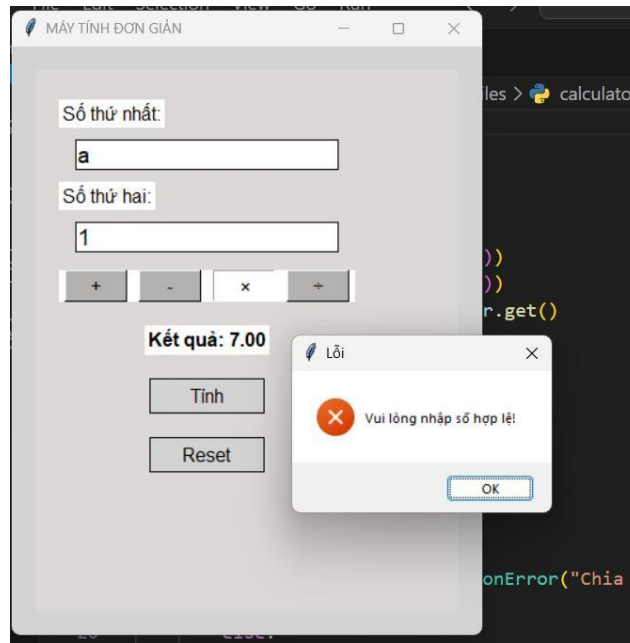
- Sau khi hoàn thiện chương trình, tiến hành chạy thử với các bộ dữ liệu đầu vào khác nhau để kiểm tra hoạt động của từng chức năng, bao gồm tính toán, xử lý lỗi và khả năng làm mới (reset).

a. Nhập đúng



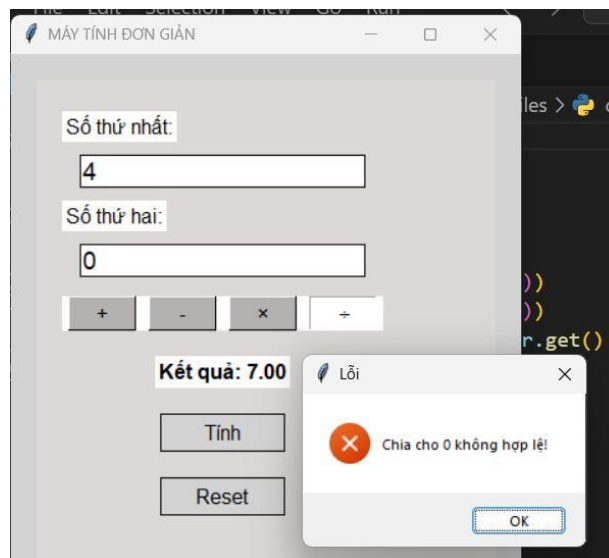
Hình 4.1.1: Kết quả nhập đúng

b. Nhập sai



Hình 4.1.2: Kết quả nhập sai định dạng số

c. Nhập chia cho 0



Hình 4.1.3: Kết quả nhập chia cho 0

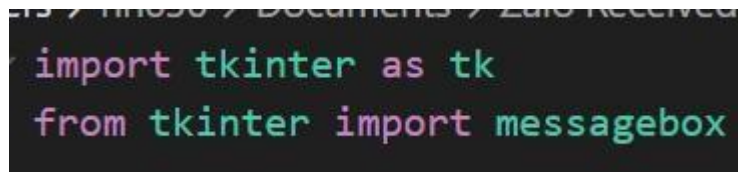
Đánh giá:

Chương trình chạy ổn định, phản hồi nhanh, xử lý đầy đủ các trường hợp thông dụng và lỗi thường gặp. Các tính năng cơ bản đều hoạt động chính xác, giao diện đơn giản, dễ sử dụng. Việc sử dụng hộp thoại khi có lỗi giúp tăng tính thân thiện và chuyên nghiệp cho sản phẩm.

4.1.2. Mô tả các tính năng

a. Giao diện đồ họa GUI

- Giao diện sử dụng tkinter, với thiết kế đơn giản, gọn gàng.
- Các thành phần bố trí bằng Grid layout giúp căn chỉnh chính xác theo hàng – cột.
- Giao diện có màu nền và khung nhẹ, dễ nhìn.

A screenshot of a code editor showing two lines of Python code. The first line is 'import tkinter as tk' and the second line is 'from tkinter import messagebox'. The code is written in a light blue font on a dark background. Above the code, a portion of a file path is visible: '... / Desktop / Documents / Lab Received'.

Hình 4.1.4: code giao diện đồ họa GUI

b. Nhập dữ liệu(entry)

- Hai ô nhập (Entry) cho phép người dùng nhập vào hai số thực cần tính toán.
- Dữ liệu được lấy thông qua lệnh: entry1.get() và entry2.get().

```
num1 = float(entry1.get())
num2 = float(entry2.get())
operation = operation_var.get()
```

Hình 4.1.5: code nhập dữ liệu cho các số

c. Chọn phép toán(radius) và tính toán lưu vào biến result

- Bốn nút Radio (RadioButton) tương ứng với bốn phép toán:

- Cộng (+), Trừ (-), Nhân (\times), Chia (\div)
- Giá trị của phép toán được lưu qua biến operation_var (kiểu StringVar).

- Khi nhấn nút “Tính”, chương trình sẽ:

- Chuyển hai ô nhập thành kiểu float.
- Dựa vào lựa chọn phép toán, thực hiện phép tính tương ứng.
- Hiển thị kết quả lên Label result_label với định dạng thập phân 2 chữ số.

```
operation = operation_var.get()

if operation == "+":
    result = num1 + num2
elif operation == "-":
    result = num1 - num2
elif operation == "x":
    result = num1 * num2
elif operation == "÷":
    if num2 == 0:
        raise ZeroDivisionError("Chia cho 0 không hợp lệ!")
    result = num1 / num2
else:
    raise ValueError("Vui lòng chọn phép toán!")

result_label.config(text=f"Kết quả: {result:.2f}")
```

Hình 4.1.6: code chọn phép tính và tính toán

4.2. Kết luận

- Sau quá trình thiết kế và xây dựng, chương trình “Máy tính đơn giản” đã hoàn thành với đầy đủ các chức năng cơ bản như: nhập hai số, chọn phép toán (cộng, trừ, nhân, chia), hiển thị kết quả tính toán, xử lý lỗi nhập sai hoặc chia cho 0, và cho phép xóa/reset toàn bộ dữ liệu nhập.

4.2.1. Những điều học được trong quá trình thực hiện

- Nắm vững cách sử dụng thư viện tkinter để thiết kế giao diện đồ họa với các thành phần như Entry, Radiobutton, Label, Button.
- Biết cách sử dụng StringVar để liên kết giữa giao diện và biến xử lý nội bộ.
- Hiểu rõ cách xử lý lỗi với try – except và sử dụng messagebox để thông báo lỗi thân thiện.
- Thực hành việc tổ chức hàm chức năng theo hướng rõ ràng, tách biệt logic và giao diện.

4.2.2. Hướng cải tiến trong tương lai

- Bổ sung thêm các phép toán nâng cao như lũy thừa, căn bậc hai, phần trăm.
- Thiết kế giao diện thân thiện hơn
- Lưu lại lịch sử tính toán để người dùng có thể xem lại.
- Thêm tính năng phím tắt hoặc bàn phím ảo để thuận tiện thao tác.

TÀI LIỆU THAM KHẢO

- Python Software Foundation. Python 3 Documentation. Truy cập tại: <https://docs.python.org/3/>
- Tkinter Docs - Python Interface to Tcl/Tk. Truy cập tại: <https://tkdocs.com/>
- W3Schools. https://www.w3schools.com/python/python_try_except.asp
- ChatGPT – OpenAI (<https://chat.openai.com>)