

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



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

GIÁO VIÊN HƯỚNG DẪN	: TS. NGUYỄN VĂN HUY
HỌ TÊN SINH VIÊN	: NGUYỄN MẠNH HIẾU
MSSV	: K225480106020
LỚP	: K58KTP

**BÀI TẬP KẾT THÚC MÔN HỌC**  
**MÔN HỌC**  
**LẬP TRÌNH PYTHON**

**Sinh viên:** 1. Nguyễn Mạnh Hiếu

MSSV: K225480106020

**Lớp:** K58KTP

**Khoá:** K58

**Bộ môn:** Công Nghệ Thông Tin

**Giáo viên hướng dẫn:** TS.Nguyễn Văn Huy

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

2. Yêu cầu: Viết chương trình máy tính có giao diện GUI cho phép người dùng nhập hai số và chọn phép toán +, −, ×, ÷ để tính toán.

3. Các sản phẩm, kết quả:

—... Thuyết minh báo cáo (1 quyển)

—... Slide báo cáo

—... Demo sản phẩm

4. Ngày giao nhiệm vụ :

5. Ngày hoàn thành nhiệm vụ :

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

## PHIẾU GHI ĐIỂM

**Sinh viên:** Nguyễn Mạnh Hiếu

**Lớp:** K58KTP

**GVHD:** TS.Nguyễn Văn Huy

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

### 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)*

# MỤC LỤC

MỤC LỤC .....	3
LỜI CAM ĐOAN .....	4
DANH MỤC BẢNG VẼ VÀ ĐỒ THỊ .....	5
LỜI NÓI ĐẦU .....	5
CHƯƠNG 1: GIỚI THIỆU ĐẦU BÀI .....	7
1.1 Tên đề tài .....	7
1.2 Mô tả bài toán .....	7
1.3 Mục tiêu .....	7
1.4 Thách thức và giải pháp .....	7
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT .....	9
2.1 NGON NGỮ LẬP TRÌNH PYTHON .....	9
2.2 Giao diện đồ họa với Tkinter .....	9
2.3. Biến liên kết và sự kiện trong GUI .....	10
2.4. Kỹ thuật bắt lỗi bằng khối try – except .....	10
CHƯƠNG 3 THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH .....	12
3.1 Sơ đồ khối hệ thống .....	12
3.1.1 Các module chính .....	12
3.1.2 Biểu đồ phân cấp chức năng .....	13
3.2 Sơ đồ các thuật toán .....	15
3.2.1 Sơ đồ khối thuật toán tính toán caculate() .....	15
3.2.2 Sơ đồ thuật toán reset() .....	16
3.3 Cấu trúc dữ liệu .....	16
CHƯƠNG 4 KIỂM THỬ VÀ KẾT LUẬN .....	17
4.1 Kiểm thử .....	17
4.1.1 Kiểm thử các tính năng .....	17
4.1.2 Mô tả các tính năng .....	19
4.2 Kết luận .....	21
TÀI LIỆU THAM KHẢO .....	23

## 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 đề tài là chính xác và trung thực. Nếu có bất kỳ sai phạm nào trong quá trình thực hiện, chúng em xin hoàn toàn chịu trách nhiệm trước nhà trường và giảng viên phụ trách.

Em xin chân thành cảm ơn!

Họ và tên

*(Ký và ghi rõ họ tên)*

## **DANH MỤC BẢNG VẼ VÀ ĐỒ THỊ**

Hình 3.1.2: Biểu đồ phân cấp chức năng

Hình 3.2.1: Sơ đồ khối thuật toán tính toán caculate(

Hình 3.2.2: Sơ đồ khối thuật toán reset

Hình 4.1.1: Nhập đúng định dạng số

Hình 4.1.2: Nhập sai định dạng số

Hình 4.1.3: Nhập chia cho 0

Hình 4.2.1.a: Thư viện tkinter

Hình 4.2.2.b: Code nhập dữ liệu cho 2 số

Hình 4.2.3.c: Code chọn phép toán và tính toán

## **LỜI NÓI ĐẦU**

Trong thời đại công nghệ phát triển không ngừng, việc xây dựng các ứng

dụng có giao diện người dùng (GUI) ngày càng trở nên quan trọng và phổ biến. Python, với sự hỗ trợ mạnh mẽ từ các thư viện như Tkinter, PyQt, hay Kivy đã trở thành một trong những ngôn ngữ lập trình phổ biến giúp người học dễ dàng tiếp cận và xây dựng các ứng dụng giao diện đồ họa cơ bản.

Bài tập kết thúc môn học Python này được thực hiện với mục tiêu vận dụng những kiến thức đã học vào một đề tài thực tiễn. Đề tài “*máy tính đơn giản (Simple Calculator GUI)*” cho phép người dùng nhập hai số thực, chọn phép toán (+, −, ×, ÷), thực hiện phép tính và hiển thị kết quả trực tiếp trên giao diện. Ngoài ra, chương trình còn xử lý các trường hợp ngoại lệ như nhập sai định dạng số hoặc chia cho 0, đảm bảo tính ổn định và thân thiện với người dùng.

Thông qua đề tài này, người học không chỉ ôn luyện và củng cố kiến thức về ngôn ngữ Python, mà còn rèn luyện kỹ năng lập trình, tư duy logic trong quá trình xây dựng phần mềm.

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 1: 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

- Tạo giao diện thân thiện với người dùng sử dụng thư viện Tkinter của Python.
- Cho phép nhập hai số và chọn phép toán cần thực hiện.
- Hiển thị kết quả tính toán sau khi nhấn nút “Tính”.
- Xử lý các lỗi nhập sai dữ liệu (ví dụ: không phải số) hoặc lỗi chia cho 0.
- Cung cấp nút “Reset” để xóa dữ liệu đã nhập.
- Hiển thị thông báo lỗi bằng hộp thoại thông báo (messagebox) khi có lỗi xảy ra.

## 1.4 Thách thức và giải pháp

*Thách thức:*

**Xử lý lỗi nhập liệu:** Người dùng có thể nhập dữ liệu không hợp lệ như ký tự, chuỗi, hoặc để trống ô nhập, dẫn đến lỗi trong quá trình xử lý phép toán.

**Chia cho 0:** Đây là lỗi phổ biến trong toán học và nếu không được xử lý đúng cách sẽ khiến chương trình dừng đột ngột hoặc trả về kết quả sai.

**Giao diện thân thiện:** Việc bố trí các thành phần giao diện (Entry, RadioButton, Button, Label) sao cho hợp lý, dễ sử dụng và dễ nhìn là một thử thách đối với người mới tiếp cận lập trình GUI.

**Hiển thị thông báo lỗi:** Việc tạo ra các thông báo lỗi rõ ràng, dễ hiểu và hiển thị bằng hộp thoại (messagebox) yêu cầu phải nắm vững cách xử lý ngoại lệ và tương tác với người dùng trong Tkinter.

*Giải pháp:*

**Kiểm tra và ép kiểu dữ liệu đầu vào:** Sử dụng khối lệnh try-except để bắt các ngoại lệ như ValueError (khi nhập không phải số) và ZeroDivisionError (khi chia cho 0), từ đó hiển thị thông báo phù hợp bằng hộp thoại messagebox.

**Thiết kế giao diện theo lưới (Grid Layout):** Sử dụng phương pháp bố cục lưới (grid()) của Tkinter để sắp xếp các thành phần giao diện một cách khoa học và dễ sử dụng.

**Tách hàm xử lý logic:** Tách riêng phần tính toán thành một hàm calculate() để dễ quản lý, bảo trì và gắn kết với sự kiện khi người dùng nhấn nút "Tính".

**Thêm chức năng “Reset”:** Cung cấp nút xóa nội dung nhập giúp người dùng dễ dàng làm mới dữ liệu và tránh nhầm lẫn khi thực hiện phép toán mới.

# CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

## 2.1 NGON NGỮ LẬP TRÌNH PYTHON

Python là một trong những ngôn ngữ lập trình hiện đại, dễ tiếp cận và được sử dụng rộng rãi trong nhiều lĩnh vực như phát triển phần mềm, trí tuệ nhân tạo, và tự động hóa. Đặc điểm nổi bật của Python là cú pháp ngắn gọn, rõ ràng, rất phù hợp với người mới học lập trình.

Trong ứng dụng này, các thành phần Python cơ bản sau đã được sử dụng:

- **Biến và kiểu dữ liệu:** bao gồm float (số thực), str (chuỗi ký tự) và bool (kiểu logic).
- **Hàm (function):** Giúp tổ chức mã nguồn thành các khối chức năng rõ ràng. Các hàm như calculate() dùng để tính toán, reset() để xóa dữ liệu.
- **Câu lệnh điều kiện:** if, elif, và else dùng để xử lý lựa chọn phép toán.
- **Khối xử lý lỗi:** Sử dụng try – except để đảm bảo chương trình không gặp lỗi khi người dùng nhập dữ liệu không hợp lệ.

## 2.2 Giao diện đồ họa với Tkinter

Tkinter là bộ công cụ GUI có sẵn trong Python, cho phép lập trình viên xây dựng giao diện tương tác mà không cần thư viện ngoài. Đây là công cụ chính được dùng để thiết kế phần giao diện trong chương trình.

Một số **widget** (thành phần giao diện) quan trọng đã được áp dụng:

**Entry:** Dùng để nhập hai số thực từ người dùng.

**Label:** Hiển thị thông tin như kết quả hoặc mô tả thành phần.

**Button:** Kích hoạt thao tác như thực hiện phép tính hoặc đặt lại dữ liệu.

**Radiobutton:** Cho phép lựa chọn một trong các phép toán (+, −, ×, ÷).

**MessageBox:** Hộp thoại hiện thông báo lỗi hoặc cảnh báo.

Tkinter hỗ trợ việc bố trí các thành phần thông qua hệ thống lưới (grid layout), giúp việc sắp xếp và quản lý giao diện dễ dàng hơn theo dạng hàng – cột.

### 2.3. Biến liên kết và sự kiện trong GUI

Trong một chương trình giao diện, việc trao đổi dữ liệu giữa người dùng và mã xử lý bên trong rất quan trọng. Tkinter sử dụng các biến liên kết để phục vụ việc này:

- **StringVar()**: Được dùng để lưu dữ liệu chuỗi, ví dụ như nội dung nhập vào từ Entry.
- **IntVar()** / **DoubleVar()**: Dành cho số nguyên hoặc số thực, thích hợp cho các lựa chọn như phép toán.

Các biến này tự động cập nhật khi người dùng tương tác, và có thể được truy xuất trực tiếp trong các hàm xử lý.

Xử lý sự kiện (event handling) trong ứng dụng được thực hiện thông qua việc gán chức năng cho các nút. Khi người dùng nhấn nút “Tính”, hàm calculate() được gọi để xử lý toàn bộ phép toán.

Trong hàm calculate() sẽ thực hiện:

- Đọc dữ liệu từ các ô nhập.
- Ép kiểu dữ liệu sang float.
- Dựa vào phép toán được chọn để tính toán.
- Xuất kết quả ra Label hoặc hiển thị hộp thoại khi gặp lỗi.

### 2.4. Kỹ thuật bắt lỗi bằng khối try – except

Một ứng dụng GUI thân thiện không chỉ cần tính năng đầy đủ mà còn phải xử lý tốt các lỗi phát sinh từ phía người dùng. Trong chương trình này, việc bắt lỗi được triển khai bằng khối **try – except**, giúp phát hiện và phản hồi khi dữ liệu không đúng định dạng hoặc khi xảy ra chia cho 0.

Hai lỗi phổ biến đã được xử lý:

**ValueError:** Phát sinh khi người dùng nhập ký tự không phải là số (ví dụ: “abc”, “a”,...).

**ZeroDivisionError:** Xảy ra khi phép chia có mẫu số bằng 0, điều này cần ngăn chặn để tránh lỗi chạy.

Khi phát hiện lỗi, chương trình không bị dừng mà thay vào đó sẽ hiển thị hộp thoại thông báo để hướng dẫn người dùng sửa lỗi.

# CHƯƠNG 3 THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

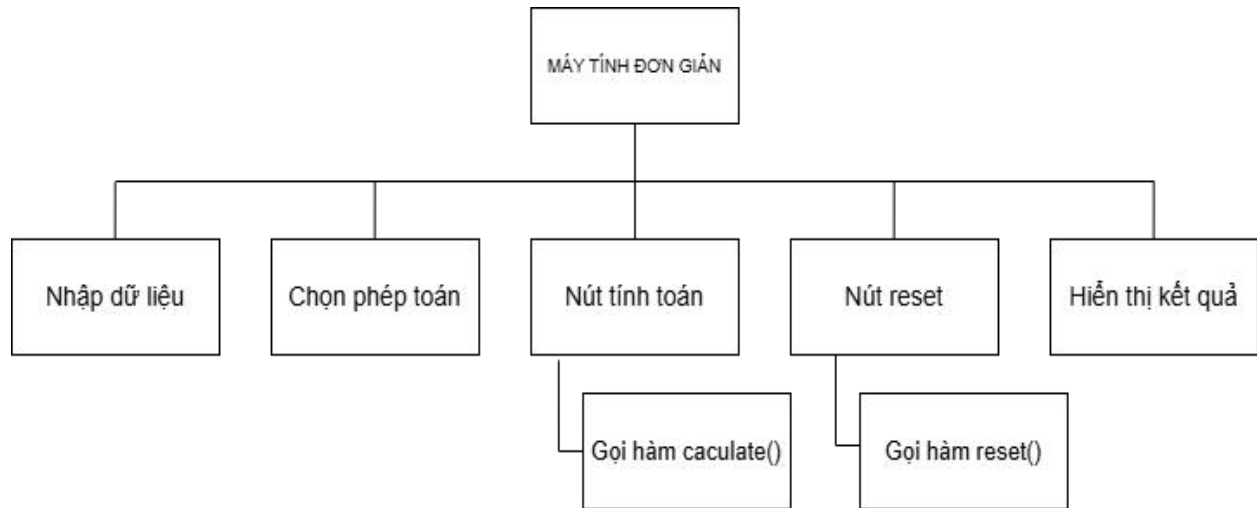
## 3.1 Sơ đồ khối hệ thống

### 3.1.1 Các module chính

Chương trình được chia thành các module chức năng chính, đảm nhiệm từng phần việc cụ thể để đảm bảo tính dễ hiểu, dễ quản lý và bảo trì. Dưới đây là mô tả các thành phần chính:

- **Giao diện người dùng (GUI):** Xây dựng bằng Tkinter, gồm các thành phần như Entry, Button, Radiobutton, Label.
- **Xử lý dữ liệu đầu vào:** Kiểm tra và chuyển đổi kiểu dữ liệu từ chuỗi sang số thực.
- **Thực hiện phép toán:** Dựa vào lựa chọn của người dùng để thực hiện  $+$ ,  $-$ ,  $\times$ , hoặc  $\div$ .
- **Xử lý lỗi:** Bắt các lỗi như nhập sai định dạng, chia cho 0.
- **Hiển thị kết quả và thông báo:** Cập nhật kết quả vào Label hoặc hiển thị hộp thoại lỗi.

### 3.1.2 Biểu đồ phân cấp chức năng



Hình 3.1.2: Biểu đồ phân cấp chức năng

#### Mô tả:

##### 1. Nhập dữ liệu (Entry)

Đây sẽ là nơi người dùng nhập các số để thực hiện phép toán. Trong Tkinter, chúng ta sử dụng widget Entry để tạo các ô nhập liệu. Bạn sẽ cần hai ô nhập liệu cho hai số.

##### 2. Chọn phép toán (Radiobutton)

Để người dùng chọn giữa các phép toán (cộng, trừ, nhân, chia), Radiobutton là lựa chọn phù hợp. Mỗi radiobutton sẽ tương ứng với một phép toán và người dùng chỉ có thể chọn một tại một thời điểm. Chúng ta sẽ liên kết các radiobutton này với một biến điều khiển chung để biết phép toán nào đã được chọn.

##### 3. Nút tính toán (Button)

Nút "Tính toán" sẽ kích hoạt logic chính của chương trình. Khi nhấn nút này, một hàm (ví dụ: calculate()) sẽ được gọi.

#### Hàm calculate()

- Đọc dữ liệu: Nó sẽ đọc các giá trị từ hai ô nhập liệu (Entry).
- Kiểm tra lỗi: Đây là bước quan trọng để đảm bảo tính ổn định của chương trình.

Cần kiểm tra:

- Dữ liệu rỗng: Người dùng có để trống ô nhập liệu nào không?
- Dữ liệu không hợp lệ: Người dùng có nhập chữ cái hoặc ký tự không phải số không?
- Chia cho 0: Nếu phép toán là chia, bạn phải kiểm tra xem số chia có bằng 0 hay không để tránh lỗi.
- Thực hiện phép toán: Sau khi xác nhận dữ liệu hợp lệ, chương trình sẽ thực hiện phép toán đã chọn (cộng, trừ, nhân, chia) dựa trên lựa chọn của radiobutton.

#### 4. Nút reset (Button)

Nút "Reset" sẽ đưa giao diện về trạng thái ban đầu, xóa các ô nhập liệu và kết quả hiển thị. Khi nhấn nút này, một hàm (ví dụ: `reset()`) sẽ được gọi.

Hàm `reset()` sẽ:

- Xóa nội dung của các ô nhập liệu (Entry).
- Xóa lựa chọn radiobutton.
- Xóa hoặc đặt lại thông báo kết quả.

#### 5. Hiển thị kết quả (Label + MessageBox)

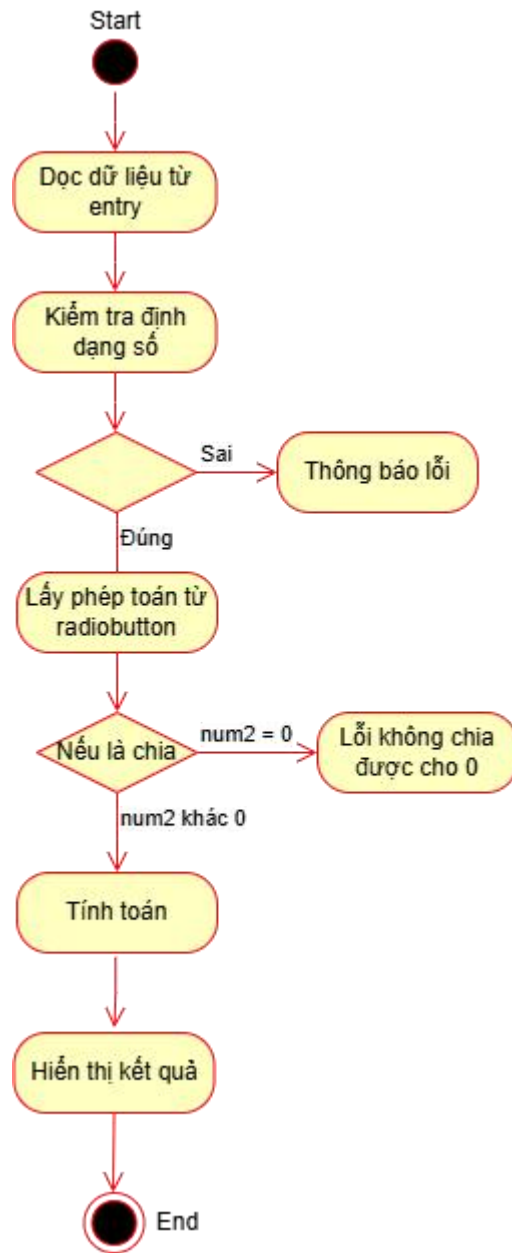
Kết quả của phép toán có thể được hiển thị theo hai cách:

- Label: Một widget Label có thể được sử dụng để hiển thị kết quả trực tiếp trên cửa sổ ứng dụng. Đây là cách phổ biến và đơn giản nhất cho kết quả thông thường.
- MessageBox: Đối với các thông báo lỗi hoặc cảnh báo (ví dụ: "Vui lòng nhập số hợp lệ", "Không thể chia cho 0"), MessageBox của Tkinter là rất hiệu quả. Nó sẽ hiển thị một cửa sổ pop-up nhỏ với thông báo.



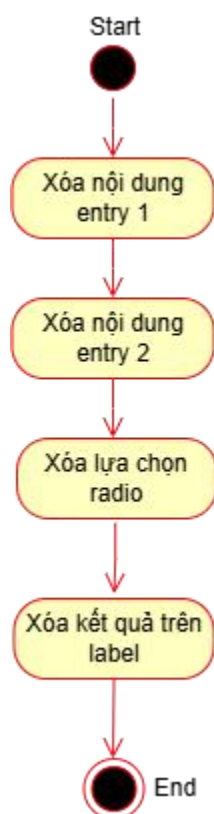
## 3.2 Sơ đồ các thuật toán

### 3.2.1 Sơ đồ khối thuật toán tính toán caculate()



Hình 3.2.1: Sơ đồ khối thuật toán tính toán caculate()

### 3.2.2 Sơ đồ thuật toán reset()



Hình 3.2.2: Sơ đồ khối thuật toán reset()

### 3.3 Cấu trúc dữ liệu

Chương trình không sử dụng cơ sở dữ liệu, mà sử dụng các biến đơn giản và biến liên kết của Tkinter để lưu trữ tạm thời dữ liệu người dùng nhập và kết quả xử lý.

*Biến sử dụng:*

Tên biến	Kiểu	Mô tả
entry1	Entry	Nhập số thứ nhất
entry2	Entry	Nhập số thứ hai
operation	StringVar	Biến lưu phép toán đã chọn (1: +, 2: −, 3: ×, 4: ÷)
result_label	Label	Hiển thị kết quả tính toán

## CHƯƠNG 4 KIỂM THỬ VÀ KẾT LUẬN


### 4.1 Kiểm thử

#### 4.1.1 Kiểm thử 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).

#### Các tính năng:

- Nhập đúng:



Máy tính đơn giản

Nhập số thứ nhất: 3.5

Nhập số thứ hai: 2

Chọn phép toán:

☐ +

☐ -

☒ x

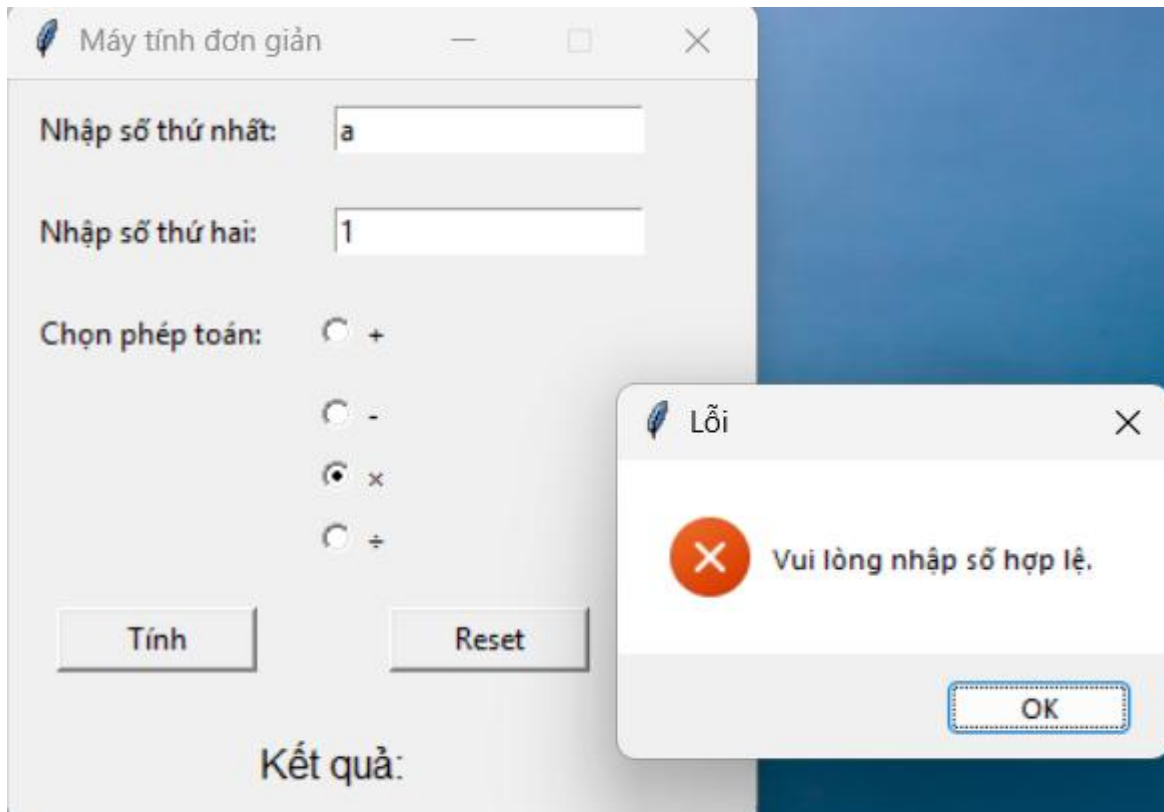
☐ ÷

Tính Reset

Kết quả: 7.0

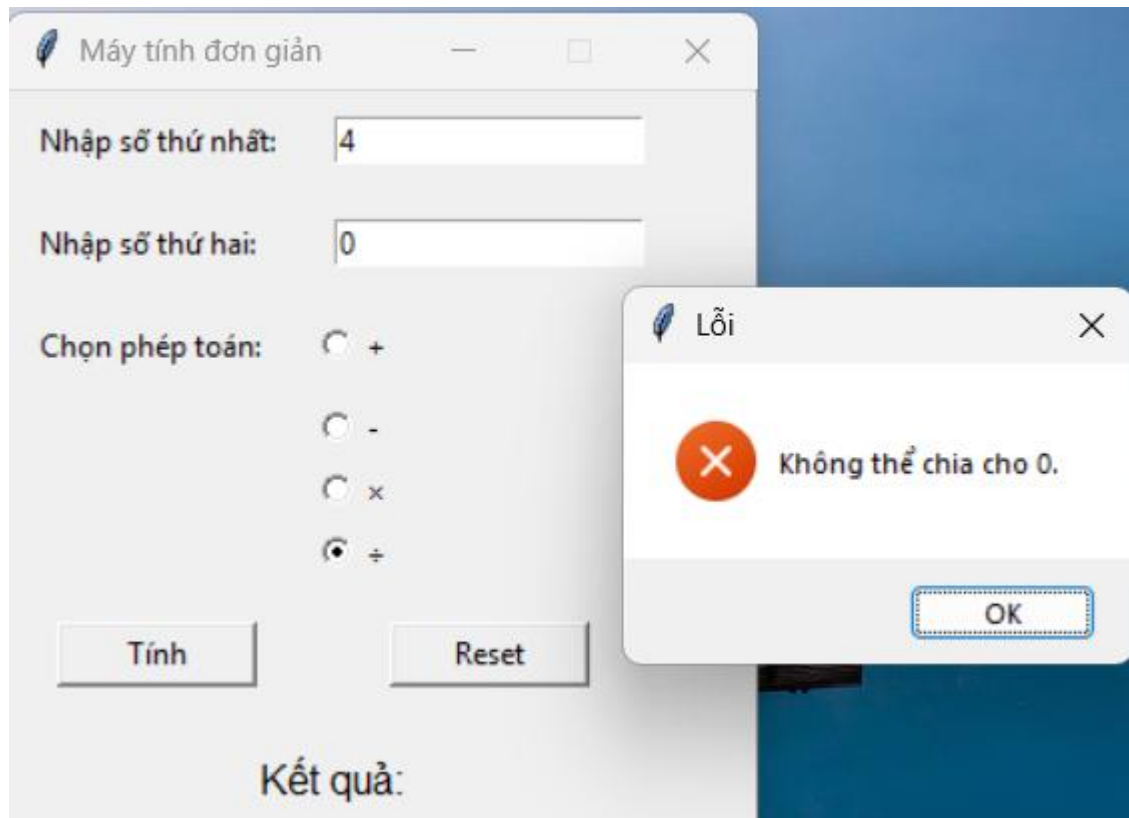
Hình 4.1.1: Nhập đúng định dạng số

- Nhập sai:



*Hình 4.1.2: Nhập sai định dạng số*

- Nhập chia cho 0



*Hình 4.1.3: 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.

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

*Hình 4.2.1.a: Thư viện tkinter*

#### 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 4.2.2.b: Code nhập dữ liệu cho 2 số*

#### c) Chọn phép toán(radio) 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.

```

if operation == "+":
    result = num1 + num2
elif operation == "-":
    result = num1 - num2
elif operation == "*":
    result = num1 * num2
elif operation == "/":
    if num2 == 0:
        raise ZeroDivisionError
    result = num1 / num2
else:
    result = "Chọn phép toán"

result_label.config(text="Kết quả: " + str(result))

```

*Hình 4.2.3.c: Code chọn phép toán và tính toán*

## 4.2 Kết luận

Chương trình Máy tính đơn giản sử dụng Python và thư viện Tkinter đã được xây dựng và hoàn thiện với giao diện trực quan, dễ sử dụng. Các chức năng cơ bản như cộng, trừ, nhân, chia đã được triển khai đầy đủ cùng với khả năng xử lý lỗi hiệu quả.

### Ưu điểm:

- Giao diện đơn giản, dễ thao tác.
- Xử lý tính toán chính xác.
- Có khả năng phát hiện và thông báo lỗi kịp thời.
- Dễ dàng mở rộng hoặc nâng cấp.

**Hạn chế:**

- Chưa hỗ trợ tính toán nâng cao như lũy thừa, căn bậc hai.
- Chưa hỗ trợ nhập biểu thức nhiều toán hạng.
- Giao diện đơn giản, chưa tùy biến nhiều về thẩm mỹ.

**Định hướng phát triển:**

- Bổ sung thêm các phép toán nâng cao.
- Cho phép nhập và tính biểu thức phức hợp.
- Cải tiến giao diện theo hướng hiện đại và thân thiện hơn với người dùng.



## **TÀI LIỆU THAM KHẢO**

1. W3Schools.
2. Tkinter Docs - Python Interface to Tcl/Tk.
3. Python Tkinter Tutorial.
4. ChatGPT – OpenAI