

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIA ĐỊNH
KHOA CÔNG NGHỆ THÔNG TIN



TIỂU LUẬN
THIẾT KẾ VÀ XÂY DỰNG
PHẦN MỀM BASIC CALCULATOR
MÔN: LẬP TRÌNH
ỨNG DỤNG VỚI JAVA

Chuyên ngành: **Kỹ thuật phần mềm**

Ngành: **Công nghệ thông tin**

Giảng viên hướng dẫn: **ThS. Nguyễn Thành Sơn**

Sinh viên thực hiện: **Nguyễn Gia Bảo**

MSSV: **22150450**

Lớp: **221407**

TP. Hồ Chí Minh, tháng 11 năm 2024

Khoa Công nghệ thông tin

NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN

TIỂU LUẬN MÔN: LẬP TRÌNH ỨNG DỤNG VỚI JAVA

1. **Sinh viên:** Nguyễn Gia Bảo
2. **Tên đề tài:** Thiết kế và xây dựng phần mềm Basic Calculator
3. **Nhận xét:**

a) Những kết quả đạt được:

.....

.....

.....

.....

.....

b) Những hạn chế:

.....

.....

.....

.....

.....

4. **Điểm đánh giá** (*theo thang điểm 10, làm tròn đến 0.5*):

Sinh viên:.....

Điểm số: Điểm chữ:

TP. HCM, ngày ... tháng ... năm 20.....

Giảng viên chấm thi

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Em xin chân thành cảm ơn sâu sắc và được bày tỏ lòng biết ơn đến với Quý Thầy Cô của khoa Công Nghệ Thông Tin Trường Đại học Gia Định và đặc biệt đối với thầy **ThS. Nguyễn Thành Sơn** đã tận tình hướng dẫn, đồng hành, động viên và chỉ bảo giúp tôi hoàn thành tốt bài báo cáo này cũng như truyền đạt những kiến thức bổ ích và rất quan trọng đối với quá trình đi làm của chúng em sau này.

Trong thời gian nghiên cứu đề tài tiểu luận vừa qua, em đã có nhiều cố gắng trong suốt quá trình thực hiện tiểu luận, tích cực trao đổi thông tin, sưu tầm, tham khảo tài liệu và học hỏi thêm kiến thức từ các Thầy Cô và các học viên khác, không chỉ nhận được những kiến thức đầy bổ ích về chuyên môn mà còn ở những lĩnh vực khác. Những trải nghiệm quý báu đó không chỉ giúp chúng em hoàn thành tốt bài báo cáo mà còn là hành trang quan trọng theo tôi trong suốt thời gian học tập và giúp tôi có thể tự tin bước vào đời làm việc sau này.

Do kiến thức và kinh nghiệm còn hạn chế cho nên bài báo cáo của em còn nhiều thiếu sót, kính mong được sự đánh giá, góp ý của thầy.

Cuối cùng, xin kính chúc thầy luôn luôn khỏe mạnh, vui vẻ và đạt được nhiều thành công cao trong công tác giảng dạy. Chúc trường Đại học Gia Định sẽ luôn là nền tảng vững chắc cho nhiều thế hệ sinh viên tiếp bước trên con đường học tập.

Xin chân thành cảm ơn!

Nguyễn Gia Bảo

MỤC LỤC

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	1
1.1. TỔNG QUAN MÔN HỌC	1
1.2. KIẾN THỨC ÁP DỤNG VÀO ĐỀ TÀI	2
CHƯƠNG 2: THỰC HIỆN ĐỀ TÀI.....	4
2.1. KHẢO SÁT HIỆN TRẠNG	4
2.2. ĐẶT RA YÊU CẦU	5
2.3. SƠ ĐỒ USE CASE	6
2.4. SƠ ĐỒ CLASS	8
2.5. CHẠY CHƯƠNG TRÌNH	11
CHƯƠNG 3: TỔNG KẾT	24
3.1. KẾT QUẢ ĐẠT ĐƯỢC	24
3.2. VỀ ƯU ĐIỂM KHI THỰC HIỆN ĐỀ TÀI.....	24
3.3. VỀ NHƯỢC ĐIỂM KHI THỰC HIỆN ĐỀ TÀI.....	24
3.4. VỀ HƯỚNG PHÁT TRIỂN, MỞ RỘNG ĐỀ TÀI	24
PHỤ LỤC: TÀI LIỆU THAM KHẢO	26

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan môn học

Lập trình ứng dụng Java là một lĩnh vực tập trung vào việc phát triển phần mềm sử dụng ngôn ngữ Java – một ngôn ngữ lập trình mạnh mẽ, linh hoạt, và đa nền tảng. Với lịch sử phát triển lâu đời, Java đã trở thành một trong những ngôn ngữ được lựa chọn hàng đầu để xây dựng các ứng dụng doanh nghiệp, hệ thống ngân hàng, ứng dụng di động Android, và nhiều lĩnh vực khác.

Khái niệm cốt lõi của lập trình Java bao gồm lập trình hướng đối tượng (OOP), một phương pháp tổ chức mã nguồn giúp mã dễ dàng mở rộng, bảo trì, và tái sử dụng. Khi học lập trình Java, người học sẽ nắm vững các yếu tố cơ bản như lớp, đối tượng, kế thừa, đa hình, và trừu tượng hóa. Những khái niệm này không chỉ là nền tảng của Java mà còn giúp xây dựng các cấu trúc phần mềm có tính tổ chức, dễ hiểu, và có thể mở rộng, một yếu tố quan trọng trong quá trình phát triển phần mềm chuyên nghiệp.

Java và ứng dụng thực tế mang lại nhiều lợi ích lớn. Java có tính năng "viết một lần, chạy mọi nơi" (Write Once, Run Anywhere), nghĩa là ứng dụng Java có thể chạy trên nhiều hệ điều hành mà không cần thay đổi mã nguồn. Khả năng này mang lại hiệu quả cao trong phát triển phần mềm, đặc biệt trong môi trường doanh nghiệp hoặc tổ chức lớn nơi cần phần mềm có tính tương thích đa nền tảng. Java cũng nổi bật nhờ khả năng bảo mật và hiệu suất cao, đặc biệt trong các ứng dụng đòi hỏi xử lý phức tạp và bảo mật cao như trong hệ thống ngân hàng và thương mại điện tử.

Phát triển ứng dụng Java và các kỹ năng cần thiết gồm việc xây dựng từ các ứng dụng console cơ bản cho đến các ứng dụng giao diện đồ họa bằng JavaFX. Việc nắm bắt JavaFX giúp người học phát triển các ứng dụng có giao diện người dùng thân thiện và khả năng tương tác cao. Ngoài ra, người học sẽ biết cách kết nối với cơ sở dữ liệu, một kỹ năng quan trọng để tạo ra các ứng dụng lưu trữ và quản lý dữ liệu hiệu quả. Kỹ năng này đáp ứng được nhu cầu của nhiều ứng dụng hiện đại và mở rộng cơ hội làm việc trong nhiều lĩnh vực công nghệ.

Kết luận, lập trình ứng dụng Java cung cấp cho người học kiến thức và kỹ năng cần thiết để phát triển các ứng dụng từ ý tưởng đến sản phẩm hoàn thiện. Với nền tảng Java, người học sẽ tự tin hơn khi bước vào ngành công nghệ thông tin, đáp ứng

yêu cầu đa dạng của thị trường, và phát triển khả năng sáng tạo, tư duy logic trong việc giải quyết các vấn đề công nghệ phức tạp.

1.2. Kiến thức áp dụng vào đề tài

Để áp dụng kiến thức lập trình Java vào đề tài "Thiết kế và Xây dựng Phần mềm Basic Calculator," bạn sẽ sử dụng nhiều yếu tố quan trọng trong Java, từ lập trình hướng đối tượng đến các thành phần giao diện đồ họa cơ bản.

1.2.1. Lập trình hướng đối tượng (OOP):

Trong Java, việc thiết kế phần mềm máy tính cơ bản với các lớp (class) đại diện cho các thành phần như màn hình hiển thị kết quả và các nút số và phép toán. Tạo các lớp như Calculator để quản lý toàn bộ hoạt động của máy tính, hiển thị trên màn hình, và các lớp tương ứng cho các phép toán (cộng, trừ, nhân, chia) giúp tách biệt và tổ chức mã tốt hơn.

1.2.2. Giao diện người dùng (JavaFX):

JavaFX sẽ là một công cụ hữu ích để tạo giao diện người dùng trực quan. Bạn có thể thiết kế bố cục máy tính bằng cách sử dụng các thành phần như Button cho các phím số và phép toán, TextField để hiển thị kết quả, và các bố cục như GridPane để sắp xếp các nút một cách gọn gàng. Từ đó, phần mềm sẽ có một giao diện đơn giản và dễ sử dụng.

1.2.3. Xử lý sự kiện:

Với máy tính, mỗi phím nhấn tương ứng với một thao tác tính toán hoặc hiển thị. JavaFX cung cấp các tính năng xử lý sự kiện mạnh mẽ để bạn có thể lập trình cho các nút, giúp phần mềm đáp ứng mỗi lần người dùng nhấn một phím. Điều này đảm bảo tính toán được thực hiện ngay lập tức, tạo ra trải nghiệm mượt mà.

1.2.4. Tính năng xử lý lỗi và ngoại lệ:

Khi xây dựng phần mềm máy tính, việc xử lý lỗi như chia cho 0 hoặc nhập sai là rất quan trọng. Tận dụng tính năng xử lý ngoại lệ của Java để hiển thị thông báo hoặc xử lý hợp lý khi gặp phải những lỗi như vậy, giúp phần mềm chạy ổn định và không bị gián đoạn.

1.2.5. Tính toán cơ bản và quản lý logic:

Dự án này giúp phát triển kỹ năng tư duy logic trong việc thiết kế các phép tính cơ bản và cách lưu trữ, quản lý kết quả trung gian. Khi người dùng nhập một dãy

phép toán, phần mềm sẽ lưu trữ giá trị và phép toán hiện tại, sau đó thực hiện phép tính theo thứ tự khi người dùng nhấn dấu bằng.

Kết hợp các kỹ năng trên sẽ tạo ra một ứng dụng máy tính cơ bản nhưng đầy đủ chức năng và dễ sử dụng. Đề tài này là một bước khởi đầu tốt để luyện tập khả năng lập trình, tư duy tổ chức mã nguồn, và phát triển giao diện người dùng, đồng thời mở rộng cơ hội để bạn ứng dụng kiến thức Java vào các dự án phức tạp hơn trong tương lai.

CHƯƠNG 2: THỰC HIỆN ĐỀ TÀI

2.1. Khảo sát hiện trạng

Trong bối cảnh hiện tại, phần mềm máy tính cơ bản là một công cụ thiết yếu trong hầu hết mọi lĩnh vực, từ học tập, văn phòng đến đời sống hàng ngày. Máy tính cơ bản phục vụ các nhu cầu tính toán đơn giản nhưng quan trọng, chẳng hạn như thực hiện phép cộng, trừ, nhân, chia nhanh chóng mà không cần máy tính vật lý. Người dùng có thể là học sinh, sinh viên, nhân viên văn phòng, hoặc bất kỳ ai cần thực hiện các phép tính nhanh gọn trên máy tính cá nhân hay thiết bị di động.

2.1.1. Hiện trạng và các giải pháp hiện tại:

Trên thị trường hiện nay, đa phần các hệ điều hành phổ biến như Windows, macOS, và Android đều tích hợp các ứng dụng máy tính cơ bản. Tuy nhiên, các ứng dụng này thường khá hạn chế về chức năng hoặc không tùy chỉnh được giao diện và trải nghiệm người dùng theo ý muốn của từng cá nhân hay đơn vị. Một số ứng dụng trực tuyến cũng có sẵn, nhưng yêu cầu người dùng phải có kết nối internet, điều này gây bất tiện khi sử dụng trong môi trường không có mạng.

2.1.2. Nhu cầu thực tế:

Trong môi trường học tập và làm việc, nhiều người cần một ứng dụng máy tính đơn giản nhưng trực quan, dễ thao tác và có khả năng thực hiện các phép tính cơ bản một cách hiệu quả mà không yêu cầu cấu hình phức tạp. Đặc biệt, nhu cầu về một công cụ có thể dễ dàng tùy biến và có tính ổn định cao trong quá trình sử dụng là rất lớn. Ngoài ra, các công ty hoặc tổ chức giáo dục cũng có thể cần các ứng dụng máy tính tích hợp vào hệ thống riêng của mình để dễ dàng hỗ trợ các hoạt động tính toán nội bộ mà không phụ thuộc vào các ứng dụng bên ngoài.

2.1.3. Sự cần thiết phải xây dựng ứng dụng Basic Calculator:

Việc xây dựng một ứng dụng máy tính cơ bản sẽ giúp đáp ứng nhu cầu sử dụng một công cụ tính toán nhanh, dễ dùng và không phụ thuộc vào môi trường mạng. Ứng dụng này không chỉ phù hợp với người dùng cá nhân mà còn có thể tích hợp vào các hệ thống phần mềm lớn hơn trong tương lai. Ngoài ra, dự án này sẽ giúp ích cho việc thực hành và phát triển kỹ năng lập trình, thiết kế giao diện người dùng và xử lý logic tính toán – những yếu tố cần thiết cho các lập trình viên khi bước vào lĩnh vực phát triển phần mềm.

2.2. Đặt ra yêu cầu

2.2.1. Yêu cầu về chức năng:

Thực hiện các phép tính cơ bản: Phần mềm phải có khả năng thực hiện các phép tính cộng, trừ, nhân, chia giữa hai số.

Lưu trữ và thao tác trên số thứ nhất và số thứ hai: Khi người dùng nhập vào một phép toán, phần mềm phải lưu số thứ nhất, đợi nhập số thứ hai, và sau đó tính toán khi nhấn dấu bằng.

Kiểm tra phép tính chia cho 0: Khi thực hiện phép chia, nếu số thứ hai là 0, phần mềm cần hiển thị thông báo lỗi thay vì thực hiện phép tính để tránh lỗi toán học.

Xóa toàn bộ và xóa từng ký tự: Phần mềm cần có các nút "C" để xóa toàn bộ màn hình và nút "Backspace" để xóa từng ký tự khi người dùng nhập sai.

Hiển thị kết quả và thao tác liên tiếp: Kết quả của phép toán sẽ được hiển thị trên màn hình và có thể tiếp tục sử dụng để thực hiện các phép toán tiếp theo.

2.2.2. Yêu cầu về giao diện người dùng (UI):

TextField hiển thị kết quả: Phần mềm cần có một TextField để hiển thị kết quả và các phép toán người dùng đã nhập. TextField này phải không cho phép chỉnh sửa trực tiếp.

Sắp xếp các nút giống máy tính tiêu chuẩn: Các nút số và phép toán phải được bố trí theo bố cục quen thuộc của máy tính, giúp người dùng dễ dàng thao tác.

Thiết kế nút có kích thước hợp lý: Các nút cần có kích thước đồng đều và dễ bấm, giúp tạo trải nghiệm tốt cho người dùng.

Thông báo lỗi: Giao diện cần hiển thị thông báo lỗi rõ ràng khi xảy ra lỗi, chẳng hạn như khi chia cho 0 hoặc nhập dữ liệu không hợp lệ.

2.2.3. Yêu cầu về xử lý sự kiện và tương tác:

Thiết lập xử lý sự kiện cho từng nút: Mỗi nút số và nút phép toán cần có sự kiện `setOnClickListener` để xử lý thao tác của người dùng.

Phân tách giữa phép toán và số: Khi nhấn nút phép toán, phần mềm cần lưu trữ số thứ nhất và phép toán trước khi xóa màn hình để nhập số thứ hai.

Tính toán ngay khi nhấn dấu bằng: Khi nhấn nút "=", phần mềm phải thực hiện phép toán và hiển thị kết quả ngay lập tức, đồng thời chuyển kết quả thành số thứ nhất để người dùng tiếp tục thực hiện các phép tính khác.

2.2.4. Yêu cầu về xử lý lỗi và ngoại lệ:

Kiểm tra dữ liệu đầu vào hợp lệ: Khi thực hiện phép toán, phần mềm phải xác minh rằng người dùng đã nhập một số hợp lệ để tránh lỗi không mong muốn.

Xử lý phép chia cho 0: Khi người dùng nhập phép chia cho 0, phần mềm sẽ hiển thị thông báo “Không thể chia cho 0” thay vì thực hiện phép tính.

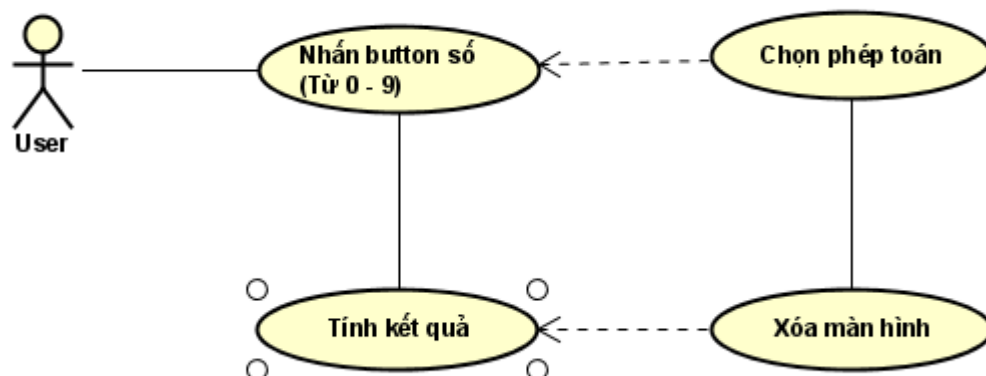
Thông báo lỗi rõ ràng: Mỗi khi có lỗi nhập liệu hoặc lỗi phép toán, phần mềm sẽ hiển thị thông báo lỗi để người dùng biết cần thao tác lại.

2.2.5. Yêu cầu về tính ổn định và bảo trì:

Mã nguồn dễ bảo trì và mở rộng: Mã nguồn phải được viết rõ ràng và có cấu trúc để dễ dàng sửa đổi và nâng cấp trong tương lai. Điều này bao gồm việc sử dụng các phương thức tách biệt trong CalculatorController để xử lý các thao tác riêng lẻ như chọn phép toán và tính kết quả.

Tính khả dụng trên nhiều hệ điều hành: Ứng dụng cần chạy ổn định trên hệ điều hành phổ biến như Windows để đảm bảo tính tiện dụng cho nhiều người dùng.

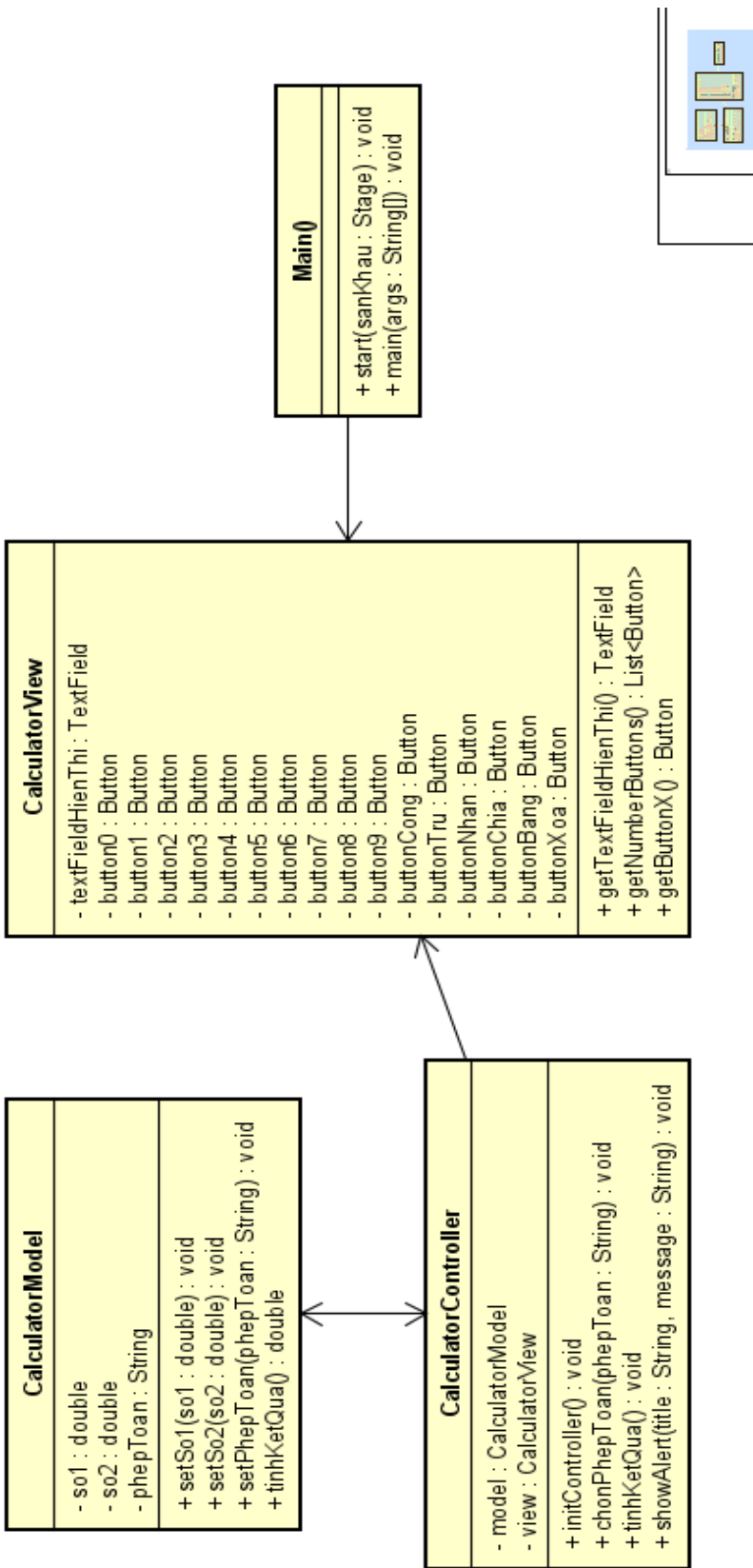
2.3. Sơ đồ Use case



STT	Tên UseCase	Ý nghĩa
1	Nhập Số	Người dùng nhấn các nút số để nhập vào các giá trị số.
2	Chọn Phép Toán	Người dùng chọn một trong các phép toán (+, -, *, /).
3	Tính Kết Quả	Người dùng nhấn nút "=" để tính toán kết quả dựa trên phép toán đã chọn và các số đã nhập.
4	Xóa Nhập Liệu	Người dùng nhấn nút "C" để xóa các dữ liệu nhập vào.

2.4. Sơ đồ Class

2.4.1.Sơ đồ Class tổng thể



2.4.2. Sơ đồ class CalculatorModel

- Thuộc tính:
- + so1: double
- + so2: double
- + phepToan: String
- Phương thức:
- + setSo1(double so1): Thiết lập số thứ nhất.
- + setSo2(double so2): Thiết lập số thứ hai.
- + setPhepToan(String phepToan): Thiết lập phép toán.
- + tinhKetQua(): Thực hiện phép tính và trả về kết quả.

2.4.3. Sơ đồ class CalculatorView

- Thuộc tính:
- + textFieldHienThi: Trường nhập hiển thị kết quả và số.
- + button0 - button9: Các nút số.
- + buttonCong, buttonTru, buttonNhan, buttonChia, buttonBang, buttonXoa: Các nút chức năng.

- Phương thức:
- + getTextFieldHienThi(): Lấy TextField hiện tại.
- + getButtonX(): Các phương thức lấy nút tương ứng cho từng phím.
- + getNumberButtons(): Lấy danh sách các nút số.

2.4.4. Sơ đồ class CalculatorController

- Thuộc tính:
- + model: Đối tượng CalculatorModel để lưu trữ các dữ liệu.
- + view: Đối tượng CalculatorView để hiển thị và lấy các thao tác từ người dùng.
- Phương thức:
- + chonPhepToan(String phepToan): Lưu phép toán đã chọn vào model.
- + tinhKetQua(): Tính toán kết quả và hiển thị trong view.
- + showAlert(String title, String message): Hiển thị thông báo lỗi khi cần thiết.

2.4.5. Sơ đồ class Main

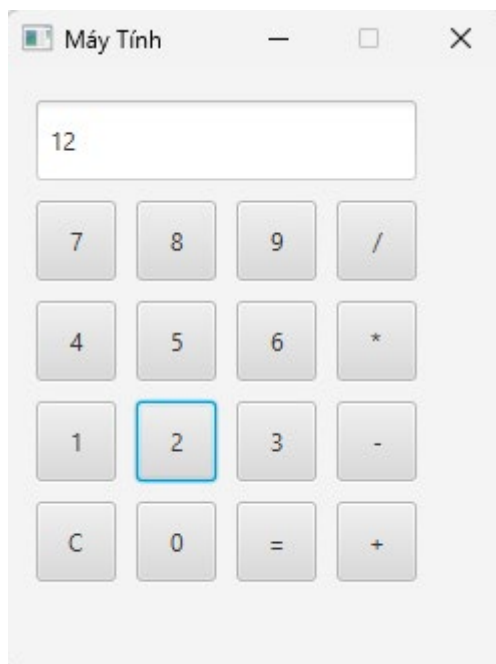
- Phương thức:

- + `start(Stage sanKhau)`: Tạo và kết nối các lớp trong mô hình MVC.
- + `main(String[] args)`: Điểm khởi chạy ứng dụng.

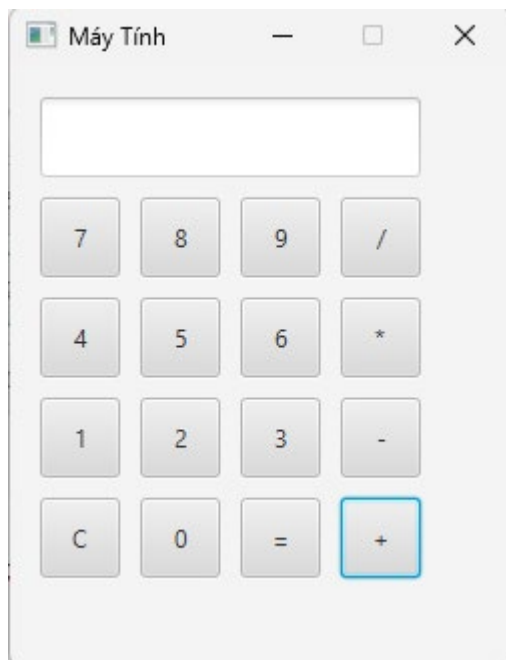
2.5. Chạy chương trình

2.5.1. Làm phép tính cộng

Bước 1: Nhập số nguyên thứ nhất



Bước 2: Nhấn buttonCong để lưu số nguyên thứ nhất, chuyển sang nhập số nguyên thứ hai



Bước 3: Nhập số nguyên thứ hai



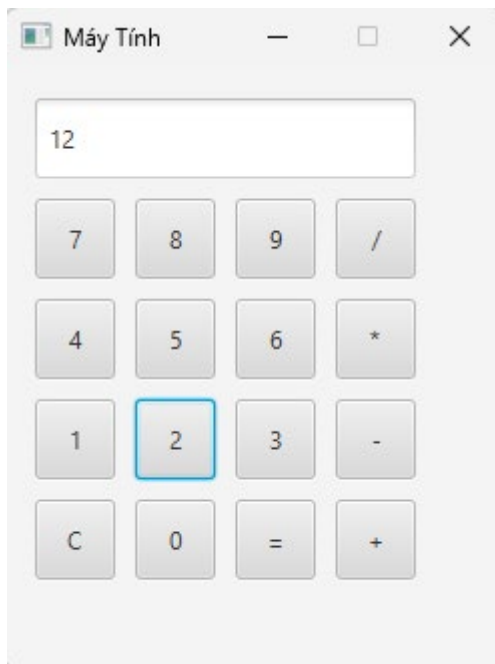
Bước 4: Nhấn button `Bang` để cho ra kết quả ($12+3=5$)



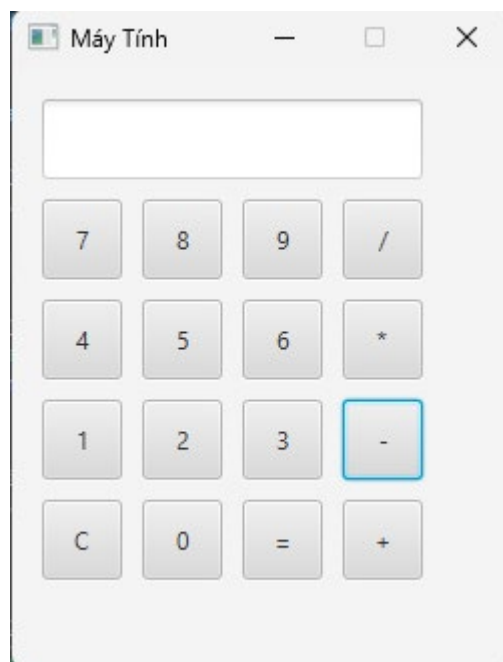
2.5.2. Làm phép tính trừ

a) Khi kết quả là số nguyên

Bước 1: Nhập số nguyên thứ nhất



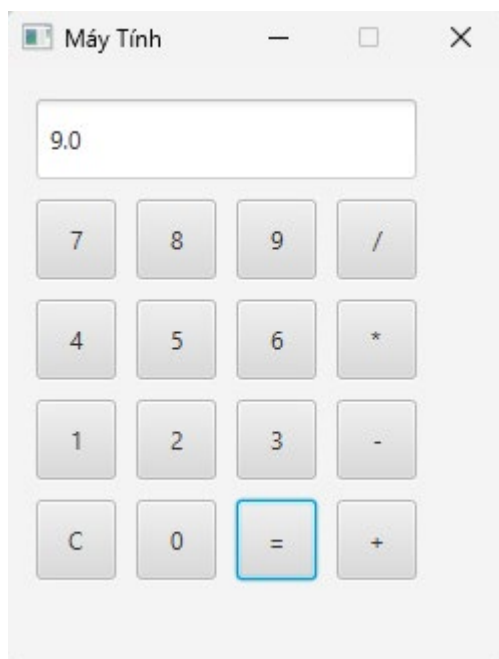
Bước 2: Nhấn buttonTrừ để lưu số nguyên thứ nhất, chuyển sang nhập số nguyên thứ hai



Bước 3: Nhập số nguyên thứ hai

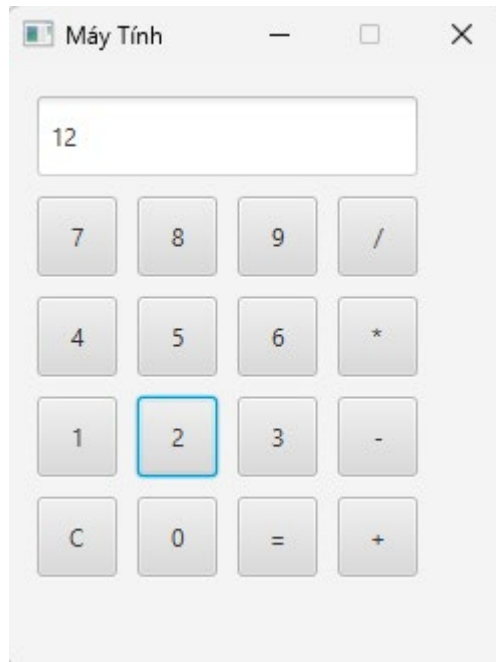


Bước 4: Nhấn buttonBằng để cho ra kết quả (12-3=9)

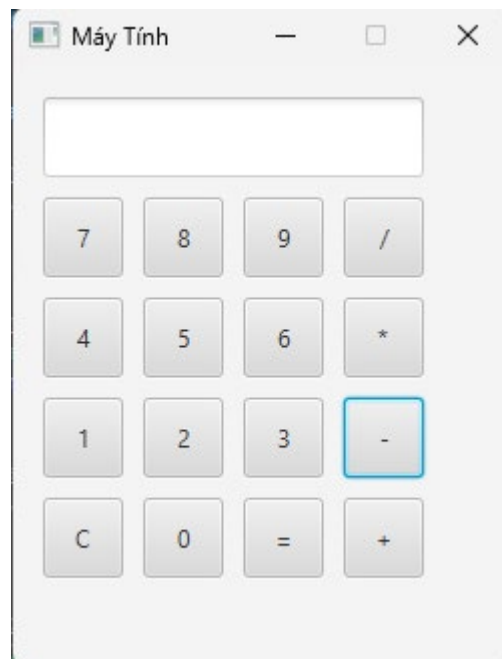


b) Khi kết quả là số nguyên

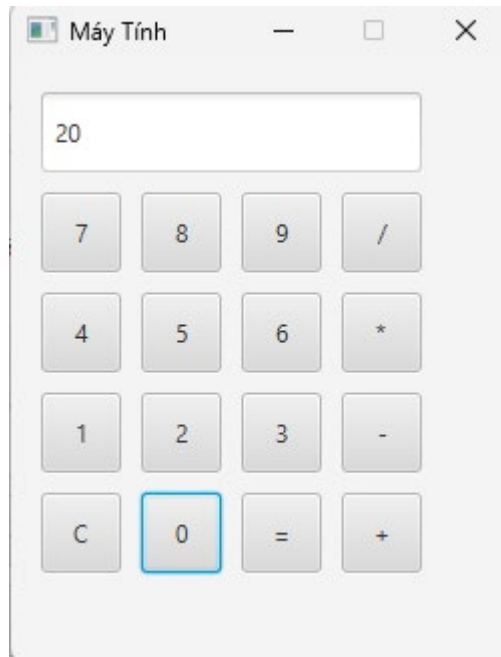
Bước 1: Nhập số nguyên thứ nhất



Bước 2: Nhấn button `Tru` để lưu số nguyên thứ nhất, chuyển sang nhập số nguyên thứ hai



Bước 3: Nhập số nguyên thứ hai

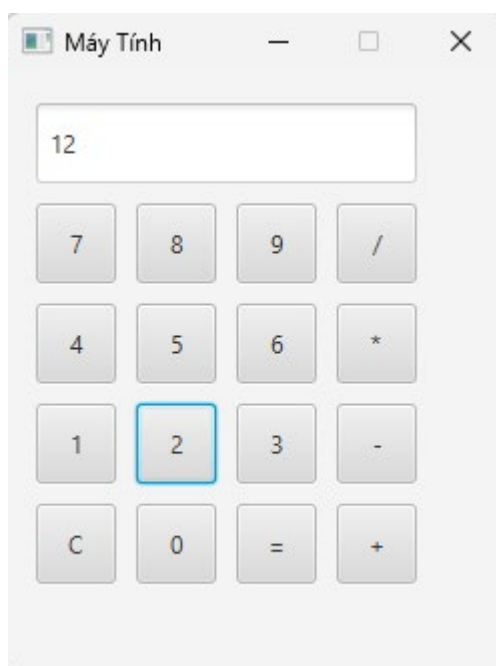


Bước 4: Nhấn buttonBằng để cho ra kết quả ($12-20 = -8$)

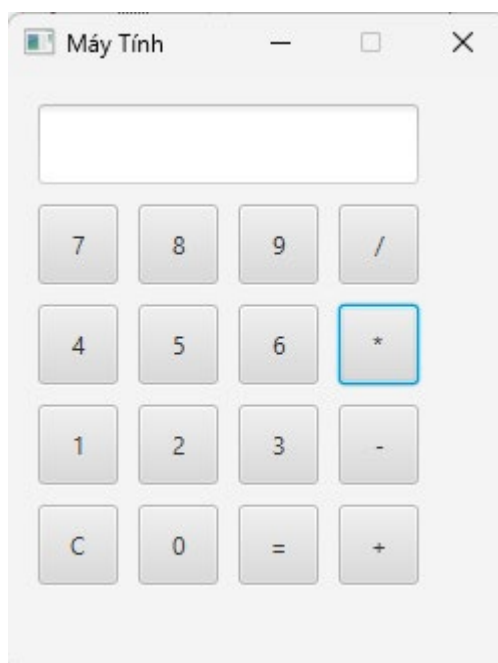


2.5.3. Làm phép tính nhân

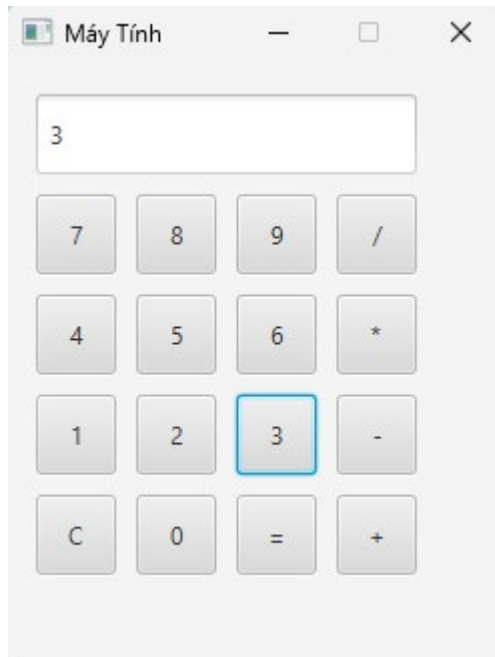
Bước 1: Nhập số nguyên thứ nhất



Bước 2: Nhấn buttonNhân để lưu số nguyên thứ nhất, chuyển sang nhập số nguyên thứ hai



Bước 3: Nhập số nguyên thứ hai



Bước 4: Nhấn buttonBang để cho ra kết quả ($12 * 3 = 36$)



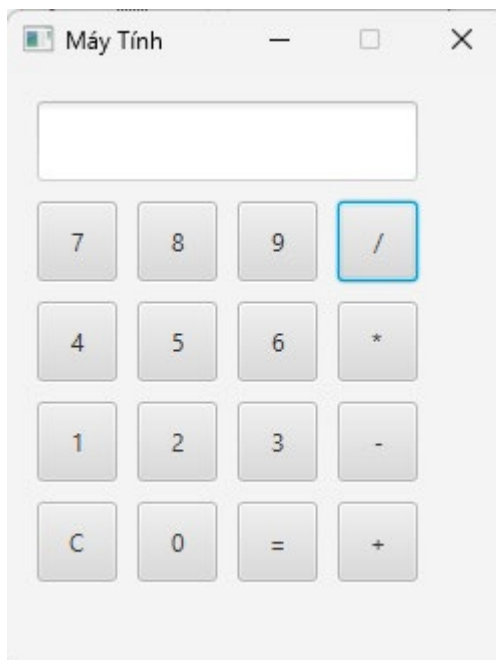
2.5.4. Làm phép tính chia

a) Khi chia cho số nguyên

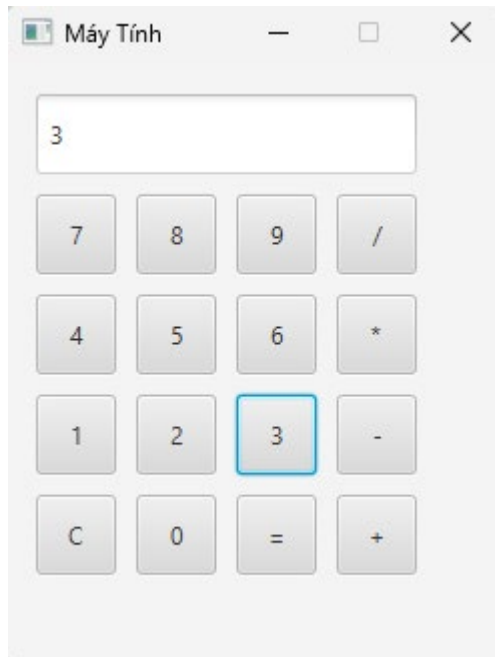
Bước 1: Nhập số nguyên thứ nhất



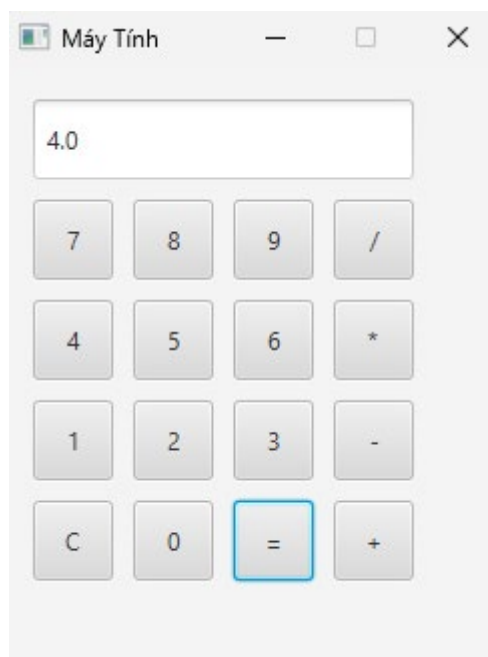
Bước 2: Nhấn buttonChia để lưu số nguyên thứ nhất, chuyển sang nhập số nguyên thứ hai



Bước 3: Nhập số nguyên thứ hai



Bước 4: Nhấn buttonBằng để cho ra kết quả ($12 / 3 = 4$)

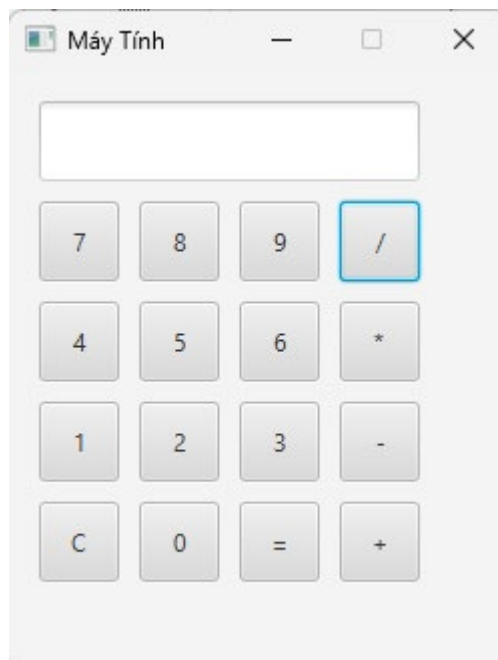


b) Khi chia cho số 0

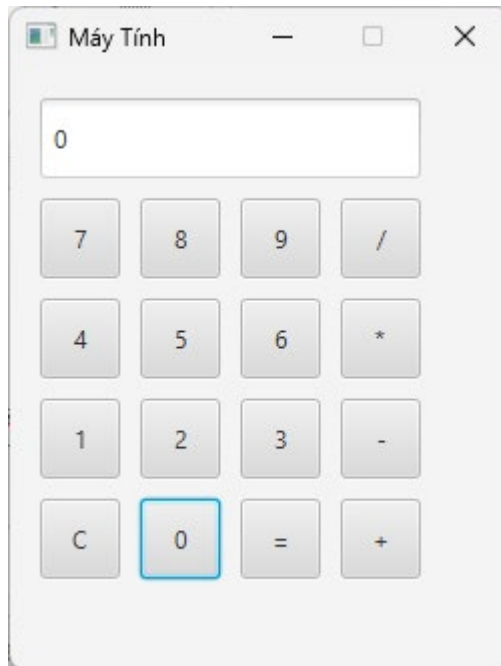
Bước 1: Nhập số nguyên thứ nhất



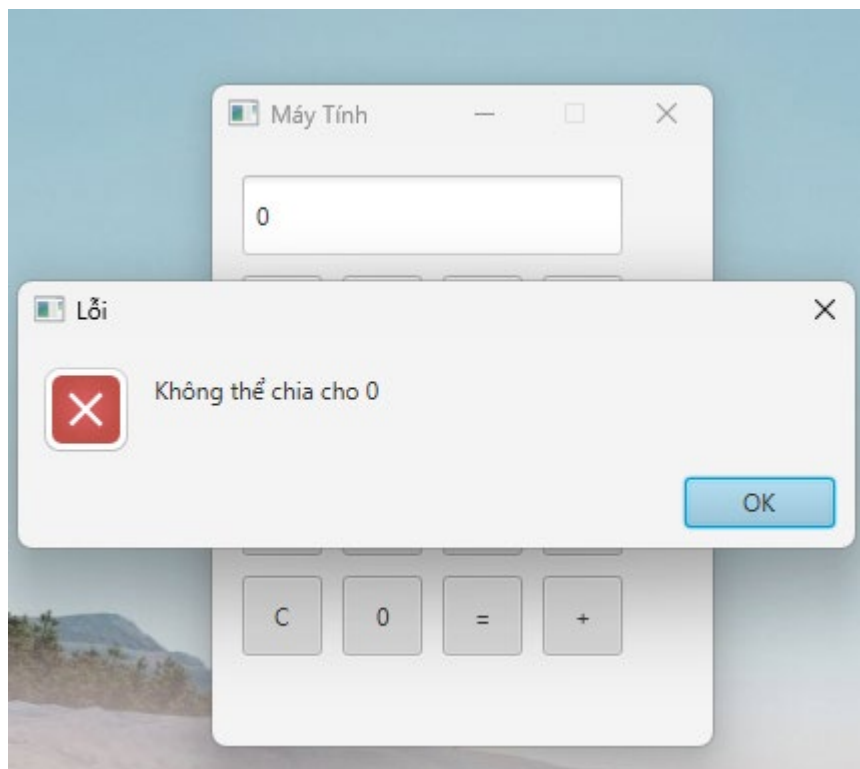
Bước 2: Nhấn buttonChia để lưu số nguyên thứ nhất, chuyển sang nhập số nguyên thứ hai



Bước 3: Nhập số nguyên thứ hai

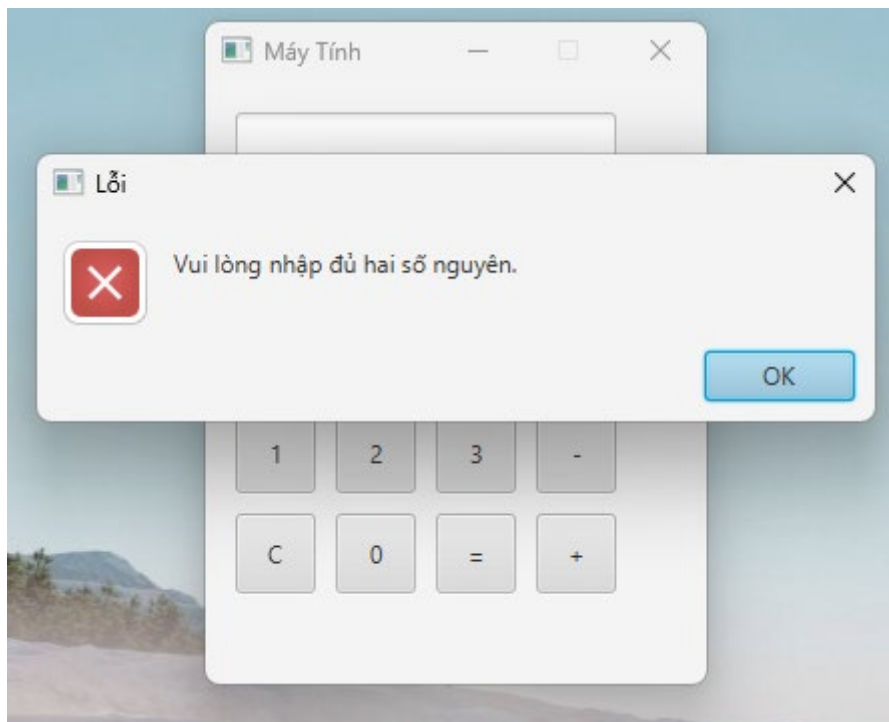


Bước 4: Nhấn buttonBang, khi này sẽ hiện thông báo “Không thể chia cho 0”



2.5.5. *Hiện lỗi thông báo*

Khi không nhập đủ 2 số nguyên để thực hiện phép tính sẽ hiện thông báo "Vui lòng nhập đủ hai số nguyên"



CHƯƠNG 3: TỔNG KẾT

3.1. Kết quả đạt được

Đề tài "Thiết kế và Xây dựng Phần mềm Basic Calculator" đã thành công trong việc xây dựng một ứng dụng máy tính cơ bản với các chức năng tính toán chính bao gồm cộng, trừ, nhân, chia. Người dùng có thể nhập số liệu, chọn phép toán, và nhận kết quả hiển thị trên giao diện trực quan. Ứng dụng đáp ứng tốt các yêu cầu cơ bản về thao tác tính toán cũng như khả năng xử lý lỗi, đảm bảo người dùng không bị gián đoạn khi nhập các giá trị không hợp lệ (như phép chia cho 0). Giao diện đơn giản, dễ sử dụng giúp người dùng dễ dàng thao tác và kiểm soát các phép tính cần thực hiện.

Tuy nhiên, hiện phần mềm chỉ làm được ở mức cơ bản do thiếu quá trình tìm hiểu, cũng như thời gian có hạn đã không làm được những chức năng như tính số thập phân, hàm lượng giác; xem lại phép tính trước kia thực hiện; thực hiện nhiều phép tính cùng một lúc, ... Sau khi thực hiện đề tài này sẽ cố gắng cải thiện và thêm nhiều tính năng để người dùng có thể trải nghiệm một cách hiệu quả hơn.

3.2. Về ưu điểm khi thực hiện đề tài

Đề tài đã đạt được yêu cầu của một ứng dụng máy tính cơ bản với giao diện dễ sử dụng và khả năng xử lý chính xác các phép toán cơ bản. Sử dụng mô hình MVC giúp mã nguồn dễ dàng bảo trì và mở rộng, đồng thời đảm bảo tính rõ ràng trong việc phân chia trách nhiệm giữa các thành phần Model, View, và Controller. Phần mềm phản hồi nhanh chóng và hoạt động mượt mà, đáp ứng các thao tác cơ bản một cách hiệu quả.

3.3. Về nhược điểm khi thực hiện đề tài

Do tính chất cơ bản của đề tài, phần mềm còn hạn chế ở các chức năng mở rộng như tính toán khoa học, hỗ trợ nhiều phép tính phức tạp hơn, hoặc khả năng lưu lại lịch sử tính toán. Bên cạnh đó, giao diện tuy dễ sử dụng nhưng còn đơn giản, chưa hấp dẫn về mặt trực quan, đặc biệt thiếu các hiệu ứng và yếu tố giao diện hiện đại.

3.4. Về hướng phát triển, mở rộng đề tài

Để cải thiện và phát triển đề tài, có thể sẽ mở rộng theo một số hướng như:

1. Bổ sung chức năng tính toán khoa học: Thêm các phép toán nâng cao như lũy thừa, căn bậc hai, hàm lượng giác (\sin , \cos , \tan) để đáp ứng nhu cầu sử dụng đa dạng hơn.
2. Lịch sử tính toán: Phát triển tính năng lưu lại lịch sử tính toán, cho phép người dùng xem lại các phép tính đã thực hiện trước đó, giúp tiện ích hơn trong quá trình sử dụng.
3. Nâng cấp giao diện: Tối ưu giao diện đồ họa để tạo trải nghiệm người dùng trực quan và chuyên nghiệp hơn, có thể sử dụng các hiệu ứng chuyển đổi, thay đổi màu sắc, hoặc các biểu tượng trực quan hơn để cải thiện trải nghiệm thị giác.
4. Tích hợp tính năng bộ nhớ: Cung cấp các nút lưu giá trị vào bộ nhớ (Memory Save, Memory Recall) để người dùng có thể lưu và sử dụng lại giá trị trong các phép toán khác mà không cần nhập lại.

PHỤ LỤC: TÀI LIỆU THAM KHẢO

1. Java là gì? Những điều bạn cần biết về ngôn ngữ lập trình Java. (n.d.). TechWorks - Việc Làm IT. <https://techworks.vn/blog/java-la-gi>
2. Minh T. P. (2023, June 18). Hiểu rõ về JavaFX và những nhược điểm của JavaFX. Z.com Cloud VPS - Tốc Độ Cao Khởi Tạo Trong 1 Phút |. <https://cloud.z.com/vn/news/javafx/>
3. Tài liệu và bài tập được ThS. Nguyễn Thành Sơn giảng dạy và thực hành trên lớp
4. CalculatorMVC/src/application at master · TsukiZombina/CalculatorMVC. (n.d.). GitHub. <https://github.com/TsukiZombina/CalculatorMVC/tree/master/src/application>
5. Almas Baim (AlmasB). (2015, August 1). JavaFX Software Tutorial: Calculator (MVC) [Video]. YouTube. <https://www.youtube.com/watch?v=y1ZaBalVZic>