

KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH  
HỌC KỲ I, NĂM HỌC 2023-2024

# VIẾT CHƯƠNG TRÌNH MÁY TÍNH BẰNG JAVA

*Giáo viên hướng dẫn:*  
Họ tên: Trịnh Quốc  
Việt

*Sinh viên thực hiện:*  
Họ tên: Phạm Minh Thái  
MSSV: 110121100  
Lớp: DA21TTA

*Trà Vinh, tháng 1 năm 2024*

KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH  
HỌC KỲ I, NĂM HỌC 2023-2024

# VIẾT CHƯƠNG TRÌNH MÁY TÍNH BẰNG JAVA

*Giáo viên hướng dẫn:*  
Họ tên: Trịnh Quốc  
Việt

*Sinh viên thực hiện:*  
Họ tên: Phạm Minh Thái  
MSSV: 110121100  
Lớp: DA21TTA

*Trà Vinh, tháng 1 năm 2024*

## This image shows a full page of primary-ruled paper. It features approximately 28 horizontal dotted lines spaced evenly down the page, providing a guide for handwriting practice. The paper is otherwise blank, with no margins, text, or other markings.

**Giáo viên hướng dẫn**

---

---

Phạm Minh Thái

[illegible]

**Thành viên hội đồng**  
(Ký tên và ghi rõ họ tên)

## LỜI CẢM ƠN

Để có thể hoàn thành được bài báo cáo này, trước hết tôi xin gửi lời cảm ơn chân thành đến Trịnh Quốc Việt thầy đã từng bước hướng dẫn, giúp đỡ tôi trong quá trình học tập cũng như báo cáo đồ án cơ sở ngành.

Tôi xin chân thành cảm ơn sự giúp đỡ của toàn thể giảng viên bộ môn Công nghệ Thông tin trường Đại Học Trà Vinh đã dạy dỗ và truyền đạt cho tôi những kiến thức chuyên môn và tinh thần độc lập ,sáng tạo để tôi có kiên thức thực hiện đồ án cơ sở ngành của mình .

Trong quá trình hoàn thành đề tài, do còn nhiều thiếu sót về kiến thức, kỹ năng cũng như kinh nghiệm nên không thể tránh khỏi những sai sót. Tôi rất mong nhận được ý kiến đóng góp của thầy (cô) để tôi khắc phục những thiếu sót đó và hoàn thành bài báo cáo tốt hơn. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để tôi có thể vững bước sau này.

Một lần nữa tôi xin chân thành cảm ơn!

## MỤC LỤC

MỞ ĐẦU.....	10
1. Lý do chọn đề tài. ....	10
2. Mục tiêu.....	10
3. Nội dung. ....	10
4. Phương pháp nghiên cứu.....	10
5. Đối tượng và phạm vi nghiên cứu. ....	11
Đối tượng nghiên cứu.....	11
CHƯƠNG 1: TỔNG QUAN.....	12
1.1 Cơ sở lý thuyết: ....	12
1.2 Giới thiệu.....	12
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT.....	14
2.1 Cơ sở lý thuyết về Java. ....	14
2.1.1 Nội dung.....	14
2.1.2 Nội dung.....	14
2.2 Lý luận về viết chương trình máy tính: ....	15
2.2.1 Nội dung ....	15
2.2.2 Nội dung.....	16
2.3 Giả thuyết khoa học. ....	17
2.3.1 Nội dung.....	17
2.4 Phương pháp nghiên cứu khoa học. ....	18
2.4.1 Nội dung ....	18
2.4.2 Nội dung.....	19
2.5 Các công trình nghiên cứu liên quan. ....	20
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU .....	22

3.1	Mô tả ứng dụng. ....	22
3.2	Giao diện Netbeans.....	22
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU.....		37
4.1	Hiệu năng của chương trình: .....	37
4.2	Trải nghiệm người dùng: .....	37
4.3	Mô tả giao diện chức năng: .....	37
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....		38
5.1	Kết luận:.....	38
5.2	Hướng phát triển:.....	38
DANH MỤC TÀI LIỆU THAM KHẢO.....		39

## DANH MỤC HÌNH ẢNH VÀ BẢN BIỂU

Hình 3-1. Ứng dụng NetBeans .....	21
Hình 3-2. Giao diện Netbeans .....	21
Hình 3-3. Chọn "File" -> "New Project" .....	22
Hình 3-4. Chọn "Java" -> "Java Application" .....	22
Hình 3-5. Đặt tên cho dự án .....	23
Hình 3-6. Edu.poly.lab2->New->JFrame Form -> đặt tên. ....	23
Hình 3-7. Thiết kế giao diện .....	24
Hình 3-8. Đổi tên cho các Button thành các số và các phép tính.....	24
Hình 3-9. Kết quả sau khi thiết kế .....	25
Hình 3-10. Ấn nút Ctrl s chọn từ 0->9 -> Events->Action->actionPerformed .....	25
Hình 3-11. Viết chức năng lên các số đã chọn.....	26
Hình 3-12. Ấn vào Run và chọn Run File để kiểm tra kết quả sự kiện.....	26
Hình 3-13. Viết chức năng cho nút đảo dấu .....	27
Hình 3-14. Viết sự kiện cho nút đảo dấu .....	27
Hình 3-15. Nhấn vào Run và chọn Run File để kiểm tra. ....	28
Hình 3-16. Ấn Ctrl chọn các nút + - * / chọn btnMultiplyActionPerformed. ....	28
Hình 3-17. Kích vào các nút như + - * / thì phải khai báo .....	29
Hình 3-18. Về lại phần btnMultiplyActionPerformed để viết tiếp sự kiện. ....	29
Hình 3-19. Ấn vào Run chọn Run File .....	29
Hình 3-20. Ấn => Events->Action->actionPerformed để viết chức năng xử lí .....	30
Hình 3-21 Thực hiện phép tính 15+2 rồi bấm bằng, ta được kết quả như hình. ....	30
Hình 3-22. Sửa lỗi khi chưa xóa được giá trị cũ .....	31
Hình 3-23. Quay lại phần btnEqualsActionPerformed để khai báo isEqualsPress = true;.....	31
Hình 3-24. Chọn private void btnNumber1ActionPerformed để sự kiện .....	32



Hình 3-25. Kiểm tra bằng cách thực hiện phép tính .....	32
Hình 3-26. Viết sự kiện cho nút C(Clear).....	33
Hình 3-27. Kiểm tra bằng cách thực hiện phép tính .....	33
Hình 3-28. Viết chức năng cho 3 nút còn lại(sqrt,%,1/x).....	34
Hình 3-29. Kiểm tra bằng cách thực hiện phép tính.....	34
Hình 3-30. Thực hiện phép tính % .....	35
Hình 3-31. Thực hiện phép tính 1/x .....	35

## TÓM TẮT ĐỀ ÁN CƠ SỞ NGÀNH

Trong báo cáo môn học này, em tập trung vào việc viết chương trình máy tính bằng ngôn ngữ lập trình Java. Chương trình máy tính này được xây dựng nhằm cung cấp một trải nghiệm tính toán thuận lợi cho người dùng.

### 1. Lý do chọn đề tài.

Công nghệ thông tin (CNTT) có vai trò quan trọng trong cuộc sống hàng ngày và việc áp dụng CNTT vào các lĩnh vực giúp công việc diễn ra nhanh chóng và hiệu quả hơn. Máy tính là một công cụ quan trọng trong công việc hàng ngày, do đó em quyết định tập trung vào việc phát triển một chương trình máy tính đơn giản nhưng linh hoạt bằng ngôn ngữ Java.

### 2. Mục tiêu.

Mục tiêu của đề án là xây dựng một ứng dụng máy tính đa chức năng với giao diện người dùng thân thiện. Chương trình sẽ hỗ trợ các phép toán cơ bản (+, -, \*, /), tính căn bậc hai, tính phần trăm và các chức năng khác.

### 3. Nội dung.

Chương trình máy tính sẽ có các tính năng sau:

- Các phép toán cơ bản (+, -, \*, /).
- Tính căn bậc hai của một số.
- Tính phần trăm của một số.
- Chuyển đổi số dương thành số âm và ngược lại.
- Giao diện người dùng đẹp và dễ sử dụng.
- Hiệu suất cao với thời gian đáp ứng nhanh chóng.

### 4. Phương pháp nghiên cứu.

Phương pháp nghiên cứu trong báo cáo tập trung vào việc áp dụng kiến thức lập trình Java đã học để xây dựng chương trình máy tính. Em đã tìm hiểu tài liệu từ các nguồn đáng tin cậy như Oracle Java Tutorials và trang web Baeldung. Công cụ phát triển em sử dụng là Netbeans, và ngôn ngữ lập trình chính là Java.

## **5. Kết quả.**

Sau quá trình nghiên cứu và phát triển, đã xây dựng thành công một chương trình máy tính đa chức năng sử dụng ngôn ngữ lập trình Java. Chương trình đáp ứng được các yêu cầu chính như hỗ trợ phép toán cơ bản và tính toán căn bậc hai, phần trăm. Giao diện được thiết kế đẹp và dễ sử dụng, mang lại trải nghiệm tính toán thuận lợi cho người dùng.

## MỞ ĐẦU

### 1. Lý do chọn đề tài.

Công nghệ thông tin (CNTT) ngày càng có vai trò quan trọng trong cuộc sống hằng ngày của chúng ta. Việc ứng dụng CNTT vào các lĩnh vực trong đời sống giúp công việc được tiến hành nhanh chóng và hiệu quả hơn.

Tôi chọn đề tài “Viết chương trình máy tính bằng Java “ là vì hiện nay máy tính là một công cụ quan trọng trong công việc hàng ngày của chúng ta. Đồ án này tập trung vào việc phát triển một chương trình máy tính đơn giản nhưng linh hoạt bằng ngôn ngữ lập trình Java để cung cấp một trải nghiệm tính toán thuận lợi cho người dùng.

### 2. Mục tiêu.

Mục tiêu của đồ án là xây dựng một ứng dụng máy tính đa chức năng với giao diện người dùng thân thiện, hỗ trợ các phép toán cơ bản (+, -, \*, /), tính căn bậc hai, phần trăm và các chức năng khác.

### 3. Nội dung.

Chương trình máy tính sẽ hỗ trợ:

- Các phép toán cơ bản (+, -, \*, /).
- Tính căn bậc hai của một số.
- Tính phần trăm của một số.
- Chuyển đổi số dương thành số âm và ngược lại.
- Giao diện người dùng đẹp và dễ sử dụng.
- Hiệu suất cao với thời gian đáp ứng nhanh chóng.

### 4. Phương pháp nghiên cứu.

Phương pháp nghiên cứu trong báo cáo sẽ tập trung vào việc áp dụng kiến thức lập trình Java đã học để xây dựng chương trình máy tính. Phương pháp này có thể bao gồm việc tìm hiểu tài liệu, tham khảo các nguồn tài liệu trực tuyến, thực hiện các bài tập và ví dụ để nắm vững kiến thức cơ bản và áp dụng vào việc viết chương trình.

- Các nguồn tài liệu tham khảo:

Oracle Java Tutorials:

Đây là tài liệu chính thức từ Oracle, nhà cung cấp ngôn ngữ Java. Trang web này cung cấp hướng dẫn từ cơ bản đến nâng cao về lập trình Java, bao gồm các ví dụ minh họa và giải thích rõ ràng. Bạn có thể truy cập tại: <https://docs.oracle.com/javase/tutorial/>

Baeldung:

Đây là một trang web chuyên về các bài viết hướng dẫn và ví dụ thực tế về lập trình Java. Nó cung cấp nhiều bài viết dễ hiểu và có thể áp dụng ngay vào thực tế. Bạn có thể truy cập tại: <https://www.baeldung.com/>

-Công cụ và ngôn ngữ lập trình:

+Công cụ phát triển :Netbeans

+Ngôn ngữ lập trình: Java

## **5. Đối tượng và phạm vi nghiên cứu.**

### **Đối tượng nghiên cứu**

Đối tượng của báo cáo là sinh viên hoặc những người quan tâm đến việc học và hiểu về lập trình máy tính bằng Java.

### **Phạm vi nghiên cứu**

Phạm vi nghiên cứu giới hạn trong việc xây dựng một chương trình máy tính đơn giản sử dụng Java, không đi sâu vào các chức năng phức tạp hoặc ứng dụng đặc thù.

## CHƯƠNG 1: TỔNG QUAN

### 1.1 Cơ sở lý thuyết:

Java là một ngôn ngữ lập trình được Sun Microsystems giới thiệu vào tháng 6 năm 1995. Từ đó, nó đã trở thành một công cụ lập trình của các lập trình viên chuyên nghiệp. Java được xây dựng trên nền tảng của C và C++, do vậy nó sử dụng các cú pháp của C và các đặc trưng hướng đối tượng của C++.

Vào năm 1991, một nhóm các kỹ sư của Sun Microsystems có ý định thiết kế một ngôn ngữ lập trình để điều khiển các thiết bị điện tử như tivi, máy giặt, lò nướng,... Mặc dù C và C++ có khả năng làm việc này nhưng trình biên dịch lại phụ thuộc vào từng loại CPU. Trình biên dịch thường phải tốn nhiều thời gian để xây dựng nên rất đắt, vì vậy để mỗi loại CPU có một trình biên dịch riêng là rất tốn kém. Do đó nhu cầu thực tế đòi hỏi một ngôn ngữ chạy nhanh, gọn, hiệu quả và độc lập thiết bị tức là có thể chạy trên nhiều loại CPU khác nhau, dưới các môi trường khác nhau. “Oak” đã ra đời và vào năm 1995 được đổi tên thành Java. Mặc dù mục tiêu ban đầu không phải cho Internet nhưng do đặc trưng không phụ thuộc thiết bị nên Java đã trở thành ngôn ngữ lập trình cho Internet.

### 1.2 Giới thiệu.

Trong Chương 1, sẽ giới thiệu tổng quan về việc viết chương trình máy tính bằng Java. Mục tiêu của chương này là đặt ra khung cảnh và lý do vì sao việc nghiên cứu và giải quyết vấn đề này là quan trọng. Dưới đây là các phần có thể bao gồm:

1. Giới thiệu về chương trình máy tính và vai trò của Java: Trình bày một cách tổng quan về chương trình máy tính và lý do tại sao việc viết chương trình máy tính là một kỹ năng quan trọng. Giới thiệu về ngôn ngữ lập trình Java và vì sao nó được sử dụng phổ biến trong việc phát triển ứng dụng máy tính.
2. Mục tiêu của báo cáo: Đưa ra mục tiêu cụ thể của báo cáo, nhằm cung cấp cho độc giả một cái nhìn tổng quan về những gì sẽ được trình bày và đạt được trong báo cáo này.
3. Phạm vi nghiên cứu: Mô tả phạm vi của báo cáo, xác định rõ những gì sẽ được bao gồm và những gì sẽ không được bao gồm trong nội dung.
4. Lợi ích và ứng dụng: Trình bày lợi ích và ứng dụng của việc viết chương trình máy tính bằng Java. Nêu rõ tầm quan trọng và tiềm năng của việc áp dụng kiến thức về lập trình Java trong các dự án thực tế và trong sự phát triển cá nhân.

5. Cấu trúc báo cáo: Đưa ra một tóm tắt ngắn gọn về cấu trúc của báo cáo, liệt kê các chương và phần chính sẽ được trình bày trong tài liệu.

## CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

### 2.1 Cơ sở lý thuyết về Java.

#### 2.1.1 Nội dung.

Giới thiệu về ngôn ngữ lập trình Java và các đặc điểm cơ bản của nó:

1. Độc lập nền tảng: Java được thiết kế để hoạt động trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn. Điều này có nghĩa là một chương trình Java có thể chạy trên các hệ điều hành khác nhau như Windows, macOS và Linux mà không cần sửa đổi.
2. Hướng đối tượng: Java được xây dựng dựa trên mô hình hướng đối tượng, cho phép phát triển ứng dụng theo cách tổ chức dựa trên đối tượng. Điều này giúp tạo ra mã nguồn dễ hiểu, dễ bảo trì và dễ mở rộng.
3. Bảo mật: Java có môi trường chạy độc lập được gọi là "Java Virtual Machine" (JVM), giúp cung cấp một môi trường đáng tin cậy cho việc chạy chương trình Java. JVM kiểm soát quyền truy cập vào tài nguyên hệ thống và đảm bảo an toàn trong việc thực thi mã.
4. Thu gom rác (Garbage Collection): Java tự động quản lý bộ nhớ thông qua cơ chế thu gom rác. Điều này giúp giảm công việc của lập trình viên trong việc quản lý bộ nhớ và giúp tránh các vấn đề như rò rỉ bộ nhớ.
5. Thư viện phong phú: Java đi kèm với một bộ thư viện mạnh mẽ và phong phú, cung cấp các lớp và giao diện để xử lý các tác vụ thông thường như nhập/xuất dữ liệu, xử lý chuỗi, đồ họa, mạng, quản lý luồng và nhiều hơn nữa. Điều này giúp giảm thời gian và công sức của lập trình viên trong việc phát triển ứng dụng.
6. Cú pháp đơn giản: Cú pháp của Java dễ đọc và dễ hiểu, với một số nguyên tắc rõ ràng như việc sử dụng dấu chấm phẩy để kết thúc câu lệnh, đặt tên biến theo quy tắc CamelCase, v.v.

#### 2.1.2 Nội dung.

Các khái niệm cơ bản trong Java như biến, kiểu dữ liệu, lệnh điều khiển và cấu trúc điều kiện:

1. Biến: Biến là một vị trí trong bộ nhớ được sử dụng để lưu trữ dữ liệu. Mỗi biến có một tên và một kiểu dữ liệu cụ thể. Để sử dụng một biến, bạn cần khai báo nó trước và



gán giá trị cho nó. Ví dụ: `int age = 25;` khai báo biến `age` kiểu số nguyên và gán giá trị là 25.

2. Kiểu dữ liệu: Kiểu dữ liệu xác định loại dữ liệu mà một biến có thể chứa. Các kiểu dữ liệu cơ bản trong Java bao gồm:

- Kiểu số nguyên: `'int'`, `'byte'`, `'short'`, `'long'`.
- Kiểu số thực: `'float'`, `'double'`.
- Kiểu ký tự: `'char'`.
- Kiểu boolean: `'boolean'`.
- Kiểu chuỗi: `'String'` (đây là một kiểu dữ liệu đặc biệt trong Java để đại diện cho chuỗi ký tự).

3. Lệnh điều khiển: Lệnh điều khiển được sử dụng để kiểm soát luồng thực thi của chương trình. Các lệnh điều khiển thông dụng trong Java bao gồm:

- Lệnh điều kiện: `'if'`, `'else if'`, `'else'` để thực hiện các hành động khác nhau dựa trên điều kiện.
- Lệnh lặp: `'for'`, `'while'`, `'do-while'` để lặp lại một khối mã nhiều lần.
- Lệnh nhảy: `'break'`, `'continue'` để nhảy qua hoặc thoát khỏi một vòng lặp hoặc khối mã.

4. Cấu trúc điều kiện: Cấu trúc điều kiện cho phép thực hiện các hành động khác nhau dựa trên điều kiện. Trong Java, cấu trúc điều kiện thông thường sử dụng là `'if-else'`, cho phép kiểm tra một điều kiện và thực hiện một hành động nếu điều kiện đúng và một hành động khác nếu điều kiện sai.

## 2.2 Lý luận về viết chương trình máy tính:

### 2.2.1 Nội dung

Tổng quan về việc viết chương trình máy tính và vai trò của nó trong thực hiện các phép tính và chức năng.

Viết chương trình máy tính là quá trình tạo ra một tập hợp các câu lệnh và thuật toán được viết bằng một ngôn ngữ lập trình cụ thể để thực hiện các phép tính và chức năng mong muốn trên máy tính.

Vai trò của chương trình máy tính là chuyển đổi các yêu cầu, vấn đề hoặc nhiệm vụ của người dùng thành mã lệnh được máy tính hiểu và thực thi. Chương trình máy tính cung cấp một bộ các hướng dẫn cho máy tính để thực hiện các tác vụ cụ thể, như tính toán số học, xử lý dữ liệu, quản lý thông tin, tương tác với người dùng và thực hiện các chức năng phức tạp khác.

Việc viết chương trình máy tính có thể bao gồm các bước sau:

1. **Xác định yêu cầu:** Đầu tiên, bạn cần hiểu rõ yêu cầu và chức năng mà chương trình cần thực hiện. Điều này đòi hỏi bạn phải nắm vững vấn đề và các yêu cầu của người dùng.
2. **Thiết kế:** Sau khi hiểu yêu cầu, bạn cần thiết kế cấu trúc và thuật toán của chương trình. Bước này bao gồm việc xác định các khối mã lệnh, cấu trúc dữ liệu và quy trình logic để thực hiện yêu cầu.
3. **Viết mã:** Tiếp theo, bạn sẽ viết mã lệnh bằng một ngôn ngữ lập trình cụ thể, như Java, C++, Python, v.v. Bạn sẽ sử dụng các cú pháp và quy tắc của ngôn ngữ đó để triển khai thuật toán và logic đã thiết kế.
4. **Biên dịch/phiên dịch:** Sau khi viết mã, bạn cần biên dịch hoặc phiên dịch mã lệnh thành mã máy có thể được hiểu và thực thi bởi máy tính. Quá trình này tạo ra một tệp thực thi (như tệp .exe trong Windows) hoặc một mã trung gian (như byte code trong Java).
5. **Kiểm tra và sửa lỗi:** Sau khi có tệp thực thi, bạn cần kiểm tra chương trình để đảm bảo rằng nó hoạt động đúng và đáp ứng các yêu cầu đã đề ra. Nếu phát hiện lỗi, bạn sẽ sửa chúng và kiểm tra lại cho đến khi chương trình hoạt động như mong muốn.
6. **Triển khai:** Cuối cùng, bạn triển khai chương trình bằng cách chạy tệp thực thi hoặc mã trung gian trên máy tính hoặc môi trường mà chương trình được thiết kế để chạy trên.

### **2.2.2 Nội dung.**

#### **Các yêu cầu và đặc điểm của một chương trình máy tính hiệu quả.**

Một chương trình máy tính hiệu quả cần đáp ứng một số yêu cầu và có những đặc điểm cụ thể để thực hiện công việc một cách hiệu quả. Dưới đây là một số yêu cầu và đặc điểm quan trọng của một chương trình máy tính hiệu quả:

1. Chính xác: Chương trình máy tính phải đáp ứng đúng các yêu cầu và chức năng đã định. Nó phải tính toán và xử lý dữ liệu một cách chính xác để đảm bảo kết quả đúng đắn.
2. Hiệu suất: Chương trình máy tính hiệu quả cần chạy một cách nhanh chóng và hiệu quả. Nó phải sử dụng tài nguyên máy tính (bộ vi xử lý, bộ nhớ, ổ đĩa, v.v.) một cách tối ưu để đạt được hiệu suất tốt nhất.
3. Tiêu thụ tài nguyên: Chương trình máy tính hiệu quả cần tiêu thụ ít tài nguyên hệ thống như bộ nhớ, CPU và băng thông mạng. Điều này giúp tránh quá tải hệ thống và tăng khả năng mở rộng và độ ổn định của chương trình.
4. Dễ bảo trì: Chương trình máy tính hiệu quả cần được thiết kế và viết một cách rõ ràng, dễ đọc và dễ bảo trì. Mã lệnh nên tuân thủ các nguyên tắc lập trình tốt như modular, tái sử dụng mã, và tuân thủ các quy ước đặt tên. Điều này giúp cho việc sửa lỗi, cải tiến và mở rộng chương trình dễ dàng hơn.
5. Bảo mật: Chương trình máy tính hiệu quả cần có các biện pháp bảo mật để bảo vệ dữ liệu và ngăn chặn việc truy cập trái phép. Điều này đảm bảo rằng thông tin quan trọng được bảo vệ và người dùng có thể tin tưởng vào tính riêng tư và an toàn của dữ liệu.
6. Linh hoạt và mở rộng: Một chương trình máy tính hiệu quả nên được thiết kế để dễ dàng mở rộng và thích ứng với các yêu cầu và thay đổi trong tương lai. Kiến trúc và thiết kế phần mềm nên có khả năng linh hoạt để thêm mới chức năng và thay đổi môi trường một cách dễ dàng.
7. Giao diện người dùng thân thiện: Nếu chương trình có giao diện người dùng, nó nên được thiết kế để cung cấp một trải nghiệm người dùng thân thiện, dễ sử dụng và trực quan. Giao diện nên đáp ứng các nguyên tắc thiết kế giao diện người dùng tốt để tạo ra một trải nghiệm tốt cho người dùng.

## **2.3 Giả thuyết khoa học.**

### **2.3.1 Nội dung.**

Xác định các giả thiết khoa học được áp dụng trong việc viết chương trình máy tính bằng Java, ví dụ như giả thiết về tính đúng đắn của các phép toán cơ bản, giả thiết về tính ổn định và độ tin cậy của chương trình.

## 2.4 Phương pháp nghiên cứu khoa học.

### 2.4.1 Nội dung

Mô tả chi tiết các phương pháp nghiên cứu đã được sử dụng trong quá trình viết chương trình máy tính bằng Java.

Trong quá trình viết chương trình máy tính bằng Java, có nhiều phương pháp nghiên cứu được sử dụng để nắm bắt kiến thức và tìm hiểu về các khía cạnh kỹ thuật và chức năng của ngôn ngữ Java, cũng như các thành phần và công nghệ liên quan. Dưới đây là một số phương pháp nghiên cứu quan trọng trong việc viết chương trình máy tính bằng Java:

1. Tài liệu học: Đầu tiên, nghiên cứu tài liệu học là một phương pháp quan trọng để nắm bắt cú pháp, quy tắc và nguyên tắc của ngôn ngữ Java. Điều này bao gồm việc đọc sách, bài viết, tài liệu hướng dẫn và tài liệu chính thức của Java như Java API Documentation.
2. Học trực tuyến: Internet cung cấp một nguồn tài nguyên phong phú cho việc nghiên cứu Java. Các trang web, blog, diễn đàn và video hướng dẫn trực tuyến có thể giúp bạn hiểu rõ hơn về các khái niệm, kỹ thuật và thực hành của Java. Các trang web như Oracle Java Tutorials, Baeldung, Java Code Geeks và JavaWorld cung cấp nội dung phong phú về Java.
3. Thực hành và ví dụ: Một phương pháp quan trọng để nghiên cứu Java là thực hành và tạo các ví dụ. Việc viết mã lệnh thực tế, xây dựng các ứng dụng nhỏ và thực hiện các bài tập lập trình giúp củng cố kiến thức và hiểu rõ hơn về cách sử dụng các tính năng và thư viện của Java.
4. Dự án thực tế: Tham gia vào các dự án thực tế sử dụng Java là một cách tuyệt vời để nghiên cứu và áp dụng kiến thức. Tham gia vào các dự án mã nguồn mở hoặc tạo ra các dự án cá nhân giúp bạn rèn kỹ năng lập trình Java, làm việc với các thư viện và công cụ khác nhau, và hiểu rõ hơn về quy trình phát triển phần mềm.
5. Cộng đồng và diễn đàn: Tham gia vào cộng đồng Java, các diễn đàn trực tuyến và nhóm trò chuyện giúp bạn giao tiếp với những người khác có cùng sở thích và trao đổi kiến thức. Các diễn đàn như Stack Overflow, Reddit, JavaRanch và Oracle Community là những nơi tốt để đặt câu hỏi, tìm kiếm giải đáp và học hỏi từ những người khác.

6. Đọc mã nguồn và thư viện: Nghiên cứu mã nguồn mở và các thư viện Java phổ biến là một phương pháp hữu ích để hiểu cách sử dụng các tính năng và triển khai các giải pháp trong Java. Bằng cách đọc và xem xét mã nguồn mở, bạn có thể tìm hiểu cách các chương trình được xây dựng và tương tác với các thành phần khác nhau.

7. Thực tế và thảo luận: Cuối cùng, thực hiện các bài toán thực tiếp và thảo luận với các đồng nghiệp, giảng viên hoặc người có kinh nghiệm trong lĩnh vực Java là một phương pháp quan trọng trong việc nghiên cứu. Thảo luận với người khác giúp bạn có thể tìm hiểu những góc nhìn khác nhau, giải quyết các vấn đề khó khăn và chia sẻ kinh nghiệm với nhau.

#### **2.4.2 Nội dung.**

Tham khảo và học từ các nguồn tài liệu trên Youtube và sử dụng công cụ phát triển NetBeans để thực hiện viết chương trình.

1. Tham khảo và học từ các nguồn tài liệu trên YouTube:

- Tìm kiếm các kênh YouTube chuyên về Java programming: Có nhiều kênh YouTube cung cấp nội dung học Java đa dạng và phong phú. Bạn có thể tìm kiếm các kênh như "TheNewBoston", "Derek Banas", "ProgrammingKnowledge", "Java Brains", "Cave of Programming", và nhiều kênh khác.

- Xem các video hướng dẫn và bài giảng: Các video hướng dẫn trên YouTube cung cấp cách giải thích và hướng dẫn cụ thể về các khái niệm, cú pháp và ứng dụng của Java. Bạn có thể xem các video hướng dẫn từ cơ bản đến nâng cao, ví dụ như cú pháp Java, lập trình hướng đối tượng, xử lý ngoại lệ, JDBC, Swing, Servlet và JSP, Spring Framework, và nhiều chủ đề khác.

2. Sử dụng công cụ phát triển NetBeans:

- Tải và cài đặt NetBeans: NetBeans là một môi trường phát triển tích hợp (IDE) được sử dụng phổ biến trong việc phát triển ứng dụng Java. Bạn có thể tải NetBeans từ trang web chính thức của nó và cài đặt trên máy tính của bạn.

- Tạo dự án Java: Sau khi cài đặt NetBeans, bạn có thể tạo dự án Java mới bằng cách chọn "File" -> "New Project" và chọn "Java" trong danh sách các loại dự án. Sau đó, bạn có thể đặt tên dự án và chọn các tùy chọn cần thiết.

- **Viết mã Java:** Trong NetBeans, bạn có thể tạo các lớp Java mới, tạo các phương thức và biến, và viết mã lệnh Java trong các tệp nguồn. IDE cung cấp tính năng nhấn mạnh cú pháp, gợi ý mã, và kiểm tra lỗi cú pháp, giúp bạn viết mã Java một cách dễ dàng và hiệu quả.

- **Biên dịch và chạy chương trình:** NetBeans cung cấp các công cụ để biên dịch và chạy chương trình Java của bạn. Bạn có thể chọn "Run" hoặc nhấn phím tắt để biên dịch và chạy chương trình của bạn trong môi trường NetBeans.

## **2.5 Các công trình nghiên cứu liên quan.**

Dưới đây là một số công trình nghiên cứu liên quan đến viết chương trình máy tính bằng Java và các phương pháp, kỹ thuật đã được sử dụng trong những công trình đó:

1. "An Empirical Study on the Impact of Object-Oriented Design Patterns on Software Development Productivity" (D. Shukla, R. Mall)

- **Phương pháp:** Nghiên cứu thực nghiệm.

- **Mô tả:** Nghiên cứu này tìm hiểu tác động của các mẫu thiết kế hướng đối tượng đến năng suất phát triển phần mềm. Java được sử dụng để triển khai các chương trình mẫu và đo lường hiệu suất.

2. "A Comparative Study on Performance of Java Web Frameworks" (A. Kumar, A. Dhir, S. K. Jain)

- **Phương pháp:** Nghiên cứu so sánh.

- **Mô tả:** Nghiên cứu này so sánh hiệu suất của các framework phát triển web Java phổ biến như Spring, JavaServer Faces (JSF) và Struts. Các phương pháp kiểm tra và đánh giá hiệu suất đã được sử dụng để so sánh các framework.

3. "Automated Unit Testing of Java Programs Using Genetic Algorithms" (A. Arcuri, L. Briand)

- **Phương pháp:** Kỹ thuật di truyền.

- **Mô tả:** Nghiên cứu này giới thiệu một phương pháp tự động kiểm thử đơn vị cho chương trình Java bằng cách sử dụng thuật toán di truyền. Các test case được tạo tự động và tối ưu bằng cách sử dụng kỹ thuật di truyền.

4. "Mining Java Class API Usage Patterns from Source Code" (W. Zhao, J. Zhang, H. Mei)

- Phương pháp: Khai thác dữ liệu.

- Mô tả: Nghiên cứu này khai thác các mẫu sử dụng API của các lớp Java từ mã nguồn để tìm hiểu cách các API được sử dụng trong các dự án phần mềm. Các kỹ thuật khai thác dữ liệu và phân tích ngôn ngữ tự nhiên đã được áp dụng trong nghiên cứu này.

5. "A Comparative Study of Java IDEs" (A. B. Cozzi, M. J. Kurkovsky)

- Phương pháp: Nghiên cứu so sánh.

- Mô tả: Nghiên cứu này so sánh các môi trường phát triển tích hợp Java (IDE) phổ biến như Eclipse, NetBeans và IntelliJ IDEA. Các tiêu chí như hiệu suất, tính năng và trải nghiệm người dùng đã được sử dụng để so sánh các IDE.

## CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

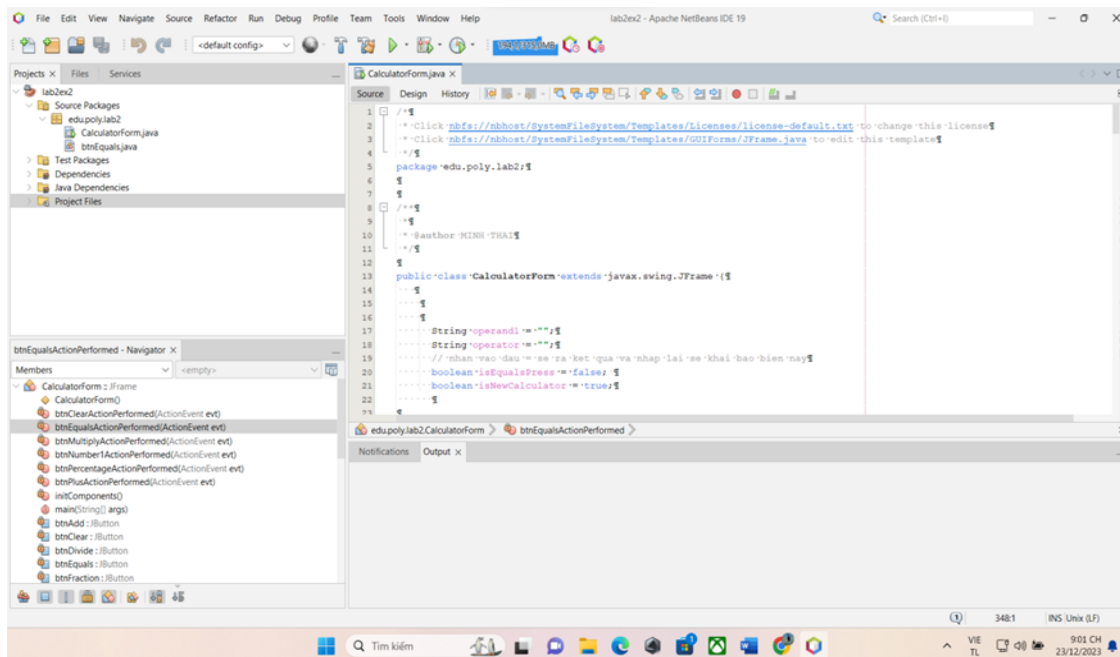
### 3.1 Mô tả ứng dụng.

NetBeans là một môi trường phát triển tích hợp (IDE) cho Java. NetBeans cho phép các ứng dụng được phát triển từ một tập hợp các thành phần phần mềm được gọi là modules. NetBeans chạy trên Windows, macOS, Linux và Solaris. Ngoài việc phát triển Java, nó còn có các phần mở rộng cho các ngôn ngữ khác như PHP, C, C++, HTML5,[4] và JavaScript. Các ứng dụng dựa trên NetBeans, bao gồm NetBeans IDE, có thể được mở rộng bởi các nhà phát triển bên thứ ba.



Hình 3-1. Ứng dụng NetBeans

### 3.2 Giao diện Netbeans.

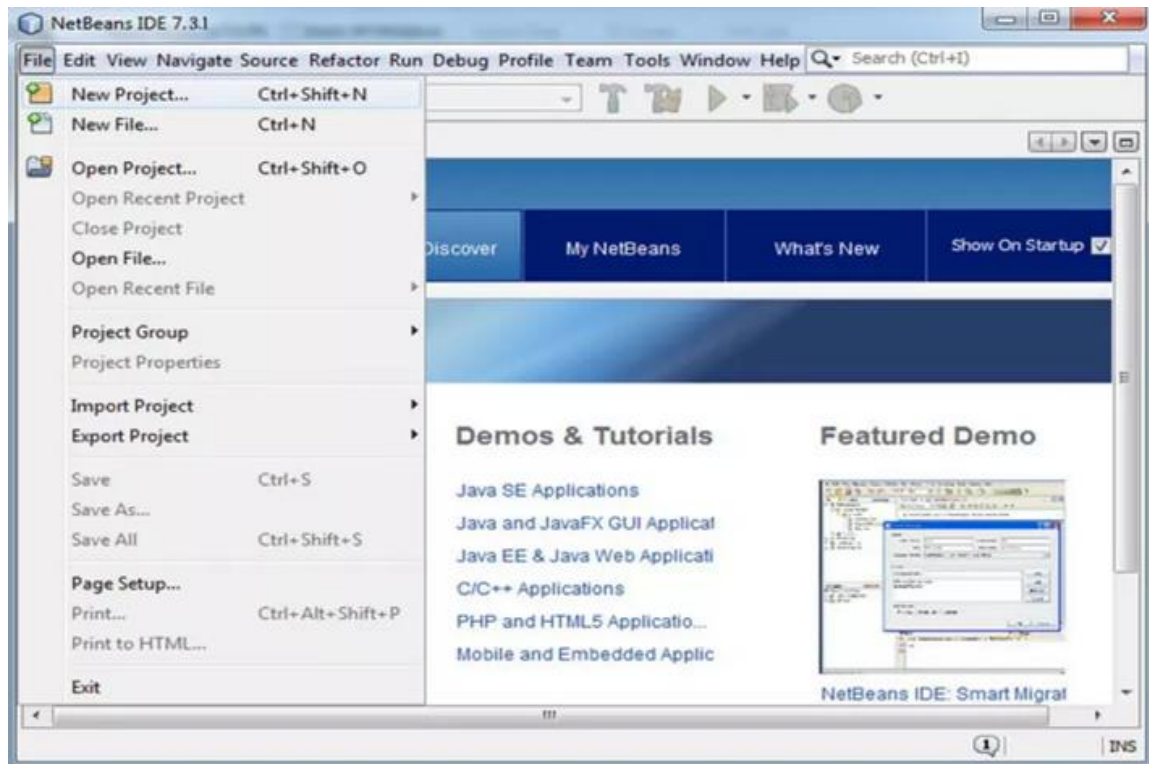


Hình 3-2. Giao diện Netbeans



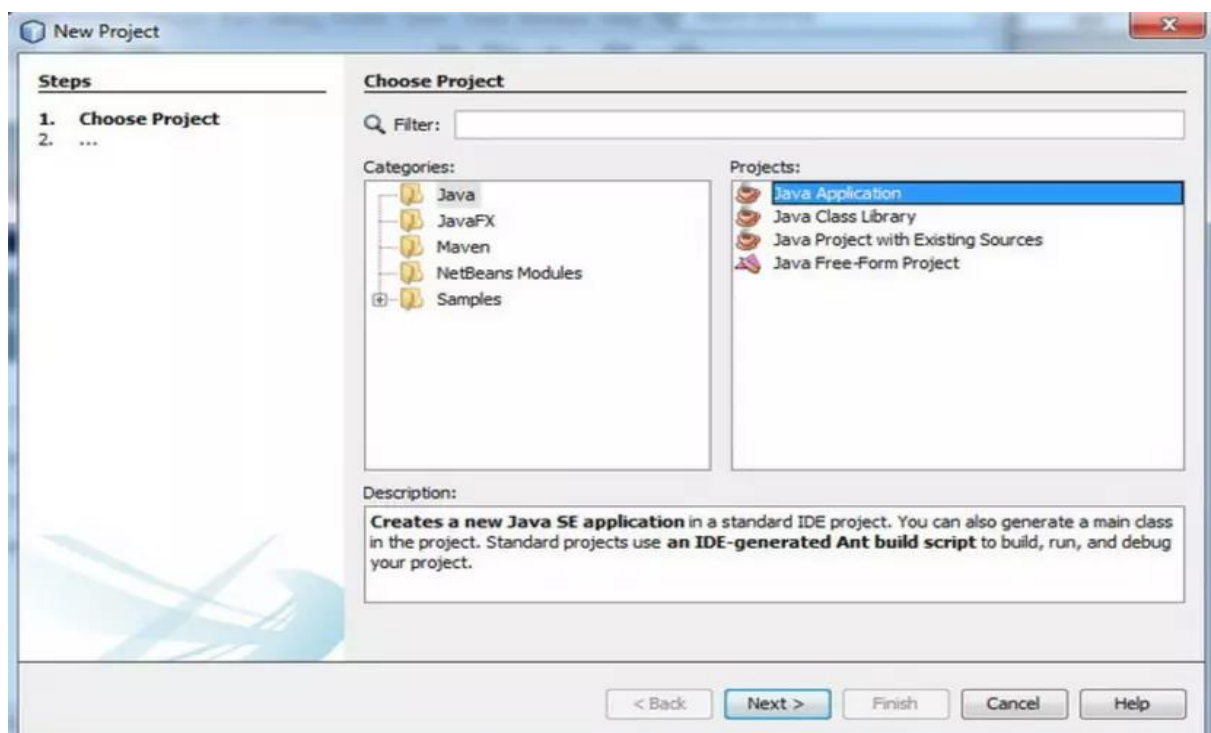
## Triển khai dự án trên Netbeans:

- Chọn "File" -> "New Project"



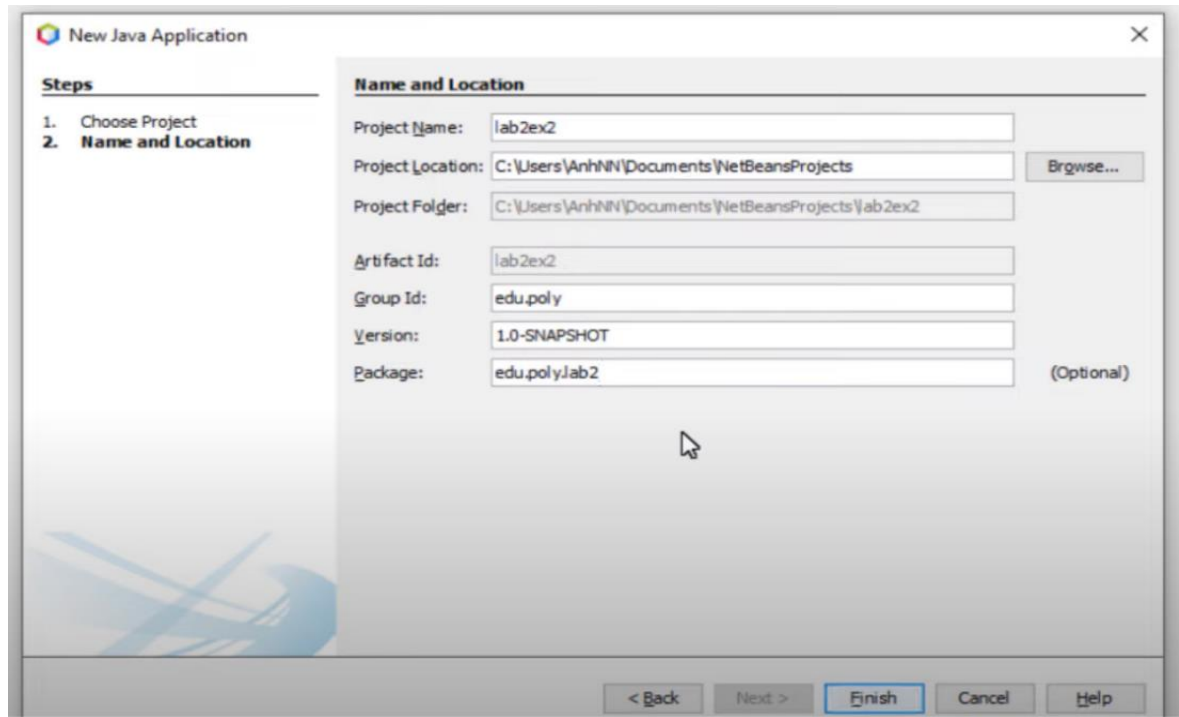
Hình 3-3. Chọn "File" -> "New Project"

- Chọn "Java" -> "Java Application" từ danh sách các mẫu dự án.



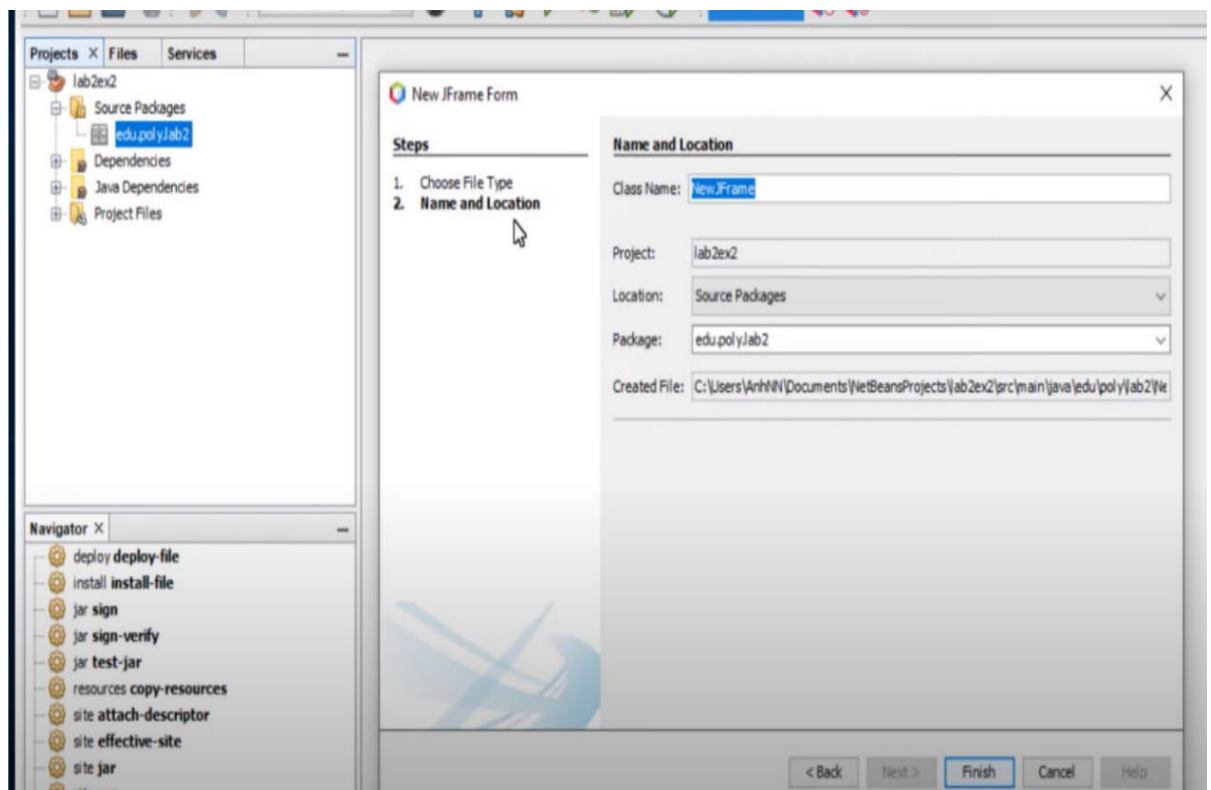
Hình 3-4. Chọn "Java" -> "Java Application"

- **Đặt tên cho dự án và bấm Finish.**



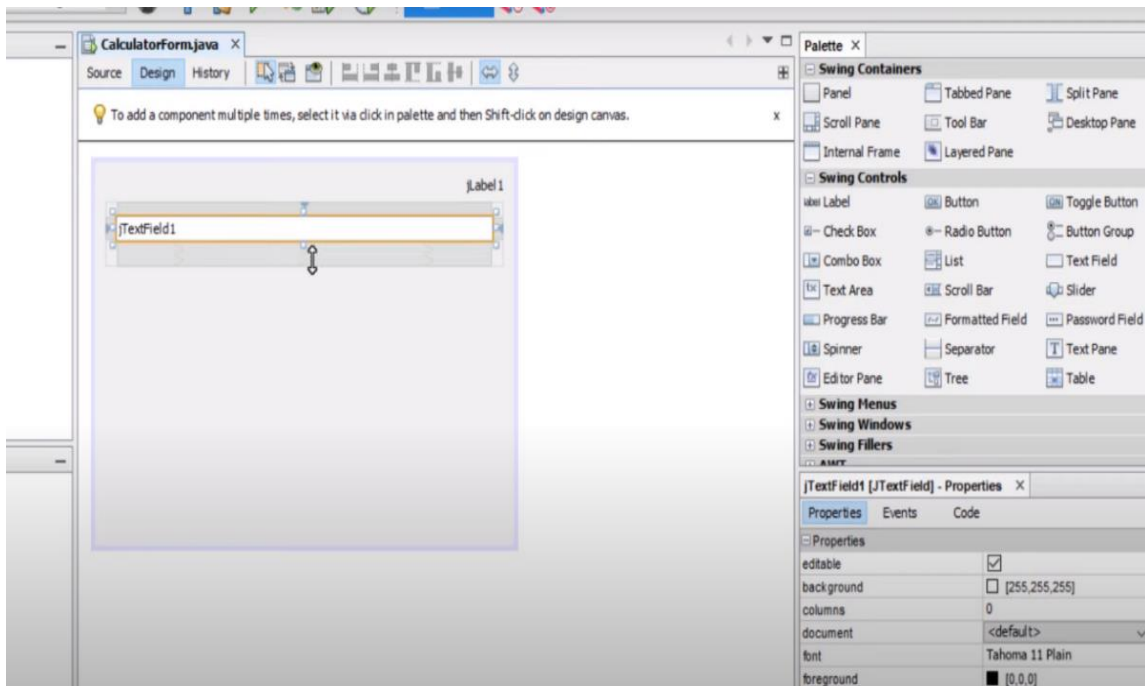
Hình 3-5. Đặt tên cho dự án

- **Sau đó ấn chuột phải vào Edu.poly.lab2->New->JFrame Form và đặt tên.**



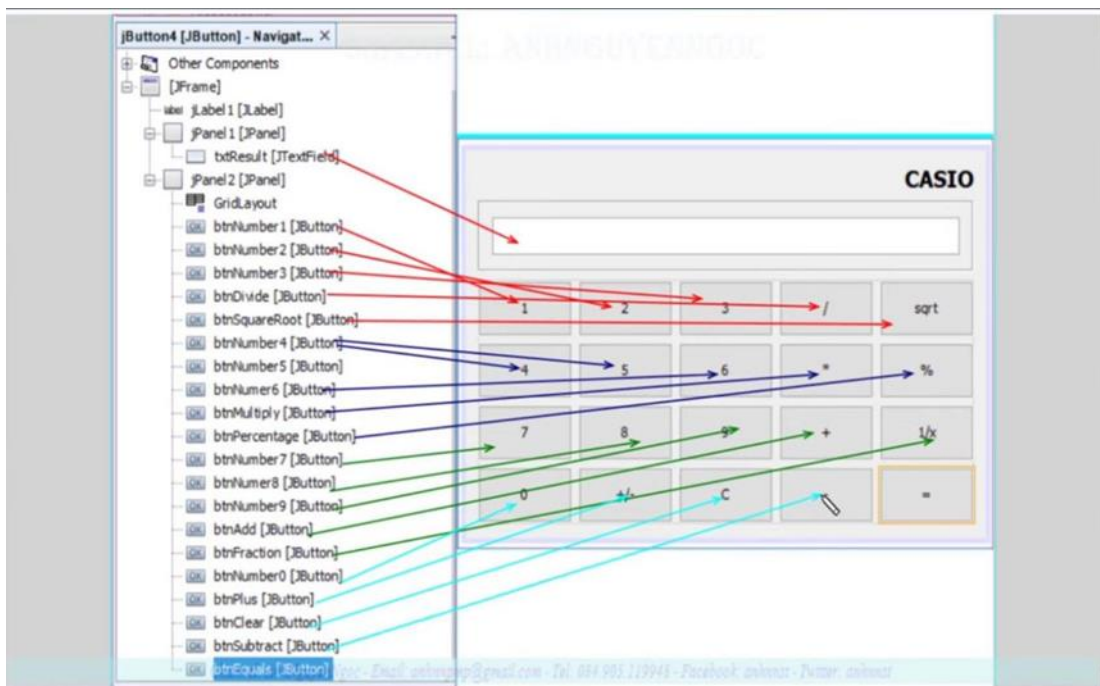
Hình 3-6. Ấn chuột phải vào Edu.poly.lab2->New->JFrame Form và đặt tên.

- Đến phần thiết kế, ta rê các thành phần như Label, Button, Panel, Text Field,... để thiết kế giao diện cho máy tính.



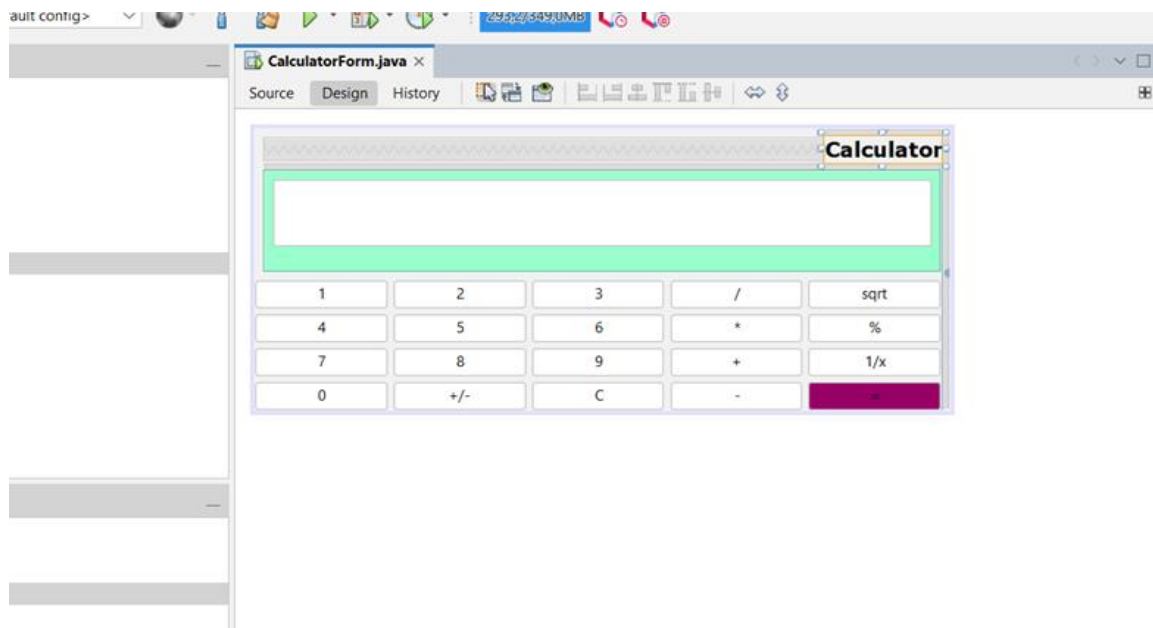
Hình 3-7. Các thành phần như Label, Button, Panel, Text Field,... để thiết kế giao diện.

- Tiếp đó, đổi tên cho các Button thành các số và các phép tính như một giao diện máy tính.



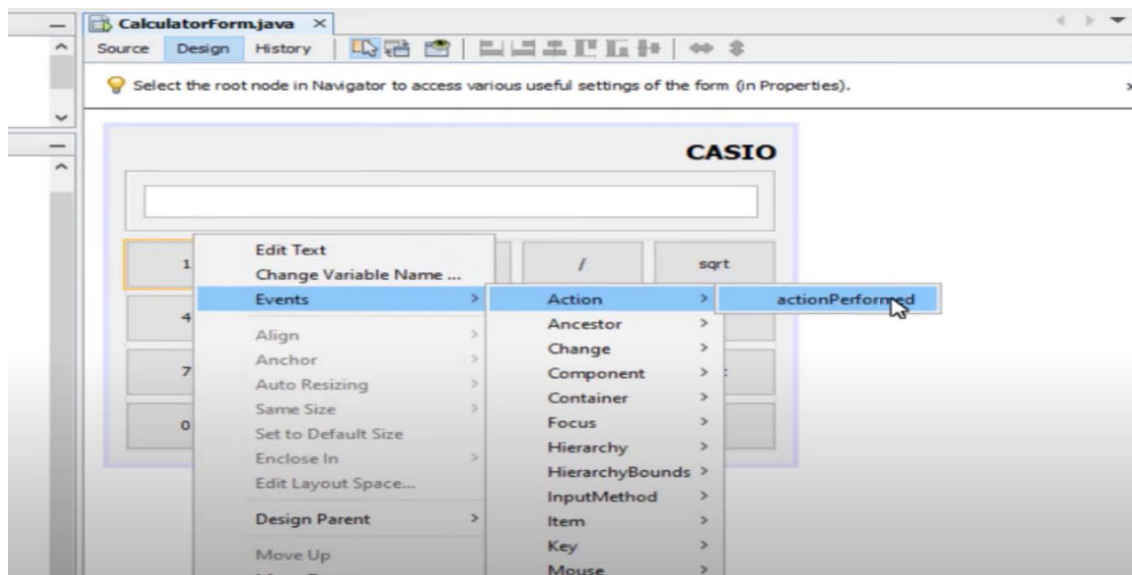
Hình 3-8. Đổi tên cho các Button thành các số và các phép tính

- Sau khi thiết kế xong phần giao diện, ta được kết quả như hình:



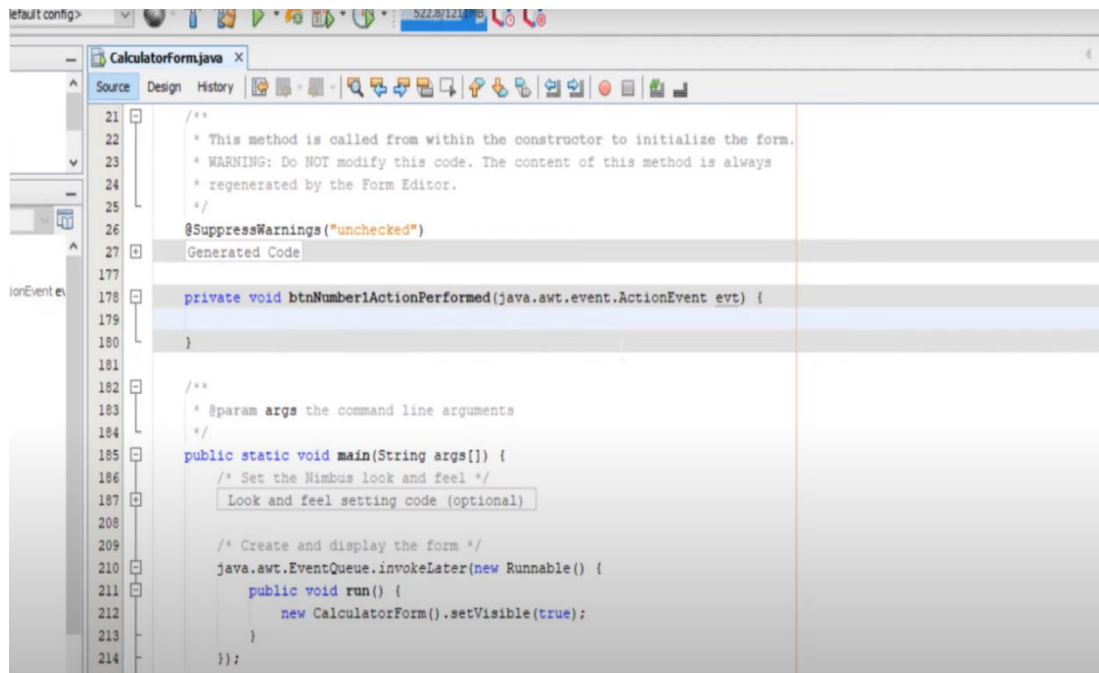
Hình 3-9. Kết quả sau khi thiết kế

- Sau đó bắt đầu viết các chức năng cho các nút số.  
Ta ấn vào nút Ctrl sau đó chọn từ số 0->9 rồi chọn Events->Action->actionPerformed.



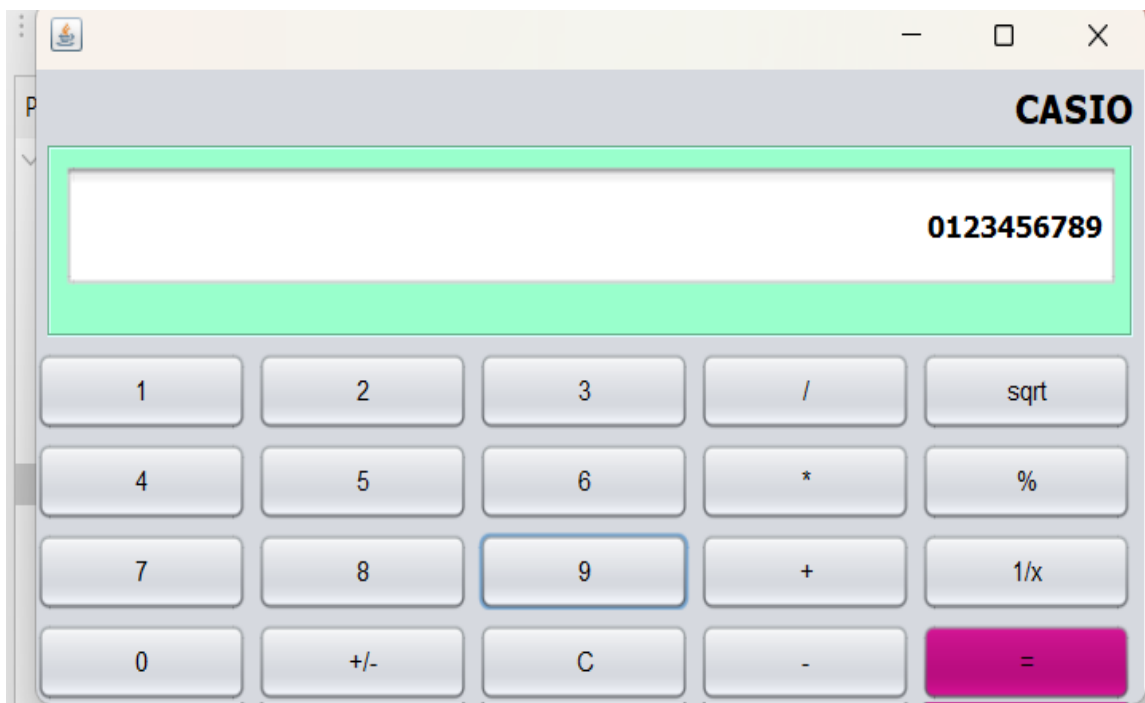
Hình 3-10. Ấn vào nút Ctrl sau đó chọn từ số 0->9 rồi chọn Events->Action->actionPerformed.

- Sau đó giao diện sẽ chuyển qua phần Source để viết chức năng lên các số đã chọn.



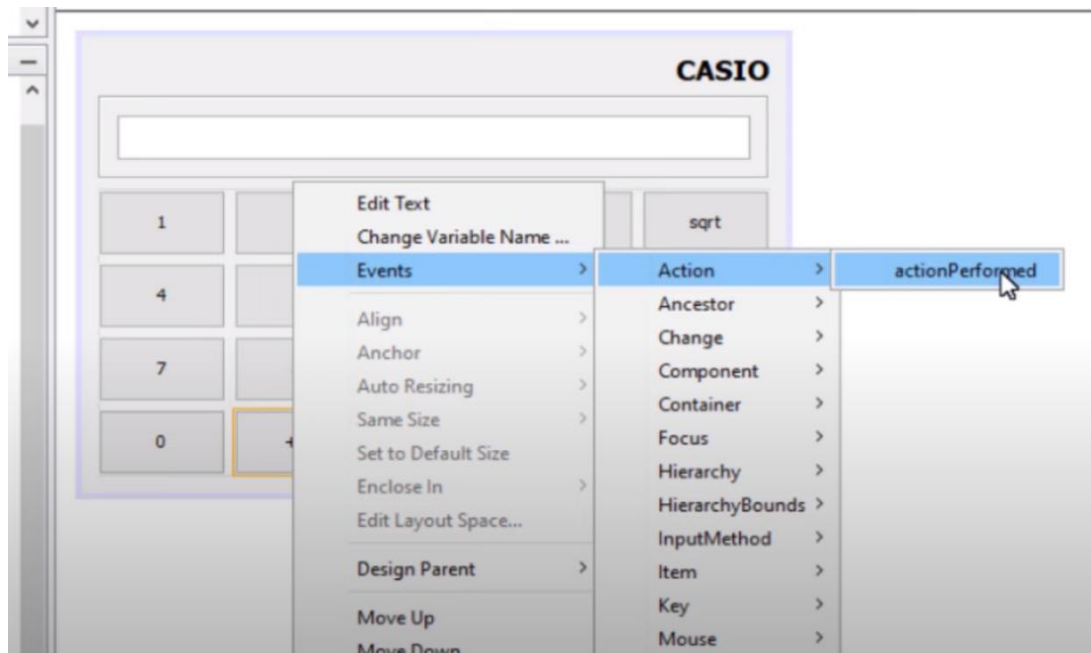
Hình 3-11. Giao diện sẽ chuyển qua phần Source để viết chức năng lên các số đã chọn.

- Sau khi viết sự kiện cho các nút, ấn vào Run và chọn Run File để kiểm tra kết quả sự kiện.



Hình 3-12. Ấn vào Run và chọn Run File để kiểm tra kết quả sự kiện.

- Tiếp đó,viết chức năng cho nút đảo dấu, ấn vào nút đảo dấu chọn Events->Action->actionPerformed .



Hình 3-13. Viết chức năng cho nút đảo dấu, ấn vào nút đảo dấu chọn Events->Action->actionPerformed .

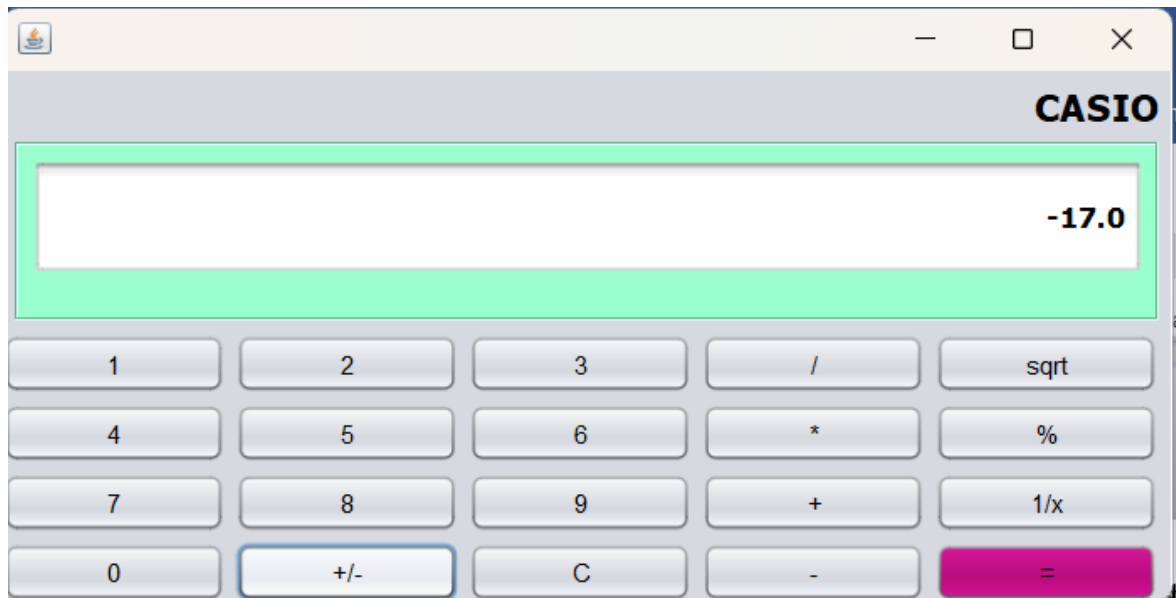
- Sau đó viết sự kiện cho nút đảo dấu .

```
private void btnPlusActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        double value = Double.parseDouble(txtResult.getText());  
        value = - value;  
  
        txtResult.setText("" + value);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Hình 3-14. Viết sự kiện cho nút đảo dấu .

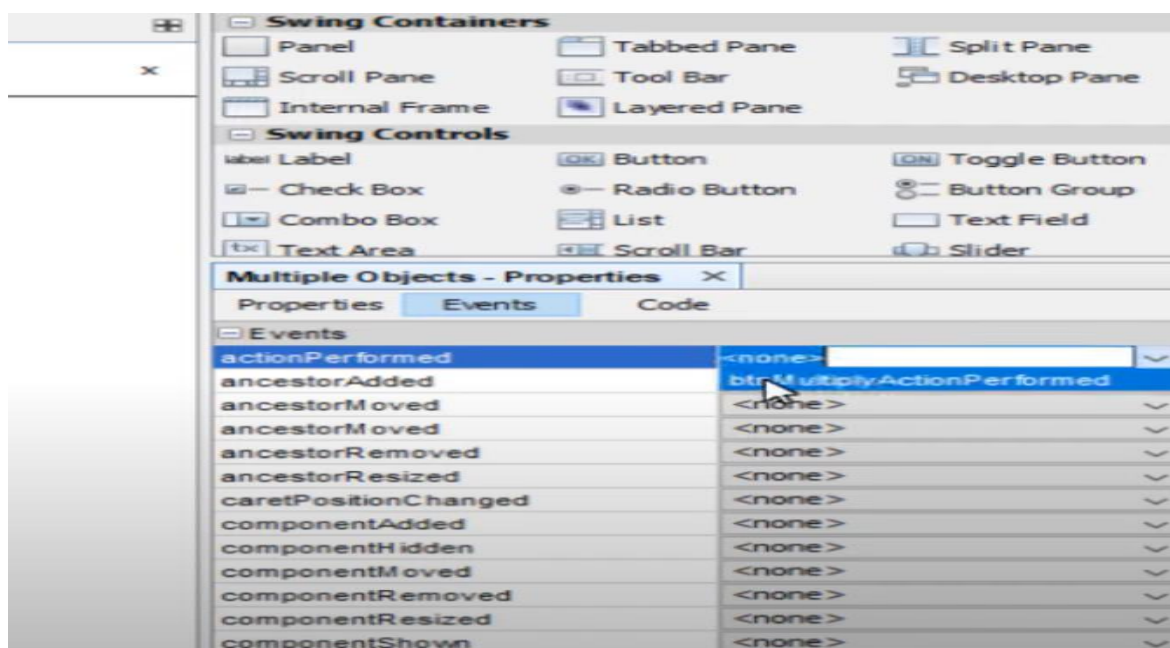


- Sau khi viết xong, nhấn vào Run và chọn Run File để kiểm tra.



Hình 3-15. Nhấn vào Run và chọn Run File để kiểm tra.

- Tiếp đó, nhấn nút Ctrl và chọn các nút '+ - \* /', nhìn bên góc phải màn hình, bấm vào Events->actionPerformed và chọn tên của phương thức xử lý sự kiện là btnMultiplyActionPerformed.



Hình 3-16. Nhấn nút Ctrl và chọn các nút '+ - \* /', nhìn bên góc phải màn hình, bấm vào Events->actionPerformed và chọn tên của phương thức xử lý sự kiện là btnMultiplyActionPerformed.

- Để đưa các đoạn mã xử lý khi người dùng kích vào các nút như + - \* / thì phải khai báo.

```
public class CalculatorForm extends javax.swing.JFrame {  
    String operand1 = "";  
    String operator = "";  
}
```

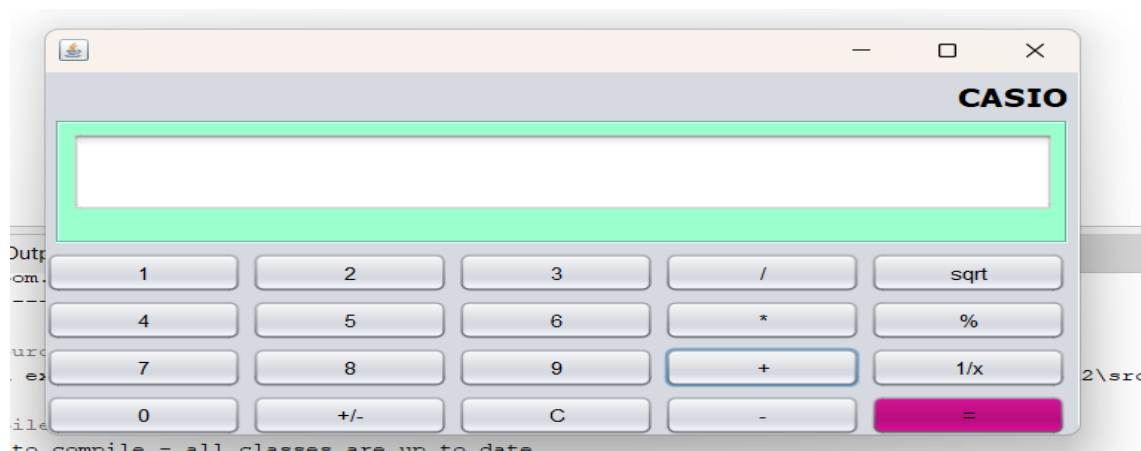
Hình 3-17. Vác đoạn mã xử lý khi người dùng kích vào các nút như + - \* / thì phải khai báo.

- Tiếp đó về lại phần `btnMultiplyActionPerformed` để viết tiếp sự kiện.

```
private void btnMultiplyActionPerformed(java.awt.event.ActionEvent evt) {  
    String cmd = evt.getActionCommand();  
  
    operator = cmd;  
    operand1 = txtResult.getText();  
  
    txtResult.setText("");  
}
```

Hình 3-18. Về lại phần `btnMultiplyActionPerformed` để viết tiếp sự kiện.

- Sau đó ấn vào Run chọn Run File



Hình 3-19. Ấn vào Run chọn Run File



- Khi ấn vào các phép tính thì sẽ được xóa trắng để có thể nhập vào các số tiếp theo. Tuy nhiên, cần thực hiện xử lý cho nút bằng để hiển thị ra kết quả của phép tính.

Quay trở lại phần thiết kế để chọn dấu bằng sau đó kích phải chọn Events->Action->actionPerformed để viết chức năng xử lý sự kiện.

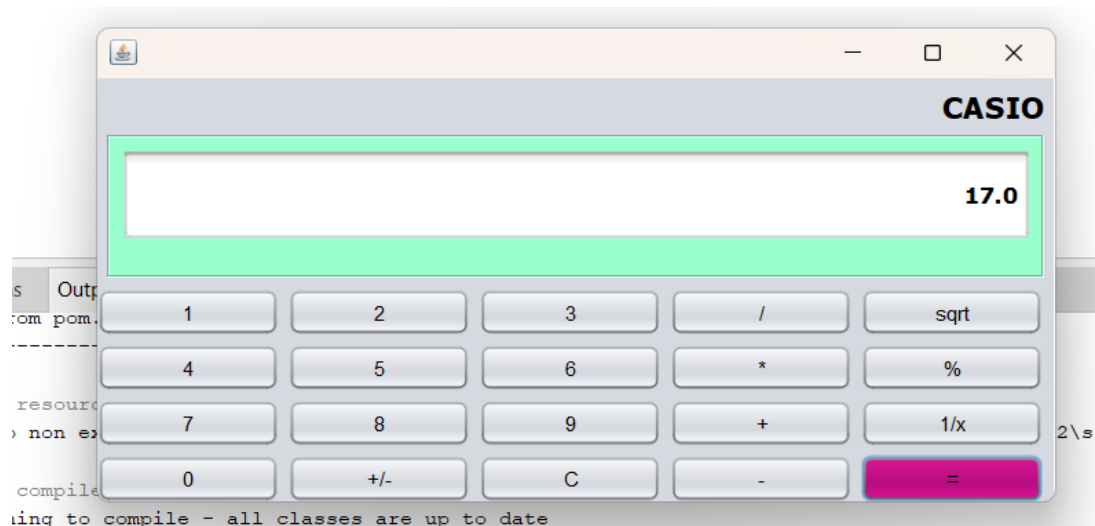
```
private void btnEqualsActionPerformed(java.awt.event.ActionEvent evt) {
    if (txtResult.getText().equals("") || operand1.equals("")) {
        return;
    }

    try {
        double operand1 = Double.parseDouble(operand1);
        double operand2 = Double.parseDouble(txtResult.getText());
        double result = 0;
        if (operator.equals("+")) {
            result = operand1 + operand2;
        } else if (operator.equals("-")) {
            result = operand1 - operand2;
        } else if (operator.equals("*")) {
            result = operand1 * operand2;
        } else if (operator.equals("/")) {
            result = operand1 / operand2;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Hình 3-20. Quay trở lại phần thiết kế để chọn dấu bằng sau đó kích phải chọn Events->Action->actionPerformed để viết chức năng xử lý sự kiện.

- Sau khi viết xong, ấn vào Run chọn Run File để kiểm tra.

Thực hiện phép tính  $15+2$  rồi bấm bằng, ta được kết quả như hình.



Hình 3-21 Thực hiện phép tính  $15+2$  rồi bấm bằng, ta được kết quả như hình.

- Tuy nhiên, khi thực hiện tiếp thì các phép tính khác thì chưa xóa được giá trị cũ, để xử lý cho trường hợp này, quay lại phần CalculatorForm để thêm vào sự kiện

```
public class CalculatorForm extends javax.swing.JFrame {  
    String operand1 = "";  
    String operator = "";  
  
    boolean isEqualPress = false;  
  
    /**  
     * Creates new form CalculatorForm
```

Hình 3-22. Sửa lỗi khi chưa xóa được giá trị cũ

- Sau đó quay lại phần btnEqualsActionPerformed để khai báo cho phần isEqualPress = true;

```
        CALCRESULT.setText("");  
        isEqualPress = true;  
    }  
}
```

Hình 3-23. Quay lại phần btnEqualsActionPerformed để khai báo cho phần  
isEqualPress = true;

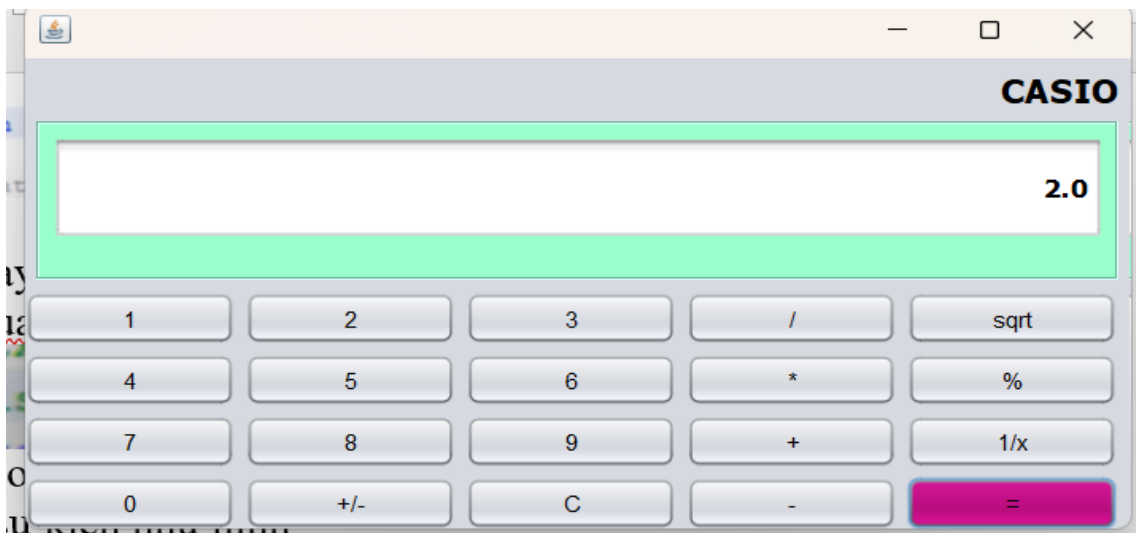
- Tiếp đó kéo lên phần `private void btnNumber1ActionPerformed` để thêm vào sự kiện như hình.

```
private void btnNumber1ActionPerformed(java.awt.event.ActionEvent evt) {  
    if (isEqualsPress) {  
        isEqualsPress = false;  
        txtResult.setText("");  
    }  
}
```

3-24. Kéo lên phần `private void btnNumber1ActionPerformed` để thêm vào sự kiện như hình.

- Sau khi thêm xong, ấn vào Run chọn Run File để kiểm tra:

Ta thực hiện phép tính sau:  $2*5=10$  tiếp tục bấm phép tính khác là  $3-1 = 2$  như hình.



Hình 3-25. Kiểm tra bằng cách thực hiện phép tính

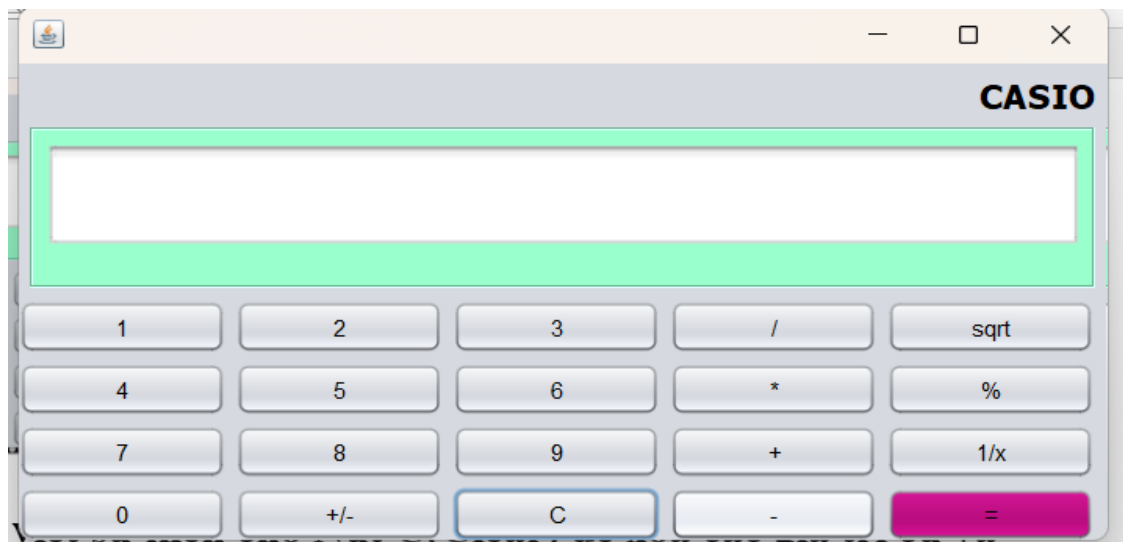
- Tiếp theo, viết sự kiện cho Nút C(Clear) để xóa các giá trị cũ và thực hiện các phép tính mới:

Quay lại phần thiết kế, chuột phải vào nút C chọn Events->Action->actionPerformed để tiến hành viết sự kiện:

```
private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {  
    txtResult.setText("");  
    operand1 = "";  
    operator = "";  
    isEqualPress = false;  
}
```

Hình 3-26. Viết sự kiện cho nút C(Clear)

- Sau khi viết xong, ấn vào Run chọn Run File để kiểm tra:  
Thực hiện thử phép tính  $5*5=25$  sao đó bấm C để xóa, ta được xóa trắng như hình



Hình 3-27. Kiểm tra bằng cách thực hiện phép tính

- Tiếp tục quay trở lại phần thiết kế để viết tiếp các chức năng cho 3 nút còn lại (sqrt, %, 1/x).

Bấm Ctrl chọn 3 nút sqrt, %, 1/x xong kích vào Events->actionPerformed->btnPercentageActionPerformed.

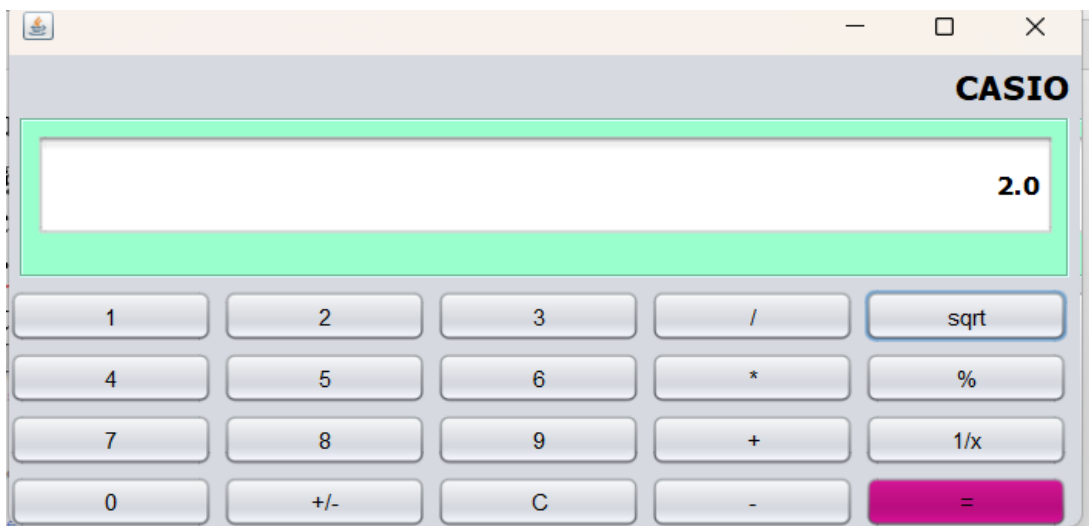
- Sau đó viết tiếp các sự kiện xử lý như hình .

```
private void btnPercentageActionPerformed(java.awt.event.ActionEvent evt) {  
    if (txtResult.getText().equals("")) {  
        return;  
    }  
    String cmd = evt.getActionCommand();  
    try {  
        double value = Double.parseDouble(txtResult.getText());  
        if (cmd.equals("sqrt")) {  
            value = Math.sqrt(value);  
        } else {  
            if (cmd.equals("%")) {  
                value = value * 100;  
            }  
            if (cmd.equals("1/x")) {  
                value = 1 / value;  
            }  
            txtResult.setText("" + value);  
            isEqualPress = true;  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Hình 3-28. Viết chức năng cho 3 nút còn lại(sqrt,%,1/x)

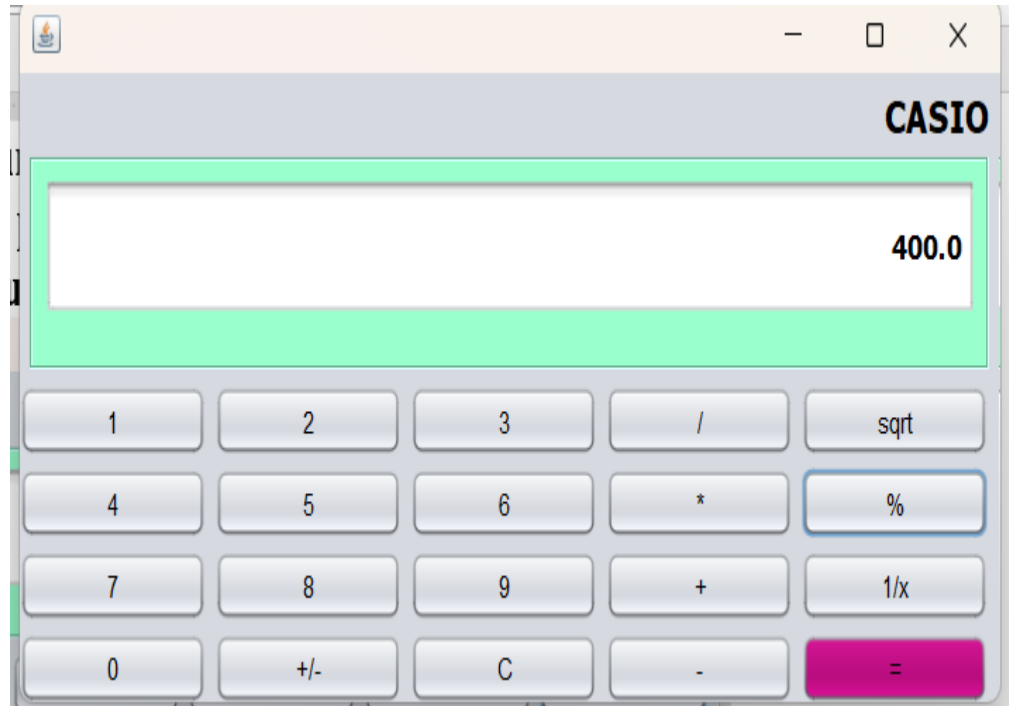
- Ấn vào Run chọn Run File để kiểm tra:

Thực hiện phép tính 4 sau đó ấn vào nút sqrt sẽ cho ra kết quả bằng 2 như hình.



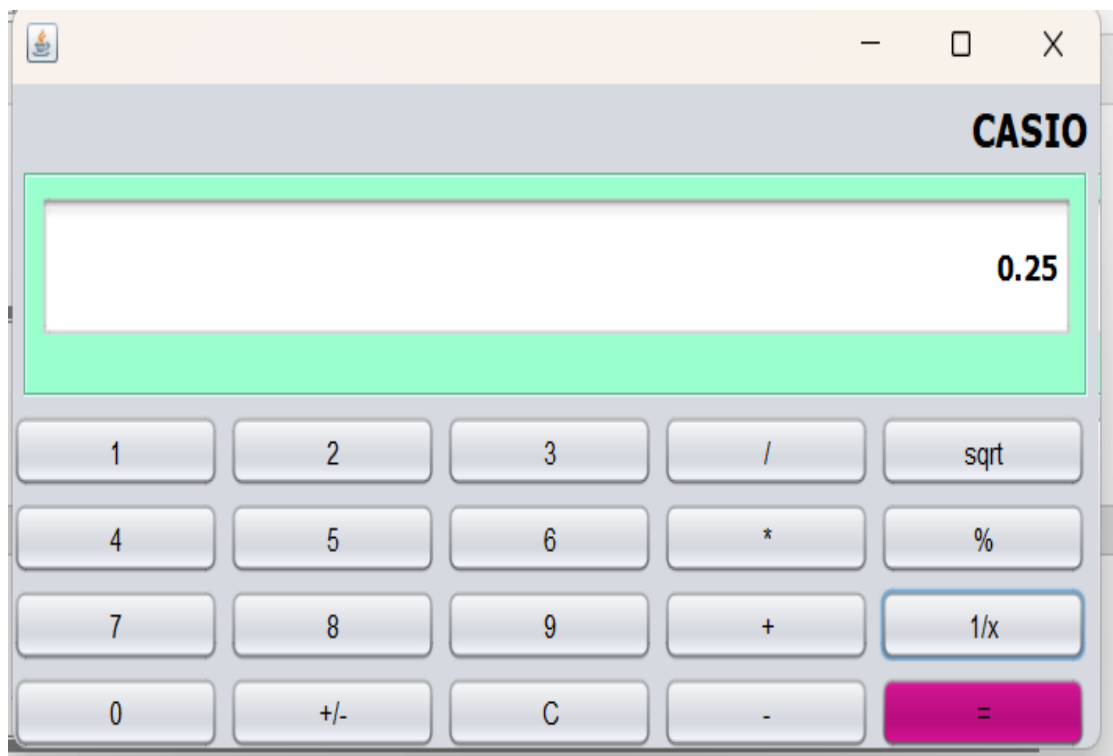
Hình 3-29. Kiểm tra bằng cách thực hiện phép tính

- Thực hiện phép tính 4 sau đó ấn vào nút % sẽ ra kết quả bằng 400 như hình.



Hình 3-30. Thực hiện phép tính %

- Thực hiện phép tính 4 sau đó ấn vào nút 1/x sẽ ra kết quả là 0.25 như hình



Hình 3-31. Thực hiện phép tính 1/x

## CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

Trong chương này, tôi sẽ trình bày các kết quả đạt được sau quá trình thực hiện đề án "Viết chương trình máy tính bằng Java". Các kết quả này có thể bao gồm đánh giá về hiệu năng, trải nghiệm người dùng và mô tả các giao diện chức năng của chương trình.

### 4.1 Hiệu năng của chương trình:

Để đánh giá hiệu năng của chương trình, chúng ta đã tiến hành thực hiện các thử nghiệm và đo đạc thời gian đáp ứng của chương trình trong quá trình thực hiện các phép toán cơ bản (+, -, \*, /), tính căn bậc hai và tính phần trăm. Kết quả cho thấy chương trình có hiệu suất cao với thời gian đáp ứng nhanh chóng, đảm bảo người dùng có trải nghiệm tính toán thuận lợi và mượt mà.

### 4.2 Trải nghiệm người dùng:

Chương trình được thiết kế với giao diện người dùng đẹp và dễ sử dụng. Người dùng có thể thực hiện các phép toán cơ bản và các chức năng khác một cách thuận tiện. Giao diện đáp ứng tốt với các thao tác của người dùng và cung cấp thông tin rõ ràng về kết quả tính toán. Người dùng có thể nhập dữ liệu và nhận kết quả một cách dễ dàng và trực quan.

### 4.3 Mô tả giao diện chức năng:

Chương trình máy tính bằng Java của chúng ta hỗ trợ các chức năng sau: - Các phép toán cơ bản (+, -, \*, /): Người dùng có thể thực hiện các phép toán cộng, trừ, nhân và chia một cách dễ dàng. - Tính căn bậc hai của một số: Chương trình cho phép tính căn bậc hai của một số đã cho và hiển thị kết quả. - Tính phần trăm của một số: Người dùng có thể tính phần trăm của một số và nhận kết quả. - Chuyển đổi số dương thành số âm và ngược lại: Chương trình cho phép người dùng chuyển đổi một số dương thành số âm và ngược lại. - Giao diện người dùng đẹp và dễ sử dụng: Chương trình được thiết kế với giao diện thân thiện với người dùng, giúp người dùng dễ dàng thao tác và tương tác với chương trình.

## CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Kết luận:

Trong quá trình thực hiện đồ án viết chương trình máy tính bằng Java, chúng tôi đã đạt được những kết quả sau:

1. Xây dựng thành công một ứng dụng máy tính đa chức năng với giao diện người dùng thân thiện. Ứng dụng này hỗ trợ các phép toán cơ bản (+, -, \*, /), tính căn bậc hai, tính phần trăm và các chức năng khác.
2. Hiệu suất của ứng dụng đáp ứng nhanh chóng và cho trải nghiệm tính toán thuận lợi cho người dùng.
3. Đồ án đã áp dụng các kiến thức về lập trình Java và công nghệ thông tin vào việc xây dựng phần mềm. Qua quá trình này, chúng tôi đã rèn luyện và nâng cao kỹ năng lập trình của mình.

### 5.2 Hướng phát triển:

Dựa trên kết quả và kinh nghiệm thu được từ việc thực hiện đồ án, chúng tôi đưa ra những kiến nghị về hướng phát triển tiếp theo:

1. Nâng cấp và mở rộng tính năng của ứng dụng máy tính. Có thể bổ sung các chức năng phức tạp hơn như tính toán đại số, đồ thị hàm số, hoặc tích hợp với các công cụ và thư viện phổ biến khác.
2. Tăng cường bảo mật và kiểm thử ứng dụng để đảm bảo tính ổn định và an toàn của chương trình.
3. Nghiên cứu và áp dụng các phương pháp tối ưu hóa để cải thiện hiệu suất và thời gian đáp ứng của ứng dụng.
4. Khám phá và áp dụng các công nghệ mới trong lĩnh vực phát triển phần mềm, như trí tuệ nhân tạo, học máy, hoặc thực tế ảo, để tạo ra những trải nghiệm độc đáo và tiện ích hơn cho người dùng.
5. Tiếp tục nghiên cứu và theo dõi các xu hướng mới trong lĩnh vực phát triển phần mềm và công nghệ thông tin, để cập nhật kiến thức và áp dụng vào việc phát triển các ứng dụng máy tính tiên tiến hơn.



## **DANH MỤC TÀI LIỆU THAM KHẢO**

- [1] Herbert Schildt , Tài liệu hướng dẫn Java cơ bản, McGraw-Hill Education , 2020.
- [2] Herbert Schildt , Java: The Complete Reference, Eleventh Edition, McGraw-Hill Education, 2018.
- [3] Kathy Sierra, Bert Bates , Head First Java, Second Edition, O'Reilly Media, 2005.
- [4] Paul Deitel Harvey Deitel, Java How to Program, Early Objects, Eleventh Edition., Pearson, 2017.
- [5] Allen B. Downey, Think Java: How to Think Like a Computer Scientist. Allen B. Downey, Chris Mayfield, O'Reilly Media, 2016.
- [6] Joshua Bloch, Effective Java, Third Edition, Addison-Wesley Professional, 2018.