ĐẠI HỌC BÁCH KHOA HÀ NỘI  
 TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

A red and yellow sign

Description automatically generated with low confidence

**Báo cáo Mini Project**

**Học phần: Object-oriented Programming - 151965 - 20241**

Đề tài: Mô phỏng và minh họa thuật toán sắp xếp trong Java  
Thuật toán: Selection Sort, Merge Sort và Shell Sort

Nhóm: 13

GVHD: Nguyễn Thị Thu Trang

Thành viên:

|  |  |
| --- | --- |
| Lê Tuấn Khải | 20225861 |
| Trương Phạm Ngọc Khánh | 20225641 |
| Đặng Đức Khải | 20226086 |
| Nguyễn Tuấn Khải | 20225862 |
| Phùng Quang Khải | 20225639 |

Hà Nội, 12/2024

# TABLE OF CONTENT

[TABLE OF CONTENT 2](#_Toc186038457)

[**I.** **Phân công các thành viên** 3](#_Toc186038458)

[**II.** **Mô tả chi tiết về Mini Project** 4](#_Toc186038459)

[1. Tổng quan 4](#_Toc186038460)

[2. Yêu cầu của Project 4](#_Toc186038461)

[a. Main Menu 4](#_Toc186038462)

[b. Màn hình minh họa thuật toán 5](#_Toc186038463)

[3. Use case diagram 6](#_Toc186038464)

[**III.** **Design** 7](#_Toc186038465)

[**IV.** **Kết luận** 9](#_Toc186038466)

# **Phân công các thành viên**

1. Lê Tuấn Khải

* Email: [Khai.lt225861@sis.hust.edu.vn](mailto:Khai.lt225861@sis.hust.edu.vn)
* MSSV: 20225861

Use case diagram, class diagram, HelpController, GeneralController

2. Trương Phạm Ngọc Khánh

* Email: [Khanh.tpn225641@sis.hust.edu.vn](mailto:Khanh.tpn225641@sis.hust.edu.vn)
* MSSV: 20225641

SortAlgorithm và các lớp con của nó

3. Đặng Đức Khải

* Email: [Khai.dd226086@sis.hust.edu.vn](mailto:Khai.dd226086@sis.hust.edu.vn)
* MSSV: 20226086

Use case diagram, giao diện, AlgorithmScreenController, MainController, Main

4. Nguyễn Tuấn Khải

* Email: [Khai.nt225862@sis.hust.edu.vn](mailto:Khai.nt225862@sis.hust.edu.vn)
* MSSV: 20225862

Class diagram, GeneralController, HelpController

5. Phùng Quang Khải

* Email: [Khai.pq225639@sis.hust.edu.vn](mailto:Khai.pq225639@sis.hust.edu.vn)
* MSSV: 20225639

Class diagram, báo cáo, slide, fix lỗi

***Ghi chú về việc sao chép mã nguồn: Không có***

# **Mô tả chi tiết về Mini Project**

## Tổng quan

Mảng là một trong những cấu trúc dữ liệu cơ bản nhất trong khoa học máy tính. Hầu hết các phép toán và các cấu trúc dữ liệu phức tạp khác đều được xây dựng và thực thi dựa trên mảng. Chính vì tầm quan trọng này, việc hiểu và nắm vững các thuật toán sắp xếp trên mảng là vô cùng cần thiết.

Project này hướng đến việc xây dựng một ứng dụng để minh họa và giải thích ba thuật toán sắp xếp cơ bản trên mảng, bao gồm**: Selection, Merge Sort và Shell Sort.** Thông qua chương trình này, người dùng có thể quan sát trực quan từng bước thực hiện của các thuật toán, từ đó hiểu rõ hơn về cách thức hoạt động và sự khác biệt giữa chúng.

## Yêu cầu của Project

Những tính năng có trong ứng dụng:

1. Main Menu

* Hiển thị tên ứng dụng: Your Agorithm.
* Choose Algorithm: Cung cấp 3 lựa chọn thuật toán sắp xếp cho người dùng, bao gồm: Selection Sort, Merge Sort và Shell Sort.
* Help: Hiển thị hướng dẫn cơ bản về cách sử dụng chương trình và mục đích của ứng dụng.
* Quit: Cung cấp tùy chọn thoát khỏi ứng dụng. Trước khi thoát, hệ thống cần hiển thị hộp thoại xác nhận để tránh thoát nhầm.

A screenshot of a computer

Description automatically generated

Figure 1: Main Menu

1. Màn hình minh họa thuật toán

Sau khi người dùng lựa chọn 1 trong 3 thuật toán để sắp xếp, ứng dụng đưa người dùng đến màn hình minh họa thuật toán:

* Tiêu đề giao diện: Dòng chữ cho biết thuật toán nào đã được chọn, ví dụ như hình ở dưới đây là Merge Sort.
* Chức năng tạo mảng: Người dùng có thể sử dụng nút “Generate” tạo ra một mảng ngẫu nhiên, hoặc nhập một mảng tùy ý vào ô nhập liệu và chọn “Upload” để lưu. Sau khi tạo mảng, mảng sẽ được hiển thị sau dòng “Your array is: ”.
* Nút “Sort!”: Khi nhấn vào nút này, thuật toán đã chọn (ví dụ: Merge Sort) sẽ bắt đầu thực hiện sắp xếp mảng. Mảng sau từng bước sắp xếp được hiển thị ở cửa sổ bên dưới.
* Dòng “Total time: “ hiển thị thời gian để chương trình thực hiện sắp xếp
* Chức năng điều hướng: “Return to Menu” cho phép quay lại Main Menu để lựa chọn thuật toán khác, và nút “Quit” để thoát khỏi chương trình. Trước khi thoát, hệ thống sẽ hỏi xác nhận để tránh thoát nhầm.
* Help: Hiển thị hướng dẫn cơ bản về cách sử dụng chương trình và mục đích của ứng dụng.

A screenshot of a computer

Description automatically generated

Figure 2: Màn hình minh họa thuật toán

* 1. Use case diagram

Figure 3: Use case diagram

A diagram of a person with text

Description automatically generated

Trong ứng dụng, người dùng tương tác với phần mềm thông qua các use case chính sau:

* Lựa chọn thuật toán sắp xếp: Người dùng chọn một trong ba thuật toán sắp xếp (Selection Sort, Merge Sort, Shell Sort) để thực hiện.
* Tạo mảng dữ liệu: Người dùng có thể lựa chọn tạo mảng dữ liệu một cách ngẫu nhiên hoặc nhập vào một mảng riêng.
* Chạy thuật toán sắp xếp: Sau khi mảng được tạo, người dùng có thể chạy thuật toán sắp xếp đã chọn, và hệ thống sẽ hiển thị các bước sắp xếp rõ ràng.
* Xem hướng dẫn: Người dùng có thể xem phần trợ giúp để hiểu rõ hơn về mục đích và cách sử dụng chương trình.
* Thoát chương trình: Khi hoàn tất, người dùng có thể thoát khỏi ứng dụng sau khi xác nhận.

Những use case này giúp người dùng dễ dàng tìm hiểu, theo dõi và tương tác với các thuật toán sắp xếp trong ứng dụng.

# **Design**

A diagram of a computer program

Description automatically generated with medium confidence

Figure 4: Class diagram

Chi tiết về chức năng, thuộc tính, phương thức và quan hệ của các lớp được mô tả trong bảng dưới đây:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên lớp | Chức năng | Attributes | Method | Quan hệ với lớp khác |
| AlgorithmScreenController | Lớp điều khiển chính cho màn hình hiển thị thuật toán. | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | ~**userOption: String** - Lựa chọn của người dùng. -**labelArray: Label** - Nhãn hiển thị mảng. -**labelTotalTime: Label** - Nhãn hiển thị thời gian xử lý. -**header: Label** - Tiêu đề giao diện. -**textField: TextField** - Ô nhập dữ liệu. -**scrollPaneResult: ScrollPane** - Khu vực hiển thị kết quả. -**intList: List<Integer>** - Danh sách các số nguyên cần sắp xếp. **~sortAlgorithm: SortAlgorithm** - Thuật toán sắp xếp được chọn. |  |  | | **+setUserOption(String: int)**: **void** - Gán giá trị lựa chọn của người dùng. **+uploadButton(ActionEvent event: int): void** – Nhận mảng do người dùng tạo **+generateButton(ActionEvent event: int): void** - Tạo mảng ngẫu nhiên. **+sortButton(ActionEvent event: int): void** - Bắt đầu sắp xếp. | Kế thừa (Inheritance) từ GeneralController |
| General  Controller | Lớp điều khiển tổng quát cung cấp chức năng chung cho giao diện. | **#scenePane: AnchorPane** - Được liên kết với một AnchorPane trong file fxml, dùng để thao tác với Stage  **#stage: Stage** - Lưu trữ cửa sổ hiện tại, giúp chuyển đổi giao diện hoặc đóng cửa sổ. **#scene: Scene** - Quản lý giao diện (cảnh) đang hiển thị trong cửa sổ. **#root: Parent** - Gốc của giao diện mới khi tải file fxml. | **+returnToMenu(ActionEvent event: int): void** - Quay lại menu chính. **+quit(ActionEvent event: int): void** - Thoát ứng dụng. **+help(ActionEvent event: int): void** - Hiển thị trợ giúp. | Kế thừa (Inheritance) cho AlgorithmScreenController, HelpController và MainController |
| HelpController | Lớp điều khiển hiển thị trợ giúp. |  | **+HelpController()** - Khởi tạo lớp để hiển thị giao diện trợ giúp. | Kế thừa (Inheritance) từ GeneralController |
| MainController | Lớp điều khiển chính, cung cấp chức năng lựa chọn thuật toán. | **-myComboBox: ComboBox<String>** - Hộp chọn các thuật toán. **-myLabel: Label** - Nhãn hiển thị thông báo. **~algorithms: String[]** - Danh sách các thuật toán có sẵn. | **+initialize(URL arg: int, ResourceBundle agr1: int):** **void** - Khởi tạo các thành phần trong giao diện. **+selectAction(ActionEvent event: int): void** - Xử lý lựa chọn của người dùng. | Kế thừa (Inheritance) từ GeneralController  Liên kết (Association) với Main |
| Main |  |  | **+Main(): void** - Hàm khởi tạo. **+start(): void** - Khởi động ứng dụng và chuẩn bị giao diện người dùng. | Liên kết (Association) với MainController |
| SortAlgorithm | Lớp trừu tượng (abstract class) đại diện cho một thuật toán sắp xếp. | # **array**: int[]: Mảng các số nguyên được sắp xếp, có thể truy cập và chỉnh sửa bởi các lớp kế thừa (Protected). | **+sort(): void** - Phương thức sắp xếp (phải được lớp con triển khai). **+displaySteps(): void** - Hiển thị từng bước sắp xếp. **+getSteps(): void** - Trả về các bước thực hiện. **+#saveStep(): void** - Lưu lại trạng thái của mảng tại từng bước. | Kế thừa (Inheritance) cho SelectionSort, MergeSort, và ShellSort |
| SelectionSort | Triển khai thuật toán sắp xếp Selection Sort. |  | **+ sort()**: void: Triển khai thuật toán Selection Sort: Tìm phần tử nhỏ nhất trong mảng và hoán đổi chúng vào vị trí thích hợp. | - Kế thừa từ SortAlgorithm  (Inheritance): Triển khai phương thức **sort()** từ lớp cơ sở. |
| MergeSort | Triển khai thuật toán sắp xếp Merge Sort. |  | + **sort()**: void: Triển khai thuật toán Merge Sort. + **merge(left: int, mid: int, right: int)**: void: Hợp nhất hai mảng con đã sắp xếp. + **mergeSort(left: int, right: int):** void: Sắp xếp đệ quy chia mảng thành 2 phần. |
| ShellSort | Triển khai thuật toán sắp xếp Shell Sort. |  | + **sort():** void: Triển khai thuật toán Shell Sort: Sắp xếp mảng với khoảng cách giảm dần. |

# **Kết luận**

Kết quả của project nhóm em đã đáp ứng được các yêu cầu cần thiết: Tạo ra một ứng dụng có thể mô phỏng và minh họa 3 thuật toán sắp xếp bằng Java: Selection Sort, Merge Sort và Shell Sort. Nhóm em rất cảm ơn cô Nguyễn Thị Thu Trang và anh Phạm Phan Anh đã giúp đỡ và hướng dẫn rất chi tiết để chúng em có thể hoàn thành project này. Tuy vậy, project vẫn còn nhiều thiếu sót và cần phải cải thiện như thiết kế giao diện hiện còn khá đơn giản, cần thêm hỗ trợ quản lí vào ra: cho phép nhập và xuất dữ liệu dưới dạng file, … Trong tương lai, nhóm em sẽ tiếp tục phát triển project với những tính năng mới nhằm nâng cao trải nghiệm người dùng và chất lượng của ứng dụng nhằm nắm vững, nâng cao kiến thức về môn học lập trình hướng đối tượng.