



Demonstration of sorting algorithms on array

Selection Sort, Merge Sort and Shell Sort



Members and Assignment

Lê Thái Sơn - 20225919

Xây dựng khung của dự án, cụ thể bao gồm các lớp Sort, StateSorting, StateSwap và cách 3 controller tô màu. Triển khai thuật toán Selection Sort, cụ thể là lớp SelectionSort

Thiết kế giao diện dự án, cụ thể bao gồm các lớp HelpController, HomeController và 2 file homepage.fxml và get_help_page.fxml

Thực hiện đóng gói các sourcecode thành 3 package model, view, controller

Tham gia sửa lớp tô MergeController, InsertionController, ShellController (cụ thể sửa 2 phương thức là SortWithColor và performSortingStep)

Tham gia làm slide và viết báo cáo (Chỉ viết báo cáo và làm slide phần mình đã làm nên đóng góp 20%)

Members and Assignment

Nguyễn Trung Sơn -20226124

- **Code giao diện của sort, điều khiển,đầu vào mảng ,sort trực tiếp (100 %)**
- **Tham gia xây dựng Use case diagram và Class Diagram (20%)**

Members and Assignment

Chu Văn Thành - 20194673

- **Chỉnh sửa UI giao diện**
- **Tô màu phần UI của các thuật toán Sort**

Members and Assignment

Lê Đức Thắng - 20225925

- Triển khai thuật toán ShellSort
- Làm báo cáo
- Tham gia xây dựng use case diagram và class diagram(30%)

Members and Assignment

Nguyễn Thanh Tân - 20225923

- Triển khai thuật toán MergeSort(100%)
- Làm khung Slide
- Tham gia xây dựng Use case diagram và Class Diagram

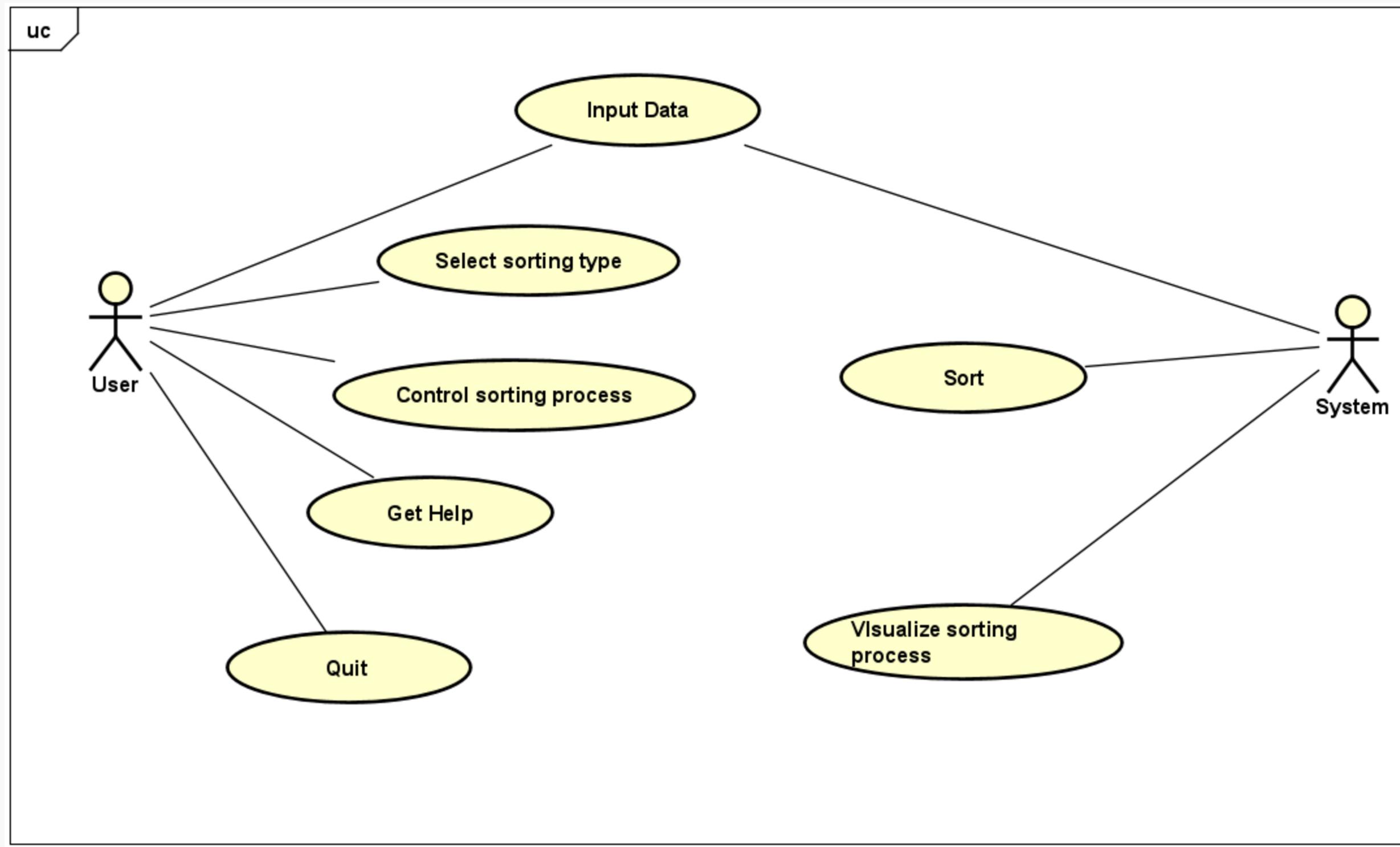
Problem description



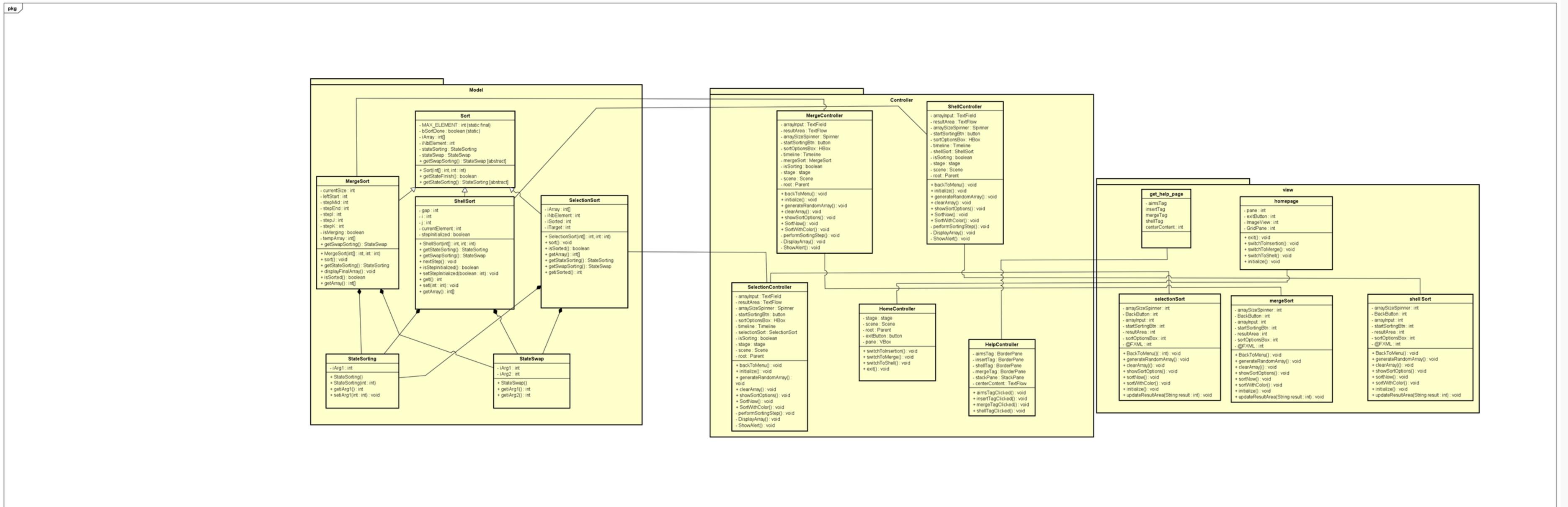
Mảng là cấu trúc cơ bản nhất của khoa học máy tính. Hầu hết các thao tác cũng như các cấu trúc dữ liệu khác đều được xây dựng và thực hiện trên mảng. Trong dự án này, nhóm 23 chúng tôi sẽ tạo một ứng dụng để giải thích ba thuật toán sắp xếp trên mảng: Selection Sort, Merge Sort và Shell Sort



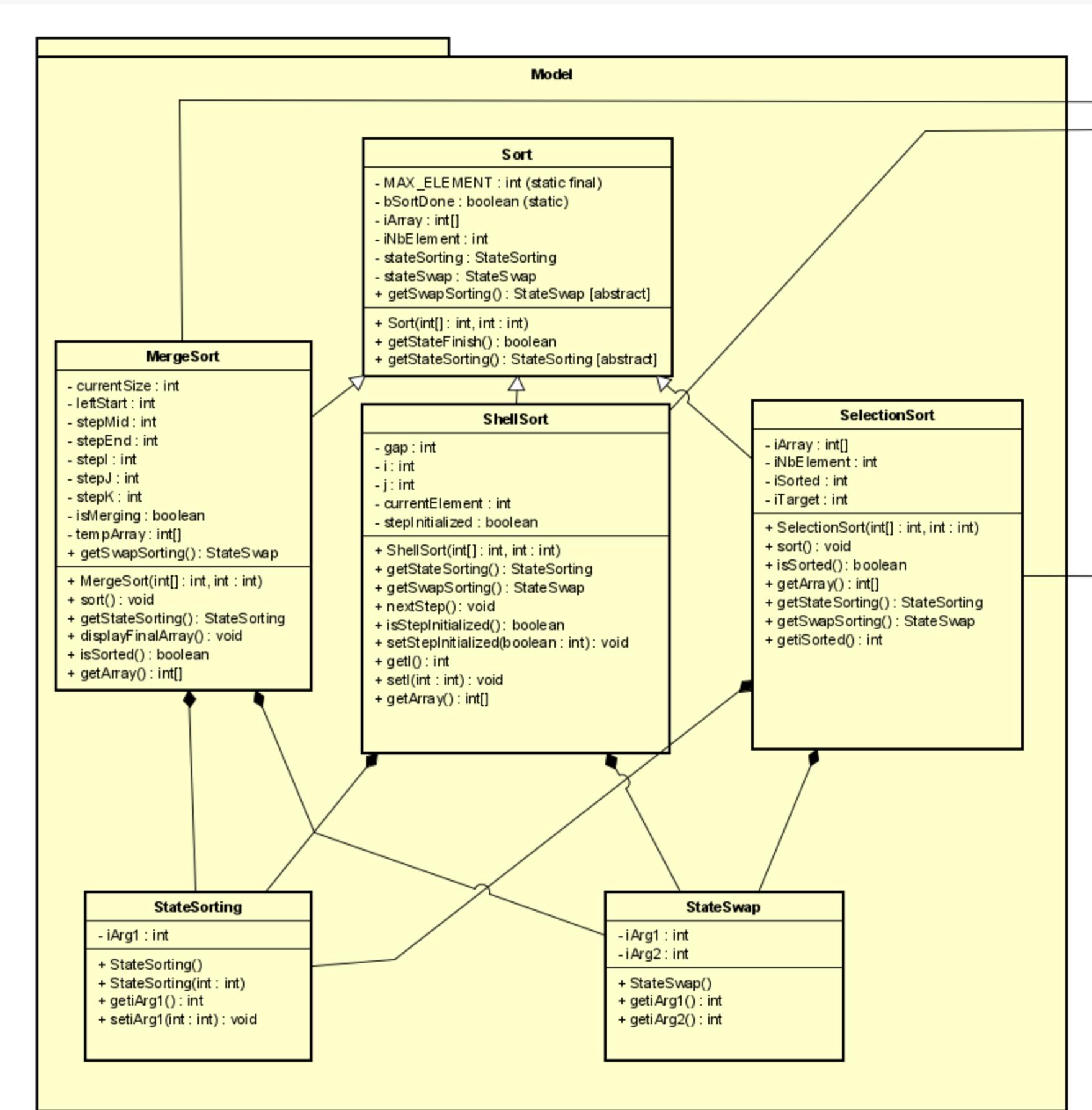
Use Case Diagram



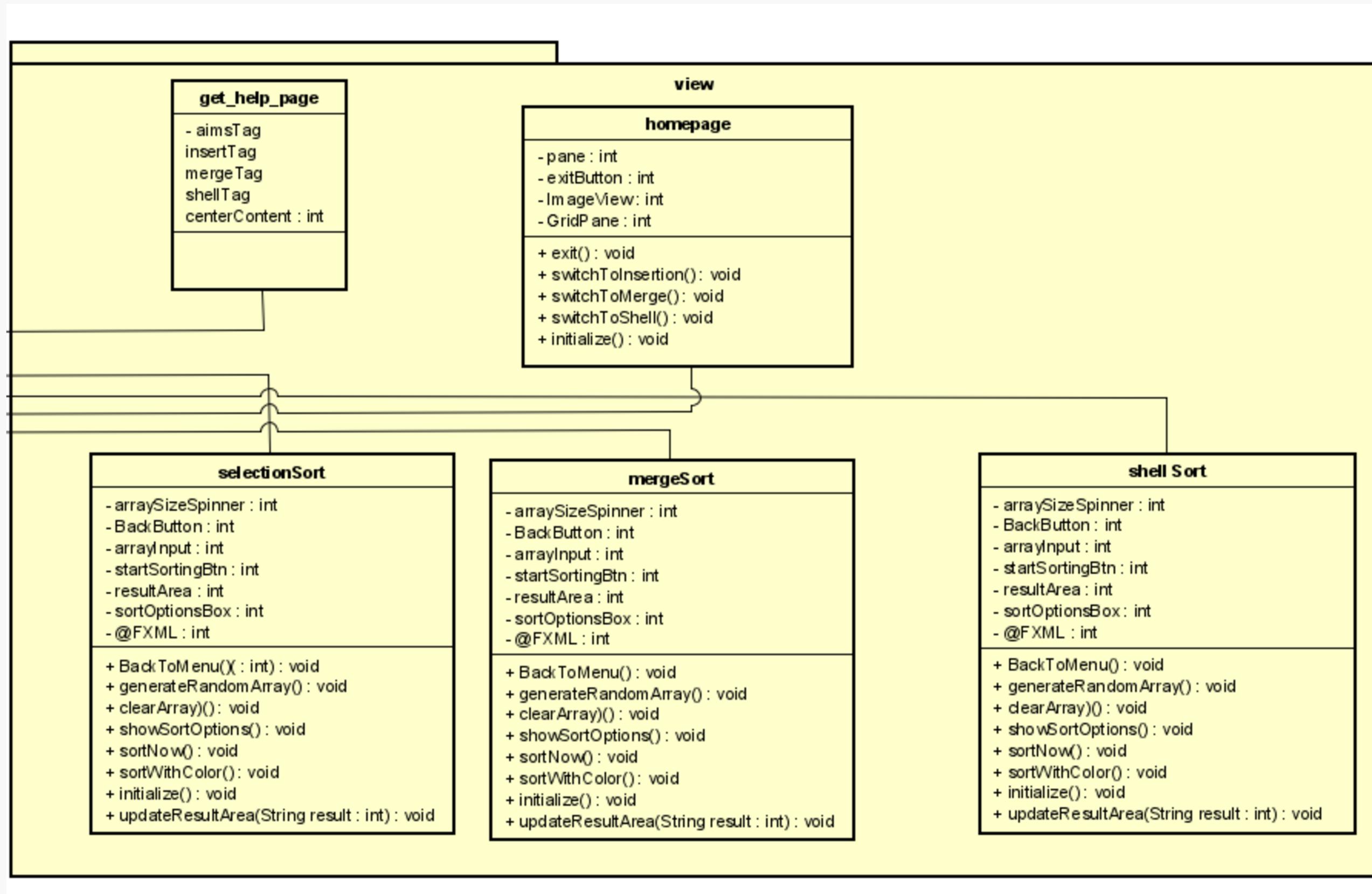
Class Diagram



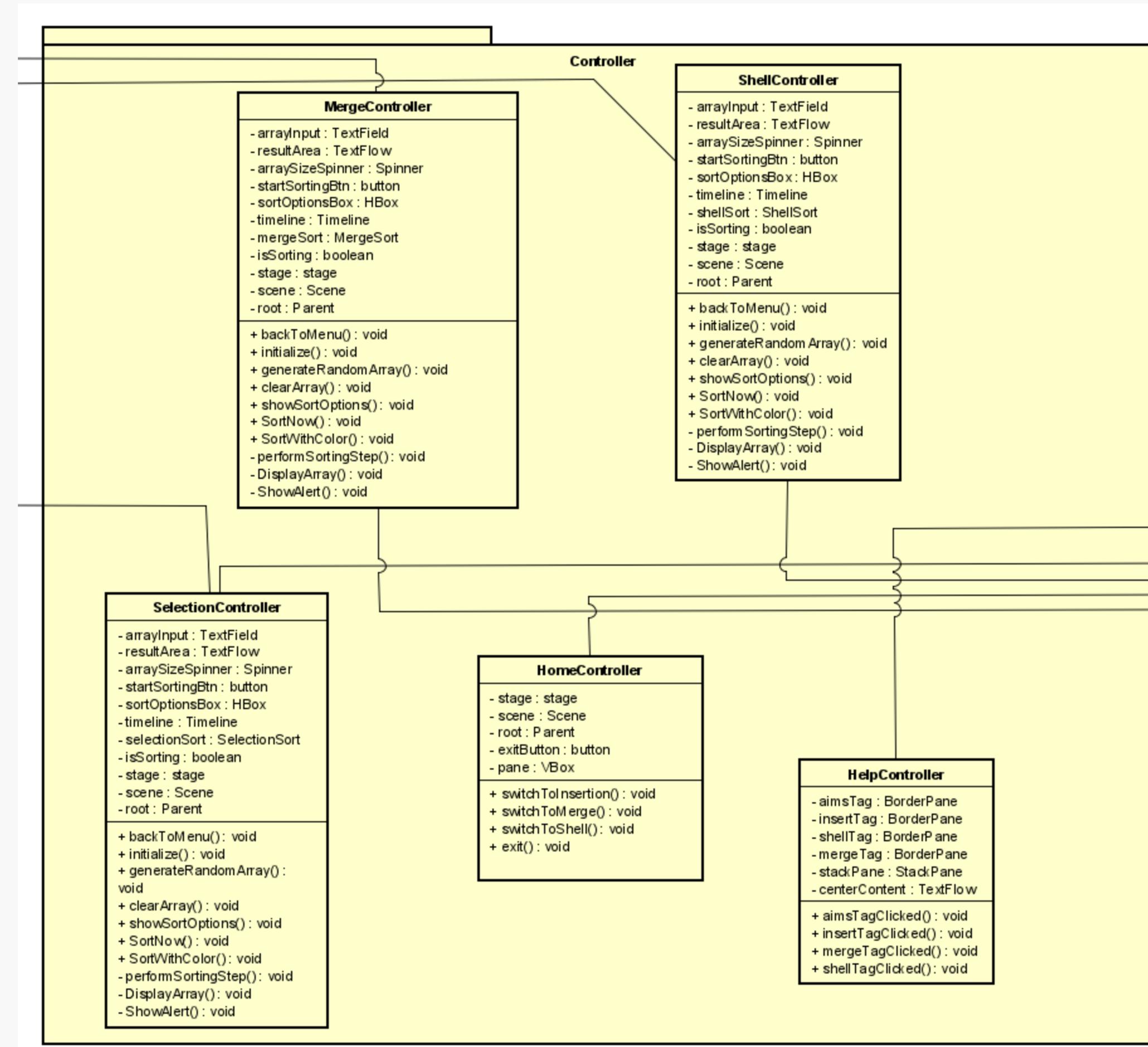
Model



View



Controller



Detail Class Diagram

Lê Thái Sơn - 20225919

I.Constructor: (public) Sort(int[] iArray, int iNbElement): Khởi tạo đối tượng Sort và khởi tạo mảng, số phần tử của mảng, khởi tạo 2 trạng thái stateSorting và stateSwap

II.Phương thức: (public) Cung cấp trạng thái hiện tại của thuật toán, bao gồm:

- + getStateFinish: Phương thức để lấy trạng thái đã sort xong hay chưa
- +getStateSorting(): Phương thức để lấy về chỉ số phần tử đang được xét tại bước hiện tại .
- +getSwapSorting(): Phương thức để lấy về chỉ số của 2 phần tử sẽ đổi chỗ tại bước này.



Detail Class Diagram

Lê Đức Thắng - 20225925

I.Constructor: (public) ShellSort(int[] iArray, int iNbElement): Khởi tạo đối tượng ShellSort và thiết lập các giá trị ban đầu cho thuật toán.

II.Phương thức ghi đè: (public) Cung cấp trạng thái hiện tại của thuật toán, bao gồm:
Phần tử đang được so sánh (getStateSorting()).
Cặp phần tử đang được hoán đổi (getSwapSorting()).

III.Getter and Setter: (public) Hỗ trợ quản lý và truy cập các biến trạng thái của thuật toán (i, stepInitialized).

IV.Phương thức xử lý chính:(public) nextStep() Thực hiện từng bước của thuật toán Shell Sort, bao gồm:Khởi tạo bước mới; So sánh, hoán đổi phần tử trong nhóm
● ● ● ● ● Giảm khoảng cách gap và kiểm tra điều kiện dừng.

Detail Class Diagram

Nguyễn Thanh Tân - 20225923

I.Constructor: (public) MergeSort(int[] iArray, int iNbElement): Khởi tạo đối tượng MergeSort và thiết lập các giá trị ban đầu cho thuật toán.

II.Phương thức : (public) Cung cấp trạng thái hiện tại của thuật toán, bao gồm:
Xử lý thuật toán, trả về phần tử đang được so sánh (getStateSorting()).
Cặp phần tử đang được hoán đổi (getSwapSorting()).
Kiểm tra xem Sort đã xong hay chưa (getStateFinish()).



OOP Techniques

Inheritance

- Lớp InsertionSort, MergeSort, và ShellSort đều kế thừa từ lớp cơ sở Sort. Điều này giúp tái sử dụng các thuộc tính và phương thức cơ bản như iArray, iNbElement, bSortDone, và các phương thức trừu tượng (getStateSorting, getSwapSorting).

Abstraction

- Lớp Sort là abstract class, chứa các phương thức trừu tượng getStateSorting và getSwapSorting. Các lớp con phải ghi đè (override) các phương thức này, cung cấp các cách triển khai riêng theo thuật toán của từng loại sắp xếp.

OOP Techniques

Association

- Lớp SortController tương tác với MainMenu và DemonstrationPanel để điều phối các thao tác từ người dùng và hiển thị giao diện tương ứng.
- Lớp SortController sử dụng lớp Sort(hoặc lớp con của nó) để thực hiện sắp xếp.

Composition

- Lớp MainMenu và DemonstrationPanel chứa các thành phần giao diện như nút bấm và vùng hiển thị để quản lý trực tiếp giao diện người dùng.

OOP Techniques

Polymorphism

- Các phương thức `getStateSorting()` và `getSwapSorting()` được định nghĩa trừu tượng (hoặc mặc định) trong lớp cha `Sort` và được ghi đè (override) trong các lớp con `SelectionSort`, `MergeSort` và `ShellSort`.

Encapsulation

- Các thuộc tính của lớp `StateSorting` và `StateSwap` (như `iArg1`, `iArg2`) được khai báo là `private`. Điều này bảo vệ dữ liệu bên trong và chỉ cho phép truy cập hoặc chỉnh sửa thông qua các phương thức công khai như `getiArg1`, `getiArg2`, `setiArg`.

Demostration Video

Video Link:

https://husteduvn-my.sharepoint.com/:v/g/personal/son_lt225919_sis_hust_edu_vn/EUWbC4StfmtFoJrZpDYlfhEBoplTDnNZ2pkz4J3ThTdEA?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoIJTdHJIYW1XZWJBcHAiLCjyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IlldlYilsInJIZmVycmFsTW9kZSI6InZpZXcifX0%3D&e=3yUo94





Thank you

