



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 (100%)

Thiết kế giao diện dự án, cụ thể bao gồm các lớp HelpController, HomeController và 2 file homepage.fxml và get_help_page.fxml (100%)

Thực hiện đóng gói các sourcecode thành 3 package model, view, controller (100%)

Tham gia sửa lớp tô màu MergeController, InsertionController, ShellController (cụ thể sửa 2 phương thức là SortWithColor và performSortingStep) (15%)

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

Tham gia thiết kế use case diagram và class diagram (20%)

Members and Assignment

Nguyễn Trung Sơn -20226124

- **Viết các View: insertionSort.fxml, mergeSort.fxml, shellSort.fxml (90%)**
- **Tham gia xây dựng Use case diagram và Class Diagram (30%)**
- **Làm slide (15%)**

Members and Assignment

Chu Văn Thành - 20194673

- **Viết các lớp Controller: MergeController, InsertionController, ShellController (85%) và ControllerBase (100%)**
- **Viết các View: insertionSort.fxml, mergeSort.fxml, shellSort.fxml (10%)**
- **Tham gia xây dựng Use case diagram và Class Diagram (20%)**
- **Làm slide (15%)**

Members and Assignment

Lê Đức Thắng - 20225925

- Triển khai thuật toán ShellSort (80%)
- Tham gia xây dựng use case diagram và class diagram(20%)
- Làm báo cáo (100%)
- Làm slide (15%)

Members and Assignment

Nguyễn Thanh Tân - 20225923

- Triển khai thuật toán MergeSort(100%)
- Gỡ lỗi thuật toán ShellSort(20%)
- Làm Slide (40%)
- Tham gia xây dựng Use case diagram và Class Diagram (10%)

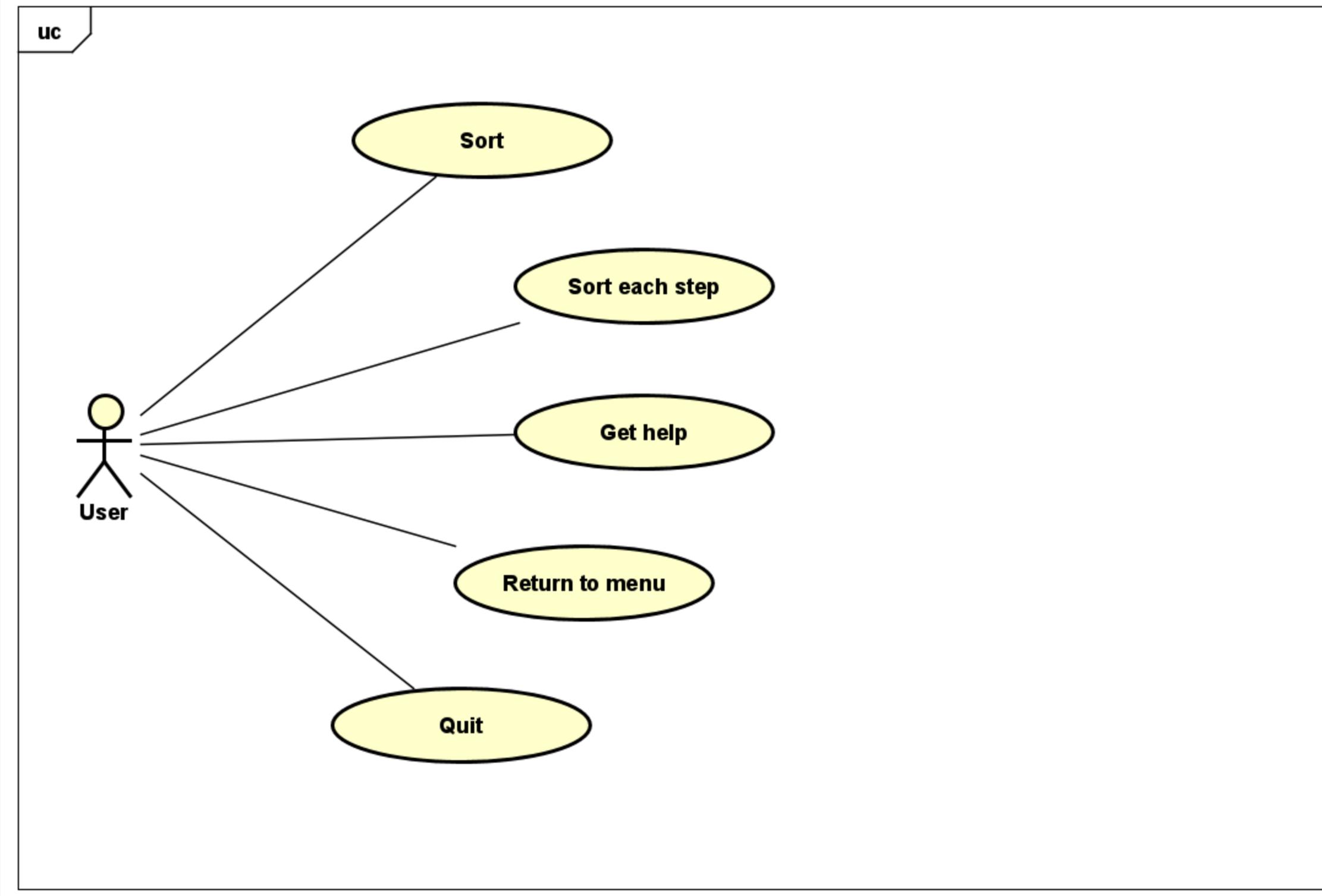
Problem description



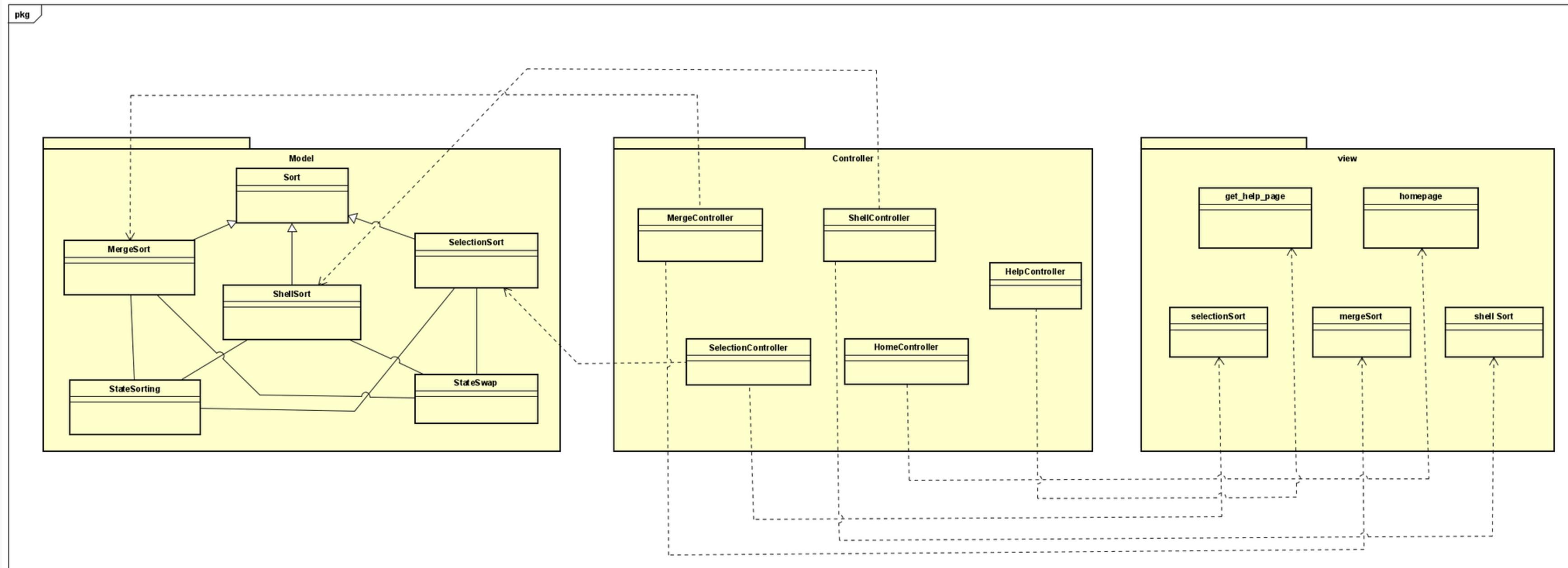
Hiện nay, có rất nhiều thuật toán sắp xếp khác nhau đang được sử dụng. Trong dự án này, nhóm 23 chúng tôi sẽ tạo một ứng dụng để mô phỏng từng bước của ba giải thuật sắp xếp trên mảng khá phổ biến, đó là: Selection Sort, Merge Sort và Shell Sort



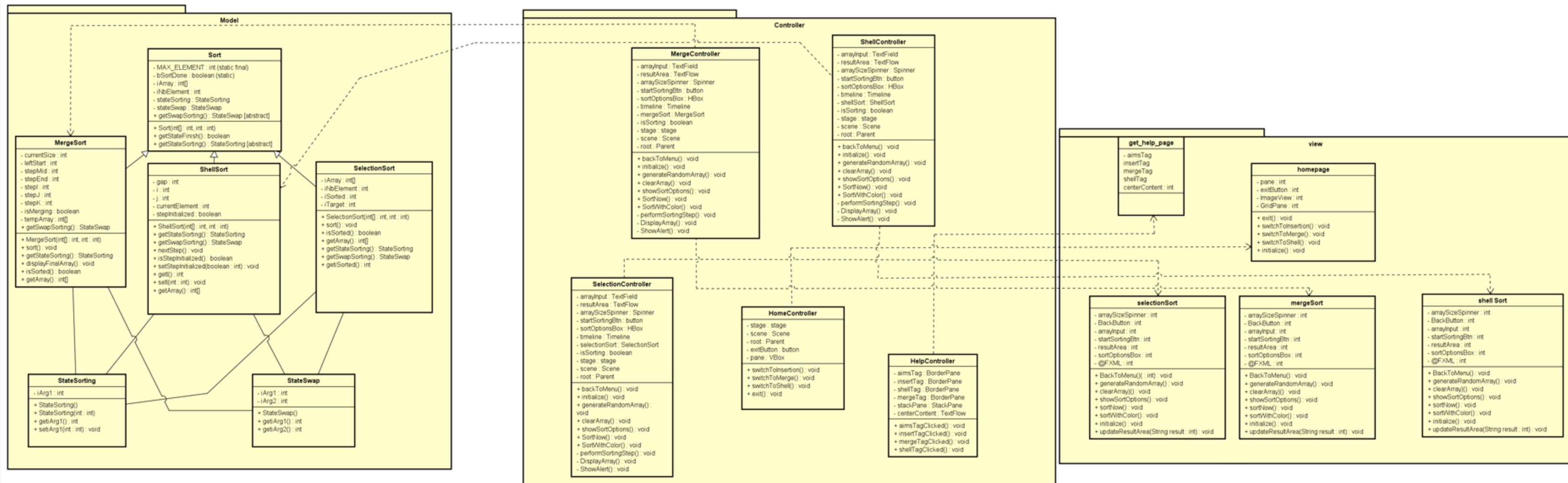
Use Case Diagram



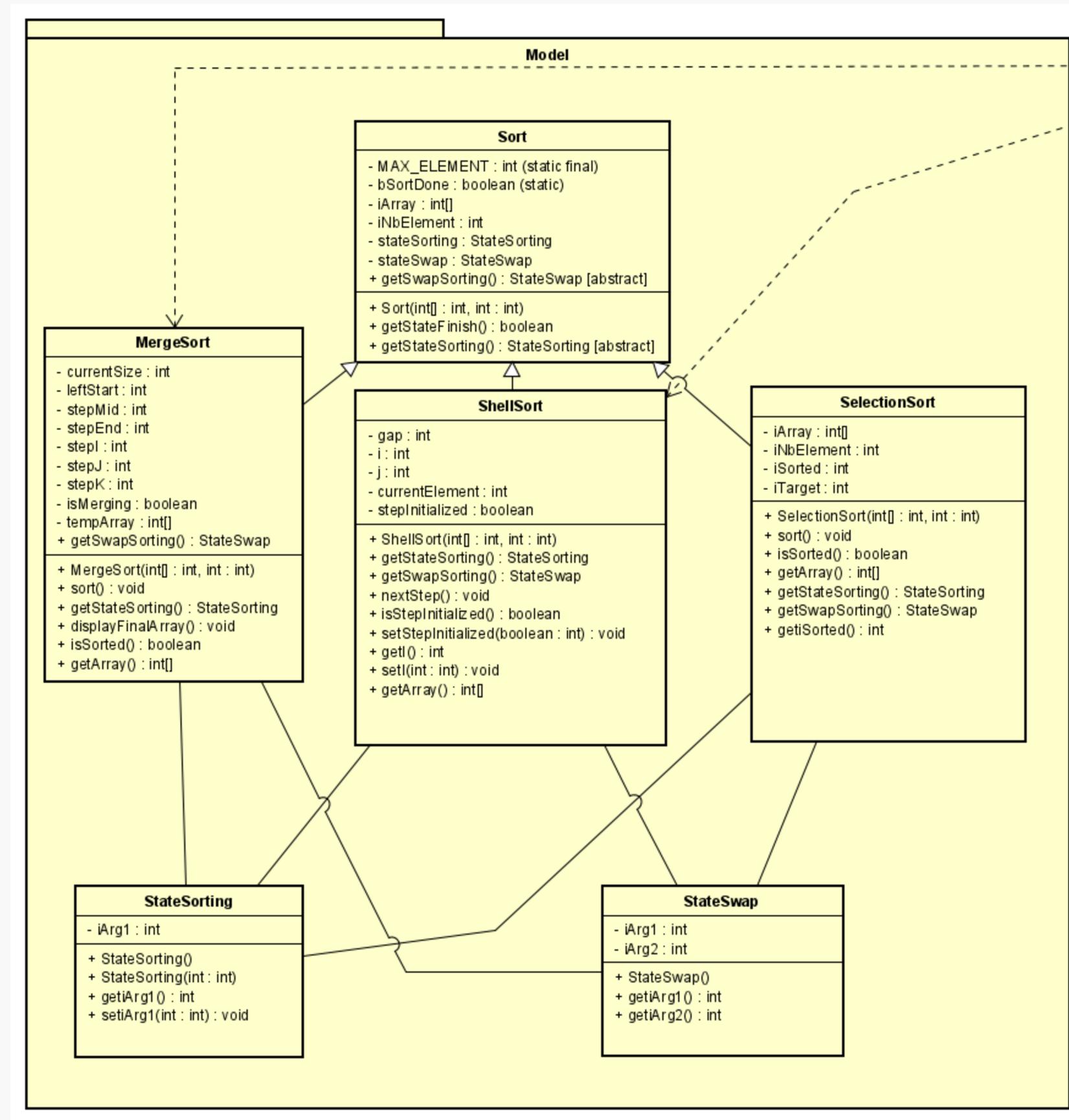
General Class Diagram



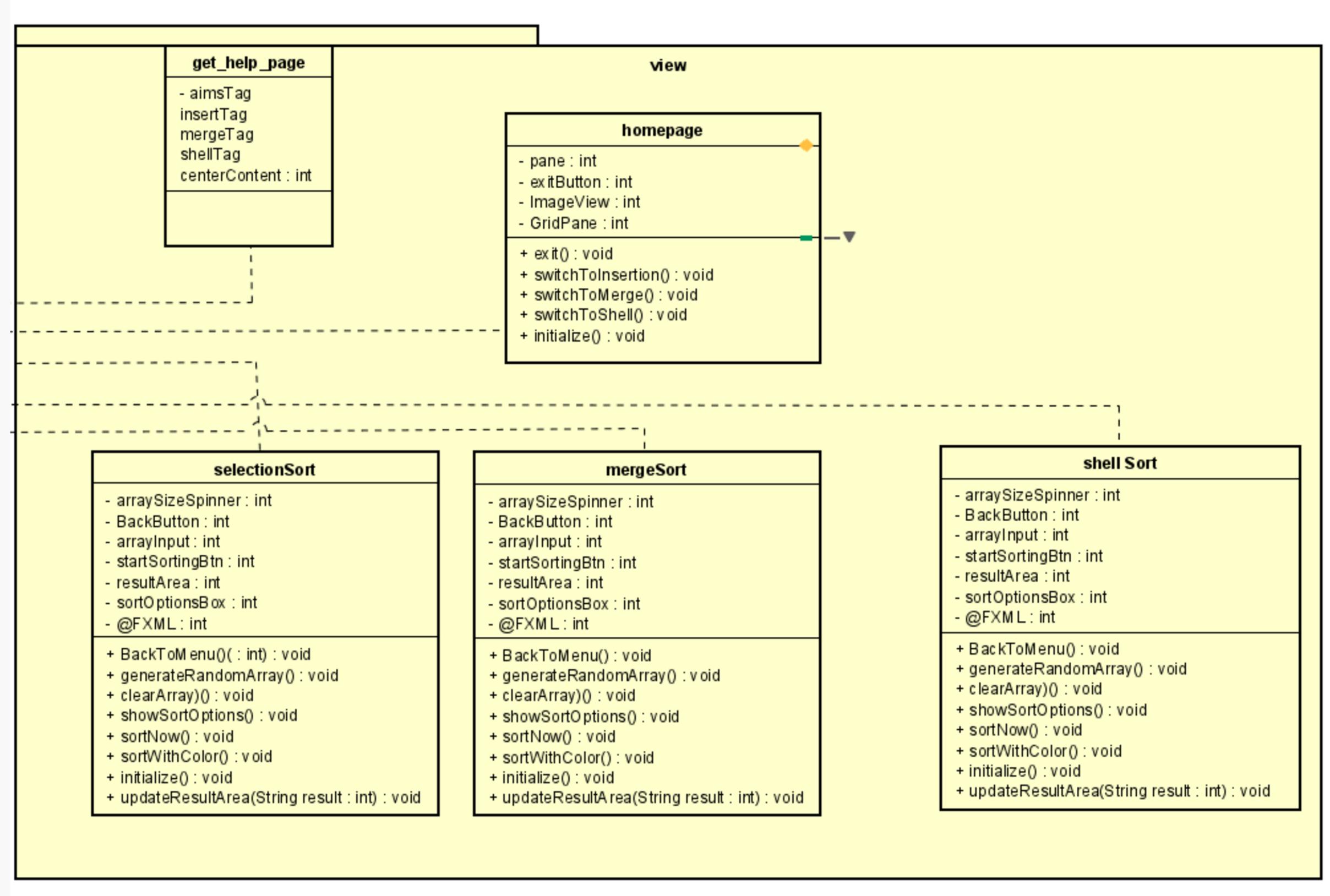
Class Diagram



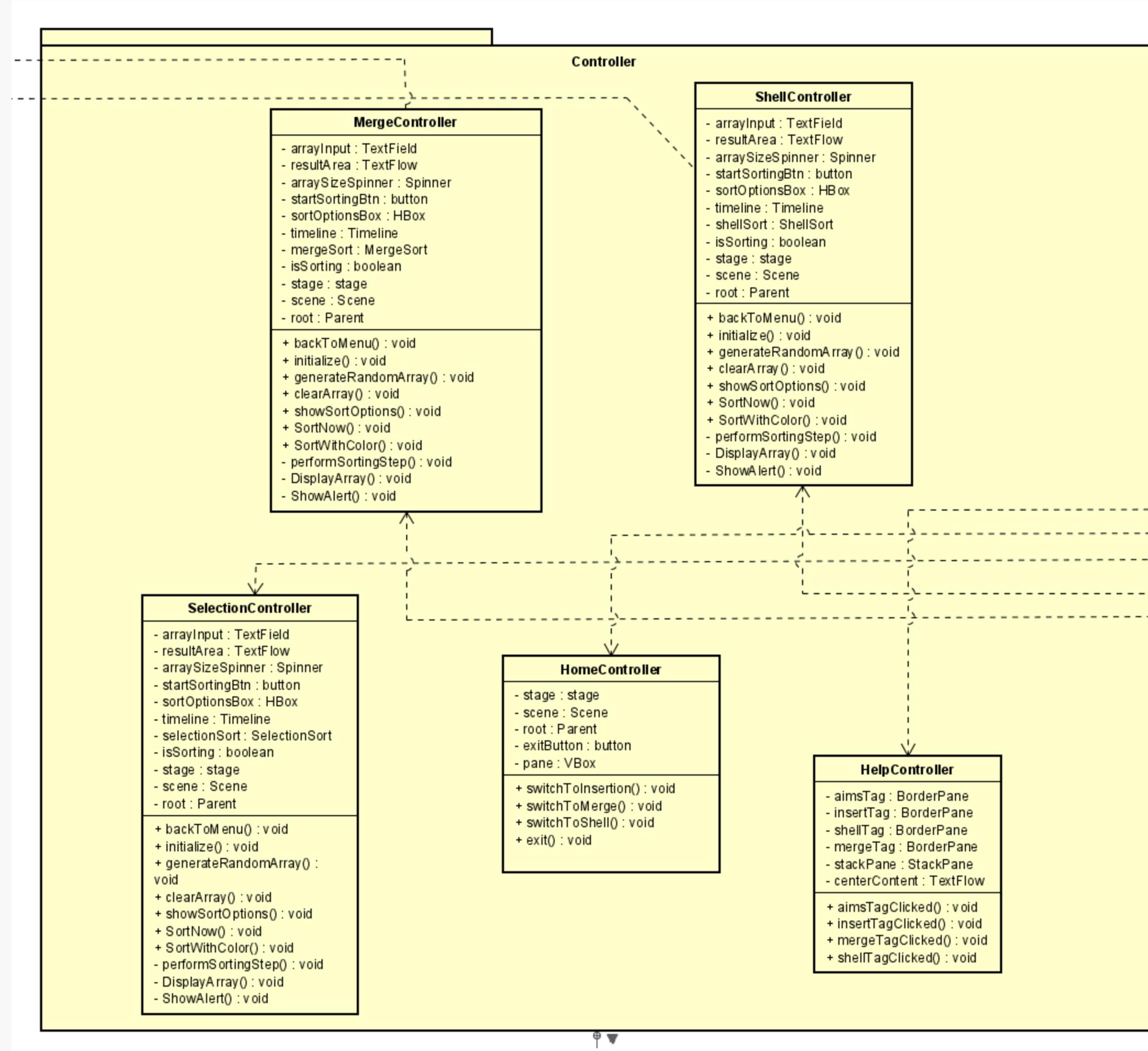
Model



View



Controller



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).
- Lớp MergeController, InsertionController và ShellController đều kế thừa từ lớp ControllerBase.

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

Dependency

- Các controller như InsertionController, MergeController và ShellController lần lượt sử dụng các phương thức từ các lớp Insertion, Merge và Shell để thực hiện logic sắp xếp tương ứng

Composition

- Các Lớp thuật toán Sort sử dụng StateSorting và StateSwap để biểu thị trạng thái sắp xếp.

OOP Techniques

Polymorphism

- Các phương thức `getStateSorting()` và `getSwapSorting()` được định nghĩa trừu tượng 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:





Thank you



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

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

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()).

