



HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Demonstration of sorting algorithms on array

ONE LOVE. ONE FUTURE.

Phần 1: Phân công nhiệm vụ

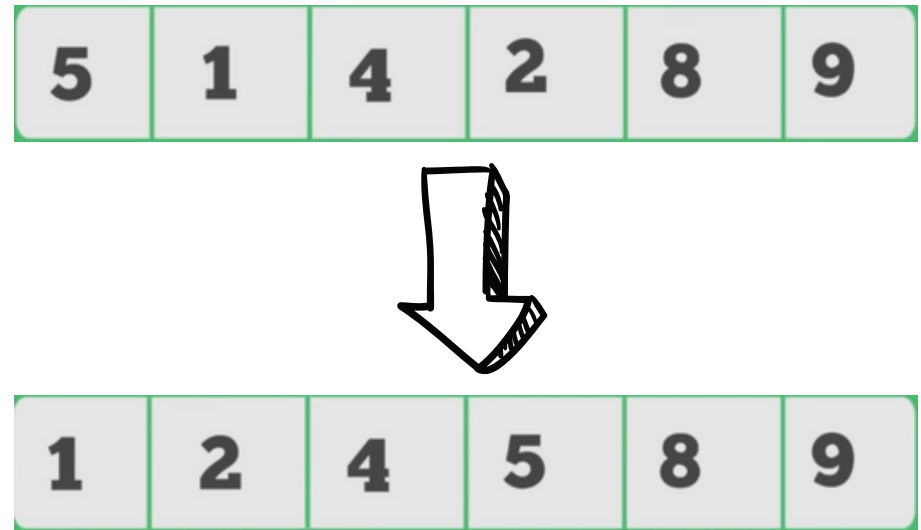
- Phạm Tuấn Minh: Class diagram (60%), Giao diện input, Slide (20%), Fix bug
- Nguyễn Đức Minh: Class diagram (20%), Cài đặt package SortAlgorithm (40%) , Slide (40%)
- Nguyễn Tuấn Minh: Class diagram (20%), Cài đặt package SortAlgorithm (40%) , Slide (40%)
- Trần Đại Quang Minh: Use case(50%), Báo cáo(50%), Giao diện Menu (50%), Cài đặt packet SortAlgorithm (10%)
- Nguyễn Nhật Minh: Use case(50%),Báo cáo(50%), Giao diện Menu (50%), Cài đặt packet SortAlgorithm (10%)

Phần 2: Phát biểu vấn đề

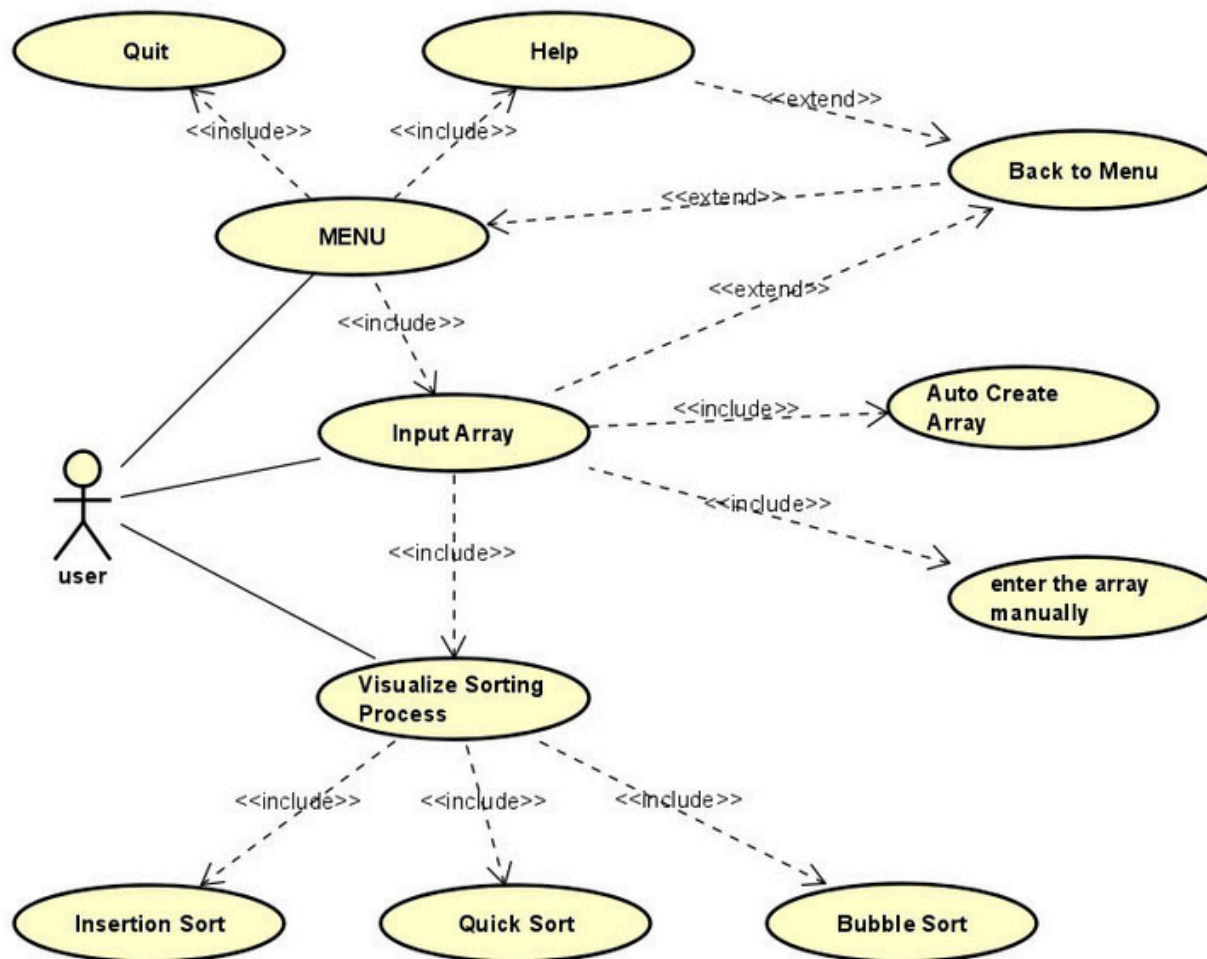
Trình diễn thuật toán sắp xếp trên mảng

Trình diễn 3 thuật toán sắp xếp cơ bản:

- Bubble sort
- Quick sort
- Insertion sort

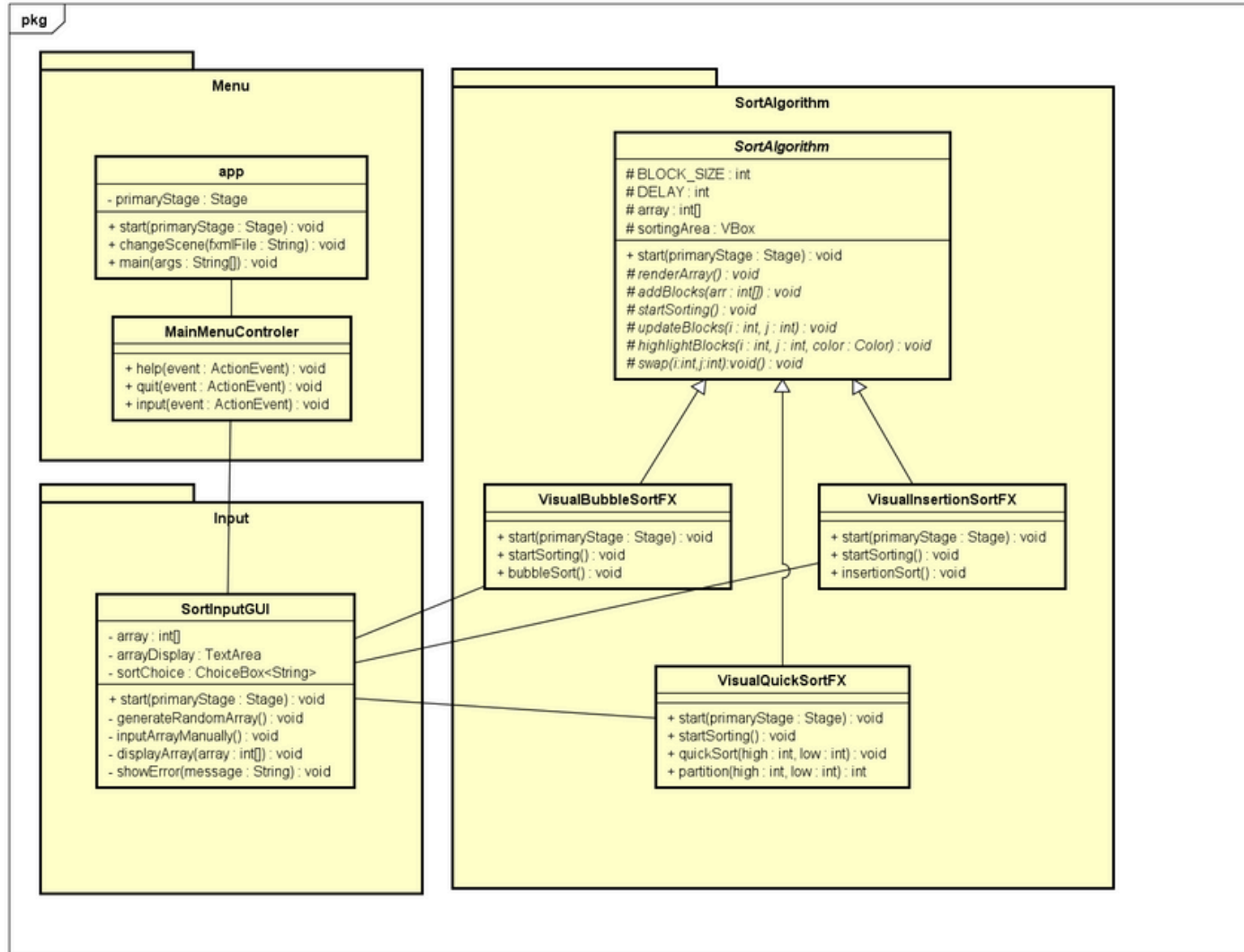


Phần 3: Sơ đồ use case



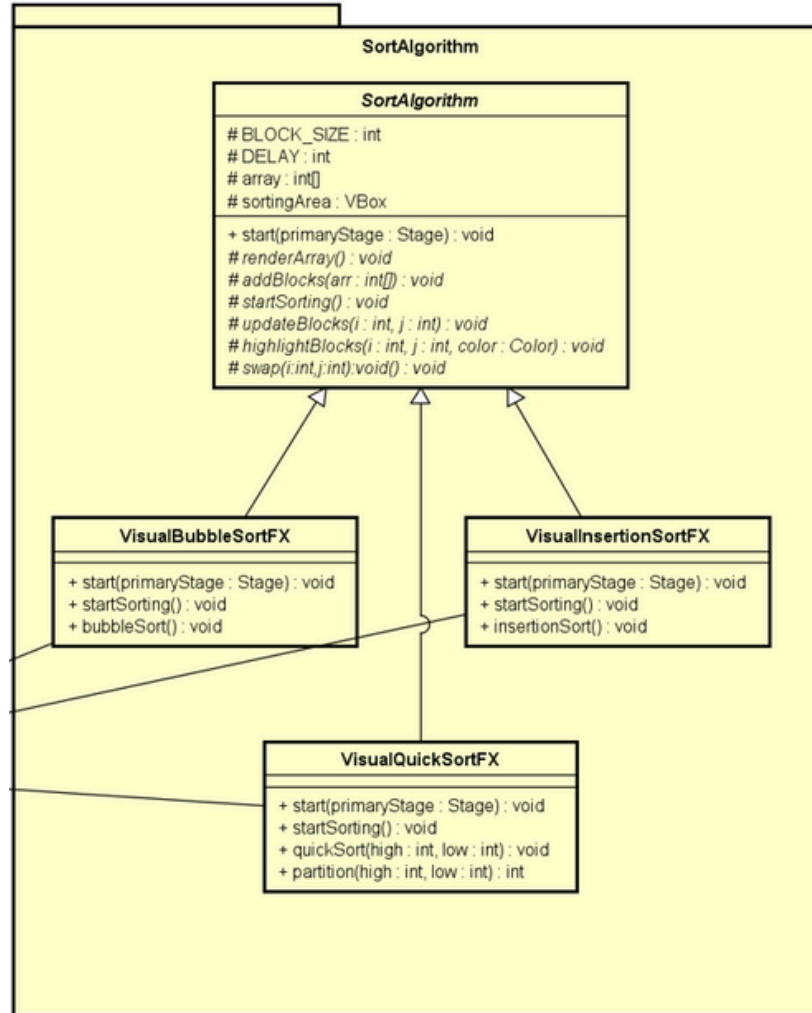
Phần 4: Sơ đồ class diagram

4.1. Sơ đồ tổng quát



Phần 4: Sơ đồ class diagram

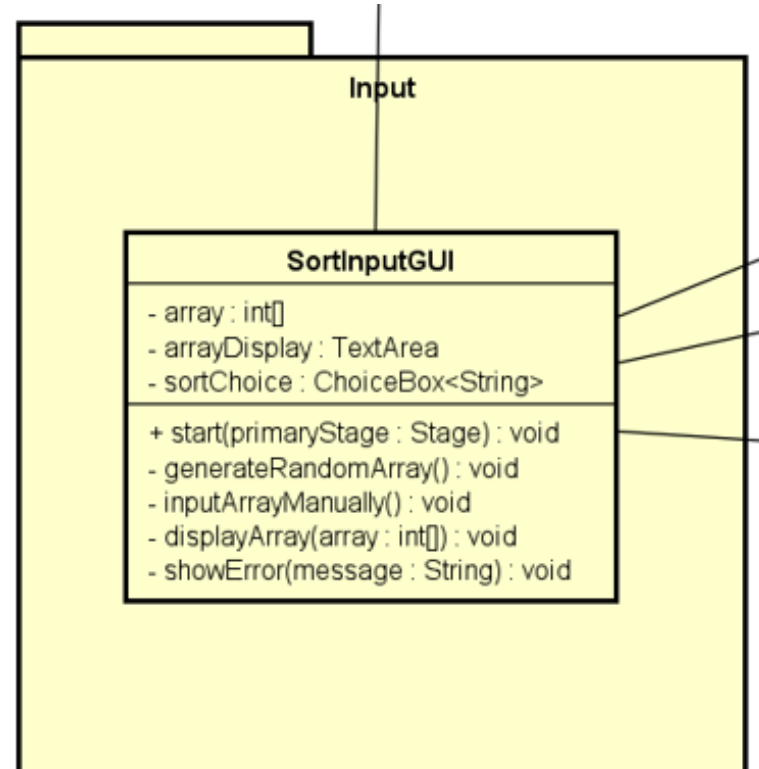
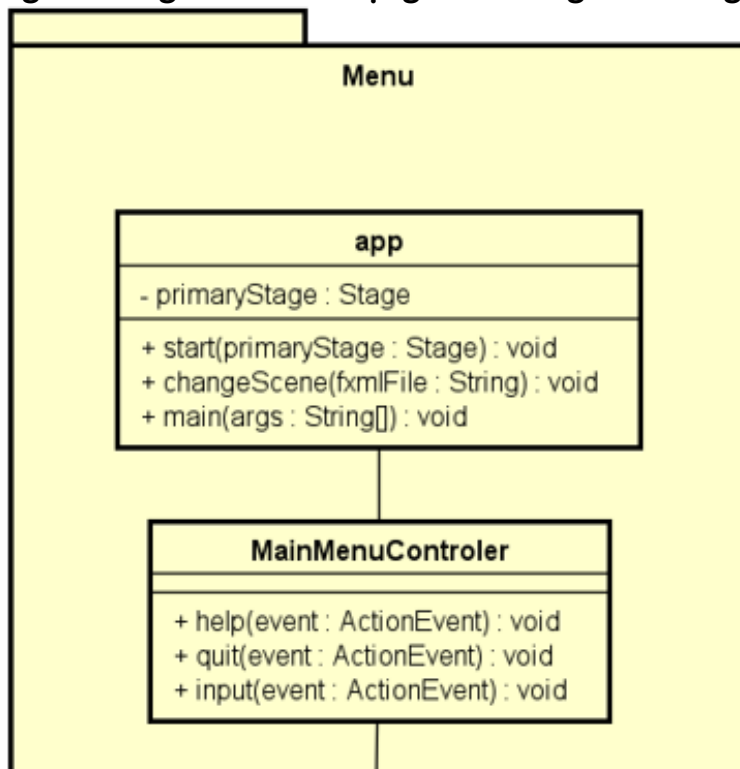
4.2. Sơ đồ lớp SortAlgorithm



Phần 5: Kỹ thuật OOP áp dụng trong Project

5.1. Đóng gói (Encapsulation):

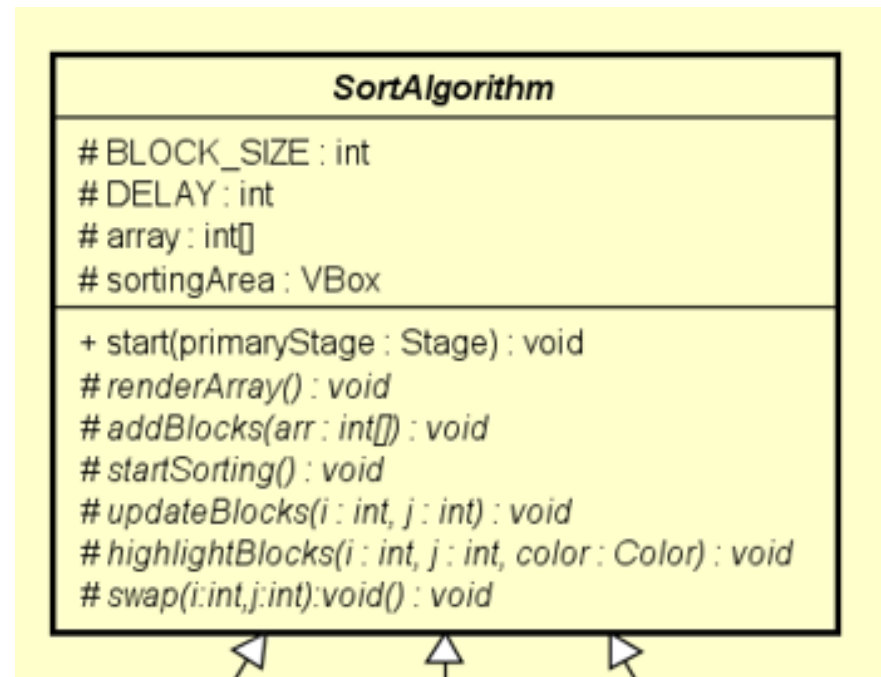
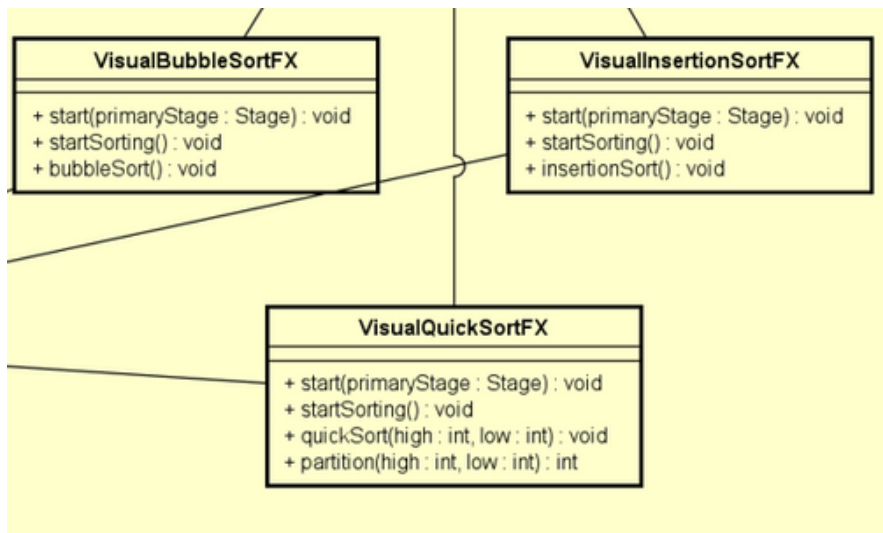
- Bảo vệ dữ liệu: Mảng array và vùng hiển thị arrayDisplay trong lớp SortInputGUI được đóng gói, tránh việc truy cập trực tiếp từ bên ngoài. Các phương thức như generateRandomArray() và inputArrayManually() quản lý và thay đổi mảng, bảo vệ dữ liệu.
- Đóng gói hành vi: Hành vi như tạo mảng ngẫu nhiên hoặc nhập mảng thủ công được đóng gói trong các phương thức, giúp người dùng chỉ cần sử dụng chức năng mà không cần biết chi tiết cách thức.



Phần 5: Kỹ thuật OOP áp dụng trong Project

5.2. Kế thừa (Inheritance):

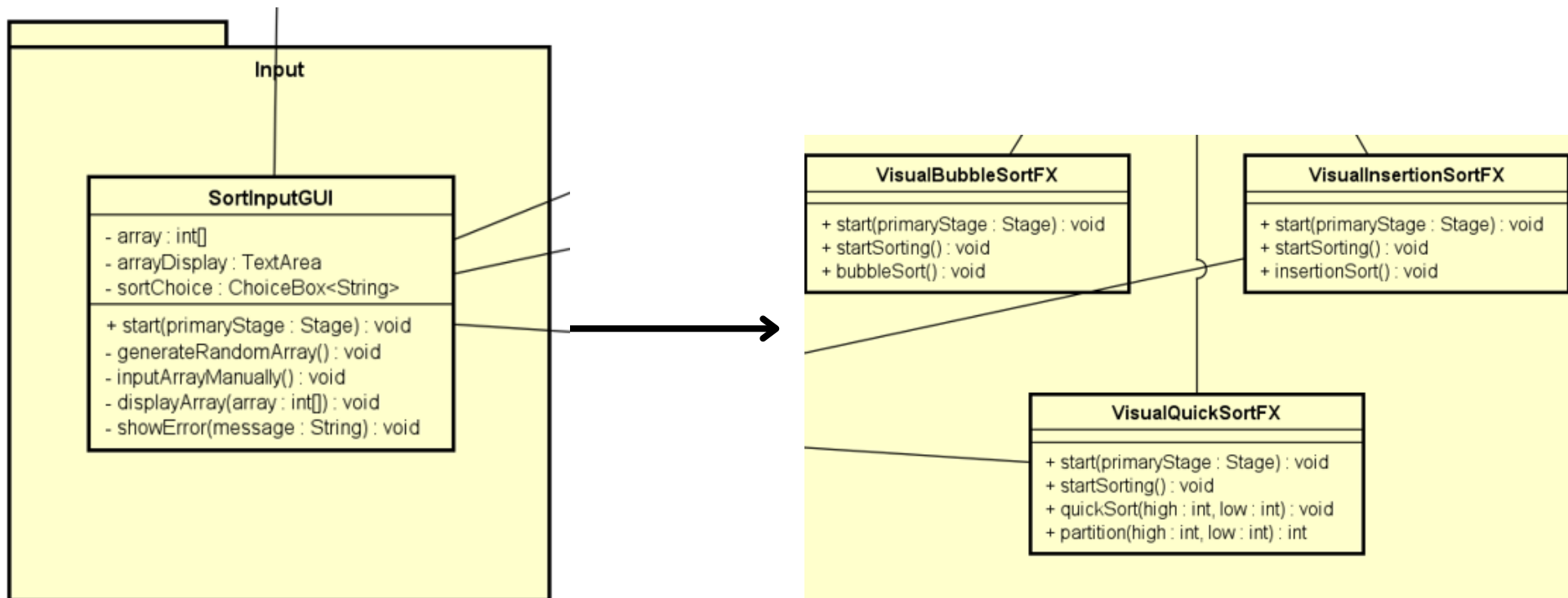
- Tái sử dụng mã: Các lớp `VisualBubbleSortFX`, `VisualInsertionSortFX`, và `VisualQuickSortFX` có thể kế thừa từ lớp `SortAlgorithm`, tái sử dụng các phương thức chung như `start()` mà không phải viết lại mã cho từng thuật toán.
- Mở rộng dễ dàng: Việc kế thừa giúp mở rộng mã mà không thay đổi lớp `SortInputGUI`. Thêm thuật toán mới chỉ cần tạo lớp mới kế thừa từ `SortAlgorithm`, giúp giao diện người dùng không cần thay đổi.



Phần 5: Kỹ thuật OOP áp dụng trong Project

5.3. Đa hình (Polymorphism):

- Thay đổi hành vi đối tượng: Đa hình được thể hiện khi người dùng chọn thuật toán sắp xếp. Các đối tượng thuật toán sẽ sử dụng phương thức start() để thực hiện thuật toán, nhưng hành vi sẽ khác nhau tùy thuộc vào thuật toán chọn.
- Đa hình trong GUI: Các đối tượng thuật toán có hành vi khác nhau khi gọi phương thức start(), giúp tăng tính linh hoạt và không cần thay đổi mã trong giao diện người dùng.



Phần 5: Kỹ thuật OOP áp dụng trong Project

5.4.Trừu tượng (Abstraction):

- Che giấu chi tiết thực hiện:Các chi tiết thuật toán như Bubble Sort, Insertion Sort, và Quick Sort được ẩn trong các lớp như VisualBubbleSortFX, người dùng chỉ cần chọn thuật toán.
- Giao diện rõ ràng:Lớp SortInputGUI cung cấp giao diện rõ ràng để người dùng nhập mảng và chọn thuật toán mà không cần biết các chi tiết nội bộ của các thuật toán.

<i>SortAlgorithm</i>
<i># BLOCK_SIZE : int</i> <i># DELAY : int</i> <i># array : int[]</i> <i># sortingArea : VBox</i>
<i>+ start(primaryStage : Stage) : void</i> <i># renderArray() : void</i> <i># addBlocks(arr : int[]) : void</i> <i># startSorting() : void</i> <i># updateBlocks(i : int, j : int) : void</i> <i># highlightBlocks(i : int, j : int, color : Color) : void</i> <i># swap(i:int,j:int):void() : void</i>

A large, stylized graphic of the HUST logo, composed of a circular arrangement of red dots of varying sizes, creating a halftone effect. The dots are arranged in a way that they form a large, irregular circle, with the word 'HUST' centered within it.

HUST

*Thank
you!*



hust.edu.vn



fb.com/dhbkhn