Hanoi University of Science & Technology

PROJECT REPORT

Demonstration of sorting algorithms on array

Object – Oriented Programming

Instructor: Prof. Nguyen Thi Thu Trang

Group 15 : Nguyễn Xuân Hưng 20194585

Nguyễn Huy Hoàng 20210373

Nguyễn Huy Hoàng 20215056

Lê Vũ Hoàng 20205082

1 . Member and Assignment

Nguyễn Xuân Hưng 20194585

* BubbleSort Class
* Demo video
* Slide
* BubbleSort.fxml, SortController

Nguyễn Huy Hoàng 20210373

* HeapSort Class
* CreateArray Class
* Main Class
* Report
* MainMenu.fxml, About.fxml, MainScreen Controller

Nguyễn Huy Hoàng 20215056

* ShellSort Class
* GeneralSort Class
* HeapSort.fxml, ShellSort.fxml

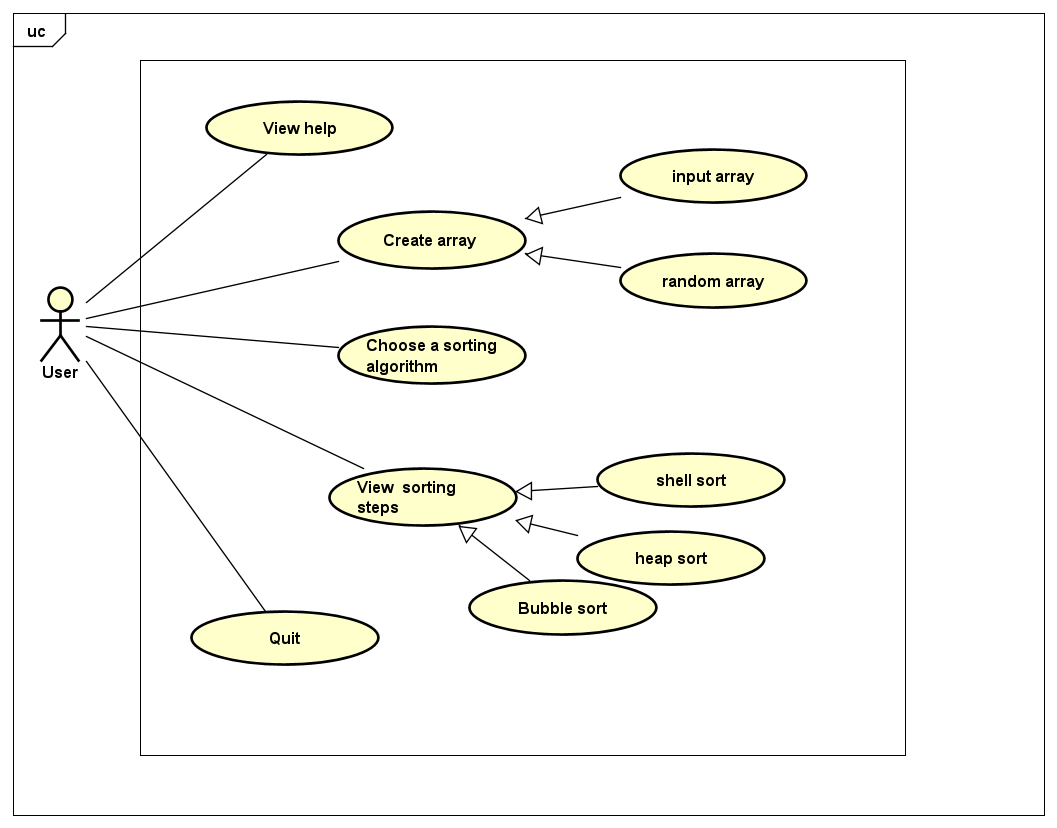
Lê Vũ Hoàng 20205082 : X

2. Description

2.1 Requirement of the project

* Sorting algorithm is an algorithm that puts elements of a list in a certain order. Efficient sorting is important for optimizing the efficiency of many other algorithms. A lots of sorting algorithm have been developed so far, making it difficult to understanding and remembering all of them. This project purpose is to help people get a better understanding 3 basic sorting algorithms: bubble sort, heap sort, and shell sort through a visualizing application showing each step of the algorithms. The application is built in Java with the GUI created using JavaFX.

2.2 Use case explaination



Sau khi người dùng khởi động ứng dụng, sẽ xuất hiện 1 cửa sổ menu . Người dùng có thể thao tác :

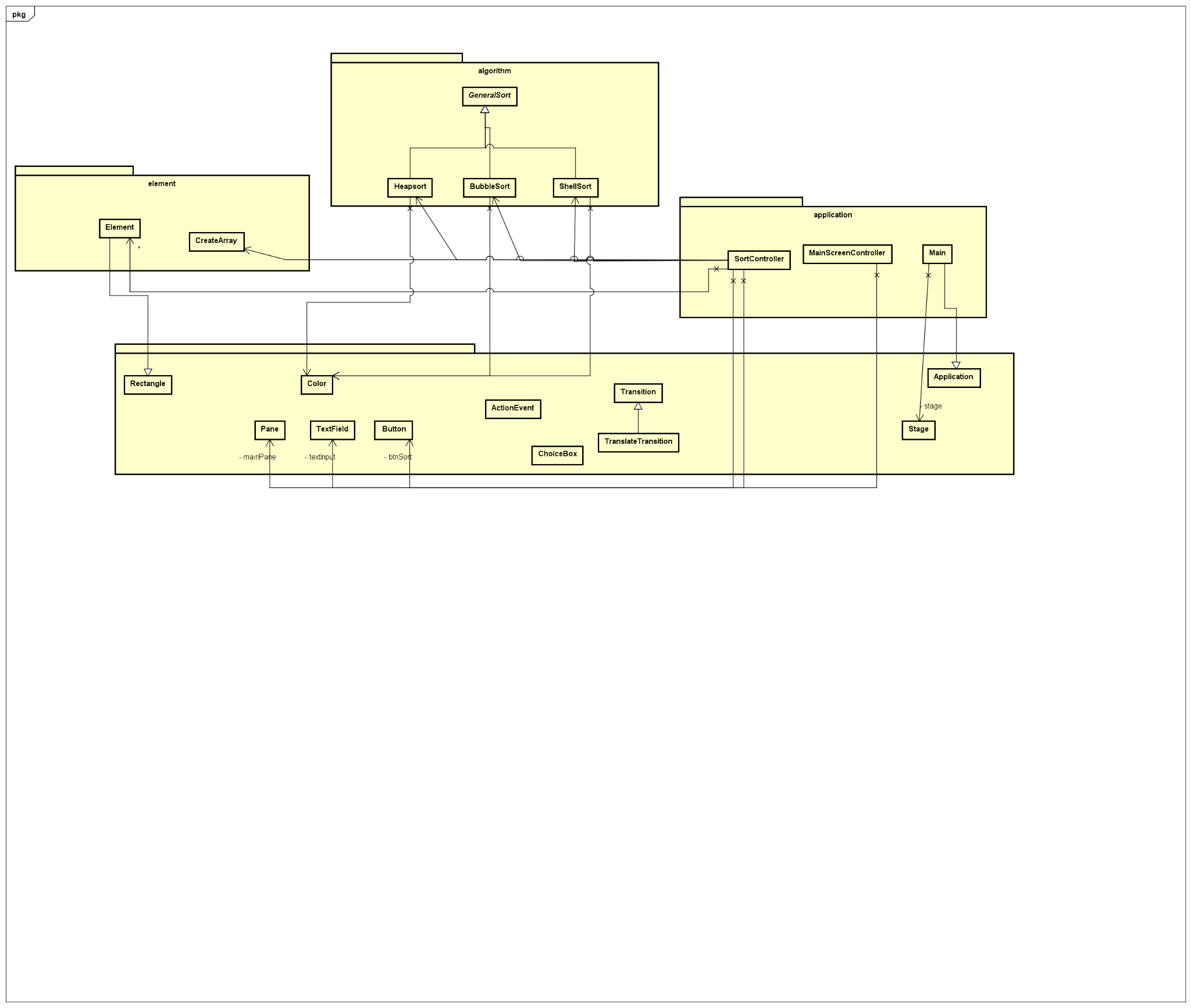
* View help : Sẽ cho biết chức năng cơ bản và mục đích của chương trình
* Quit : Thoát khỏi chương trình
* Choose a sorting algorithms : Chọn 1 trong 3 nút hiển thị để xem cửa sổ của thuật toán sắp xếp

Sau khi chọn 1 trong 3 thuật toán, sẽ xuất hiện màn hình của thuật toán, Tại đây người dùng có thể :

* Khởi tạo 1 mảng ngẫu nhiên : Bằng cách chọn độ dài và kiểu của mảng
* Nhập mảng: Nhập trực tiếp mảng thông qua bàn phím ( Chương trình chỉ chấp nhận dữ liệu là số nguyên)
* Xem các bước sắp xếp : sau khi tạo mảng xong người dùng có thể thao tác bằng việc click vào nút sort button để xem tiến trình thực hiện việc sắp xếp

3. Design

3.1 General class diagram



Quan hệ giữa các gói:

- Element class kế thừa từ Rectangle class

- bubble sort, heap sort, shell sort kế thừa từ lớp GeneralSort

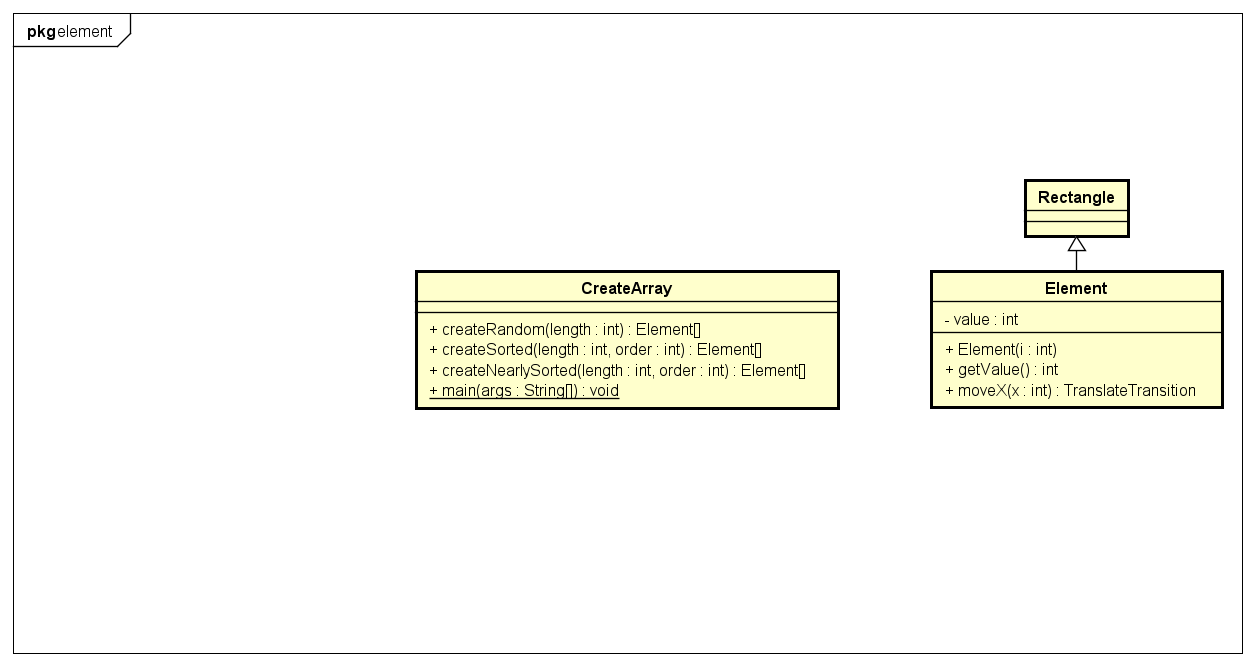
- Lớp SortController liên kết với CreateArray class, bubble sort, heap sort, and shell sort

- Main class thừa từ Application class và liên kết với Stage class.

- SortController và MainScreenController được liên kết với nhiều phần tử từ JavaFX để tạo GUI.

3.2 Several class diagrams

Element :

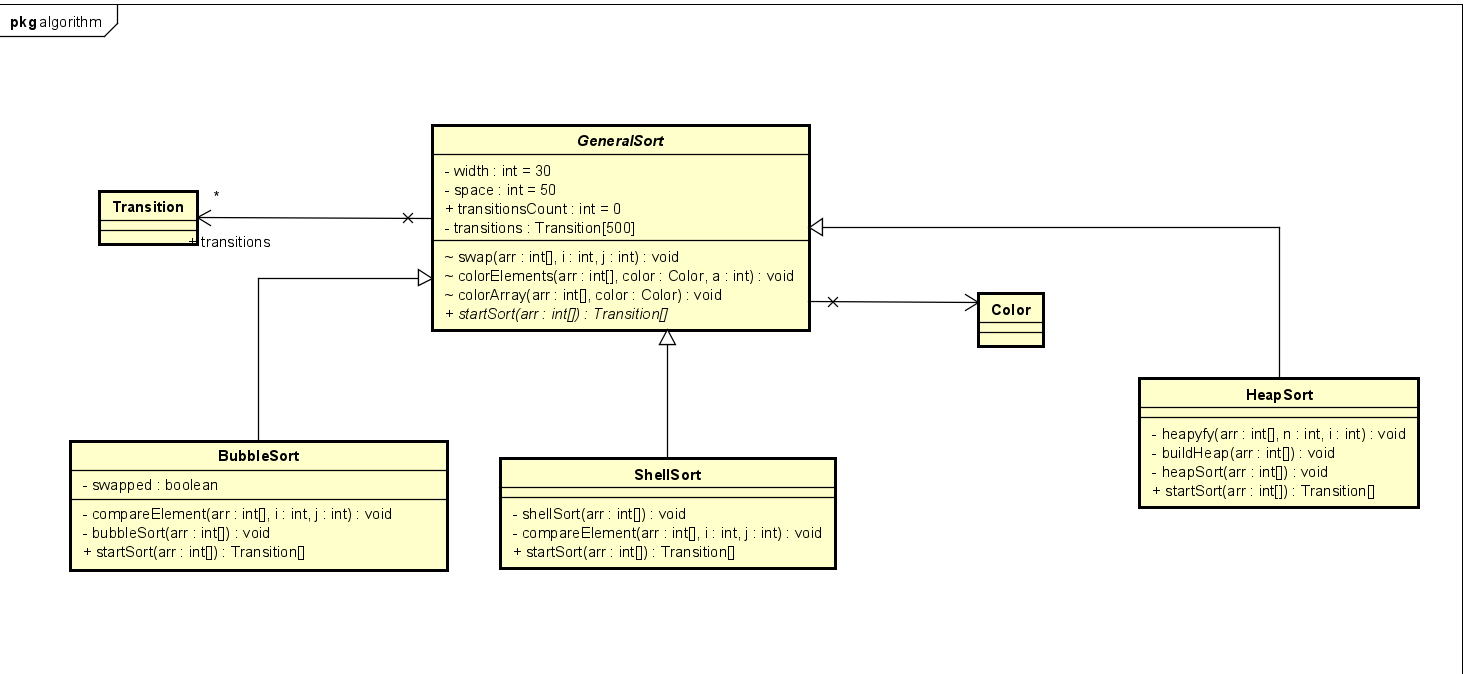


Một số method quan trọng:

⦁ createRandom: tạo một mảng phần tử ngẫu nhiên với độ dài, giá trị nhất định của mỗi phần tử nằm trong khoảng từ 1 đến length.

⦁ createSorted: tạo một mảng với các phần tử từ 1 đến độ dài, thứ tự giảm dần hoặc tăng dần.

⦁ createNearlySorted: tạo một mảng được sắp xếp, sau đó hoán đổi ngẫu nhiên một số phần tử

Algorithm : 

Một số phương pháp quan trọng:

⦁ Swap : Hoán đổi hai phần tử trong mảng và thêm một chuyển tiếp song song hoán đổi hai hình chữ nhật tương ứng trên màn hình.

⦁ colorElements: thêm chuyển tiếp song song, chứa chuyển tiếp tô của một số phần tử cụ thể.

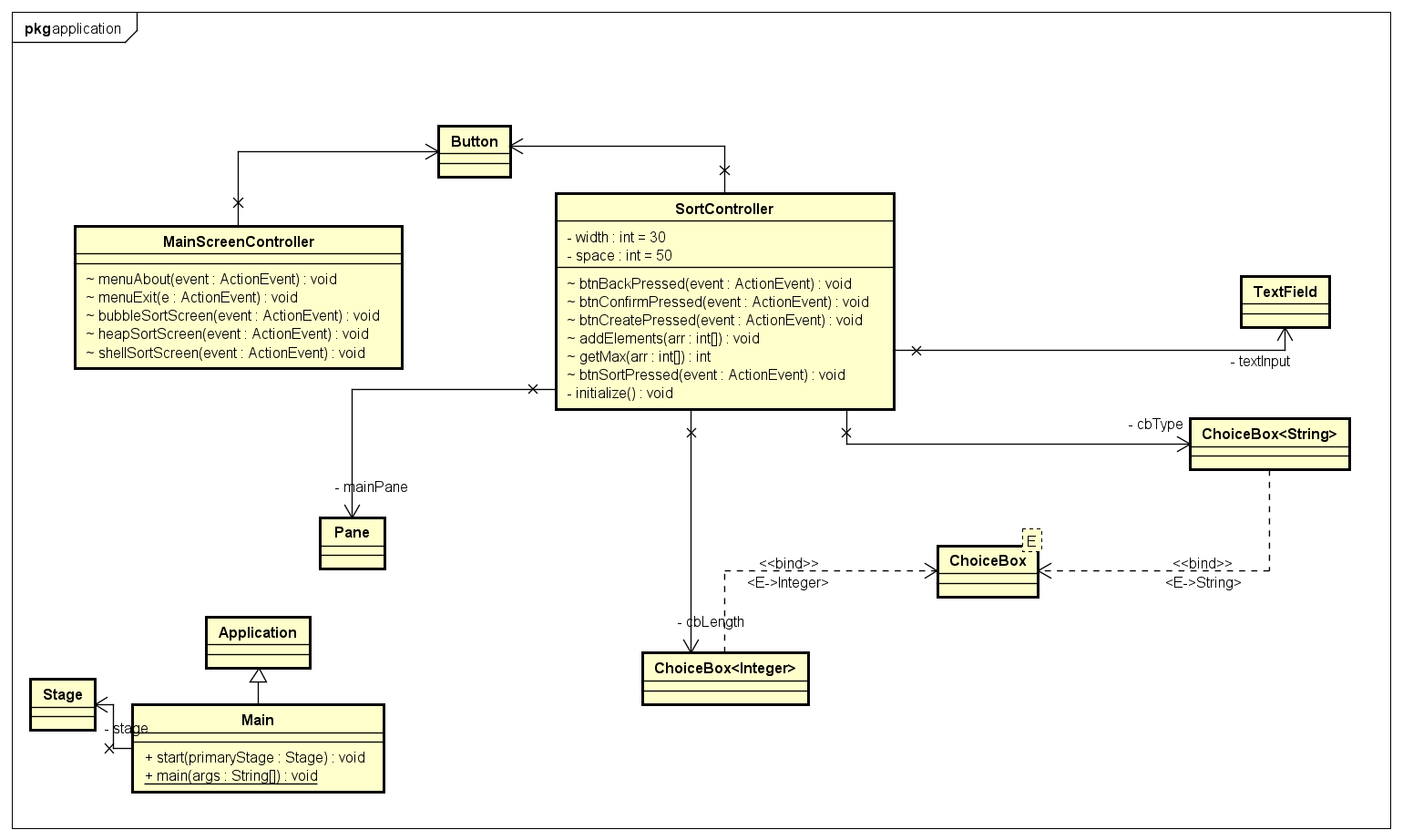
⦁ colorArray: thêm một chuyển tiếp song song, chứa các chuyển tiếp tô của tất cả các phần tử trong mảng.

Bubble Sort: Sắp xếp bằng cách so sánh và đổi chỗ các phần tử kề nhau cho đến khi mảng được sắp xếp hoàn toàn.

Heap Sort: Sử dụng cấu trúc heap nhị phân để xây dựng một mảng sắp xếp bằng cách đổi chỗ liên tục nút gốc với phần tử cuối cùng.

Shell Sort: Giảm dần khoảng cách giữa các phần tử để thực hiện sắp xếp chèn, bắt đầu với khoảng cách lớn và kết thúc bằng một lần đi qua cuối cùng với khoảng cách là 1.

Application :



Một số method quan trọng:

⦁ btnSortPressed: Vì chúng ta chỉ sử dụng một bộ điều khiển cho 3 tệp fmxl, chúng ta tạo một thể hiện của một lớp sắp xếp dựa trên văn bản trên nút sắp xếp (bubble sort, heap sort, shell sort). Chúng ta sử dụng thể hiện lớp sort này để thực hiện thuật toán sắp xếp trên mảng đã tạo.