ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

----------------------------------



**BÁO CÁO MINI-PROJECT**

Môn: Lập Trình Hướng Đối Tượng

*Mã học phần: IT3103*

*Mã lớp học: 151965*

*GVHD: Nguyễn Thị Thu Trang*

*Nhóm 27*

|  |  |
| --- | --- |
| Bùi Minh Tuấn | 20225678 |
| Trần Xuân Tú | 20225677 |
| Đoàn Thanh Tùng | 20225946 |
| Lương Nguyễn Quốc Tùng | 20215161 |
| Đặng Xuân Tùng | 20205042 |

*Hà Nội, 24/12/2024*

Table of Contents

[I. Assignment of Members 3](#_Toc185981416)

[II. Mini-Project Description 4](#_Toc185981417)

[1. Tên đề tài 4](#_Toc185981418)

[2. Mô tả chi tiết yêu cầu của dự án 4](#_Toc185981419)

[3. Use Case Diagram 4](#_Toc185981420)

[3.1. Actor: 5](#_Toc185981421)

[3.2. Use Cases: 5](#_Toc185981422)

[3.3. Giải thích cách người dùng tương tác 5](#_Toc185981423)

[III. Design 7](#_Toc185981424)

[1. Sơ đồ lớp chung (General class diagram) 7](#_Toc185981425)

[2. Một số sơ đồ lớp cho mỗi gói (Class diagrams for each package) 8](#_Toc185981426)

[2.1. Sơ đồ lớp gói datastructure 8](#_Toc185981427)

[2.2. Sơ đồ lớp gói view 9](#_Toc185981428)

[2.3. Sơ đồ lớp gói controller 10](#_Toc185981429)

[3. Mối quan hệ giữa các lớp 11](#_Toc185981430)

[3.1. Kế thừa 11](#_Toc185981431)

[3.2. Liên kết 11](#_Toc185981432)

[3.3. Hợp thành 12](#_Toc185981433)

[3.4. Đa hình 12](#_Toc185981434)

Table of Figures

[Figure 1: Use Case Diagram 5](#_Toc185980521)

[Figure 2: General class diagram 8](#_Toc185980522)

[Figure 3: Class diagram for datastructure package 9](#_Toc185980523)

[Figure 4: Class diagram for view package 10](#_Toc185980524)

[Figure 5: Class diagram for controller package 11](#_Toc185980525)

# Assignment of Members

* Bùi Minh Tuấn – 20225678: List; ListScene; (Create, Delete, Sort, Find) MenuController; SceneController; utility package; modify the Design; make Presentation. (35%)
* Trần Xuân Tú – 20225677: Queue, QueueScene; modify CreateMenuController, modify the Design, write Report. (25%)
* Đoàn Thanh Tùng – 20225946: Datastructure; Stack; StackScene; BaseScene; HelpScene; modify CreateMenuController; PushMenuController; Design. (35%)
* Lương Nguyễn Quốc Tùng – 20215161: MenuScene. (5%)
* Đặng Xuân Tùng – 20205042: Can not contact (0%)

# Mini-Project Description

## Tên đề tài

* Topic: Demonstration of operations on stack, queue, list. (Trình bày cách hoạt động của ngăn xếp, hàng đợi, danh sách.)

## Mô tả chi tiết yêu cầu của dự án

Trong lĩnh vực khoa học máy tính, các cấu trúc dữ liệu cơ bản như ngăn xếp (stack), hàng đợi (queue) và danh sách (list) được sử dụng thường xuyên. Vì vậy hãy thiết kế một chương trình để hiển thị và giải thích một số hoạt động cơ bản trên các cấu trúc này. Chương trình được thiết kế sẽ cung cấp giao diện để thực hiện các hoạt động như tạo (create), chèn (insert), sắp xếp (sort), tìm kiếm (find), xóa (delete) các phần tử trong các cấu trúc cơ bản của ngăn xếp (stack), hàng đợi (queue) và danh sách (list); giúp người dùng hiểu rõ hơn về cách thức hoạt động của chúng trong lập trình và khoa học máy tính.

## Use Case Diagram

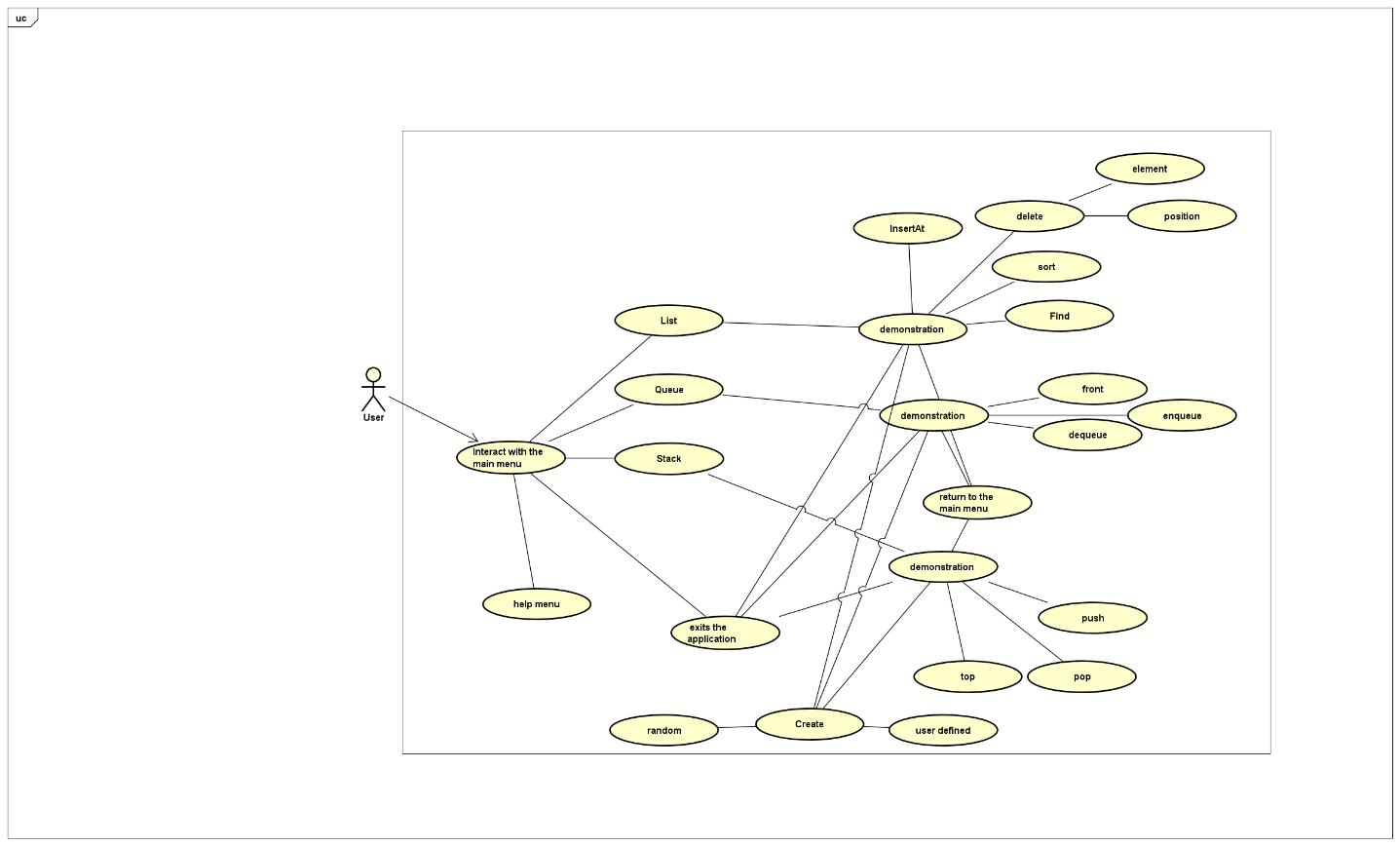


Figure 1: Use Case Diagram

### ****Actor:****

* **User:** Người dùng sử dụng hệ thống, là đối tượng trực tiếp tương tác với chương trình.

### ****Use Cases:****

* **Interact with the main menu:** Người dùng tương tác với menu chính để bắt đầu sử dụng các chức năng trong chương trình.
* **Select Data Structure Type:** Chọn loại cấu trúc dữ liệu cụ thể mà người dùng muốn làm việc.
* **Stack, Queue, List:** Các cấu trúc dữ liệu cụ thể mà người dùng có thể tương tác và làm việc.
* **Create:** Tạo một cấu trúc dữ liệu mới (Stack, Queue, List) có thể random hoặc ngườ dùng tự nhập.
* **Insert:** Chèn một phần tử vào cấu trúc dữ liệu.
* **Delete:** Xóa một phần tử khỏi cấu trúc dữ liệu.
* **Find:** Tìm kiếm một phần tử trong cấu trúc dữ liệu.
* **Sort:** Sắp xếp các phần tử trong cấu trúc dữ liệu.
* **Enqueue, Dequeue:** Thêm, lấy phần tử trong Queue.
* **Push, Pop:** Thêm, lấy phần tử trong Stack.
* **Top, Front:** Đưa ra phần tử đầu của Queue/ Stack.
* **Return to the main menu:** Quay lại menu chính để thực hiện các thao tác khác.
* **Help menu:** Truy cập vào mục trợ giúp để tìm hiểu thêm về cách sử dụng chương trình.
* **Exit the application:** Thoát khỏi chương trình.

### Giải thích cách người dùng tương tác

* **Khởi động chương trình:** Người dùng mở chương trình và sẽ thấy giao diện chính với menu chứa các chức năng cơ bản.
* **Chọn loại cấu trúc dữ liệu:** Người dùng chọn loại cấu trúc dữ liệu mà họ muốn làm việc (ngăn xếp, hàng đợi hoặc danh sách).
* **Thực hiện các thao tác:** Sau khi chọn loại cấu trúc dữ liệu, người dùng có thể thực hiện các thao tác như:
  + **Tạo:** Tạo một cấu trúc dữ liệu mới, ban đầu thường rỗng.
  + **Chèn:** Thêm một phần tử vào một vị trí cụ thể trong cấu trúc dữ liệu.
  + **Xóa:** Xóa một phần tử khỏi cấu trúc dữ liệu.
  + **Tìm kiếm:** Tìm kiếm một phần tử trong cấu trúc dữ liệu.
  + **Sắp xếp:** Sắp xếp các phần tử trong cấu trúc dữ liệu theo một thứ tự nhất định.
* **Chuyển đổi giữa các cấu trúc:** Nếu cần, người dùng có thể quay trở lại menu chính để chuyển đổi giữa các loại cấu trúc dữ liệu để thực hiện các thao tác khác.
* **Kết thúc:** Khi hoàn thành công việc, người dùng có thể thoát khỏi chương trình hoặc quay lại menu chính để thực hiện các thao tác khác.

# Design

## Sơ đồ lớp chung (General class diagram)

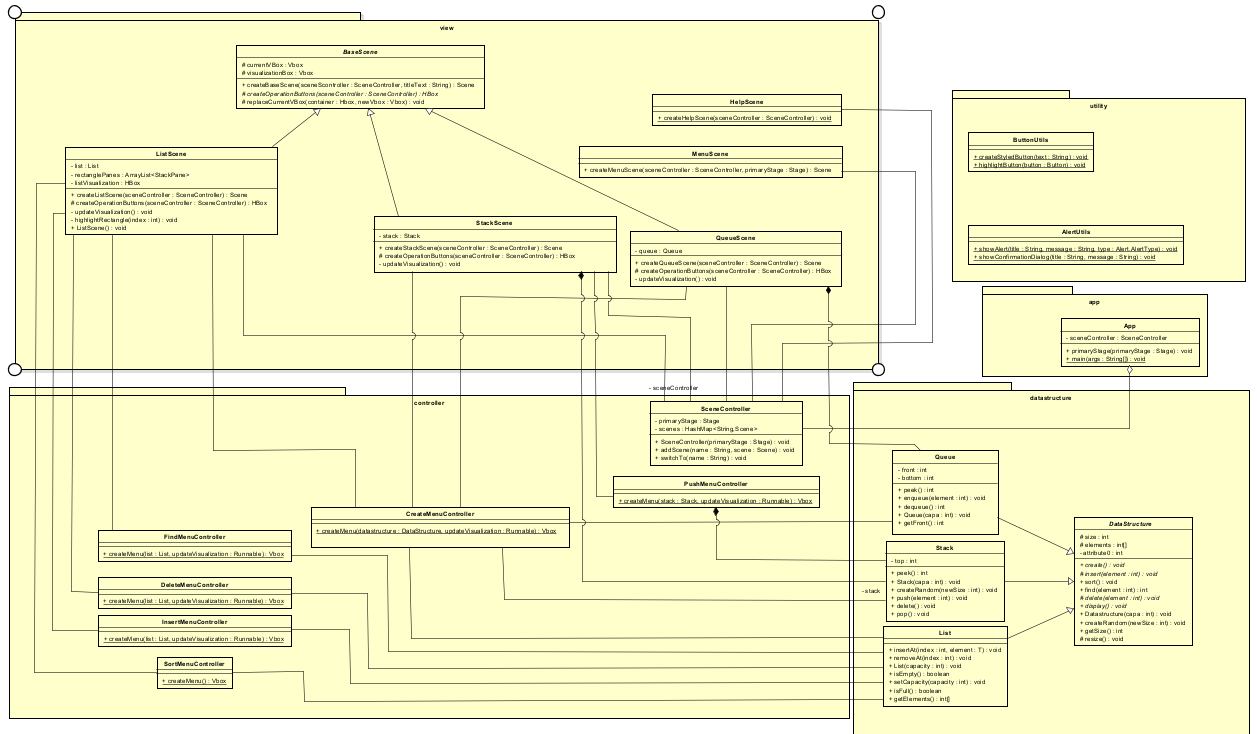


Figure 2: General class diagram

* Các lớp thuộc gói (package) datastructure thể hiện các cấu trúc dữ liệu ngăn xếp (stack), hàng đợi (queue) và danh sách (list) được sử dụng trong hệ thống.
* Các lớp thuộc gói (package) view định nghĩa các giao diện màn hình như màn hình menu, help, ngăn xếp (stack), hàng đợi (queue) và danh sách (list) và các thành phần giao diện (ví dụ: nút, nhãn) và các logic liên quan đến việc hiển thị và tương tác với người dùng.
* Gói (package) controller đóng vai trò trung tâm trong việc điều khiển luồng của chương trình. Các lớp trong gói này sẽ nhận các yêu cầu từ người dùng (thông qua giao diện), xử lý các yêu cầu đó và cập nhật giao diện tương ứng.
* Lớp App là lớp chính của chương trình
* Ngoài ra còn có các lớp thuộc gói(package) utility cung cấp các chức năng hỗ trợ như tạo nút, hiển thị thông báo.

## Một số sơ đồ lớp cho mỗi gói (Class diagrams for each package)

### Sơ đồ lớp gói datastructure

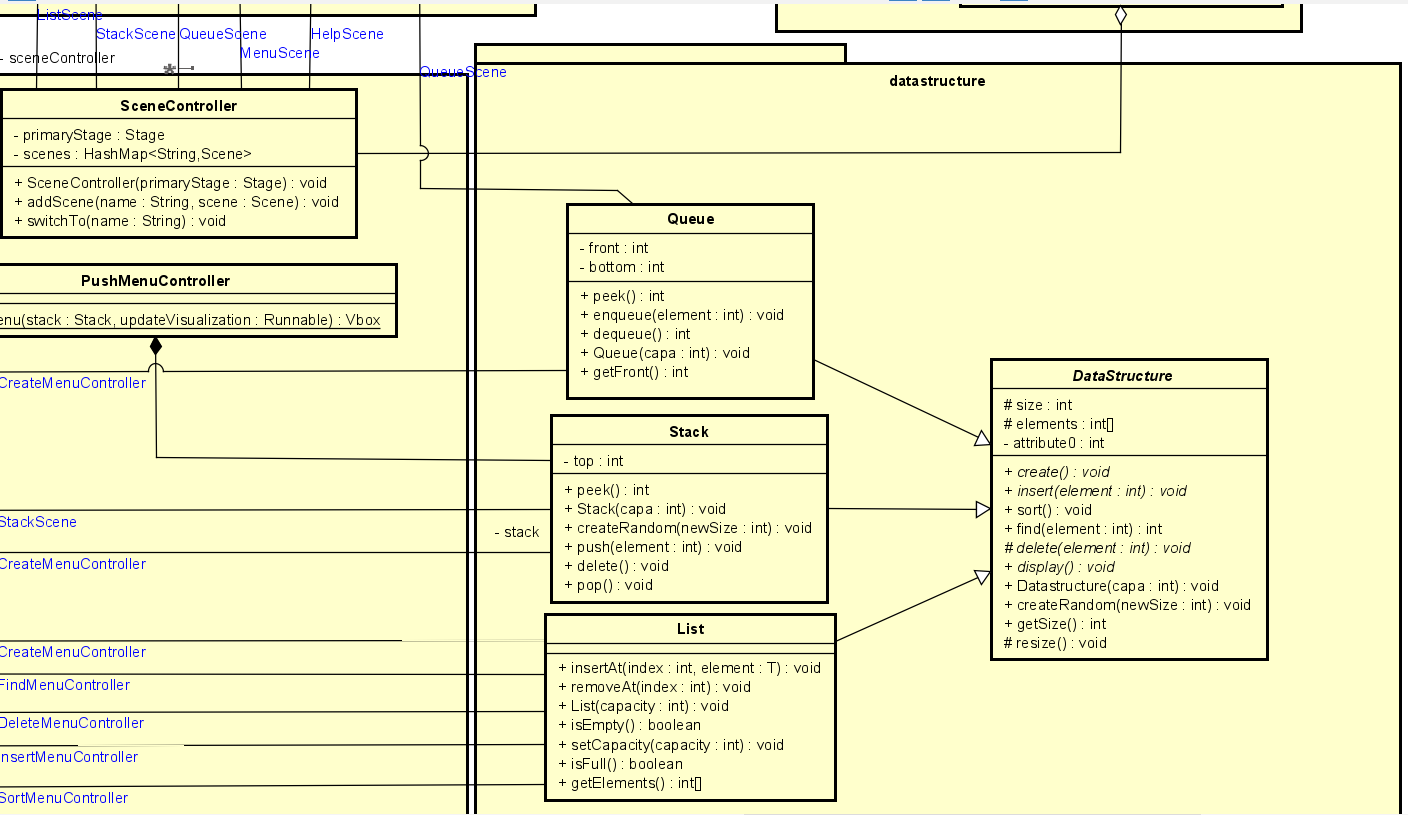


Figure 3: Class diagram for datastructure package

#### **Lớp DataStructure:**

* **Thuộc tính:**
  + elements Mảng lưu trữ các phần tử.
  + size: Số lượng phần tử hiện có.
* **Phương thức:**
  + Các phương thức cơ bản của các cấu trúc dữ liệu: create, insert, sort, find, delete, createRandom, resize, ...
* **Chức năng:**
  + Là lớp cơ sở cho các cấu trúc dữ liệu như Stack, Queue, List.

#### **Lớp Stack, Queue, List:**

* **Kế thừa:** Kế thừa từ lớp DataStructure.
* **Phương thức:**
  + Các phương thức cụ thể cho từng cấu trúc dữ liệu (ví dụ: push, pop cho stack; enqueue, dequeue cho queue; insert, remove cho list).
* **Chức năng:**
  + Đại diện cho các cấu trúc dữ liệu cụ thể của Stack, Queue, List.

### Sơ đồ lớp gói view

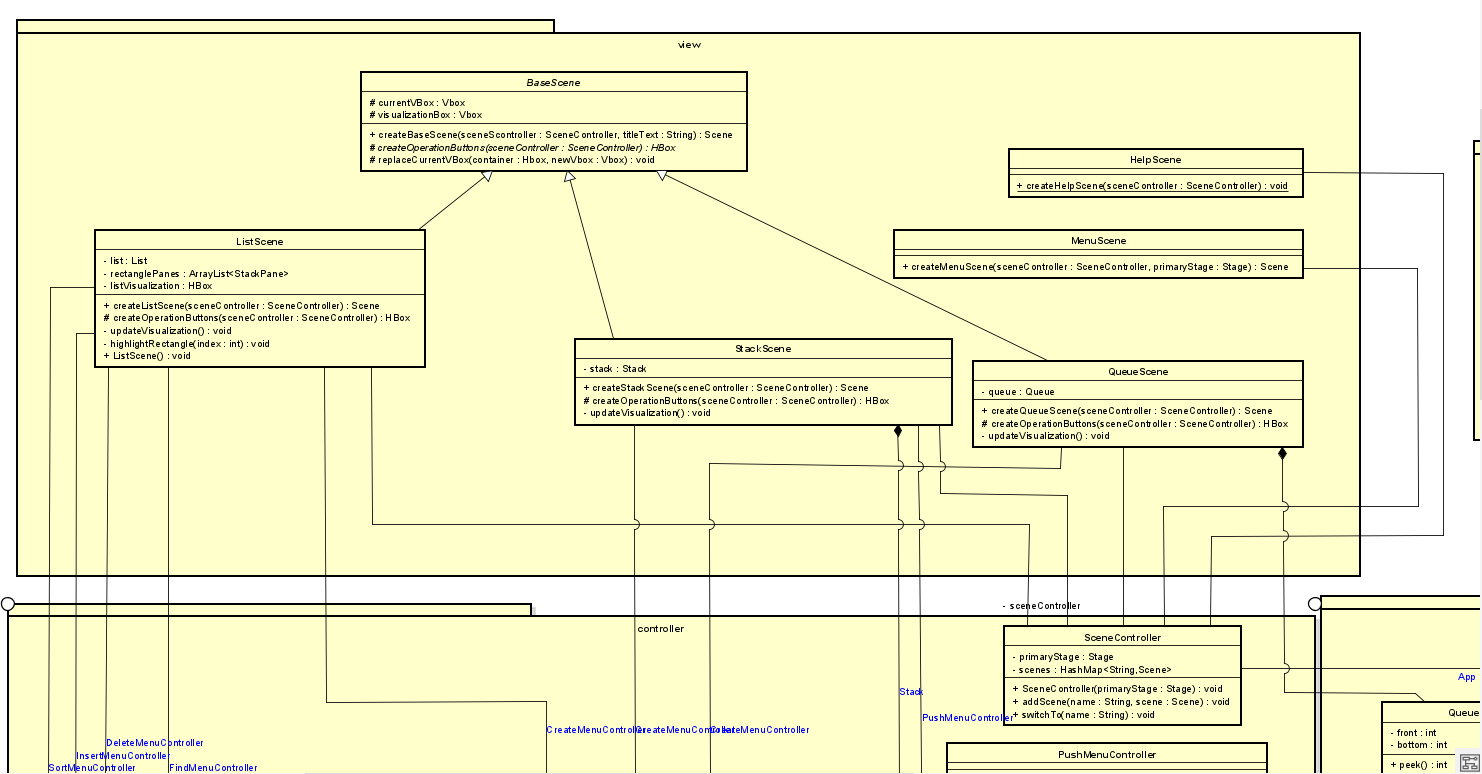


Figure 4: Class diagram for view package

#### **Lớp BaseScene:**

* **Thuộc tính:**
  + currentVBox: Một VBox (một container trong JavaFX) dùng để chứa các thành phần giao diện chính của scene.
  + visualizationBox: Một VBox khác, có thể được sử dụng để hiển thị trực quan hóa dữ liệu.
* **Phương thức:**
  + createBaseScene: Phương thức này tạo ra một Scene mới, thiết lập các thuộc tính cơ bản và trả về Scene đó.
  + replaceCurrentVBox: Phương thức này thay thế VBox hiện tại (currentVBox) bằng một VBox mới, cho phép cập nhật giao diện một cách linh hoạt.
* **Chức năng:** Là lớp cơ sở cho các scene (màn hình) trong chương trình. Nó cung cấp các thuộc tính và phương thức chung cho tất cả các scene, tạo ra một khuôn mẫu để các scene con kế thừa và mở rộng.

#### **Lớp StackScene, QueueScene, ListScene:**

* **Kế thừa:** Kế thừa từ lớp BaseScene.
* **Thuộc tính:**
  + list, stack, queue: Các đối tượng đại diện cho cấu trúc dữ liệu tương ứng.
  + rectangles: Một danh sách các hình chữ để trực quan hóa các phần tử trong cấu trúc dữ liệu.
* **Phương thức:**
  + Các phương thức để thực hiện các thao tác trên cấu trúc dữ liệu (ví dụ: thêm, xóa, tìm kiếm).
  + Các phương thức updateVisualization() để cập nhật giao diện khi cấu trúc dữ liệu thay đổi.
* **Chức năng:** Các lớp này đại diện cho các scene hiển thị và tương tác với các cấu trúc dữ liệu tương ứng (list, stack, queue).

#### **Lớp HelpScene, MenuScene:**

* **Chức năng:**
  + **HelpScene:** Hiển thị thông tin trợ giúp cho người dùng
  + **MenuScene:** Hiển thị menu chính của ứng dụng.

### Sơ đồ lớp gói controller

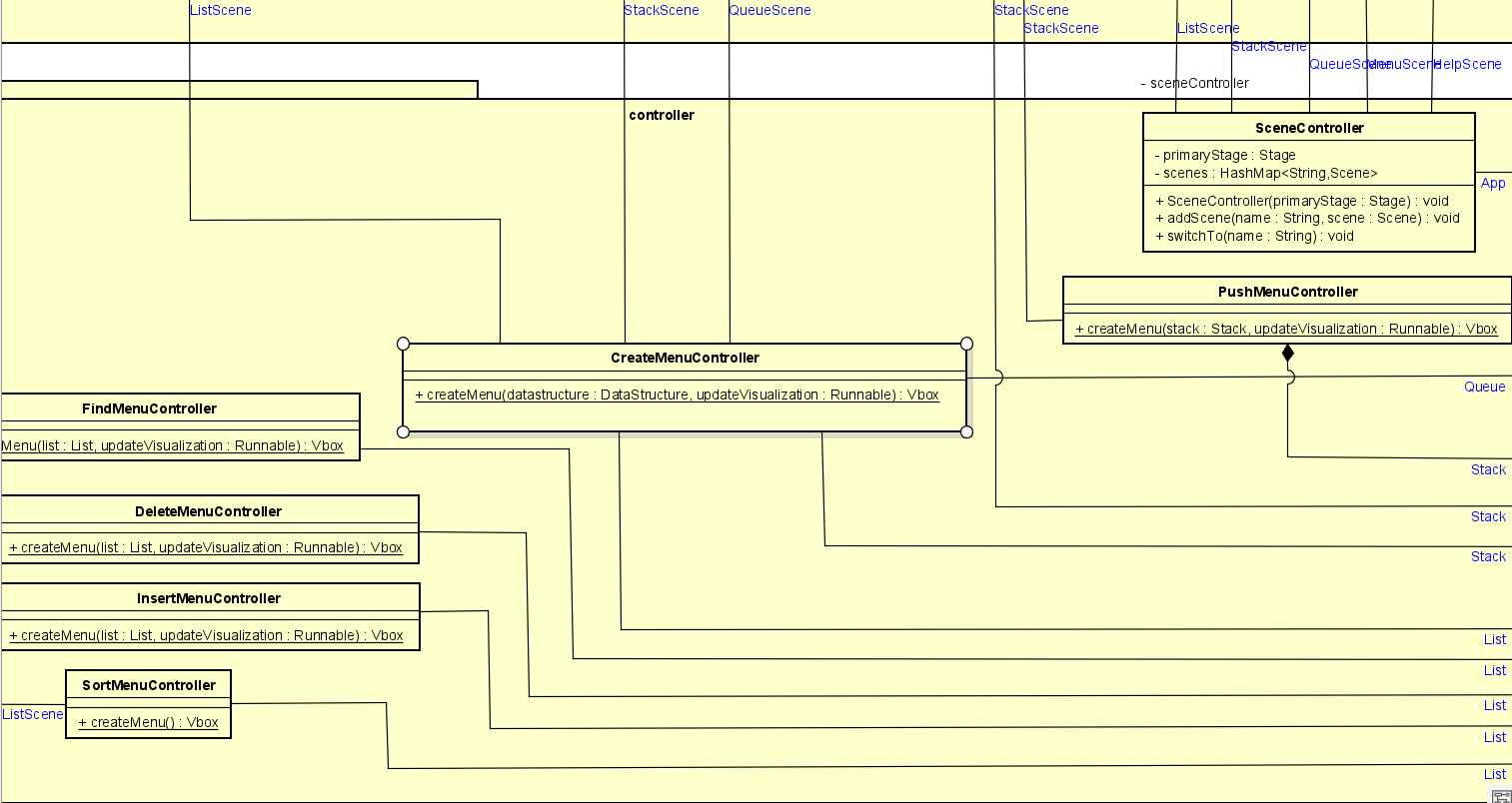


Figure 5: Class diagram for controller package

#### **Lớp SceneController:**

* **Thuộc tính:**
  + primaryStage: Stage chính của ứng dụng.
  + scenes: Một map lưu trữ các scene và tên của chúng..
* **Phương thức:**
  + addScene: Thêm một scene mới vào danh sách các scene.
  + switchTo: Chuyển đổi sang một scene cụ thể.
* **Chức năng:** Là lớp điều khiển chính của chương trình, quản lý toàn bộ luồng điều khiển giữa các scene và lưu trữ thông tin về các scene đã tạo.

#### **Lớp CreateMenuController, FindMenuController, DeleteMenuController, InsertMenuController, SortMenuController, PushMenuController:**

* **Phương thức:**
  + createMenu: Phương thức này tạo ra một menu (có thể là một VBox hoặc HBox) chứa các nút hoặc các thành phần giao diện khác để thực hiện các thao tác tương ứng với cấu trúc dữ liệu. Các tham số truyền vào: dataStructure (Đối tượng đại diện cho cấu trúc dữ liệu cần thao tác List, Stack, Queue) , updateVisualization (Một hàm callback để cập nhật giao diện sau khi thực hiện thao tác trên cấu trúc dữ liệu).
* **Chức năng:** Các lớp này cung cấp các chức năng cụ thể cho từng loại thao tác trên dữ trúc dữ liệu, như tạo, tìm kiếm, xóa, chèn, sắp xếp..

## Mối quan hệ giữa các lớp

### Kế thừa

* Các lớp ListScene, StackScene, QueueScene kế thừa từ BaseScene.
* Các lớp Stack, Queue, và List kế thừa từ lớp DataStructure.

### Liên kết

* Lớp CreateMenuController liên kết với các lớp Stack, Queue, List và liên kết với các lớp ListScene, StackScene, QueueScene.
* Lớp ListScene liên kết với các lớpFindMenuController, DeleteMenuController, InsertMenuController, SortMenuController.
* Lớp StackScene liên kết với lớp PushMenuController
* Lớp SceneController liên kết với các lớp ListScene, StackScene, QueueScene, MenuScene, HelpScene và lớp App.

### Hợp thành

* Lớp Stack với các lớp PushMenuController và lớp StackScene.
* Lớp List với các lớp ListScene, FindMenuController, DeleteMenuController, InsertMenuController, SortMenuController.
* Lớp Queue với lớp QueueScene.

### Đa hình

* Overriding lớp Datastructure bởi các lớp kế thừa như lớp Stack, Queue, List với các phương thức create, insert, delete, createRandom.
* Overriding lớp Basescene bởi các lớp kế thừa như lớp StackScene, QueueScene, ListScene với phương thức createBaseScene.
* Overloading với phương thức createMenu thuộc lớp CreateMenuController tùy thuộc vào tham số đầu vào là list, stack hay queue.