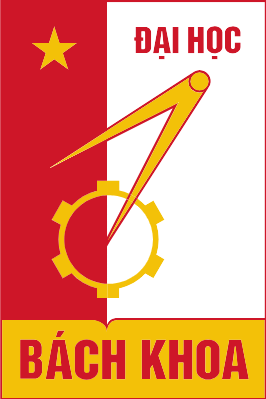
**ĐẠI HỌC BÁCH KHOA HÀ NỘI  
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Mini-Project: Topic 1 – Trình bày các thao tác trên Stack, Queue , List.**

GVHD : Nguyễn Thị Thu Trang

TA : Phạm Phan Anh

Nhóm 19 : Vũ Thị Quỳnh Như - 20215110

Nguyễn Thị Nhung – 20225754

Nguyễn Thị Kiều Oanh – 20225899

Bùi Tuấn Phong – 20225900

Hoàng Nguễn Phong – 20215112

Hà Nội 12/2024

# Phân công thành viên:

Vũ Thị Quỳnh Như

* Xây dựng lớp Stack(Push, Pop, Peek).
* Extend lớp abstract DataStructure và tạo các phương thức.

Nguyễn Thị Nhung

* Thiết kế và triển khai giao diện chính.
* Xử lý điều hướng giữa các màn hình.
* Thiết kế giao diện để hiển thị các nút hoặc danh sách thả xuống cho các chức năng.

Nguyễn Thị Kiều Oanh

* Xây dựng lớp Queue(Enqueue, Dequeue, Peek).
* Extend lớp abstract DataStructure và tạo các phương thức.

Bùi Tuấn Phong

* Xây dựng lớp List(Add, Get, Remove).
* Extend lớp abstract DataStructure và tạo các phương thức.

Hoàng Nguễn Phong

* Tạo lớp abstract DataStructure ,triển khai các hoạt động minh họa cho từng thao tác create(), insert()…
* Thống nhất GUI với logic demonstration.
* Link các nút đến phương thức đúng. Design animation.

# Mô tả Mini-Project

## Yêu cầu

Thiết kế một chương trình để hiển thị và giải thích một số thao tác cơ bản trên các cấu trúc Stack, Queue, List. Kiến thức cơ bản: tạo, chèn, sắp xếp, tìm, xóa các phần tử trong cấu trúc

Thông số kỹ thuật:

- GUI: Tự do thiết kế GUI, tuy nhiên, không cần tập trung quá nhiều vào giao diện vì mục đích chính của dự án là sử dụng OOP để thiết kế ứng dụng

+ Trên menu chính: tiêu đề của ứng dụng, các tùy chọn để người dùng lựa chọn giữa ba loại

cấu trúc dữ liệu, menu trợ giúp và thoát

• Người dùng phải chọn một loại cấu trúc dữ liệu trước khi vào phần trình diễn

• Menu trợ giúp hiển thị cách sử dụng cơ bản và mục đích của dự án

• Nút thoát thoát khỏi ứng dụng.

+ Trong phần trình diễn

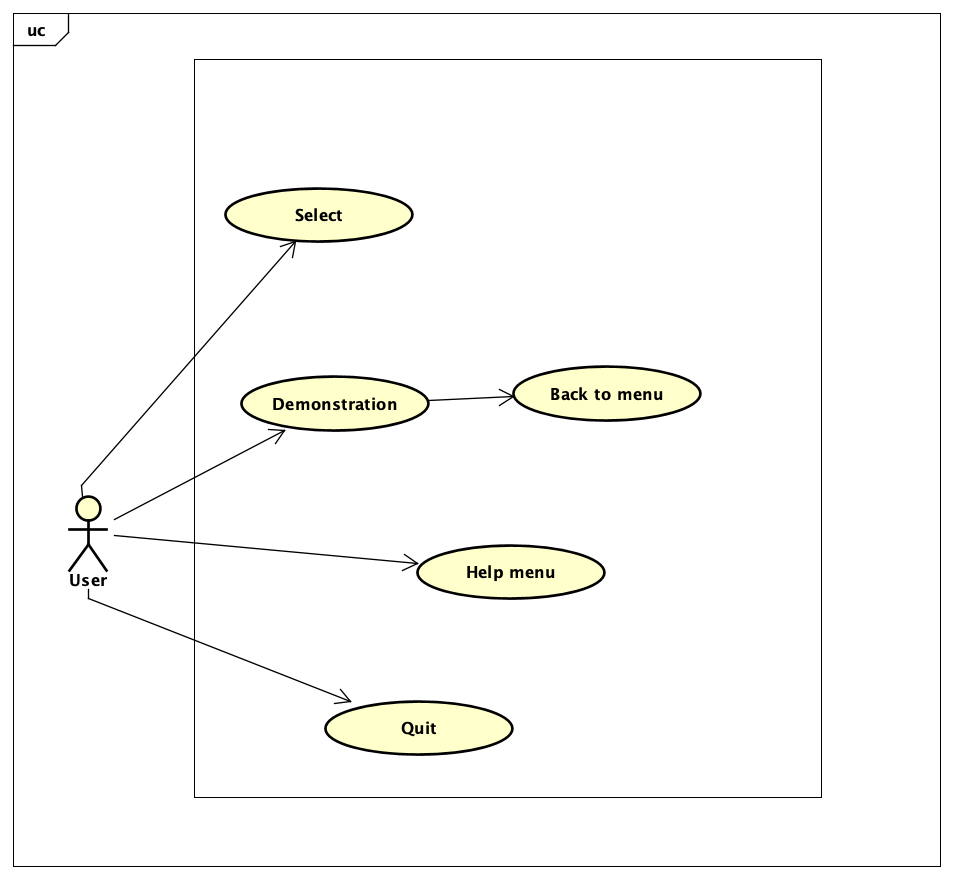
• Phần trình diễn chính hiển thị các tùy chọn để người dùng lựa chọn: tạo, chèn, sắp xếp, tìm và

xóa các phần tử trong cấu trúc.Có thể chọn đưa vào danh sách thả xuống hoặc các nút riêng biệt.

• Sau khi người dùng đã chọn một lựa chọn, hãy trình bày nó trên màn hình.

• Luôn có nút Quay lại để người dùng quay lại menu chính bất kỳ lúc nào

Usecase diagram



## Giải thích

* Khi bắt đầu chương trình trên menu chính sẽ có 3 loại options.

+ Đầu tiên là các lựa chọn để demo Stack, Queue, List người dùng chọn 1 trong 3 lựa chọn trên người dùng phải chọn một loại cấu trúc dữ liệu trước khi vào phần trình diễn

+ Tiếp đến là Help, Menu trợ giúp hiển thị cách sử dụng cơ bản và mục đích của dự án + Cuối cùng là Quit, Nút thoát thoát khỏi ứng dụng.

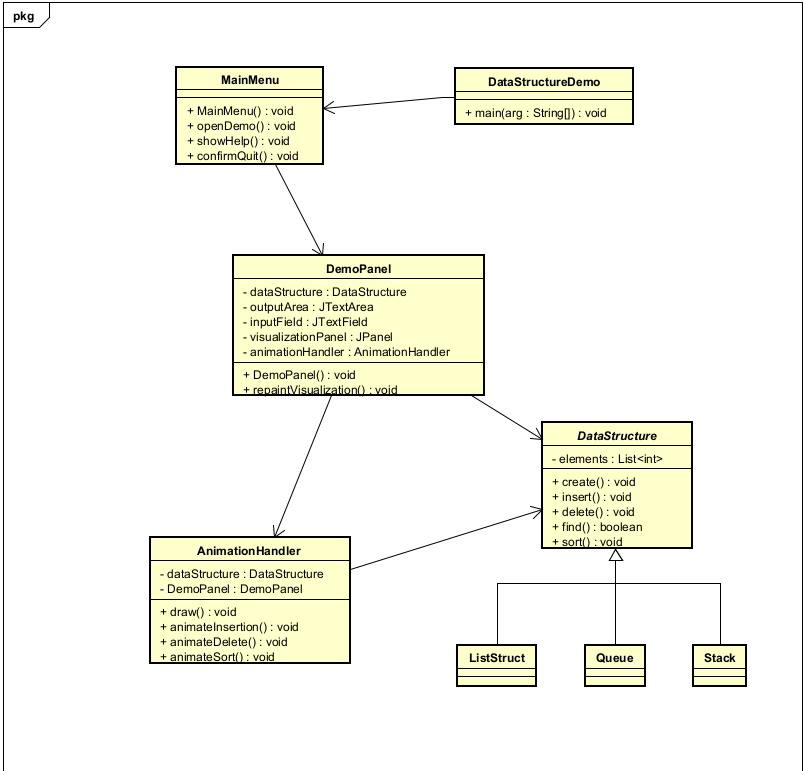
* Trong phần trình diễn sau khi đã chọn 1 trong 3 kiểu dữ liệu Stack,Queue, List

+ Đối với lựa chọn Stack, Queue bản demo sẽ hiện lên người dùng sẽ nhập giá trị vào phần value sau đó chọn các lựa chọn như create, insert, sort, find,delete

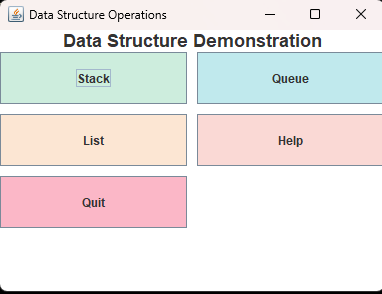
+ Đối với lựa chọn List, bản demo sẽ hiện lên và người dùng sẽ nhập các giá trị vào phần value, (index) sau đó chọn các lựa chọn như create, insert, sort, find,delete

# Thiết kế

Class diagram



* Lớp Data Structure là lớp abstract với các phương thức cần Override như insert, delete, sort, find
* Các lớp như Stack, Queue, ListStruct kế thừa lớp Data Structue, và override các phương thức trên
* Lớp Mainmenu sử dụng java swing để hiển thị màn hình menu như sau



* Lớp Demopannel là một phần của giao diện người dùng đồ họa trong Java, được sử dụng java Swing. Lớp này là một JPanel, cung cấp giao diện thao tác trên các cấu trúc dữ liệu như Stack, Queue hoặc List. Người dùng có thể tạo, chèn, sắp xếp, tìm kiếm và xóa phần tử trong cấu trúc dữ liệu thông qua các nút bấm.

**+** Xử lý Nút Bấm:

* Create: Xóa tất cả phần tử trong cấu trúc dữ liệu và cập nhật giao diện.
* Insert:
  + Thêm giá trị vào Stack/Queue (không cần chỉ mục).
  + Thêm giá trị vào List (cho phép thêm vào vị trí cụ thể hoặc cuối danh sách).
* Sort: Gọi hiệu ứng sắp xếp trên cấu trúc dữ liệu.
* Find: Tìm kiếm một giá trị và hiển thị kết quả trong outputArea.
* Delete:
  + Xóa phần tử đầu/cuối của Stack/Queue.
  + Xóa phần tử tại chỉ mục cụ thể trong List.
* Back: Đóng cửa sổ GUI
* Lớp AnimationHandler trong Java chịu trách nhiệm điều khiển các hoạt động trực quan và hiệu ứng động trên giao diện khi thực hiện các thao tác (chèn, xóa, sắp xếp) trên cấu trúc dữ liệu (Stack, Queue, List).

+ Phương thức drawElements(Graphics g):

* Vẽ các phần tử của cấu trúc dữ liệu lên visualizationPanel theo dạng hộp.
* Các phần tử được hiển thị dạng hình chữ nhật, màu vàng (#fffd7a), mỗi phần tử cách nhau 10px.
* Các phần tử vừa được tráo đổi sẽ có tam giác đỏ bên dưới.

+ Phương thức animateStackQueueInsertion(int value):

* Mô phỏng quá trình chèn phần tử vào Stack/Queue.
* Sử dụng javax.swing.Timer để tạo hiệu ứng phần tử di chuyển dần từ bên trái vào đúng vị trí.
* Khi phần tử đến vị trí, nó sẽ được chèn vào cấu trúc dữ liệu và panel sẽ được vẽ lại.

+ Phương thức animateListInsertion(int value, int targetIndex):

* Mô phỏng quá trình chèn phần tử vào List.
* Phần tử di chuyển theo hướng ngang đến đúng vị trí chỉ định (targetIndex).
* Nếu chỉ số ngoài phạm vi, phần tử sẽ được thêm vào cuối danh sách.

+ Phương thức animateStackDelete():

* Xóa phần tử khỏi Stack với hiệu ứng phần tử "trượt lên trên" và biến mất.
* Nếu Stack rỗng, hiển thị thông báo lỗi.

+ Phương thức animateQueueDelete():

* Xóa phần tử khỏi Queue với hiệu ứng phần tử "trượt sang trái" và biến mất.
* Phần tử đầu tiên của hàng đợi sẽ được xóa khỏi giao diện và cấu trúc dữ liệu.

+ Phương thức animateListDelete(int index):

* Mô phỏng xóa phần tử ở vị trí cụ thể trong List.
* Phần tử sẽ di chuyển sang trái trước khi biến mất.
* Nếu chỉ mục không hợp lệ, hiển thị thông báo lỗi.

+ Phương thức animateSort(JTextArea outputArea, String name):

* Mô phỏng thuật toán sắp xếp nổi bọt (Bubble Sort).
* Hai phần tử được tráo đổi sẽ được đánh dấu bằng tam giác đỏ.
* Sau khi sắp xếp xong, kết quả sẽ được hiển thị trong outputArea.