**Queue Interface**

Queue (hàng đợi) là một cấu trúc dữ liệu FIFO (First In, First Out), tức là phần tử được thêm vào trước sẽ được lấy ra trước.

Thuộc gói java.util và kế thừa từ Collection.

Các phương thức chính:

offerThêm phần tử vào cuối hàng đợi (trả về false nếu không thành công).

poll: Lấy và xóa phần tử đầu tiên, trả về null nếu rỗng.

peek: Lấy phần tử đầu tiên mà không xóa.

Ứng dụng:

Thường được dùng trong hệ thống hàng đợi, ví dụ như xử lý yêu cầu từ người dùng, hàng chờ công việc trong hệ điều hành.

**Deque Interface**

Deque (Double-Ended Queue) là một hàng đợi hai đầu, cho phép thêm và xóa phần tử từ cả hai đầu (đầu và cuối).

Có thể hoạt động như FIFO (hàng đợi) hoặc LIFO (ngăn xếp).

Các phương thức chính:

addFirst(E e), addLast(E e): Thêm phần tử vào đầu/cuối hàng đợi.

pollFirst(), pollLast(): Lấy và xóa phần tử từ đầu/cuối.

peekFirst(), peekLast(): Xem phần tử ở đầu/cuối mà không xóa.

Ứng dụng:

Thường được dùng trong các bài toán cấu trúc ngăn xếp hoặc xử lý tuần tự hai đầu như duyệt cây nhị phân hoặc thuật toán BFS (Breadth-First Search).

**Các Class Triển Khai từ Queue và Deque Interface**

Queue

PriorityQueue, LinkedList: Sử dụng hàng đợi ưu tiên và liên kết kép.

Deque

LinkedList, ArrayDeque: Hỗ trợ thao tác hai đầu bằng danh sách liên kết và mảng.

**1. PriorityQueue**

Đặc điểm:

* Là hàng đợi ưu tiên, phần tử có độ ưu tiên cao hơn sẽ được xử lý trước, không theo thứ tự thêm vào.
* Mặc định, phần tử được sắp xếp theo thứ tự tự nhiên (tăng dần), nhưng có thể tùy chỉnh với Comparator.

Khi nào dùng?

* Khi cần xử lý các công việc hoặc yêu cầu theo độ ưu tiên, ví dụ: hệ thống CPU ưu tiên các tiến trình có độ ưu tiên cao hơn.

**2. LinkedList (Queue và Deque)**

Đặc điểm:

* LinkedList có thể hoạt động như Queue hoặc Deque.
* Thêm/xóa phần tử nhanh ở đầu hoặc cuối do sử dụng danh sách liên kết.

Khi nào dùng?

* Khi cần hàng đợi hoặc ngăn xếp với thao tác thêm/xóa nhanh ở cả hai đầu.

**3. ArrayDeque**

Đặc điểm:

* Sử dụng mảng động để lưu trữ các phần tử.
* Nhanh hơn LinkedList khi thao tác với Deque vì không cần cấp phát bộ nhớ mới cho từng phần tử liên kết.
* Không cho phép null trong danh sách.

Khi nào dùng?

* Khi cần ngăn xếp hoặc hàng đợi hai đầu với hiệu suất cao hơn so với LinkedList.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cấu trúc | Truy cập | Thêm/Xóa | Khi nào dùng |
| PriorityQueue | Mảng động ưu tiên | Theo thứ tự ưu tiên | Chậm khi tìm phần tử | Khi cần Deque hiệu suất cao |
| LinkedList | Danh sách liên kết kép | Tuần tự | Nhanh ở đầu/cuối | Khi cần Queue hoặc Deque linh hoạt |
| ArrayDeque | Mảng động | Tuần tự | Nhanh ở đầu/cuối | Khi cần xử lý theo độ ưu tiên |