**1.Đặc điểm của Queue:**

* Lưu trữ theo thứ tự FIFO:Phần tử được thêm vào trước sẽ được lấy ra trước(First In,First Out)
* Chứa phần tử null:Một số implementation của Queue có thể chứa các phần tử null
* Hỗ trợ các thao tác:Cung cấp các phương thức thêm,xóa và kiểm tra phần tử hàng đầu
* Queue không đảm bảo tính đồng bộ,do đó cần được xử lý đồng bộ hóa khi sử dụng trong môi trường đa luồng

**2.Đặc điểm của Deque:**

* Deque cho phép thêm và xóa phần tử từ cả hai đầu của hang đợi
* Hỗ trợ các thao tác bổ sung: Cung cấp các phương thức để thêm, xóa và kiểm tra phần tử từ cả hai đầu.
* Có thể chứa null: Tương tự như Queue, một số implementation của Deque có thể chứa các phần tử null
* Không đồng bộ: Deque cũng không đồng bộ, cần xử lý đồng bộ hóa khi sử dụng trong môi trường đa luồng.

**3.Các triển khai từ Queue:**

**a. LinkedList**

* **Đặc điểm**:
  + Cung cấp một implementation của cả Queue và Deque.
  + Hỗ trợ các thao tác thêm, xóa từ cả hai đầu.
* **Trường hợp sử dụng**:
  + Sử dụng LinkedList khi bạn cần một hàng đợi với khả năng thêm và xóa từ cả hai đầu và không yêu cầu tính đồng bộ.

**b. PriorityQueue**

* **Đặc điểm**:
* Hàng đợi ưu tiên, nơi các phần tử được sắp xếp theo thứ tự tự nhiên hoặc theo một comparator.
* Không đảm bảo thứ tự FIFO.
* **Trường hợp sử dụng**:
  + Sử dụng PriorityQueue khi bạn cần một hàng đợi mà các phần tử được xử lý theo độ ưu tiên thay vì theo thứ tự vào.

**4.Các triển khai từ deque**

**a. ArrayDeque**

* **Đặc điểm**:
  + Cung cấp một implementation của Deque sử dụng một mảng động.
  + Hỗ trợ thêm và xóa phần tử từ cả hai đầu với hiệu suất tốt.
* **Trường hợp sử dụng**:
  + Sử dụng ArrayDeque khi bạn cần một hàng đợi có thể mở rộng và muốn thao tác từ cả hai đầu. Nó thường nhanh hơn LinkedList cho các thao tác thêm/xóa.

**b. LinkedList**

* **Đặc điểm**:
  + Như đã đề cập, LinkedList cũng triển khai Deque.
  + Cho phép thêm và xóa phần tử từ cả hai đầu, có thể là FIFO hoặc LIFO.
* **Trường hợp sử dụng**:
  + Sử dụng LinkedList khi bạn cần tính linh hoạt tương tự như ArrayDeque, nhưng muốn sử dụng cấu trúc dữ liệu danh sách liên kết.

**Một vài phương thức cơ bản:**

* add(E e): Thêm một phần tử vào cuối hàng đợi (nếu không có giới hạn).
* offer(E e): Thêm một phần tử vào hàng đợi, trả về false nếu không thành công.
* remove(): Xóa và trả về phần tử đầu tiên trong hàng đợi.
* poll(): Xóa và trả về phần tử đầu tiên trong hàng đợi, trả về null nếu hàng đợi rỗng.
* peek(): Trả về phần tử đầu tiên trong hàng đợi mà không xóa nó, trả về null nếu hàng đợi rỗng.
* size(): Trả về số lượng phần tử trong hàng đợi.
* isEmpty(): Kiểm tra xem hàng đợi có rỗng hay không.
* clear(): Xóa tất cả các phần tử trong hàng đợi.