Set Interface

**1. Đặc điểm của Set Interface:**

* Set là một interface trong Java, đại diện cho một tập hợp các phần tử không trùng lặp (no duplicates). Điều này có nghĩa là trong một Set, mỗi phần tử chỉ xuất hiện một lần.
* Set không đảm bảo thứ tự của các phần tử (trừ các triển khai cụ thể có thể hỗ trợ thứ tự riêng như LinkedHashSet).
* Các phương thức phổ biến:
  + add(E e): Thêm phần tử vào tập hợp, trả về true nếu phần tử chưa tồn tại, trả về false nếu phần tử đã tồn tại.
  + remove(Object o): Xóa một phần tử khỏi tập hợp.
  + contains(Object o): Kiểm tra xem một phần tử có nằm trong tập hợp hay không.
  + size(): Trả về số lượng phần tử trong tập hợp.
  + isEmpty(): Kiểm tra xem tập hợp có rỗng không.
  + clear(): Xóa toàn bộ phần tử khỏi tập hợp.

**2. Các lớp triển khai từ Set Interface:**

**a. HashSet:**

* **Đặc điểm**:
  + HashSet sử dụng cơ chế băm (hashing) để lưu trữ các phần tử. Điều này giúp cho việc tìm kiếm, thêm, xóa phần tử diễn ra rất nhanh (trung bình là O(1)).
  + HashSet không đảm bảo thứ tự các phần tử.
* **Trường hợp sử dụng**:
  + Sử dụng khi bạn cần lưu trữ một tập hợp không trùng lặp và không quan tâm đến thứ tự các phần tử.
  + Ví dụ: Lưu trữ danh sách các đối tượng duy nhất như email, số điện thoại hoặc mã ID.

**b. LinkedHashSet:**

* **Đặc điểm**:
  + LinkedHashSet là một phiên bản của HashSet nhưng có duy trì thứ tự chèn các phần tử. Thứ tự này được duy trì theo trình tự mà các phần tử được thêm vào tập hợp.
  + Hiệu suất chậm hơn HashSet một chút vì phải duy trì thứ tự chèn.
* **Trường hợp sử dụng**:
  + Sử dụng khi cần đảm bảo thứ tự chèn của các phần tử và vẫn không có các phần tử trùng lặp.
  + Ví dụ: Lưu trữ danh sách các mục đã duyệt qua trong một ứng dụng với thứ tự thời gian.

**c. TreeSet:**

* **Đặc điểm**:
  + TreeSet sắp xếp các phần tử theo thứ tự tự nhiên hoặc theo một comparator được cung cấp.
  + Việc thêm, xóa và tìm kiếm phần tử có độ phức tạp O(log n) do cấu trúc của nó là cây nhị phân cân bằng (Red-Black Tree).
* **Trường hợp sử dụng**:
  + Sử dụng khi cần một tập hợp không trùng lặp với các phần tử được sắp xếp theo thứ tự tự nhiên hoặc thứ tự do người dùng chỉ định.
  + Ví dụ: Lưu trữ danh sách sinh viên theo thứ tự điểm hoặc tên.

**d. EnumSet:**

* **Đặc điểm**:
  + EnumSet chỉ sử dụng cho các kiểu enum (kiểu liệt kê).
  + Nó rất hiệu quả về mặt bộ nhớ và hiệu suất vì sử dụng các bit để biểu diễn các phần tử của enum.
  + Không hỗ trợ các phần tử null.
* **Trường hợp sử dụng**:
  + Sử dụng khi cần lưu trữ tập hợp các giá trị enum.
  + Ví dụ: Lưu trữ các ngày trong tuần hoặc các trạng thái của một tiến trình.