

🕒 Cập nhật tháng 8 năm 2024

## [Bài đọc] Giới thiệu Set Interface và các lớp triển khai

### 1. Set là gì trong java

- **Set** là một interface thuộc gói **java.util**, được sử dụng để lưu trữ tập hợp các phần tử không trùng lặp.
- **Set** không lưu trữ các phần tử theo thứ tự cụ thể và không cho phép các phần tử trùng lặp.
- Các lớp phổ biến triển khai Set trong Java gồm:
  - **HashSet**: Lưu trữ phần tử dựa trên bảng băm, không đảm bảo thứ tự
  - **LinkedHashSet**: Giữ thứ tự chèn của phần tử
  - **TreeSet**: Lưu trữ phần tử theo thứ tự tự nhiên hoặc theo comparator được cung cấp

### 2. Các phương thức quan trọng trong Set

- Dưới đây là các phương thức phổ biến của Set (được định nghĩa trong interface **java.util.Set**):
- **Phương thức thêm và xóa phần tử**
  - `boolean add(E e)` – Thêm mới phần tử
  - `boolean remove(Object o)` – Xóa phần tử
  - `void clear()` – Xóa tất cả phần tử khỏi collection
- **Phương thức kiểm tra**
  - `boolean contains(Object o)` – Kiểm tra phần tử tồn tại trong mảng không
  - `boolean isEmpty()` – Kiểm tra collection có rỗng hay không
- **Phương thức lấy thông tin**
  - `int size()` – Lấy ra độ dài của collection
  - `Iterator<E> iterator()` – Trả về một iterator để duyệt qua các phần tử trong Set
- **Các phương thức khác**
  - `boolean addAll(Collection<? extends E> c)` – Thêm tất cả các phần tử từ Collection vào Set
  - `boolean retainAll(Collection<?> c)` – Giữ lại các phần tử của Set có trong Collection

- `boolean removeAll(Collection<?> c)` - Xóa tất cả các phần tử trong Collection ra khỏi Set

### 3. Ưu và nhược điểm của Set

- **Ưu điểm**

- Đảm bảo không có phần tử trùng lặp
- Dễ dàng quản lý dữ liệu tập hợp

- **Nhược điểm**

- Không hỗ trợ truy cập phần tử theo chỉ số (index).
- Hiệu suất có thể bị ảnh hưởng tùy thuộc vào kiểu Set (HashSet nhanh hơn TreeSet, nhưng không duy trì thứ tự)

Link tài nguyên đọc thêm: <https://www.geeksforgeeks.org/set-in-java/>

Danh sách các bài học

