**Map Interface**

Trong java, map được sử dụng để lưu trữ và truy xuất dữ liệu theo cặp khóa (key) và giá trị (value). Mỗi cặp key và value được gọi là entry.

Map chỉ chứa các giá trị key duy nhất, không chứa các key trùng lặp. Nếu thêm cùng một key, giá trị cũ sẽ bị thay thế.

Value có thể trùng lặp, nhưng key không được trùng.

Các lớp cài đặt Map interface :

HashMap không đảm bảo thứ tự các entry được thêm vào.

LinkedHashMap đảm bảo thứ tự các entry được thêm vào.

TreeMap duy trình thứ tự các phần tử dựa vào bộ so sánh Comparator.

**HashMap**

Đặc điểm:

* Không duy trì thứ tự của các phần tử.
* Thao tác tìm kiếm, thêm, xóa có hiệu suất O(1) (trung bình) nhờ sử dụng cơ chế hashing.
* Cho phép key và value là null (chỉ một key null, nhiều value null).

Khi nào nên dùng?

* Khi không quan tâm đến thứ tự của các phần tử.
* Phù hợp cho các tác vụ cần hiệu suất cao, như đếm số lần xuất hiện của các từ, hoặc lưu dữ liệu tạm thời

**LinkedHashMap**

Đặc điểm:

* Duy trì thứ tự các phần tử theo thứ tự thêm vào.
* Hỗ trợ một tùy chọn thứ tự truy cập – nghĩa là nếu bạn truy xuất một phần tử, nó sẽ được đưa về cuối danh sách (thường dùng trong các cache LRU - Least Recently Used).
* Hiệu suất gần bằng HashMap.

Khi nào nên dùng?

* Khi cần bảo toàn thứ tự thêm vào hoặc theo thứ tự truy cập gần nhất.
* Phù hợp để triển khai cache với chính sách LRU.

**TreeMap**

Đặc điểm:

* Duy trì thứ tự sắp xếp tự nhiên của key (hoặc theo comparator tùy chỉnh).
* Không cho phép key là null (nhưng value có thể là null).
* Các thao tác tìm kiếm, thêm, xóa có hiệu suất O(log n) nhờ sử dụng cây nhị phân cân bằng (Red-Black Tree).

Khi nào nên dùng?

* Khi cần sắp xếp tự động theo thứ tự tăng dần của key (hoặc theo logic sắp xếp riêng).
* Phù hợp cho các bài toán như danh bạ, bảng xếp hạng, hoặc bộ lịch.