1. Nêu ra các đặc điểm **Map Interface**

* 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.
* Có thể có nhiều key khác nhau ánh xạ tới cùng một giá trị.
* Cung cấp các phương thức để thao tác với các cặp này như put(), get(), remove(), containsKey()
* Sức chứa (compacity) mặc định khi khởi tạo map là 24 = 16. Kích thước này sẽ tự động tăng gấp đôi mỗi khi thêm phần tử vượt quá kích thước của nó.

1. Kể ra các **class**triển khai từ **Map Interface**

* HashMap
* LinkedHashMap
* TreeMap

1. Phân biệt rõ trường hợp sử dụng của từng **class**đó

HashMap:

* Sử dụng bảng băm (hash table) để lưu trữ dữ liệu.
* Không đảm bảo thứ tự của các cặp key-value.
* Cho phép một khóa null và nhiều giá trị null.
* Tốc độ tìm kiếm và truy xuất nhanh với độ phức tạp O(1).
* **Trường hợp sử dụng:** Khi không cần duy trì thứ tự của các phần tử và cần tốc độ xử lý cao, đặc biệt khi có nhiều thao tác tìm kiếm, thêm và xóa.

LinkedHashMap

* Giống HashMap, nhưng duy trì thứ tự chèn (insertion order) hoặc thứ tự truy cập (access order).
* Duy trì một danh sách liên kết kép (doubly linked list) để lưu trữ thứ tự các cặp key-value.
* Trường hợp sử dụng: Khi cần một map có khả năng truy cập nhanh và đồng thời muốn duy trì thứ tự chèn hoặc truy cập gần đây nhất (ví dụ trong cache).

TreeMap

* Lưu trữ dữ liệu theo cây đỏ-đen (Red-Black tree).
* Duy trì các cặp key-value theo thứ tự tự nhiên của khóa hoặc theo một bộ so sánh tùy chỉnh (custom comparator).
* Tốc độ truy cập là O(log n) do bản chất của cây nhị phân.
* Trường hợp sử dụng: Khi cần duy trì các phần tử theo thứ tự sắp xếp, ví dụ như khi cần sắp xếp các phần tử theo thứ tự từ điển hoặc số học.

1. Các interface triển khai map

* Map
* SortedMap