**1.Khái niệm và đặc điểm:**

* Map interfece trong Java là một cấu trúc dũ liệu quan trọng cho phép lưu trữ dữ liệu theo key-value,mỗi khóa là duy nhất và liên kết với một giá trị.
* Map không đảm bảo thứ tự của các phần tử,tuy nhiên một sô triển khai map duy trì thứ tự như LinkedHashMap
* Mỗi khóa trong Map phải là duy nhất,nếu cố gắng thêm khóa đã tồn tại,giá trị sẽ bị cập nhật

**2.Các triển khai của Map:**

**a.HashMap:**

**Đăc điểm:**

* Dựa trên cấu trúc dữ liệu Hash table
* Không đảm bảo thứ tự các phần tử
* Cho phép môt khóa null và nhiều giá trị

**Trường hợp sử dụng:**

* Sử dụng khi cần tìm kiếm,thêm hoặc xóa các phần tử một cách nhanh chóng
* Thích hợp các trường hợp không yêu cầu thứ tự

**b.LinkedHashMap**

**Đặc điểm:**

* Giống như HashMap,nhưng duy trì thứ tự của các phần tử theo thứ tự chúng đc truyền vào
* Hỗ trợ hiệu xuất tốt cho tìm kiếm,thêm và xóa

**Trường hợp sử dụng:**

* Sử dụng LinkedHashMap khi cần một Map duy trì thứ tự của các phần tử mà hiệu suất tốt như HashMap

**c.TreeMap**

**Đặc điểm**:

* Dựa trên cấu trúc cây nhị phân tìm kiếm
* Duy trì thứ tự của các phần tử theo thứ tự tự nhiên của khóa
* Không cho phép khóa null

**Trường hợp sử dụng:**

* Sử dụng TreeMap khi cần một Map truy cập theo thứ tự hoặc khi cần tìm kiếm theo phạm vi

**d. Hashtable**

**Đặc điểm**:

* + Dựa trên cấu trúc hash table, giống như HashMap, nhưng là đồng bộ (synchronized).
  + Không cho phép khóa và giá trị là null.

**Trường hợp sử dụng**:

* + Sử dụng Hashtable khi bạn cần một Map đồng bộ hóa và an toàn trong môi trường đa luồng. Tuy nhiên, thường thì ConcurrentHashMap được ưa chuộng hơn vì hiệu suất tốt hơn.

**Một vài phương thức trong Map:**

* put(K key, V value): Thêm hoặc cập nhật một cặp khóa-giá trị.
* get(Object key): Trả về giá trị tương ứng với khóa chỉ định.
* remove(Object key): Xóa cặp khóa-giá trị tương ứng với khóa chỉ định.
* containsKey(Object key): Kiểm tra xem bản đồ có chứa khóa chỉ định hay không.
* containsValue(Object value): Kiểm tra xem bản đồ có chứa giá trị chỉ định hay không.
* size(): Trả về số lượng cặp khóa-giá trị trong bản đồ.
* isEmpty(): Kiểm tra xem bản đồ có rỗng hay không.
* keySet(): Trả về một Set chứa tất cả các khóa trong bản đồ.
* values(): Trả về một Collection chứa tất cả các giá trị trong bản đồ.
* entrySet(): Trả về một Set chứa tất cả các cặp khóa-giá trị.
* clear(): Xóa tất cả các cặp khóa-giá trị trong bản đồ.