|  |  |  |  |
| --- | --- | --- | --- |
| Cấu trúc dữ liệu dạng Map là gì ? | là một giao diện (interface) trong gói java.util đại diện cho cấu trúc dữ liệu ánh xạ (key-value). Nó được sử dụng để lưu trữ các cặp giá trị, trong đó **key** là duy nhất và được ánh xạ tới một **value** tương ứng.  **Các đặc điểm chính của Map**   1. **Key** là duy nhất: Không thể có hai phần tử trong Map có cùng key. 2. **Value** không cần phải duy nhất: Nhiều key có thể ánh xạ đến cùng một value. 3. Map không thuộc loại **Collection**, nhưng nó là một phần của Java Collections Framework. | | |
| Các lớp triển khai từ interface Map ? | **HashMap**  **Đặc điểm**:  Dựa trên bảng băm (hash table).  Không duy trì thứ tự các phần tử.  Cho phép **null key** (chỉ một) và nhiều **null value**.  Không đồng bộ (non-synchronized), không an toàn cho các luồng (thread-unsafe).  **Trường hợp sử dụng**: Khi bạn cần một ánh xạ nhanh và không quan tâm đến thứ tự các phần tử.  **LinkedHashMap**  **Đặc điểm**:  Kế thừa từ HashMap nhưng duy trì thứ tự các phần tử (theo thứ tự thêm vào).  Hiệu suất tương tự HashMap.  Cho phép **null key** và nhiều **null value**.  **Trường hợp sử dụng**: Khi cần duy trì thứ tự các phần tử trong ánh xạ.  **TreeMap**  **Đặc điểm**:  Dựa trên cây nhị phân cân bằng (red-black tree).  Duy trì các phần tử theo thứ tự tự nhiên (natural order) hoặc theo một Comparator được chỉ định.  Không cho phép **null key**, nhưng cho phép nhiều **null value**.  **Trường hợp sử dụng**: Khi cần một ánh xạ sắp xếp. | | |
| HashMap là gì ? | HashMap là một lớp trong Java, thuộc gói java.util, được sử dụng để lưu trữ dữ liệu dưới dạng **cặp key-value**. Nó hoạt động dựa trên một **bảng băm** (hash table) để quản lý và tổ chức dữ liệu, cho phép truy xuất nhanh các phần tử. | | |
| LinkedHashMap có điểm gì khác so với HashMap ? |  | HashMap | LinkedHashMap |
| Thứ tự phần tử | Không duy trì thứ tự phần tử. | Duy trì thứ tự chèn vào hoặc thứ tự truy cập (nếu bật accessOrder = true). |
| Cấu trúc bên trong | Dựa trên bảng băm (hash table). | Bảng băm kết hợp với danh sách liên kết kép (doubly-linked list). |
| Hiệu suất | Nhanh hơn do không duy trì thứ tự. | Chậm hơn một chút vì cần duy trì danh sách liên kết kép. |
| Thứ tự truy cập (accessOrder) | Không hỗ trợ. | Hỗ trợ duy trì thứ tự truy cập nếu bật chế độ accessOrder. |
| Bộ nhớ sử dụng | Tiêu tốn ít bộ nhớ hơn. | Tốn thêm bộ nhớ để lưu liên kết giữa các phần tử. |
| |  | | --- | |  |   Trường hợp sử dụng | Khi cần ánh xạ key-value nhanh mà không quan tâm đến thứ tự | Khi cần duy trì thứ tự chèn hoặc thứ tự truy cập, như trong các ứng dụng cache |
| Null Key và Null Value | Cho phép **một null key** và nhiều null value | Tương tự HashMap |
| Đồng bộ (Synchronization) | Không đồng bộ. | Không đồng bộ. |
| TreeMap là gì ? các đặc điểm của nó ? | TreeMap là một lớp trong Java, thuộc gói java.util, triển khai interface Map. Nó lưu trữ các cặp **key-value** dưới dạng **cây nhị phân cân bằng** (**Red-Black Tree**). Vì vậy, các phần tử trong TreeMap luôn được sắp xếp theo thứ tự tự nhiên của **key** hoặc theo thứ tự của một Comparator tùy chỉnh được cung cấp. | | |
| Trình bày khái niệm binary tree ? | **Binary Tree** (cây nhị phân) là một cấu trúc dữ liệu cây (tree data structure), trong đó mỗi **node** có tối đa **hai node con**:   * **Node con trái** (*left child*). * **Node con phải** (*right child*).   Cây nhị phân thường được sử dụng để tổ chức và quản lý dữ liệu theo thứ tự để dễ dàng thực hiện các thao tác tìm kiếm, thêm, xóa, hoặc duyệt qua các phần tử. | | |