Bài 12

|  |  |
| --- | --- |
| 1. Nhắc lại Set? Các class triển khai của Set? | * Set là một collection (tập hợp) trong Java, chứa các phần tử không trùng lặp và không có thứ tự cụ thể. * Không cho phép phần tử null lặp lại. * Các class triển khai của Set :   + HashSet: lưu trữ phần tử không theo thứ tự, hiệu năng cao.  + LinkedHashSet: giữ nguyên thứ tự thêm vào.  + TreeSet: lưu phần tử theo thứ tự tăng dần tự nhiên hoặc theo comparator. |
| 1. Map là gì? Đặc điểm của Map? Cho ví dụ ứng dụng? | * Map<K, V> là một cấu trúc dữ liệu ánh xạ giữa key và value. * Mỗi key là duy nhất, còn value có thể trùng nhau. * Đặc điểm :   + Không cho phép key trùng.  + Có thể chứa value null, key null (tùy class).  + Không thuộc Collection, nhưng được dùng tương tự.   * Ví dụ :   + Lưu danh bạ: Map<String, String> phoneBook = new HashMap<>(); // key: tên, value: số  + Đếm số lần xuất hiện từ trong văn bản. |
| 1. Các lớp triển khai Map? So sánh giữa các class đó? | * HashMap:  + Hiệu suất cao, không có thứ tự  + Thứ tự duyệt không xác định  + 1 null key, nhiều null value * LinkedHashMap:  + Duy trì thứ tự thêm vào  + Có thứ tự duyệt  + Cho phép null * TreeMap:  + Các key được sắp xếp tăng dần  + Thứ tự duyệt có thứ tự tăng  + Không cho null key |
| 1. Tree là gì? BinaryTree là gì? Binary Search Tree (BST) là gì? | * Tree (Cây) là cấu trúc dữ liệu dạng phân cấp gồm nhiều node, node cha - con. * Node gốc là root, không có cha. * Binary Tree (Cây nhị phân) là cây mà mỗi node có tối đa 2 con: trái và phải. * BST là Binary Tree có sắp xếp:   + Node trái < node hiện tại  + Node phải > node hiện tại  + Áp dụng trong tìm kiếm, chèn, xóa hiệu quả: O(log n) trong trường hợp tốt. |
| 1. Các cách duyệt BST? | * Duyệt theo thứ tự (Depth-First Traversal):   + In-order (Trái - Gốc - Phải): Kết quả các node tăng dần  + Pre-order (Gốc - Trái - Phải)  + Post-order (Trái - Phải - Gốc)   * Duyệt theo tầng (Breadth-First / Level Order):   + Duyệt từng tầng từ trên xuống dưới, trái sang phải. |