|  |  |
| --- | --- |
| **1. Java collection framework là gì ? Các interface chính ?** | Là một khuôn khổ cung cấp các class và interface để lưu chữ và thao tác với 1 nhóm các đối tượng  - Các interface chính :  + List : cấu trúc dữ liệu dạng tuyến tính và có thể lưu các phần tử trùng lặp .  + Queue : Cấu trúc hàng đợi (FIFO 🡺 fist in - fist out)  + Set : Cấu trúc dữ liệu không tuyến tính và không cho phét lưu các phần tử trùng lặp  + Map : Lưu trữ theo kiểu <Key-value> |
| **2. So sánh List và Set** | |  |  |  | | --- | --- | --- | | **Đặc điểm** | **List** | **Set** | | Trật tự | Có thứ tự, phần tử được lưu theo vị trí | Không có thứ tự cụ thể | | Trùng lặp | Cho phép phần tử trùng lặp | Không cho phép phần tử trùng lặp | | Hiệu suất tìm kiếm | Tìm kiếm chậm hơn | Tìm kiếm nhanh hơn | | Thao tác thay đổi | Dễ dàng thay đổi từng phần tử | Thường chỉ thêm hoặc xóa, không chỉnh sửa phần tử trực tiếp | | Dùng khi nào? | Khi cần bảo toàn thứ tự và cho phép trùng lặp | Khi cần loại bỏ phần tử trùng lặp và không quan trọng thứ tự | |
| **3. Các class triển khai interface List ?** | Trong Java, List là một interface được triển khai bởi nhiều class, bao gồm:   * ArrayList – Dựa trên mảng động, truy xuất nhanh nhưng chèn/xóa chậm. * LinkedList – Danh sách liên kết đôi, thêm/xóa nhanh hơn nhưng truy xuất chậm hơn. * Vector – Giống ArrayList nhưng hỗ trợ đồng bộ hóa. * CopyOnWriteArrayList – An toàn cho đa luồng, sao chép danh sách khi có thay đổi. |
| **4. So sánh ArrayList và Array?** | |  |  | | --- | --- | | **Array** | **ArrayList** | | Độ dài cố định | Độ dài có thể thay đổi được | | Không có các method làm việc với mảng | Có các method làm việt với List | | Có thể tluue được kiểu nguyên thủy và đối tượng | Chỉ có thể lưu được đối tượng | | Nhanh hơn khi lấy dữ liệu ra | Chậm hơn khi lấy dữ liệu ra | | Không có Generic <> | Có Generic | |
| **5. So sánh ArrayList và LinkedList** | |  |  | | --- | --- | | **ArrayList** | **LinkedList** | | Dựa trên mảng động | Dựa trên các note liên kết | | Xóa phần tử chậm hơn | Xóa phần tử nhanh hơn | | Khi truy xuất dữ liệu nhanh hơn | Khi truy xuất dữ liệu chậm hơn | |
| **6. Các class triển khai interface Set ? Điểm khác nhau giữa các class đó?** | Trong Java, Set là một interface chứa các phần tử không trùng lặp. Các class phổ biến triển khai Set:   * HashSet – Hiệu suất cao , không giữ thứ tự. * LinkedHashSet – Giữ thứ tự phần tử theo thứ tự thêm vào. * TreeSet – Sắp xếp tự nhiên hoặc tùy chỉnh, hiệu suất . * ConcurrentSkipListSet – Sắp xếp tự nhiên, hỗ trợ đa luồng. |
| **7. Lưu ý gì khi sử dụng Set để lưu dữ liệu kiểu đối tượng** | Khi sử dụng Set để lưu trữ đối tượng trong Java, cần lưu ý:   * **Ghi đè** equals() **và** hashCode() để tránh lỗi khi kiểm tra trùng lặp. * **Không đảm bảo thứ tự** – nếu cần thứ tự, dùng LinkedHashSet hoặc TreeSet. * **Đối tượng có thể thay đổi** – thay đổi thuộc tính có thể ảnh hư |