**Bài soạn ArrayList, LinkedList và Set - *Đỗ Văn Hoàng CGHN C0624G1***

|  |  |
| --- | --- |
|  | Ghi chú |
|  | |  |  |  |  | | --- | --- | --- | --- | | Tiêu chí | ArrayList | LinkedList | Set | | Cấu trúc dữ liệu | ArrayList sử dụng mảng động để lưu trữ các phần tử. ArrayList là một cấu trúc dữ liệu dựa trên chỉ mục (index), trong đó mỗi phần tử (element) được liên kết với một chỉ mục. | LinkedList sử dụng danh sách liên kết (Doubly Linked List) để lưu trữ các phần tử. Các phần tử trong LinkedList được gọi là node, mỗi node cần lưu trữ 3 thông tin: tham chiếu phần tử trước nó, giá trị của phần tử và một tham chiếu tới phần tử kế tiếp. | Set được sử dụng để lưu trữ một tập hợp các phần tử không trùng lặp. Set không lưu trữ các phần tử theo chỉ số và không cho phép các phần tử trùng lặp.  3 dạng Set bao gồm : HashSet, LinkedSet, TreeSet | | Thao tác thêm và xóa | Thao tác thêm và xóa phần tử với ArrayList là chậm bởi vì nó sử dụng nội bộ mảng. Bởi vì sau khi thêm hoặc xóa các phần tử cần sắp xếp lại. | Thao tác thêm và xóa phần tử với LinkedList nhanh hơn ArrayList. Bởi vì nó không cần sắp xếp lại các phần tử sau khi thêm hoặc xóa. Nó chỉ cần cập nhật lại tham chiếu tới phần tử phía trước và sau nó. | Độ phức tạp ảnh hưởng đến hiệu suất, tăng dần theo thứ tự : HashSet < LinkedSet < TreeSet. Hashset hiệu quả hơn cho các tập hợp lớn so với ArrayList và LinkedList vì có độ phức tạp thời gian trung bình là O(1). | | Thao tác truy xuất | Truy xuất phần tử trong ArrayList nhanh hơn LinkedList. Bởi vì các phần tử trong ArrayList được lưu trữ dựa trên chỉ mục (index). | Thao tác truy xuất phần tử trong LinkedList chậm hơn nhiều so với ArrayList. Bởi vì, nó phải duyệt qua lần lượt các phần tử từ đầu tiên cho đến cuối cùng. | Độ phức tạp ảnh hưởng đến hiệu suất, tăng dần theo thứ tự : HashSet < LinkedSet < TreeSet.  HashSet hiệu quả hơn trong việc tìm kiếm phần tử so với LinkedList và ArrayList vì có độ phức tạp thời gian trung bình là O(1). | | Truy xuất ngẫu nhiên | ArrayList có thể truy xuất ngẫu nhiên phần tử. | LinkedList không thể truy xuất ngẫu nhiên. Nó phải duyệt qua tất cả các phần tử từ đầu tiên đến phần tử cuối cùng để tìm phần tử. | Set không thể truy xuất ngẫu nhiên. Nó phải duyệt qua tất cả các phần tử từ đầu tiên đến phần tử cuối cùng để tìm phần tử hoặc chuyển đổi thành List. | | Tiêu tốn bộ nhớ | ArrayList yêu cầu ít bộ nhớ hơn so với LinkedList. Bởi vì ArrayList chỉ lưu trữ dữ liệu (data) của nó và chỉ mục (index). | LinkedList yêu cầu nhiều bộ nhớ hơn so với ArrayList. Bởi vì LinkedList lưu giữ thông tin của nó và tham chiếu tới phần tử trước và sau nó. | Độ phức tạp ảnh hưởng đến bộ nhớ, tăng dần theo thứ tự : HashSet < LinkedSet < TreeSet. Nhưng tiết kiệm bộ nhớ hơn một chút ở khoản không lưu trữ phần tử trùng lặp. | | Dùng khi nào? | ArrayList tốt hơn trong việc lưu trữ và truy xuất dữ liệu | LinkedList tốt hơn trong việc thao tác dữ liệu (thêm/xóa). | Dùng Set cần một tập hợp các phần tử mà không có phần tử trùng lặp và không quan tâm đến thứ tự. | |  |  |  |  | |
|  | Kết luận |