|  |  |
| --- | --- |
| **1. Generic là gì? Ưu và nhược điểm** | Generic là kỹ thuật truyền kiểu dữ liệu vào như 1 tham số => tham số hoá dữ liệu Ưu điểm: -Phát hiện lỗi tại thời điểm biên dịch -Không cần ép kiểu -Tạo ra các class, interface, method dùng chung => tái sử dụng code. Nhược điểm: -Không thể khởi tạo đối tượng trực tiếp cho kiểu Generic -Không sử dụng từ khoá static có biến Generic -Không thể áp dụng kiểu nguyên thuỷ cho Generic |
| **2. Cấu trúc dữ liệu Stack là gì? Một số method làm việc với stack?** | - Là cấu trúc dữ liệu hoạt động dựa trên cơ chế FILO (first-in last-out) 🡺 Vào trước ra sau  - Method làm việc với stack :   * push(E item) – Thêm một phần tử vào đầu stack. * pop() – Loại bỏ và trả về phần tử trên cùng của stack. * peek() – Xem phần tử trên cùng mà không xóa. * empty() – Kiểm tra xem stack có rỗng không. * search(Object o) – Tìm vị trí của phần tử trong stack (trả về vị trí từ 1, hoặc -1 nếu không tìm thấy). |
| **3. Cấu trúc dữ liệu queue là gì? Một số method làm việc với queue?** | - Queue : Là cấu trúc dữ liệu hoạt động dựa trên cơ chế FIFO (first-in first-out) 🡺 Vào trước ra trước  - Method làm việc với queue :   * offer(E e**)** – Thêm phần tử vào cuối hàng đợi. * poll() – Lấy và xóa phần tử đầu tiên (trả về null nếu rỗng). * remove() – Giống poll(), nhưng sẽ ném lỗi nếu queue rỗng. * peek() – Xem phần tử đầu tiên mà không xóa. * isEmpty() – Kiểm tra xem queue có rỗng không. * size() – Trả về số lượng phần tử hiện có. |
| **4. Một số class triển khai của queue** | Một số class triển khai của queue :  + ArrayDeque  + LinkedLisst  + PriorityQueue |