**CÂU HỎI AUDIT MODULE 2**

1. JAVA là ngôn ngữ lập trình hướng đối tượng được phát triển bởi Sun Microsystems. Java cho phép viết một lần, chạy mọi nơi, độc lập nền tảng với Java Virtual Machine
2. JDK là Java Development Kit, bao gồm các công cụ và thư viện cần thiết để phát triển ứng dụng Java. JRE là Java Runtime Environment, cung cấp môi trường chạy cho các chương trình Java đã biên dịch. JVM là Java Virtual Machine, thực thi mã bytecode Java
3. Thông dịch thực hiện dịch từng dòng code khi chạy chương trình. Biên dịch dịch toàn bộ code thành ngôn ngữ máy trước khi chạy.
4. Java là ngôn ngữ biên dịch, code Java được biên dịch thành bytecode để chạy trên JVM
5. Java Platform bao gồm Java Virtual Machine, Java Programming Language và Java Application Programming Interface. Nó cung cấp môi trường phát triển và thực thi cho Java
6. Các kiểu dữ liệu chính trong Java bao gồm:
   1. Primitive type: boolean, byte, char, short, int, long, float, double. Giá trị mặc định là 0 (hoặc false đối với boolean)
   2. Reference type: class, interface, array, enum, etc. Giá trị mặc định là null.
7. Stack dùng cho các biến cục bộ, heap dùng cho các đối tượng được tạo bằng new. Stack nhanh hơn và nhỏ hơn heap
8. Các cách khởi tạo mảng trong Java:

int[] arr = new int[5];

int[] arr = {1, 2, 3, 4, 5};

int[] arr = new int[] {1, 2, 3, 4, 5};

1. Phần tử mảng có thể là bất kỳ kiểu dữ liệu nào trong Java. Giá trị mặc định của phần tử mảng là giá trị mặc định của kiểu dữ liệu tương ứng
2. OOP - Lập trình hướng đối tượng là một kỹ thuật lập trình dựa trên khái niệm đối tượng, bao gồm thuộc tính và phương thức
3. Class là một khuôn mẫu (template) để tạo ra các object. Object là một thể hiện cụ thể của class, chứa dữ liệu
4. Constructor là phương thức đặc biệt được gọi khi tạo mới đối tượng, dùng để khởi tạo trạng thái ban đầu cho object
5. Constructor được đặt tên trùng với tên lớp, không có kiểu trả về. Có thể overload constructor với các tham số khác nhau
6. Constructor dùng để khởi tạo đối tượng, method dùng để định nghĩa các hành vi hay chức năng của đối tượng
7. Tính bao đóng là che giấu thông tin, chi tiết bên trong lớp. Sử dụng access modifier (private, protected) để thu được tính bao đóng
8. Tham trị truyền giá trị của biến. Tham chiếu truyền địa chỉ ô nhớ của biến
9. static dùng để khai báo thuộc tính, phương thức tĩnh (class). Chúng thuộc về lớp, không cần tạo đối tượng.
10. Biến static chỉ nên dùng cho các giá trị không thay đổi. Tên biến static nên viết hoa và truy cập bằng tên lớp
11. Các loại biến: local variable, instance variable, static variable, parameter variable
12. Các access modifier: private, default, protected, public. Phạm vi truy cập từ hẹp đến rộng
13. Kế thừa cho phép một lớp có các thuộc tính và phương thức của lớp khác. Lớp con kế thừa từ lớp cha
14. Lớp con kế thừa tất cả các phương thức và thuộc tính public và protected của lớp cha. Ngoại trừ các phương thức và thuộc tính private.
15. Lớp Object là lớp cơ sở của tất cả các lớp trong Java, mọi lớp đều kế thừa từ lớp Object
16. Đa hình cho phép có nhiều hình thức thực thi khác nhau từ cùng một phương thức, dựa trên loại đối tượng
17. Overloading là nạp chồng phương thức, Overriding là ghi đè phương thức. Overloading xảy ra cùng lớp, Overriding xảy ra 2 lớp có quan hệ kế thừa
18. Ép kiểu là chuyển đổi kiểu dữ liệu tự động hoặc ép buộc từ kiểu này sang kiểu khác. Có ép kiểu ngầm định và ép kiểu tường minh.
19. Tính trừu tượng là che giấu chi tiết triển khai, chỉ thể hiện giao diện/chức năng cốt lõi cần thiết
20. Abstract class có thể có phương thức đã định nghĩa, interface chỉ định nghĩa phương thức trống. Interface thiên về tính trừu tượng hơn
21. Array có kích thước cố định, truy xuất nhanh hơn nhưng thao tác chèn/xóa phần tử chậm. ArrayList là mảng động, có thể thay đổi kích thước, truy xuất chậm hơn nhưng thao tác trên phần tử linh hoạt
22. ArrayList dùng mảng để lưu trữ, cho phép truy xuất ngẫu nhiên nhanh. LinkedList dùng danh sách liên kết để lưu trữ, cho phép thao tác thêm/xóa nhanh hơn nhưng truy xuất chậm.
23. List lưu trữ theo thứ tự, cho phép truy xuất bằng chỉ số và có phần tử trùng lặp. Set không có thứ tự, không cho phép truy xuất bằng chỉ số và không chứa phần tử trùng. Queue lưu trữ theo thứ tự, thao tác dựa trên FIFO
24. Set lưu trữ các phần tử không trùng lặp. Các lớp triển khai Set là HashSet, LinkedHashSet, TreeSet
25. Generic cho phép chỉ định kiểu dữ liệu khi khai báo class, interface. Tránh ép kiểu khi sử dụng
26. Generic giúp code rõ ràng và an toàn hơn về kiểu dữ liệu. Nhược điểm là khiến code phức tạp hơn và không thể sử dụng với kiểu dữ liệu nguyên thủy
27. Stack là cấu trúc dữ liệu LIFO. Các phương thức: push, pop, peek, empty, search
28. Queue là cấu trúc dữ liệu FIFO. Các lớp triển khai queue là LinkedList, PriorityQueue, ArrayDeque.
29. Các phương thức cơ bản của Queue: add, offer, remove, poll, element, peek
30. Comparable dùng để so sánh các đối tượng của cùng một lớp với nhau. Comparator dùng để so sánh các đối tượng khác lớp
31. Map lưu trữ dữ liệu dạng cặp key-value. Các lớp triển khai Map: HashMap, LinkedHashMap, TreeMap, Hashtable
32. equals() so sánh nội dung đối tượng. hashcode() trả về mã băm duy nhất đại diện cho đối tượng
33. Cây nhị phân là cấu trúc dữ liệu trong đó mỗi nút có nhiều nhất 2 nút con, thỏa mãn tính chất nhị phân.
34. Các cách duyệt cây nhị phân: preorder, inorder, postorder, level order
35. Exception (ngoại lệ) là sự kiện xảy ra trong quá trình thực thi làm chương trình bị đình chỉ
36. Các exception: RuntimeException, IOException, SQLException, ClassNotFoundException..
37. Error thường do lỗi hệ thống, không thể handle. Exception có thể kiểm soát và xử lý được
38. Các cách xử lý exception: try-catch, try-catch-finally, throw, throws, try-with-resources
39. Không nên dùng catch-all exception, chỉ nên catch exception cần xử lý. Tránh viết code có thể ném exception trong khối finally
40. Serialization dùng để chuyển đổi object thành dạng có thể lưu trữ (chuỗi byte). Lưu ý khai báo serialVersionUID cho lớp và loại bỏ các thuộc tính transient.