|  |  |
| --- | --- |
| 1. **Java là gì?** | * Java là ngôn ngữ lập trình hướng đối tượng * Có khả năng thực thi ở nhiều loại thiết bị(WORA) * Bảo mật cao * Được sử dụng rộng rãi |
| 1. **Phân biệt JDK, JRE, JVM** | |  |  |  |  | | --- | --- | --- | --- | | **Thành phần** | **Chức năng** | **Gồm những gì?** | **Dành cho ai** | | 1. **JVM (Java Virtual Machine) - Máy ảo Java** | * Chạy mã Java (bytecode) trên nhiều nền tảng | * Không chứa thư viện Java, chỉ thực thi bytecode | * Tích hợp trong JRE & JDK | | 1. **JRE (Java Runtime Environment) - Môi trường chạy Java** | * Cung cấp môi trường chạy ứng dụng Java | * JVM + Thư viện Java giúp chạy ứng dụng Java | * Người dùng chạy ứng dụng Java | | 1. **JDK (Java Development Kit) - Bộ công cụ phát triển Java** | * Bộ công cụ phát triển Java (viết, biên dịch, chạy) | * JRE + Trình biên dịch (javac) + Debugger | * Lập trình viên Java | |
| 1. **Javac là gì?** | * javac là trình biên dịch (compiler) của ngôn ngữ lập trình Java. Nó được sử dụng để biên dịch mã nguồn Java (.java) thành mã bytecode (.class), giúp chương trình có thể chạy trên **Máy ảo Java (JVM - Java Virtual Machine).** |
| 1. **Phân biệt  thông dịch và biên dịch?** | |  |  | | --- | --- | | **Biên dịch (Compiler)** | **Thông dịch (Interpreter)** | | * Chuyển toàn bộ mã nguồn thành mã máy trước khi chạy * Nhanh hơn (vì đã biên dịch trước khi chạy) * Báo lỗi ngay trong quá trình biên dịch | * Dịch từng dòng mã nguồn và thực thi ngay lập tức * Chậm hơn (vì phải dịch từng dòng khi chạy) * Chạy đến đâu báo lỗi đến đó | |
| 1. **Các kiểu dữ liệu trong java? Vì sao lại chia ra làm nhiều kiểu dữ liệu số?** | **-Kiểu dữ liệu nguyên thủy (Primitive Data Types)**: Là các kiểu dữ liệu cơ bản, không phải đối tượng, dùng để lưu trữ giá trị trực tiếp. Có 8 kiểu dữ liệu nguyên thủy trong Java:   * **byte**: 1 byte, giá trị từ -128 đến 127 * **short**: 2 byte, giá trị từ -32,768 đến 32,767 * **int**: 4 byte, giá trị từ -2^31 đến 2^31 - 1 * **long**: 8 byte, giá trị từ -2^63 đến 2^63 - 1 * **float**: 4 byte, số thực (dùng cho các phép toán có phần thập phân, độ chính xác thấp) * **double**: 8 byte, số thực (dùng cho các phép toán có phần thập phân, độ chính xác cao hơn float) * **char**: 2 byte, lưu trữ một ký tự Unicode * **boolean**: 1 bit, có giá trị true hoặc false   🡺Java phân chia các kiểu dữ liệu số để tối ưu hóa bộ nhớ, hiệu suất và độ chính xác cho từng trường hợp sử dụng cụ thể, giúp lập trình viên có thể chọn kiểu dữ liệu phù hợp với yêu cầu ứng dụng.  -Kiểu tham chiếu :   * Character * Interger * Long * Boolean * String |
| 1. **Tìm hiểu các loại toán tử trong java? trong java có toán tử === không?** | **-** **Các loại toán tử trong Java:**   * Toán tử số học * Toán tử so sánh * Toán tử logic * Toán tử tăng/giảm * Toán tử một ngôi   -Java **không có toán tử ===**. Sử dụng:   * **==** để so sánh giá trị (với kiểu dữ liệu nguyên thủy và tham chiếu). * **.equals()** để so sánh giá trị đối tượng. |
| 1. **Tìm hiểu Cấu trúc điều kiện trong java** | -Cấu trúc điều kiện trong Java cho phép chương trình thực hiện các hành động khác nhau dựa trên các điều kiện cụ thể.  🡺Dưới đây là các cấu trúc điều kiện phổ biến trong Java:   * Câu lệnh if * Câu lệnh if-else * Câu lệnh if lồngnhau(nestedif) * Câu lệnh if bậc thang * Câu lệnh switch-case |
| 1. **Cách input/output dữ liệu trong chương trình java** |  **Nhập dữ liệu**:   * **Scanner**: Dùng để nhập dữ liệu từ bàn phím.   + nextLine() để nhập chuỗi.   + nextInt() để nhập số nguyên.   + nextDouble() để nhập số thực.    **Xuất dữ liệu**:   * **System.out.println()**: In dữ liệu và xuống dòng. * **System.out.print()**: In dữ liệu mà không xuống dòng. * **System.out.printf()**: In dữ liệu với định dạng cụ thể. |