**1. JDK, JRE và JVM trong hệ sinh thái Java**

**Vai trò và sự khác nhau**

* **JDK (Java Development Kit)**: Bộ công cụ dành cho lập trình viên, bao gồm JRE và các công cụ phát triển như trình biên dịch (javac), trình gỡ lỗi, và các thư viện phát triển ứng dụng.
* **JRE (Java Runtime Environment)**: Môi trường chạy Java, bao gồm JVM và các thư viện lõi giúp thực thi các ứng dụng Java. JRE không có trình biên dịch (javac), nên không thể dùng để phát triển ứng dụng.
* **JVM (Java Virtual Machine)**: Máy ảo Java, có nhiệm vụ chuyển đổi bytecode (mã trung gian) thành mã máy để thực thi trên hệ điều hành cụ thể.

**Mối quan hệ giữa chúng**

* JVM là một phần của JRE, giúp chạy chương trình Java.
* JRE chứa JVM và các thư viện cần thiết để chạy Java.
* JDK chứa JRE và các công cụ phát triển, giúp lập trình viên viết và biên dịch mã nguồn Java.

**2. "Write Once, Run Anywhere" trong Java**

Khẩu hiệu **"Write Once, Run Anywhere" (WORA)** có nghĩa là một chương trình Java chỉ cần viết và biên dịch một lần nhưng có thể chạy trên nhiều nền tảng mà không cần biên dịch lại.

**Cơ chế giúp Java chạy trên nhiều nền tảng**

* Khi mã nguồn Java được biên dịch (javac), nó không chuyển thành mã máy trực tiếp mà thành **bytecode (.class)** – một dạng mã trung gian.
* JVM trên từng nền tảng sẽ dịch bytecode thành mã máy cụ thể tương thích với hệ điều hành đó.
* Nhờ JVM, Java không phụ thuộc vào hệ điều hành, giúp nó chạy trên Windows, macOS, Linux mà không cần biên dịch lại.

**3. Cài đặt và chạy chương trình Java đầu tiên bằng IntelliJ IDEA**

**Bước 1: Cài đặt IntelliJ IDEA và JDK**

* Tải và cài đặt **IntelliJ IDEA** từ [JetBrains](https://www.jetbrains.com/idea/download/).
* Cài đặt **JDK** từ [Oracle](https://www.oracle.com/java/) hoặc OpenJDK.

**Bước 2: Tạo dự án Java mới**

1. Mở IntelliJ IDEA → Chọn **New Project**.
2. Chọn **Java** và đảm bảo JDK được thiết lập.
3. Đặt tên cho dự án và chọn thư mục lưu trữ.

**Bước 3: Viết chương trình Java đầu tiên**

1. Trong src, nhấp chuột phải → **New → Java Class** → Đặt tên Main.
2. Viết mã Java:
3. public class Main {
4. public static void main(String[] args) {
5. System.out.println("Hello, Java!");
6. }
7. }

**Bước 4: Chạy chương trình**

* Nhấp chuột phải vào Main.java → **Run 'Main.main()'**.
* Kết quả **Hello, Java!** sẽ hiển thị trong console.

**4. Các phiên bản Java và sự khác nhau giữa Java SE, Java EE, Java ME**

**Các phiên bản Java**

Java có nhiều phiên bản, mỗi phiên bản được đánh số (Java 8, Java 11, Java 17, Java 21...). Các phiên bản **LTS (Long-Term Support)** như Java 8, Java 11, Java 17 được hỗ trợ lâu dài.

**Sự khác biệt giữa Java SE, Java EE và Java ME**

| **Phiên bản** | **Mục đích sử dụng** | **Đối tượng hướng đến** |
| --- | --- | --- |
| **Java SE (Standard Edition)** | Dùng để phát triển ứng dụng desktop, ứng dụng console, và các phần mềm Java cơ bản | Lập trình viên phát triển phần mềm thông thường |
| **Java EE (Enterprise Edition) (nay là Jakarta EE)** | Dùng để phát triển ứng dụng web, ứng dụng doanh nghiệp, có các API mạnh mẽ như Servlet, EJB, JPA | Các công ty, tổ chức cần hệ thống lớn, dịch vụ web, ứng dụng phân tán |
| **Java ME (Micro Edition)** | Dùng cho các thiết bị nhúng, điện thoại di động, IoT | Các nhà phát triển phần mềm cho thiết bị di động, hệ thống nhúng |

Java SE là nền tảng cơ bản, Java EE mở rộng Java SE để hỗ trợ ứng dụng doanh nghiệp, còn Java ME dùng cho thiết bị nhỏ với tài nguyên hạn chế.