**1. Vai trò và mối quan hệ của JDK, JRE và JVM**

* **JVM (Java Virtual Machine):**  
  Đây là “máy ảo” của Java, chịu trách nhiệm thực thi các file bytecode. Mỗi nền tảng (Windows, Linux, macOS) đều có một cài đặt JVM riêng, giúp chuyển đổi bytecode thành mã máy chạy trên nền tảng đó.
* **JRE (Java Runtime Environment):**  
  Là môi trường chạy Java, bao gồm JVM cùng với các thư viện lõi (core libraries) và các tài nguyên cần thiết để chạy ứng dụng Java. JRE không bao gồm các công cụ phát triển như trình biên dịch.
* **JDK (Java Development Kit):**  
  Là bộ công cụ dành cho lập trình viên, bao gồm JRE và các công cụ phát triển như trình biên dịch (javac), debugger, và tài liệu API. Nói cách khác, JDK là “hộp công cụ” cho việc phát triển ứng dụng Java.

**Mối quan hệ giữa chúng:**  
JDK chứa JRE, và bên trong JRE có JVM. Trong khi JVM chịu trách nhiệm chạy chương trình, JRE cung cấp môi trường thực thi cần thiết, và JDK cung cấp bộ công cụ phát triển cho lập trình viên.

**2. Khái niệm "Write once, run anywhere" và cơ chế hỗ trợ**

* **Khái niệm "Write once, run anywhere" (WORA):**  
  Ý nghĩa là sau khi viết và biên dịch chương trình Java thành bytecode, ứng dụng có thể chạy trên bất kỳ hệ điều hành nào có cài đặt JVM mà không cần phải chỉnh sửa hay biên dịch lại mã nguồn.
* **Cơ chế hỗ trợ:**  
  Java sử dụng bytecode – một dạng mã trung gian độc lập với nền tảng. Mỗi JVM trên các hệ điều hành khác nhau sẽ chịu trách nhiệm chuyển đổi bytecode này thành mã máy tương ứng với nền tảng đó. Điều này đảm bảo tính di động và khả năng chạy đa nền tảng của các ứng dụng Java.

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

Các bước cơ bản như sau:

1. **Cài đặt IntelliJ IDEA và JDK:**
   * Tải và cài đặt IntelliJ IDEA (phiên bản Community hoặc Ultimate) từ trang chủ của JetBrains.
   * Tải và cài đặt JDK (phiên bản phù hợp với dự án của bạn) từ trang Oracle hoặc OpenJDK.
   * Cấu hình đường dẫn JDK trong IntelliJ IDEA.
2. **Tạo dự án Java mới:**
   * Mở IntelliJ IDEA, chọn **New Project**.
   * Chọn **Java** và đảm bảo đã chọn đúng JDK.
   * Đặt tên dự án và chọn thư mục lưu trữ.
3. **Tạo file Java (ví dụ: Main.java):**
   * Trong cửa sổ dự án, chuột phải vào thư mục **src** > **New** > **Java Class**.
   * Đặt tên file (ví dụ: Main) và nhập đoạn mã mẫu, chẳng hạn:

public class Main {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

1. **Chạy chương trình:**
   * Click chuột phải vào file Main.java và chọn **Run 'Main.main()'** hoặc sử dụng nút chạy trên thanh công cụ.
   * Kết quả sẽ hiển thị trên cửa sổ console của IntelliJ IDEA.

Quy trình trên giúp sinh viên nắm được cách tạo, biên dịch và chạy một ứng dụng Java cơ bản trong môi trường phát triển hiện đại.

**4. Phân biệt các phiên bản Java: Java SE, Java EE và Java ME**

* **Java SE (Standard Edition):**
  + **Mục đích sử dụng:** Phát triển các ứng dụng desktop, ứng dụng console và các ứng dụng cơ bản.
  + **Đối tượng:** Lập trình viên làm việc với các ứng dụng thông thường và các hệ thống phần mềm độc lập.
  + **Nội dung:** Bao gồm core APIs, cấu trúc ngôn ngữ, và các thư viện tiêu chuẩn.
* **Java EE (Enterprise Edition):**
  + **Mục đích sử dụng:** Phát triển các ứng dụng doanh nghiệp, ứng dụng web, và các hệ thống phân tán.
  + **Đối tượng:** Các lập trình viên và doanh nghiệp cần xây dựng các ứng dụng có quy mô lớn, đòi hỏi tính mở rộng, bảo mật và quản lý giao dịch.
  + **Nội dung:** Bổ sung các API cho web services, Servlets, JSP, EJB, JPA, và nhiều công nghệ khác hỗ trợ phát triển ứng dụng doanh nghiệp.
* **Java ME (Micro Edition):**
  + **Mục đích sử dụng:** Phát triển ứng dụng cho các thiết bị có tài nguyên hạn chế như điện thoại di động, máy tính bảng nhỏ, và các thiết bị nhúng.
  + **Đối tượng:** Nhà phát triển ứng dụng dành cho thiết bị di động hoặc các hệ thống nhúng.
  + **Nội dung:** Cung cấp một tập hợp API nhỏ gọn, tối ưu cho các thiết bị không mạnh về phần cứng.

Sự khác biệt chủ yếu nằm ở mục đích sử dụng và quy mô của ứng dụng. Java SE là nền tảng cơ bản, Java EE mở rộng để đáp ứng nhu cầu của các ứng dụng doanh nghiệp, còn Java ME hướng đến các thiết bị có nguồn lực hạn chế.