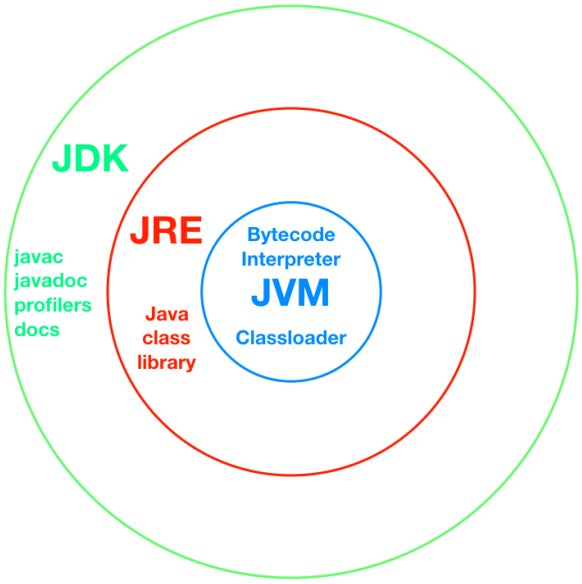
JRE, JDK, and JVM



1. JVM (Java Virtual Machine)

Role: The JVM is the **engine** that runs Java bytecode (compiled Java programs). It provides an environment where Java programs can execute.

Key Functionality:

- **Platform independence**: JVM makes Java programs platform-independent by abstracting the underlying hardware and operating system.
- **Execution**: It reads and executes Java bytecode from .class files (produced by compiling .java files).
- **Memory Management**: JVM manages memory using garbage collection, which automatically deallocates memory that is no longer in use.

2. JRE (Java Runtime Environment)

Role: The **JRE** is a package that provides the **libraries** and **JVM** necessary to run Java applications.

Key Components:

- **JVM**: The Java Virtual Machine, which executes Java bytecode.
- **Core Libraries**: A set of pre-written Java classes (like java.lang, java.util, java.io, etc.) that provide essential functionality for Java applications.
- **Other Resources**: Configuration files and other runtime resources.

3. JDK (Java Development Kit)

Role: The **JDK** is a **superset** of the JRE and includes tools needed for **developing** Java applications.

Key Components:

- **JRE**: Includes the full JRE for running Java applications.
- **Development Tools**: Includes the **Java Compiler (javac)** to convert .java source code into .class bytecode, and other tools like **debugger (jdb)**, **JavaDoc** (for generating API documentation), **jar** (for packaging classes), and more.