JDK – JAVA DEVLOPMENT KIT :

JDK (Java Development Kit) is a software development kit required to develop applications in Java. When you download JDK, JRE is also downloaded with it.

In addition to JRE, JDK also contains a number of development tools (compilers, JavaDoc, Java Debugger, etc).



**The Java Development Kit  is an implementation of  one of the Java Platform:**

* [Standard Edition](about:blank) (Java SE),
* [Java Enterprise Edition (Java EE)](about:blank),
* [Micro Edition (Java ME),](about:blank)

**Contents of JDK**

The JDK has a private Java Virtual Machine (JVM) and a few other resources necessary for the development of a Java Application.

**JDK contains:**

* Java Runtime Environment (JRE),
* An interpreter/loader (Java),
* A compiler (javac),
* An archiver (jar) and many more.

**Most Popular JDKs:**

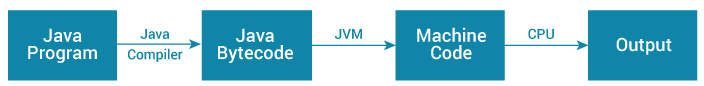
* **Oracle JDK:** the most popular JDK and the main distributor of Java11,
* **OpenJDK:** Ready for use: JDK 15, JDK 14, and JMC,
* **Azul Systems Zing:** efficient and low latency JDK for Linux os,

**JVM – JAVA VIRTUAL MACHIINE :**

JVM (Java Virtual Machine) is an abstract machine that enables your computer to run a Java program.

When you run the Java program, Java compiler first compiles your Java code to bytecode. Then, the JVM translates bytecode into native machine code (set of instructions that a computer's CPU executes directly).

Java is a platform-independent language. It's because when you write Java code, it's ultimately written for JVM but not your physical machine (computer). Since JVM executes the Java bytecode which is platform-independent, Java is platform-independent.



What are the Features of the JVM?

* **Platform independence:** The JVM includes a platform-independent runtime environment which allows for the execution of Java applications on most hardware and software architectures.
* **Memory management:** The JVM manages memory automatically vis its built-in garbage collector, which frees up memory when the program is finished with it, saving the developer from having to account for memory allocation and deallocation.
* **Security:** The JVM enforces security policies to prevent malicious code from gaining access to system resources.
* **Just-In-Time compilation:** JIT compilation improves application performance by compiling frequently executed bytecode into native machine language.
* **Multithreading:** The JVM supports multithreading, which lets Java programs run multiple threads of execution at the same time.

JRE – JAVA RUNTIME ENVIRONMRENT :

JRE (Java Runtime Environment) is a software package that provides Java class libraries, Java Virtual Machine (JVM), and other components that are required to run Java applications.

JRE is the superset of JVM.



**Components of Java JRE**

The components of JRE are mentioned below:

* Integration libraries include Java Database Connectivity (JDBC)
* Java Naming, Interface Definition Language (IDL)
* Directory Interface (JNDI)
* Remote Method Invocation Over Internet Inter-Orb Protocol (RMI-IIOP)
* Remote Method Invocation (RMI)
* Scripting

**Working of JRE**

Java Development Kit (JDK) and Java Runtime Environment (JRE)  both interact with each other to create a sustainable runtime environment that enables Java-based applications to run seamlessly on any operating system. The  JRE runtime architecture consists of the following elements as listed:

1. [ClassLoader](about:blank)
2. ByteCode verifier
3. [Interpreter](about:blank)

IDE – INTEGRATED DEVLOPMENT ENVIRONMENT :

An **IDE (Integrated Development Environment)** is software that combines commonly used developer tools into a compact GUI (graphical user interface) application. It is a combination of tools like a **code editor**, **code compiler**, and **code debugger**with an integrated terminal.

**Why are IDEs important?**

By providing frequently used **developer tools**all in one simple interface, one can directly get on to building their applications without going through the hardship of manually configuring and integrating the development environment.

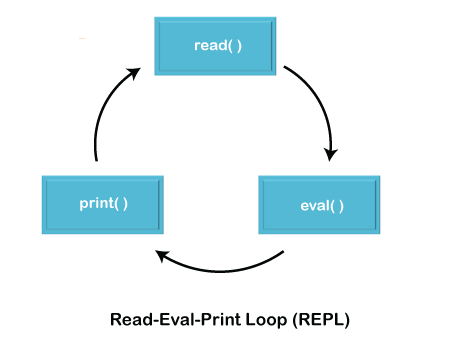
**Common Features of an IDE (Integrated Development Environment)**

IDEs provide a broad variety of features which typically consist of:

* **Editor:**Typically a text editor can help you write software code by highlighting syntax with visual cues, providing language-specific auto-completion, and checking for bugs as you type.
* **Compiler:**A compiler interprets human-readable code into machine-specific code that can be executed on different operating systems like Linux, Windows, or Mac OS. Most IDEsusually come with built-in compilers for the language it supports.
* **Debugger:**A tool that can assist developers to testand debugtheir applications and graphically point out the locations of bugsor errorsif any.

REPL – READ EVAL PRINT AND LOOP :

The **Read-Eval-Print Loop** or **REPL** is a shell interface. This interface reads and evaluates each line of input and then prints the result. The **Read-Eval-Print Loop** helps us to interact with our application runtime present in a specific state. The commands are read and evaluated by the **REPL** and print the result. After printing the result, REPL goes back to the start to read, evaluate and print our next input.



Using REPL, we can easily write and test our Java code without compiling it and can see the output directly on the console.

Why is REPL so helpful?

By using the REPL, we don't need to compile or test our Java code using the javac command. After using the REPL,

1. No need of an editor to write the Java program.
2. No need to save the Java program.
3. No need to compile Java program.
4. No need to do editing if any compile-time or runtime error comes.
5. No need to repeat the process.