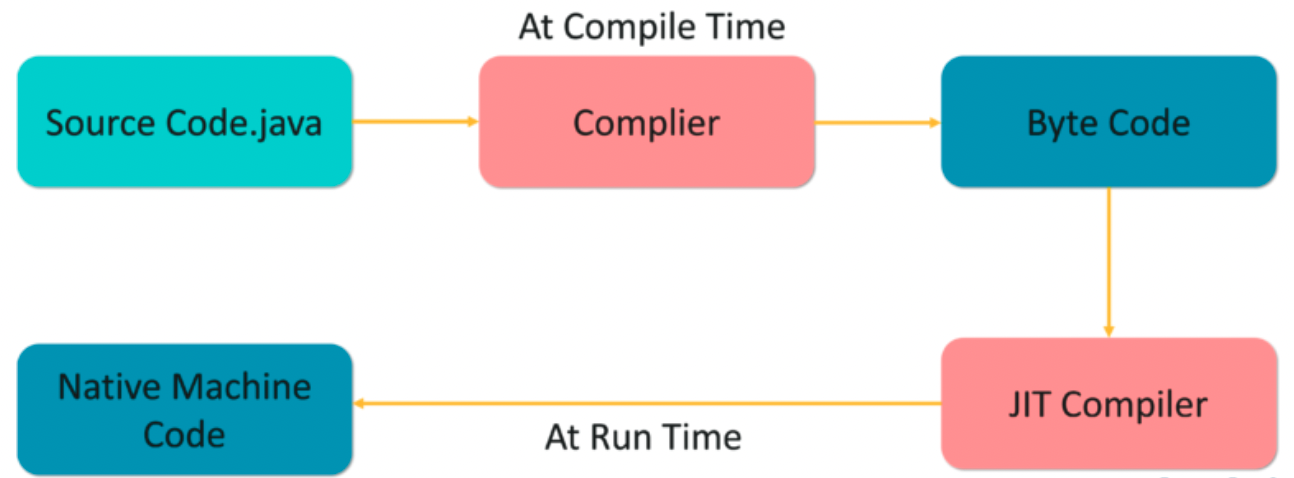
The **Just-In-Time (JIT) compiler** is a **component of the Java™ Runtime Environment** that **improves the performance of Java applications at run time**.

**Java programs consist of classes**, which contain **platform-neutral bytecodes** that can be **interpreted by a JVM** **on many different computer architectures**. **At run time**, the **JVM loads the class files**, **determines the semantics of each individual bytecode and performs the appropriate computation**. The **additional processor and memory usage** **during interpretation** **mean that a Java application** **performs more slowly than a native application**. The **JIT compiler** helps **improve the performance of Java programs** **by** **compiling bytecodes into native machine code at run time**.



The **JIT compiler is enabled by default** and is **activated when a Java method is called**. The **JIT compiler compiles the bytecodes of that method into native machine code**, compiling it "**just in time**" to run. **When a method has been compiled**, the **JVM calls the compiled code of that method directly instead of interpreting it**. Theoretically, if compilation did not require processor time and memory usage, compiling every method could allow the speed of the Java program to approach that of a native application.

**JIT compilation does require processor time and memory usage**. When the **JVM first starts up, thousands of methods are called**. **Compiling all** **of these methods can** **significantly affect startup time**, even if the program eventually achieves very good peak performance.

**In practice, methods are not compiled the first time they are called**. **For each method**, the **JVM maintains a call count**, which is **incremented every time the method is called**. The **JVM interprets a method** **until its call count exceeds a JIT compilation threshold**. Therefore, **often-used methods are compiled soon** **after the JVM has started**, and **less-used methods are compiled much later**, **or not at all**. The **JIT compilation threshold** **helps the JVM start quickly** and **still have improved performance**. The **threshold has been carefully selected** to obtain an **optimal balance between startup times and long-term performance**.

The **JIT compiler can be disabled**, in which case the **entire Java program will be interpreted**. **Disabling the JIT compiler is not recommended** **except to diagnose or workaround JIT compilation problems**.