

先从经典题入手，并给出答案。带着问题去进一步学习Java平台相关的基础概念。

Java平台

版本号：2018/8/27-1(13:00)

- [Java平台](#)
 - [经典题](#)
 - [Java平台](#)
 - [JVM运行模式](#)
 - [AOT\(提前编译\)](#)
 - [知识储备](#)
 - [参考资料](#)

经典题

1、请谈谈对Java平台的理解？

1. Java语言是一种面向对象的语言
2. 具有两个重要特性：
 1. "Write once,run anywhere", 具有优秀的跨平台能力。
 2. "GC, garbage collector", Java通过GC机制分配回收内存。
3. JRE, Java运行环境(Java Runtime Enviornment), 包含了JVM和Java类库。
4. JDK, Java开发工具(Java Development Kit), 是JRE的超集, 提供了更多工具: 如编译器、诊断工具等。

2、"Java是解释执行的语言", 是否正确？

1. 不准确。
2. 开发的Java源代码会通过 `javac` 编译成 字节码-bytecode
3. 运行时, JVM内嵌的解释器会将 字节码 转换成 机器码, 这里是 解释执行
4. 常用的JVM是Oracle JDK提供的 HotSpot JVM, 提供了 JIT-just in time 编译器。
5. JIT即时编译器, 也就是动态编译器, 会在运行时将热点代码 编译 成机器码, 这就是 编译执行

Java平台

1、Java平台包括了哪些内容？

1. 基本语言特性：面向对象、反射、泛型、lambda等
2. Java类库：
 1. 核心类库：IO、NIO、网络、utils、并发、集合等
 2. 安全类库
 3. jdk.management等类库
3. JVM：
 1. (GC)垃圾收集器: SerialGC、Parallel GC、CMS、G1等
 2. 动态编译
 3. 运行时
 4. 辅助功能：如JFR等
4. 工具
 1. 辅助工具：jlink、jar、jdeps
 2. 编译器：javac、sjavac
 3. 诊断工具：jmap、jstack、jconsole、jhsdb、jcmd
5. 生态圈
 1. Android
 2. JavaEE
 3. Spring
 4. Maven

2、Java通常分为编译期和运行时

3、Java的编译和C/C++的编译有什么不同？

1. C/C++的编译是直接变成机器码
2. Java编译时声称Java字节码(保存在.class文件中)，并不是可以直接执行的机器码。
3. JVM屏蔽了操作系统和硬件细节。实现了“一次编译，到处运行”的特性。

4、直接运行Java字节码的处理器

1. 出现过能直接运行Java字节码的处理器。
2. 但是效果并不是很好。

5、Java程序是如何运行的？

1. 运行时，JVM会通过类加载器加载字节码，解释或者编译执行。
2. 现在都是解释和编译混合的一种版本，即所谓的 混合模式(-Xmixed)

6、Java程序执行的客户端模式和服务端模式

1. Server服务端的JVM，会进行上万次调用以收集足够的信息，然后进行高效的编译。
2. Client客户端这个阈值是1500次。

7、HotSpot中的JIT编译器

1. 第一种：C1，对应于Client模式。适用于对启动速度敏感的应用。
2. 第二种：C2，对用于Server模式。适用于长时间运行的服务端应用。
3. HotSpot采用了 分层编译
4. JIT是以方法为单位。

JVM运行模式

8、JVM参数指定 `-Xint` 会只进行解释编译，会损失JIT的性能优势。

9、JVM参数指定 `-Xcomp` 会关闭解释器，启动速度会很慢，且性能不一定很高(无法利用JIT的优化方式，如：分支预测)

AOT(提前编译)

10、Java有种新的编译方式就是AOT

1. AOT-Ahead of Time,提前编译
2. 直接将字节码编译成了机器码，用于避免JIT预热的性能开销。
3. Oracle JDK 9 引入了实验性的AOT特性，并且 分层编译 和 AOT 可以协作使用。

知识储备

1、Java的GC机制是什么？

参考资料

1. [Java垃圾回收（GC）机制详解](#)