

Java的魔力：字节码 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022_Java_E7_9A_84_E9_AD_94_c104_253536.htm Java 解决上述两个问题安全性和可移植性的关键在于Java 编译器的输出并不是可执行的代码，而是字节码（bytecode）。字节码是一套设计用来在Java运行时系统下执行的高度优化的指令集，该Java运行时系统称为Java 虚拟机(JavaVirtual Machine, JVM)。在其标准形式下，JVM是一个字节码解释器。这可能有点让人吃惊，因为像C之类语言的编译结果是可执行的代码。事实上，出于对性能的考虑，许多现代语言都被设计为编译型，而不是解释型。然而，正是通过JVM运行Java 程序才有助于解决在Internet 上下载程序的主要问题。这就是Java 输出字节码的原因。将一个Java 程序翻译成字节码，有助于它更容易地在一个大范围的环境下运行程序。原因非常直接：只要在各种平台上都实现Java 虚拟机就可以了。在一个给定的系统中，只要系统运行包存在，任何Java 程序就可以在该系统上运行。记住：尽管不同平台的Java 虚拟机的细节有所不同，但它们都解释同样的Java 字节码。如果一个Java 程序被编译为本机代码，那么对于连接到Internet 上的每一种CPU 类型，都要有该程序的对应版本。这当然不是一个可行的解决方案。因此，对字节码进行解释是编写真正可移植性程序的最容易的方法。对Java 程序进行解释也有助于它的安全性。因为每个Java 程序的运行都在Java 虚拟机的控制之下，Java 虚拟机可以包含这个程序并且能阻止它在系统之外产生副作用。正如你将看到的，Java 语言特有的某些限制增强了它的安全性。被解释的程序的运

行速度通常确实会比同一个程序被编译为可执行代码的运行速度慢一些。但是对Java来说，这两者之间的差别不太大。使用字节码能够使Java运行时系统的程序执行速度比你想象的快得多。尽管Java被设计为解释执行的程序，但是在技术上Java并不妨碍动态将字节码编译为本机代码。SUN公司在Java 2发行版中提供了一个字节码编译器JIT（Just In Time，即时）。JIT是Java虚拟机的一部分，它根据需要、一部分一部分地将字节码实时编译为可执行代码。它不能将整个Java程序一次性全部编译为可执行的代码，因为Java要执行各种检查，而这些检查只有在运行时才执行。记住这一点是很重要的，因为JIT只编译它运行时需要的代码。尽管如此，这种即时编译执行的方法仍然使性能得到较大提高。即使对字节码进行动态编译后，Java程序的可移植性和安全性仍能得到保证，因为运行时系统（该系统执行编译）仍然能够控制Java程序的运行环境。不管Java程序被按照传统方式解释为字节码，还是被动态编译为可执行代码，其功能是相同的。

100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com