

1.1.2 Java 的特点

1. 简单性

Java 语言是在 C、C++ 语言的基础上发展而来的，既具有 C 和 C++ 的优点，又摒弃了运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集机制，大大简化了程序设计者的内存管理工作。另外，Java 也适合在小型机上运行，它的基本解释器及类的支持只有 40KB 左右、标准类库和线程的支持也只有 215KB 左右。同时，Java 开发工具集 Java Development Kit（简称 JDK）提供了丰富的基础类库，使得学习者能够在短时间内掌握 Java 语言开发技术。

2. 跨平台性

Java 语言一个非常重要的特点就是平台无关性。Java 虚拟机在执行字节码时，把字节码解释成具体平台上的机器指令执行。Java 的字节码文件运行于 Java 虚拟机（Java Virtual Machine，简称 JVM）之



上,实现跨平台运行,即“一次编译,到处运行”。

3. 安全性

Java 的编程类似 C++, 但舍弃了 C++ 的指针对存储器地址的直接操作, 程序运行时, 内存由 JVM 分配, 这样可以避免病毒通过指针侵入系统。Java 对程序提供了安全管理器, 防止程序的非法访问。

4. 面向对象性

Java 吸取了 C++ 面向对象的思想, 将数据封装于类中, 利用类的优点, 实现了程序的简洁性和便于维护性。封装、继承、多态等有关面向对象的特性, 使程序代码只需一次编译, 然后通过上述特性重复使用, 增强了代码的复用性。

5. 分布式

Java 建立在扩展 TCP/IP 网络平台上。类库提供了用 HTTP 和 FTP 协议传送和接受信息的方法, 这使得程序员使用网络上的文件和使用本机文件一样容易。

6. 健壮性

Java 程序在编译和运行时都进行错误检查。类型检查能够帮助检查出许多开发早期出现的错误, Java 自己操纵内存减少了内存出错的可能性, 这些功能特征大大缩短了开发 Java 应用程序的周期, 并且 Java 还提供了 Null 指针检测、数组边界检测、异常出口、Bytecode 校验等功能。

7. 解释性

C、C++ 等语言, 都是针对 CPU 芯片进行编译, 生成机器代码, 该代码的运行和特定的 CPU 有关。Java 与 C 和 C++ 不同, 它不针对 CPU 芯片进行编译, 而是把程序编译成称为字节码的“中间代码”。字节码是很接近机器码的文件, 可以在提供了 Java 虚拟机的任何系统上被解释执行。

8. 动态性

Java 程序的基本组成单元就是类, 有些类是自己编写的, 有些是从类库中引入的, 而类又是运行时动态装载的, 这就使得 Java 可以进行动态的程序维护, 不像 C++, 每当类库升级以后, 如果想让程序具有新类库提供的功能, 就需要修改程序, 重新编译。

9. 高效性

Java 编译后的中间代码在虚拟机上运行, Java 字节码的设计使之能够很容易地直接转换成对应于特定 CPU 的机器码, 从而得到较高的性能。

10. 多线程

多线程机制使应用程序能够并发执行, 而且同步机制保证了对共享数据的正确操作。通过使用多线程, 程序设计者可以分别用不同的线程完成特定的行为, 而不需要采用全局的事件循环机制, 这样很容易实现网络上的实时交互行为。

