

浮动主题

- 1.Java是HTML的扩展
- 2. 使用XML，所以不需要Java
- 3.Java 是一种非常容易学习的程序设计语言
- 4.Java 将成为适用于所有平台的通用性编程语言
- 5. Java 只不过是另外一种程序设计语言
- 6. Java 是专用的，应该避免使用
- 7. Java 是解释型的，因此对于关键的应用程序速度太慢了
- 8. 所有的 Java 程序都是在网页中运行的
- 9. Java 程序是主要的安全风险
- 10. JavaScript 是 Java 的简易版

1.5 关于Java的常见误解

1.4 Java发展简史

第一章 JAVA程序设计概述

1.1 java程序平台

1.2 java “白皮书”的关键术语

1.3 Java applet 与 Internet

- 1) 简单性 — 1. 没有头文件、指针运算、结构、联合、操作符重载、虚基类等，但没有清除所有不适当的C++特性 2. 小-----支持开发能够在小型机器上独立运行的软件
- 2) 面向对象 — 是一种程序设计技术，将重点放在数据（即对象）和对象的接口上
- 3) 分布式 — Java有一个丰富的例程库，用于处理像 HTTP 和 FIT 之类的 TCP/IP 协议
- 4) 健壮性 — Java 投入了大量的精力进行早期的问题检测、后期动态的（运行时）检测，并消除了容易出错的情况... .. Java 和 C++ 最大的不同在于 Java 采用的指针模型可以消除重写内存和损坏数据的可能性
- 5) 安全性 — 使用 Java 可以构建防病毒、防篡改的系统。从一开始，Java 就设计成能够防范各种攻击
- 6) 体系结构中立 — 编译器生成一个体系结构中立的目标文件格式，这是一种编译过的代码，只要有Java 运行时系统，这些编译后的代码可以在许多处理器上运行
- 7) 可移植性 — 与 C 和 C++ 不同，Java 规范中没有 “依赖具体实现” 的地方基本数据类型的大小以及有关运算都做了明确的说明
- 8) 解释型 — Java 解释器可以在任何移植了解释器的机器上执行 Java 字节码
- 9) 高性能 — 尽管对解释后的字节码性能已经比较满意，但在有些场合下还需要更加高效的性能。字节码可以（在运行时刻）动态地翻译成对应运行这个应用的特定 CPU 的机器码
- 10) 多线程 — 多线程可以带来更好的交互响应和实时行为
- 11) 动态性 — 从各种角度看，Java 与 C 或 C++ 相比更加具有动态性。它能够适应不断发展的环境库中可以自由地添加新方法和实例变量，而对客户端却没有任何影响

- 运行时堆栈溢出。如蠕虫和病毒常用的攻击手段。
- 破坏自己的进程空间之外的内存。
- 未经授权读写文件。