**第一章 Java程序概述**

**1.1 Java的关键术语**

* **简单性**

**Java剔除了C++中难以理解、易混淆的特性。例如指针运算、虚基类等等。另一方面是小，Java的目标之一是支持开发能够在小型机器上独立运行的软件。基本的解释器以及类支持大约仅为40KB；再加上基础的标准类库和对线程的支持大约需要增加175KB。**

* **面向对象**

**面向对象的程序设计语言可以投射成一个木匠制作椅子的过程注重的是椅子本身，而不是直接考虑使用什么工具去制作。注重椅子本身可以理解成面向对象，而第二位则是面向过程。**

* **健壮性**

**Java的设计目标之一在于使得Java编写的程序具有多方面的可靠性。Java投入了大量的精力进行早期的问题检测，后期动态的检测，并消除了有出错倾向的状态。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。**

* **安全性**

**Java适用于网络/分布式环境。为了达到这个目标，在安全方面投入了很大精力，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袭击，包括，运行时堆栈溢出、蠕虫等病毒常用的袭击手段、未经授权读写文件，在自己的处理控件之外破坏内存。**

* **可移植性**

**1. Java作为一种编程语言:源代码可移植性**

**不论Java的源代码运行在何种cpu、操作系统或者Java编译器上都产生一样的结果。相对应的使用C或者C++的人又太多的选择，由于在许多细节都没有严格的定义，这样就导致在不同环境下运行程序会有不同的结果。由于Java严密的语义结构，而使编译器不承担这一职责。**

**2. Java作为一个虚拟机:cpu可移植性**

**大多数编译器产生的目标代码只能运行在一中cpu（如Intel的X86系列），即使那种支持多种编译器的目标代码，也不能产生适合多种cpu的目标代码。而java不一样，java编译器产生的目标代码针对的是一种并不存在的cpu--java虚拟机，而不是一种具体的cpu。java虚拟机能掩盖不同cpu之间的差别，从而使java-code运行在安装有虚拟机的机器上。**

**3. Java作为一种虚拟的操作系统和图形用户界面:操作系统的可移植性**

**即使经过重新编译，大多数的用C和C++编写的Windows程序也不能在Unix或Macintosh系统上运行。这是为什么呢?因为程序员在编写Windows程序时使用了大量的WindowsAPI和中断调用，而Windows程序对系统功能的调用与Unix和Macintosh程序有很大的差别，所以除非将全套WindowsAPI移植到其它操作系统上，否则重编译的程序仍不能运行。JAVA采用了提供一套与平台无关的库函数(包括AWT、UTIL、LANG等等)的方法来解决这个问题。就象JVM提供了一个虚拟的CPU一样，JAVA库函数提供了一个虚拟的GUI环境。**

* **解释性**

**Java字节码使“写一次，到处运行”成为可能。可以在任何有Java编译器的平台上把Java程序编译成字节码。这个字节码可以运行在任何Java VM上。例如，同一个Java程序可以运行在WindowsNT、Solaris和Macintosh上。**