

第一章Java语言的产生及其特点(2) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/260/2021_2022__E7_AC_AC_E4_B8_80_E7_AB_A0J_c97_260926.htm 1.2.2 面向对象
面向对象可以说是Java最重要的特性。Java语言的设计完全是面向对象的，它不支持类似C语言那样的面向过程的程序设计技术。Java支持静态和动态风格的代码继承及重用。单从面向对象的特性来看，Java类似于SmallTalk，但其它特性、尤其是适用于分布式计算环境的特性远远超越了SmallTalk。 1.2.3 分布式Java
包括一个支持HTTP和FTP等基于TCP/IP协议的子库。因此，Java应用程序可凭借URL打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。为分布环境尤其是Internet提供动态内容无疑是一项非常宏伟的任务，但Java的语法特性却使我们很容易地实现这项目标。 1.2.4 健壮Java
致力于检查程序在编译和运行时的错误。类型检查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。Java还实现了真数组，避免了覆盖数据的可能。这功能特征大大缩短了开发Java应用程序的周期。Java提供：Null指针检测数组边界检测异常出口字节代码校验 1.2.5 结构中立的另外，为了建立Java作为网络的一个整体，Java将它的程序编译成一种结构中立的中间文件格式。只要有Java运行系统的机器都能执行这种中间代码。现在，Java运行系统有Solaris2.4(SPARC),Win32系统(Windows95和Windows NT)等。

Java源程序被编译成一种高层次的与机器无关的byte-code格式语言，这种语言被设计在虚拟机上运行，由机器相关的运行调试器实现执行。

1.2.6 安全

Java的安全性可从两个方面得到保证。一方面，在Java语言里，象指针和释放内存等C功能被删除，避免了非法内存操作。另一方面，当Java用来创建浏览器时，语言功能和一?copy.浏览器本身提?copy.的功能结合起来，使它更安全。Java语言在你的机器上执行前，要经过很多次的测试。它经过代码校验，检查代码段的格式，检测指针操作，对象操作是否过分以及试图改变一个对象的类型。

1.2.6.1 Byte-code校验

如果byte-code通过代码校验，没有返回错误，我们可知道：代码没有堆栈上溢出和下溢出所有操作代码参数类型都是正确的没有发生非法数据转换，如将整数转换成指针。访问对象操作是合法的

1.2.6.2 类装载

Class Loader通过将本机类与网络资源类的名称分开，来保持安全性。因为调入类时总要经过检查，这样避免了特洛伊木马现象的出现。从网络上下载的类型被调进一个与源相关的私有的名字域。当一个私有类访问另一个类时，build-in(本机类)首先被检查，然后检查相关的类。这样就避免了破坏本机类情况的出现。

1.2.7 可移植的同体系结构无关的特性

使得Java应用程序可以在配备了Java解释器和运行环境的任何计算机系统上运行，这成为Java应用软件便于移植的良好基础。但仅仅如此还不够。如果基本数据类型设计依赖于具体实现，也将为程序的移植带来很大

不便。例如在Windows 3.1中整数(Integer)为16bits，在Windows 95中整数为32bits，在DEC Alpha中整数为64bits，在Intel 486中为32bits。通过定义独立于平台的基本数据类型及其运算，Java数据得以在任何硬件平台上保持一致。Java语言的基本数据类型及其表示方式如下：
byte 8-bit 二进制补码 short 16-bit 二进制补码 int 32-bit 二进制补码 long 64-bit 二进制补码 float 32-bit IEEE 754浮点数 double 64-bit IEEE 754浮点数 char 16-bit Unicode字符
在任何Java解释器中，数据类型都是依据以上标准具体实现的。因为几乎目前使用的所有CPU都能支持以上数据类型、8~64位整数格式的补码运算和单/双精度浮点运算。Java编译器本身就是用Java语言编写的。Java运算系统的编制依据POSIX方便移植的限制，用ANSI C语言写成。Java语言规范中也没有任何"同具体实现相关"的内容。
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