

Java语言特点及开发工具JDK PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Java_E8_AF_AD_E8_A8_80_c104_144417.htm

一、Java语言特点 Java是一种跨平台，适合于分布式计算环境的面向对象编程语言。具体来说，它具有如下特性：简单性、面向对象、分布式、解释型、可靠、安全、平台无关、可移植、高性能、多线程、动态性等。下面我们将重点介绍Java语言的面向对象、平台无关、分布式、多线程、可靠和安全等特性。

1.面向对象 面向对象其实是现实世界模型的自然延伸。现实世界中任何实体都可以看作是对象。对象之间通过消息相互作用。另外，现实世界中任何实体都可归属于某类事物，任何对象都是某一类事物的实例。如果说传统的过程式编程语言是以过程为中心以算法为驱动的话，面向对象的编程语言则是以对象为中心以消息为驱动。用公式表示，过程式编程语言为：程序=算法 数据；面向对象编程语言为：程序=对象 消息。所有面向对象编程语言都支持三个概念：封装、多态性和继承，Java也不例外。现实世界中的对象均有属性和行为，映射到计算机程序上，属性则表示对象的数据，行为表示对象的方法（其作用是处理数据或同外界交互）。所谓封装，就是用一个自主式框架把对象的数据和方法联在一起形成一个整体。可以说，对象是支持封装的手段，是封装的基本单位。Java语言的封装性较强，因为Java无全程变量，无主函数，在Java中绝大部分成员是对象，只有简单的数字类型、字符类型和布尔类型除外。而对于这些类型，Java也提供了相应的对象类型以便与其他对象交互操作。多态性就是多种表现形式，具体来说

，可以用“一个对外接口，多个内在实现方法”表示。举一个例子，计算机中的堆栈可以存储各种格式的数据，包括整型，浮点或字符。不管存储的是何种数据，堆栈的算法实现是一样的。针对不同的数据类型，编程人员不必手工选择，只需使用统一接口名，系统可自动选择。运算符重载

(operatoroverload)一直被认为是一种优秀的多态机制体现，但由于考虑到它会使程序变得难以理解，所以Java最后还是把它取消了。继承是指一个对象直接使用另一对象的属性和方法。事实上，我们遇到的很多实体都有继承的含义。例如，若把汽车看成一个实体，它可以分成多个子实体，如：卡车、公共汽车等。这些子实体都具有汽车的特性，因此，汽车是它们的“父亲”，而这些子实体则是汽车的“孩子”。Java提供给用户一系列类(class)，Java的类有层次结构，子类可以继承父类的属性和方法。与另外一些面向对象编程语言不同，Java只支持单一继承。

2平台无关性

Java是平台无关的语言是指用Java写的应用程序不用修改就可可在不同的软硬件平台上运行。平台无关有两种：源代码级和目标代码级。C和C具有一定程度的源代码级平台无关，表明用C或C写的应用程序不用修改只需重新编译就可以在不同平台上运行。Java主要靠Java虚拟机(JVM)在目标码级实现平台无关性。JVM是一种抽象机器，它附着在具体操作系统之上，本身具有一套虚拟机指令，并有自己的栈、寄存器组等。但JVM通常是在软件上而不是在硬件上实现。(目前，SUN系统公司已经设计实现了Java芯片，主要使用在网络计算机NC上。另外，Java芯片的出现也会使Java更容易嵌入到家用电器中。)JVM是Java平台无关的基础，在JVM上，有一个Java解释器用来解

释Java编译器编译后的程序。Java编程人员在编写完软件后，通过Java编译器将Java源程序编译为JVM的字节代码。任何一台机器只要配备了Java解释器，就可以运行这个程序，而不管这种字节码是在何种平台上生成的。另外，Java采用的是基于IEEE标准的数据类型。通过JVM保证数据类型的一致性，也确保了Java的平台无关性。Java的平台无关性具有深远意义。首先，它使得编程人员所梦寐以求的事情（开发一次软件在任意平台上运行）变成事实，这将大大加快和促进软件产品的开发。其次Java的平台无关性正好迎合了“网络计算机”思想。如果大量常用的应用软件（如字处理软件等）都用Java重新编写，并且放在某个Internet服务器上，那么具有NC的用户将不需要占用大量空间安装软件，他们只需要一个Java解释器，每当需要使用某种应用软件时，下载该软件的字节代码即可，运行结果也可以发回服务器。目前，已有数家公司开始使用这种新型的计算模式构筑自己的企业信息系统。

3 分布式

分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机上，操作分布是指把一个计算分散在不同主机上处理。Java支持WWW客户机/服务器计算模式，因此，它支持这两种分布性。对于前者，Java提供了一个叫作URL的对象，利用这个对象，你可以打开并访问具有相同URL地址上的对象，访问方式与访问本地文件系统相同。对于后者，Java的applet小程序可以从服务器下载到客户端，即部分计算在客户端进行，提高系统执行效率。Java提供了一整套网络类库，开发人员可以利用类库进行网络程序设计，方便得实现Java的分布式特性。

4 可靠性和安全性

Java最初设计目的是应用于电子类消费产品，因此要求较高的可靠性

。Java虽然源于C，但它消除了许多C不可靠因素，可以防止许多编程错误。首先，Java是强类型的语言，要求显式的方法声明，这保证了编译器可以发现方法调用错误，保证程序更加可靠；其次，Java不支持指针，这杜绝了内存的非法访问；第三，Java的自动单元收集防止了内存丢失等动态内存分配导致的问题；第四，Java解释器运行时实施检查，可以发现数组和字符串访问的越界，最后，Java提供了异常处理机制，程序员可以把一组错误代码放在一个地方，这样可以简化错误处理任务便于恢复。由于Java主要用于网络应用程序开发，因此对安全性有较高的要求。如果没有安全保证，用户从网络下载程序执行就非常危险。Java通过自己的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁破坏。当Java字节码进入解释器时，首先必须经过字节码校验器的检查，然后，Java解释器将决定程序中类的内存布局，随后，类装载机负责把来自网络的类装载到单独的内存区域，避免应用程序之间相互干扰破坏。最后，客户端用户还可以限制从网络上装载的类只能访问某些文件系统。上述几种机制结合起来，使得Java成为安全的编程语言。

5多线程

线程是操作系统的一种新概念，它又被称作轻量进程，是比传统进程更小的可并发执行的单位。C和C++采用单线程体系结构，而Java却提供了多线程支持。Java在两方面支持多线程。一方面，Java环境本身就是多线程的。若干个系统线程运行负责必要的无用单元回收，系统维护等系统级操作；另一方面，Java语言内置多线程控制，可以大大简化多线程应用程序开发。Java提供了一个类Thread，由它负责启动运行，终止线程，并可检查线程状态。Java的线程还包括一组同步原语。这些原语负责对线程实

行并发控制。利用Java的多线程编程接口，开发人员可以方便得写出支持多线程的应用程序，提高程序执行效率。必须注意地是，Java的多线程支持在一定程度上受运行时支持平台的限制。例如，如果操作系统本身不支持多线程，Java的多线程特性可能就表现不出来。

二、Java小程序和应用程序用Java可以写两种类型的程序：小程序（又叫JavaApplet）和应用程序（JavaApplication）。小程序是嵌入在HTML文档中的Java程序；而Java应用程序是从命令行运行的程序。对Java而言，Java小程序的大小和复杂性都没有限制。事实上，Java小程序有些方面比Java应用程序更强大。但是由于目前Internet通讯速度有限，因此大多数小程序规模较小。小程序和应用程序之间的技术差别在于运行环境。Java应用程序运行在最简单的环境中，它的唯一外部输入就是命令行参数。另一方面，Java小程序则需要来自Web浏览器的大量信息：它需要知道何时启动，何时放入浏览器窗口，何处，何时激活关闭。由于这两种不同的执行环境，小程序和应用程序的最低要求不同。由于WWW使小程序的发布十分便利，因此小程序更适合作为Internet上的应用程序。相反，非网络系统和内存较小的系统更适合用Java应用程序而较少用Java小程序实现。另外，Java应用程序也很容易以Internet为基础环境，事实上有些优秀的Java应用程序正是如此。

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