# 第1章：初识Java

Java从诞生就注定着不平凡，借助于互联网的浪潮长期屹立在世界开发语言排行榜的榜首，这也造成了大批Java的拥趸，鄙人正是其中之一，Java衍生自C++却又超脱于C++，它看起来简单可想真正做到一个资深的developer，又并非一件易事，本篇作为Java述说的开篇将从计算机基础讲起到写出第一个Java程序HelloWorld为止，通篇重点在于以下几点：

* Java语言的发展简史；
* Java语言的特性；
* Java开发环境的搭建；
* Java跨平台的重点述说

作为Java述说的开篇，我们会先从简单了解计算机的基本理论知识讲起，这对于我们后期的开发具有很大的帮助，下面开始本篇内容。

## 计算机的基础知识

进入21世纪第二个十年，计算机已经成为生活中一个必不可少的工具了，但我们真的了解计算机吗？计算机有哪些部分构成？不同的计算机又可以做什么样的事情呢？我们的PC和用来做加减乘除的计算器都属于计算机范畴吗？下面我们就来来接认识一下。

### 1.1.1 计算机

计算机（Computer）全称：电子计算机，俗称电脑。是一种能够按照程序运行，自动、高速处理海量数据的现代化智能电子设备。由硬件和软件所组成，没有安装任何软件的计算机称为裸机。常见的形式有台式计算机、笔记本计算机、大型计算机等。

计算机的应用已渗透到社会的各个领域，正在改变着人们的工作、学习和生活的方式，推动着社会的发展。主要有一下几个方面：

1：科学计算

科学计算也称数值计算。计算机最开始是为解决科学研究和工程设计中遇到的大量数学问题的数值计算而研制的计算工具。例如，人造卫星轨迹的计算，房屋抗震强度的计算，火箭、宇宙飞船的研究设计都离不开计算机的精确计算。就连我们每天收听收看的天气预报都离不开计算机的科学计算。

2、数据处理

在科学研究和工程技术中，会得到大量的原始数据，其中包括大量图片、文字、声音等信息，数据处理就是对数据进行**收集、分类、排序、存储、计算、传输、制表**等操作。

3、自动控制

自动控制是指通过计算机对某一过程进行自动操作，它不需人工干预，能按人预定的目标和预定的状态进行过程控制。例如，无人驾驶飞机、导弹、人造卫星和宇宙飞船等飞行器的控制，都是靠计算机实现的。

4、计算机辅助设计

计算机辅助设计(Computer Aided Design，简称CAD)是指。借助计算机的帮助，人们可以自动或半自动地完成各类工程设计工作。目前CAD技术已应用于飞机设计、船舶设计、建筑设计、机械设计、大规模集成电路设计等。在京九铁路的勘测设计中，使用计算机辅助设计系统绘制一张图纸仅需几个小时，而过去人工完成同样工作则要一周甚至更长时间。可见采用计算机辅助设计，可缩短设计时间，提高工作效率，节省人力、物力和财力，更重要的是提高了设计质量。

5、人工智能

人工智能(Artificial Intelligence，简称AI)是指计算机模拟人类某些智力行为的理论、技术和应用。例如，用计算机模拟人脑的部分功能进行思维学习、推理、联想和决策，使计算机具有一定“思维能力”。我国已开发成功一些中医专家诊断系统，可以模拟名医给患者诊病开方。 还有就是机器人也是计算机人工智能的典型例子。

6、多媒体应用

随着电子技术特别是通信和计算机技术的发展，人们已经有能力把文本、音频、视频、动画、图形和图像等各种媒体综合起来，构成一种全新的概念—“多媒体”(Multimedia)。比如一些flash广告，网页游戏等。

7、计算机网络

计算机网络是由一些独立的和具备信息交换能力的计算机互联构成，以实现资源共享的系统。如在全国范围内的银行信用卡的使用，火车和飞机票系统的使用等。

### 1.1.2 计算机硬件

计算机硬件（Computer Hardware）是指计算机系统中由电子、机械和光电元件等组成的各种物理装置的总称。这些物理装置按系统结构的要求构成一个有机整体为计算机软件运行提供物理基础。

计算机通常由CPU、主板、内存、电源、主机箱、硬盘、显卡、键盘、鼠标，显示器等多个部件组成，跟多的时候我们讲起归纳为五大部件：控制单元、运算单元、输入设备、输出设备、存储器，其中控制单元和运算单元即算术逻辑运算的单元构成了完整的CPU.

### 1.1.3 计算机软件

计算机软件是使用计算机过程中必不可少的工具，计算机软件可以使计算机按照事先预定好的顺序完成特定的功能，计算机软件按照其功能划分为系统软件与应用软件

* 系统软件： DOS(Disk Operating System), Windows, Linux, Unix, Mac, Android, iOS等；
* 应用软件： Office QQ聊天 YY语言 扫雷 等

系统软件是计算机的灵魂，通过它进行计算机硬件的管理和资源的分配使用，并且系统软件提供接口供应用软件开发和调用操作系统进行硬件的使用，没有系统软件的计算机只是一台通电的普通机器，并不能提供给我们任何帮助。我们也可以认为系统软件是应用软件与计算机硬件的中间件。

### 1.1.4 软件开发

* 软件：按照特定顺序组织的计算机数据（数据结构）和指令（算法）的集合，例如：体重和身高他们属于数据的范畴单独的拿出来没有任何意义，可是如果我们通过身高和体重的一个比例计算就可以得出我们的健康状态，这个计算属于指令；这种数据和指令的集合才有意义；
* 开发：软件的制作过程
* 软件开发：借助开发工具与计算机语言制作软件。

### 1.1.5 计算机语言

* 语言：人类进行沟通交流的各种表达符号，方便人与人之间进行沟通与信息交换；
* 计算机语言：计算机语言（Computer Language）指用于人与计算机之间通讯的语言。**计算机语言是人与计算机之间传递信息的媒介。计算机系统最大特征是指令通过一种语言传达给机器。**为了使电子计算机进行各种工作，就需要有一套用以编写计算机程序的数字、字符和语法规划，由这些字符和[语法规则](http://baike.baidu.com/view/4269308.htm" \t "http://baike.baidu.com/_blank)组成计算机各种指令（或各种语句）。这些就是计算机能接受的语言。计算机语言的种类非常的多，总的来说可以分成**机器语言，[汇编语言](http://baike.baidu.com/view/49.htm" \t "http://baike.baidu.com/_blank)，高级语言**三大类。语言可以有多种多样，但他们的功能却都是一致的那就是提供人与计算机进行信息的传递和交互。

### 1.1.6 计算机语言的发展史

从计算机诞生到今天已经接近一个世纪了，在这一个世纪中计算机软硬件技术飞速发展，而人机交互语言也发生了翻天覆地的变化，从最初的可读性极差的0与1到了今天极具人性化的高级语言，但人机交互底层的东西仍旧是一样的，计算机还是只能通过0与1来理解人们的意思，下面我们通过探讨计算机语言在这近一个世纪所发生的变化来叙述这个问题。

#### 1.1.6.1 机器语言

机器语言是指一台计算机全部的**指令集合**，[电子计算机](http://baike.baidu.com/view/6373.htm" \t "http://baike.baidu.com/_blank)所使用的是由"0"和"1"组成的二进制数，二进制是计算计算机语言的基础。计算机发明之初，人们只能降贵纡尊，用计算机的语言去命令计算机干这干那，一句话，就是写出一串串由"0"和"1"组成的指令序列交由计算机执行，这种**计算机能够认识的语言，就是机器语言**。使用机器语言是十分痛苦的，特别是在程序有错需要修改时，更是如此。因此程序就是一个个的二进制文件。一条机器语言成为一条指令。指令是不可分割的最小功能单元。而且，由于每台计算机的[指令系统](http://baike.baidu.com/view/178189.htm" \t "http://baike.baidu.com/_blank)往往各不相同，所以，在一台计算机上执行的程序，要想在另一台计算机上执行，必须另编程序，造成了重复工作。但由于使用的是针对特定型号计算机的语言，故而运算效率是所有语言中最高的。机器语言，是第一代计算机语言。

#### 1.1.6.2 汇编语言

为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串，比如，用"ADD"代表加法，"MOV"代表数据传递等等，这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种[程序设计语言](http://baike.baidu.com/view/128511.htm" \t "http://baike.baidu.com/_blank)就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种[翻译程序](http://baike.baidu.com/view/922843.htm" \t "http://baike.baidu.com/_blank)被称为汇编程序。汇编语言同样十分依赖于机器硬件，移植性不好，但效率仍十分高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精炼而质量高，所以至今仍是一种常用而强有力的软件开发工具。汇编语言的实质和机器语言是相同的，都是直接对硬件操作，只不过指令采用了英文缩写的[标识符](http://baike.baidu.com/view/390932.htm" \t "http://baike.baidu.com/_blank)，更容易识别和记忆。它同样需要编程者将每一步具体的操作用命令的形式写出来。汇编程序的每一句指令只能对应实际操作过程中的一个很细微的动作，例如移动、自增，因此汇编源程序一般比较冗长、复杂、容易出错，而且使用汇编语言编程需要有更多的计算机专业知识，但汇编语言的优点也是显而易见的，用汇编语言所能完成的操作不是一般高级语言所能实现的，而且源程序经汇编生成的[可执行文件](http://baike.baidu.com/view/159830.htm" \t "http://baike.baidu.com/_blank)不仅比较小，而且执行速度很快。

#### 1.1.6.3 高级语言

高级语言（High-level programming language）相对于机器语言（machine language，是一种[指令](https://baike.baidu.com/item/%E6%8C%87%E4%BB%A4" \t "https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80/_blank)集的体系。这种指令集，称[机器码](https://baike.baidu.com/item/%E6%9C%BA%E5%99%A8%E7%A0%81" \t "https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80/_blank)（machine code），是电脑的CPU可直接解读的数据）而言。是高度封装了的[编程语言](https://baike.baidu.com/item/%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80/9845131" \t "https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80/_blank)，与低级语言相对。它是以人类的日常语言为基础的一种编程语言，使用一般人易于接受的文字来表示（例如汉字、不规则英文或其他外语），从而使程序编写员编写更容易，亦有较高的可读性，以方便对电脑认知较浅的人亦可以大概明白其内容。由于早期电脑业的发展主要在美国，因此一般的高级语言都是以英语为蓝本。在1980年代，当东亚地区开始使用电脑时，在日本、台湾及中国大陆都曾尝试开发用各自地方语言编写的高级语言，当中主要都是改编[BASIC](https://baike.baidu.com/item/BASIC" \t "https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80/_blank)或专用于[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93" \t "https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80/_blank)数据访问的语言，但是随着编程者的外语能力提升，现时的有关开发很少。

由于汇编语言依赖于[硬件](https://baike.baidu.com/item/%E7%A1%AC%E4%BB%B6/479446" \t "https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80/_blank)体系，且助记符量大难记，于是人们又发明了更加易用的所谓高级语言。在这种语言下，其语法和结构更类似汉字或者普通英文，且由于远离对硬件的[直接操作](https://baike.baidu.com/item/%E7%9B%B4%E6%8E%A5%E6%93%8D%E4%BD%9C/6064390" \t "https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80/_blank)，使得一般人经过学习之后都可以编程。高级语言通常按其基本类型、代系、实现方式、应用范围等分类。

常见的高级语言有：BASIC（True basic、Qbasic、Virtual Basic）、C、C++、PASCAL、FORTRAN、智能化语言（LISP、Prolog、[CLIPS](http://baike.baidu.com/view/5554534.htm" \t "http://baike.baidu.com/_blank)、OpenCyc、Fazzy）、动态语言(Python、PHP、Ruby、Lua)等等。- 高级语言源程序可以用解释、编译两种方式执行。通常用后一种。高级语言是绝大多数编程者的选择。和汇编语言相比，它不但将许多相关的机器指令合成为单条指令并且去掉了与具体操作有关但与完成工作无关的细节，例如使用堆栈、寄存器等，这样就大大简化了程序中的指令。由于省略了很多细节，所以编程者也不需要具备太多的专业知识。高级语言主要是相对于汇编语言而言，它并不是特指某一种具体的语言，而是包括了很多编程语言，流行的VB、VC、FoxPro、Delphi等，这些语言的语法、命令格式都各不相同

计算机软件都是用各种计算机语言（后面讲）编写的。最底层的叫机器语言，它由一些0和1组成，可以被某种电脑直接理解，但人就很难理解。上面一层叫汇编语言，它只能由某种电脑的汇编器软件翻译成机器语言程序，才能执行。人能够勉强理解汇编语言。人常用的语言是更上一层的高级语言，比如C，C++，C#， Java等。这些语言编写的程序一般都能在多种电脑上运行，但必须先由一个叫作编译器或者是解释器的软件将高级语言程序翻译成特定的机器语言程序。编写计算机软件的人员叫程序设计员、程序员、编程人员。他们当中的高手有时也自称为黑客。

### 1.1.7 人机交互方式

交互的意思就是你和程序对话.例如你点击一个按钮,然后程序弹出一个窗口"您真的要退出吗?"这样你就可以选择了.这个就叫交互.目前大部分的程序都是交互程序。软件的出现实现了人与计算机之间更好的交互。这种交互方式成为人机交互或者人机互动（英文：Human–Computer Interaction或Human–Machine Interaction，简称HCI或HMI），是一门研究系统与用户之间的交互关系的学问。系统可以是各种各样的机器，也可以是计算机化的系统和软件。人机交互界面通常是指用户可见的部分。用户通过人机交互界面与系统交流，并进行操作。小如收音机的播放按键，大至飞机上的仪表板、或是发电厂的控制室。人机交互界面的设计要包含用户对系统的理解（即[心智模型](http://baike.baidu.com/view/2333986.htm" \t "http://baike.baidu.com/_blank)），那是为了系统的[可用性](http://baike.baidu.com/view/1436.htm" \t "http://baike.baidu.com/_blank)或者用户友好性。人机交互的发展经历了几个阶段：

* 早期的手工作业阶段：作业控制语言及交互命令语言阶段即命令行方式：需要有一个控制台，输入特定的指令，让计算机完成一些操作。较为麻烦，需要记录住一些命令。
* 图形用户界面（GUI）阶段：这种方式简单直观，使用者易于接受，容易上手操作。
* 网络用户界面的出现
* 多通道、多媒体的智能人机交互阶段

人机交互的发展历史, 是从人适应计算机到计算机不断地适应人的发展史.它经历了几个阶段:

1. 早期的手工作业阶段.当时交互的特点是由设计者本人(或本部门同事)来使用计算机, 他们采用手工操作和依赖机器(二进制机器代码)的方法去适应现在看来是十分笨拙的计算机;

(2)作业控制语言及交互命令语言阶段.这一阶段的特点是计算机的主要使用者———程序员可采用批处理作业语言或交互命令语言的方式和计算机打交道, 虽然要记忆许多命令和熟练地敲键盘, 但已可用较方便的手段来调试程序、了解计算机执行情况; 

(3)图形用户界面(GUI)阶段.GUI 的主要特点是桌面隐喻、WIMP技术、直接操纵和“ 所见即所得(WYSIWYG)”.由于GU I 简明易学、减少了敲键盘、实现了“事实上的标准化” , 因而使不懂计算机的普通用户也可以熟练地使用, 开拓了用户人群.它的出现使信息产业得到空前的发展;

(4)网络用户界面的出现.以超文本标记语言HTML 及超文本传输协议HTTP 为主要基础的网络浏览器是网络用户界面的代表.由它形成的WWW 网已经成为当今Internet 的支柱.这类人机交互技术的特点是发展快, 新的技术不断出现,如搜索引擎、网络加速、多媒体动画、聊天工具等;

(5)多通道、多媒体的智能人机交互阶段.以虚拟现实为代表的计算机系统的拟人化和以手持电脑、智能手机为代表的计算机的微型化、随身化、嵌入化,是当前计算机的两个重要的发展趋势.而以鼠标和键盘为代表的GU I 技术是影响它们发展的瓶颈.

### 1.1.8 键盘功能键及快捷键介绍

常用快捷键：

|  |  |
| --- | --- |
| F键 | F1帮助 、F2改名 、F3搜索 、F4地址 、F5[刷新](http://www.hudong.com/wiki/%E5%88%B7%E6%96%B0" \o "刷新" \t "http://w.baike.com/_blank) 、F6切换 、F10菜单 |
| CTRL键 | CTRL+A全选 、CTRL+F4关闭 、CTRL+ESC开始菜单  CTRL+C复制 、CTRL+S 保存  CTRL+X剪切 、CTRL+V粘贴 、CTRL+Z撤消 、CTRL+O打开 、CTRL+F查找 、CTRL+H替换 、 |
| ALT键 | ALT+ENTER属性 、ALT+F4关闭 、ALT+TAB[切换](http://www.hudong.com/wiki/%E5%88%87%E6%8D%A2" \o "切换" \t "http://w.baike.com/_blank) 、ALT+ESC切换 、ALT+空格键 窗口菜单 |
| Windows键 | Windows快捷键  Windows徽标键+L键切换用户  Windows徽标键+TAB键在任务栏上的按钮间循环  Windows徽标键+E键启动“我的电脑”  Windows徽标键+D键快速显示桌面  Windows徽标键+R键执行“运行”命令  Windows徽标键+U键打开“工具管理器”  Windpws徽标键+M键将所有窗口最小化  Windows徽标键+F键搜索文件或者文件夹  Windows徽标键+F1键显示Windows帮助  Windows徽标键+Shift+M键将最小化的窗口还原  Windows徽标键+Ctrl+F键搜索计算机  Windows徽标键+PauseBreak键打开“系统属性”对话框 |
| 其他 | SHIFT+DELETE永久删除 、DELETE删除 、PrtSc(PrintScreen)屏幕截图 |

### 1.1.9 常用的DOS命令

DOS命令，计算机术语，是指DOS操作系统的命令，是一种面向磁盘的操作命令，主要包括目录操作类命令、磁盘操作类命令、文件操作类命令和其它命令。

目前大家常用的操作系统有windows10,windows 7等，都是图形化的界面。在有这些系统之前的人们使用的操作系统是DOS系统。

DOS命令不区分大小写，比如C盘的Program Files，在[dos](https://baike.baidu.com/item/dos" \t "https://baike.baidu.com/item/DOS%E5%91%BD%E4%BB%A4/_blank)命令中完全可以用"progra~1"代替，加上英文引号是因为名称的中间有空格（即多于一个词），这一点是初学者经常忽略的。常用的命令有如下：

Ping命令：它是用来检查网络是否通畅或者网络连接速度的命令。作为一个生活在网络上的管理员或者黑客来说，ping命令是第一个必须掌握的DOS命令，它所利用的原理是这样的：网络上的机器都有唯一确定的IP地址，我们给目标IP地址发送一个数据包，对方就要返回一个同样大小的数据包，根据返回的数据包我们可以确定目标主机的存在，可以初步判断目标主机的操作系统等。下面就来看看它的一些常用的操作。先看看帮助吧，在DOS窗口中键入：ping /? 回车，出现如图1。所示的帮助画面。在此，我们只掌握一些基本的很有用的参数就可以了（下同）。

* -t ：表示将不间断向目标IP发送数据包，直到我们强迫其停止。试想，如果你使用100M的宽带接入，而目标IP是56K的小猫，那么要不了多久，目标IP就因为承受不了这么多的数据而掉线，呵呵，一次攻击就这么简单的实现了。
* -l ：定义发送数据包的大小，默认为32字节，我们利用它可以最大定义到65500字节。结合上面介绍的-t参数一起使用，会有更好的效果哦。
* -n ：定义向目标IP发送数据包的次数，默认为3次。如果网络速度比较慢，3次对我们来说也浪费了不少时间，因为现在我们的目的仅仅是判断目标IP是否存在，那么就定义为一次吧。说明一下，如果-t 参数和-n参数一起使用，ping命令就以放在后面的参数为标准，比如“ping IP -t -n 3”，虽然使用了-t参数，但并不是一直ping下去，而是只ping 3次。另外ping命令不一定非得ping IP，也可以直接ping主机域名，这样就可以得到主机的IP。下面我们举个例子来说明一下具体用法，如图2。这里time=2表示从发出数据包到接受到返回数据包所用的时间是2秒，从这里可以判断网络连接速度的大小。从TTL的返回值可以初步判断被ping主机的操作系统，之所以说“初步判断”是因为这个值是可以修改的。这里TTL=32表示操作系统可能是win98。 （小知识：如果TTL=128，则表示目标主机可能是Win2000；如果TTL=250，则目标主机可能是Unix）至于利用ping命令可以快速查找局域网故障，可以快速搜索最快的QQ服务器，可以对别人进行ping攻击……这些就靠自己发挥了。



打开DOS控制台的方式：

1：开始--程序--附件--命令提示符

2：开始--运行—cmd—回车

3：win+r—cmd--回车

其它常用DOS命令

d: 回车 盘符切换

dir(directory):列出当前目录下的文件以及文件夹

md (make directory) : 创建目录

rd (remove directory): 删除目录

cd (change directory)改变指定目录(进入指定目录)

cd.. : 退回到上一级目录

cd\: 退回到根目录

del (delete): 删除文件,删除一堆后缀名一样的文件\*.txt

exit : 退出dos命令行

cls : (clear screen)清屏

Systeminfo:查询计算机详细信息

Ipconfig:查询IP相关内容

## 1.2 Java简介

1996年Java的第一次发布就引起了人民的极大兴趣。有人认为：Java有望成为一种最优秀的程序设计语言，事实也证明了如此，多年间Java始终位于编程语言的榜首位置。但一旦一种语言应用于某个领域，与现存代码的相容性问题就摆在了人们面前。也正是由于相容性这个严峻的问题确实存在于现实中，所以，或多或少的还是有一些“累赘”被加到语言中，这也就导致了Java并不如想象中的那么完美无暇，但这并不足以动摇Java的“铁王座”，Java仍是一种优秀的语言，尤其表现在Web编程方面。

Java是一个完整的平台，有一个庞大的库，其中包含了很多可重用的代码和一个提供诸如安全性、跨操作系统的可移植性以及自动垃圾回收等服务的执行环境（JVM）。

作为一名程序设计语言，常常希望能够有一种语言，它具有令人赏心悦目的语法和易于理解的语义（C++并不是这样的）。与许多其他的优秀语言一样，Java完全满足了这些要求。有些语言提供了可移植性、垃圾收集等。但是，没有提供一个大型的库。如果想要有奇特的绘图功能、网络连接功能和数据库存取功能就必须自己动手编写代码。Java具备这些特性，它**是一种功能齐全的出色语言，是一个高质量的执行环境，还提供了一个庞大的库。**正是因为它集多种优势于一身，所以对广大的程序设计人员有着不可抗拒的吸引力。

作为一种高级的面向对象的程序设计语言。使用Java语言编写的程序是跨平台的，从PC机到手持电脑都有Java开发的程序和游戏，Java程序可以在任何计算机、操作系统和支持Java的硬件设备上运行。

### 1.2.1 Java语言的发展简史

Java的历史要追溯到1991年，由Patrick Naughton和James Gosling（一个很全能的计算机奇才）带领的Sun公司的工程师小组想要设计一种小型的计算机语言，主要用于像有线电视转换盒这类的消费设备。由于这些消费设备的处理能力和内存都很有限，所以语言必须非常小且能够生成非常紧凑的代码。另外，由于不同的厂商会选择不同的中央处理器（CPU），因此这种语言的关键是不与任何特定的体系结构绑定在一起。这个项目被命名为“Green”。

**代码短小、紧凑且与平台无关，这些要求存促使开发团队设计一个可移植的语言，可以为虚拟机生成中间代码。**

不过，Sun公司的人都有Unix（想想吧，90年代的开发人员，又有谁不了解Unix呢？高级语言的出现帮助了我们的开发，但也使开发人员的基础技能变得十分羸弱）的应用背景。因此，所开的语言以C++为基础，而不是Lisp、Smalltalk或Pascal。不过，就像Gosling在专访中谈到的：“毕竟，语言只是实现目标的工具，而不是目标本身”。Gosling把这种语言称为“Oak”（原因大概是Games极其喜欢他办公室的那棵橡树吧）。Sun公司的人后来发现Oak是一种已有的计算机语言的名字，于是，将其改名为Java（印度尼西亚有一个盛产咖啡的海岛（中文名叫爪哇）Java Island）。事实证明这是一个很有灵感的选择。

1992年，Green项目发布了他的第一个产品，称为“\*7”。这个产品具有非常智能的远程控制。遗憾的是，Sun公司对生产这个产品并不感兴趣，Green项目组的人员必须找出其他方法来将他们的技术推向市场。然而，没有一个标准的消费品电子公司对此感兴趣。于是，Green项目组竞标了一个提供视频点播等新型服务的有线电视盒的项目，但没有成功（有趣的是，得到这个项目的公司的领导恰恰是开创Netspace公司的Jim Clark。Netspace公司后来对Java的成功给予了很大的帮助）。

Green项目（这时换了一个新名字---“First Person公司”）花费了1993年一整年以及1994年的上半年，一直在苦苦寻求其技术的买家。然而，一个也没有找到（Patrick Naughton，项目组的创立人之一，也是完成这个项目大多数市场工作的人，声称为了销售这项技术，累计飞行了300000英里）。1994年First Person公司解散了。

当这一切在Sun公司发生的时候，Internet的万维网也在日益发展壮大。万维网的关键是把超文本页面转换到屏幕上的浏览器。1994年大多数人都在使用Mosaic，这是一个1993年出自伊利诺斯大学超级计算机中心的非商业化的web浏览器（Mosaic的一部分是由MarcAndreessen编写的。当时，他作为一名参加半工半读项目的本科生，编写了这个软件，每小时的薪水只有6.85美元（实际上对于当时的中国人来说真的是高收入了）。他后来成了Netspace公司的创始人之一和技术总监，可谓名利双收）。

在接受Sun World采访的时候，Gosling说在1994年中期，Java语言的开发者意识到：“我们能够建立一个相当酷的浏览器。我们已经拥有客户机/服务器主流模型中所需要的体系结构中立、实时、可靠、安全---这些在工作站环境并不太重要，所以，我们决定开发浏览器。”

实际的浏览器是由Patrick Naughton和Jonathan Payne开发的，并演变为HotJava浏览器。为了炫耀Java语言超强的能力，HotJava浏览器采用Java编写。设计者让HotJava浏览器具有在网页中执行内嵌代码的能力。这一“技术印证”在1995年5月23日的SunWorld上得到展示，同时引发了人们延续至今的对Java的疯狂追逐。

1996年年初，Sun发布了Java的第一个版本。人们很快意识到Java1.0不能用来进行真正的应用开发。的确，可以使用Java1.0来实现在画布上随机跳动的神经质的文本Applet，但它却没有提供打印功能。坦率的说，Java1.0的确没有为其黄金时期的到来做好准备。后来的Java1.1弥补了其中的大多明显的缺陷，大大改进了反射能力，并为GUI编程增加了新的事件处理模型。不过它仍然具有很大的局限性。

1998年JavaOne会议的头号新闻是即将发布Java1.2版。这个版本取代了早期玩具式的GUI，并且它的图形工具箱更加精细而具有可伸缩性，更加接近“一次编写，到处运行”的承诺。在1998年12月Java1.2发布三天之后，Sun公司市场部将其名称改为更加吸引人的“**Java2标准版软件开发工具箱1.2版**”。

除了“标准版”之外，Sun还推出了两个其他的版本：一个是用于手机等嵌入式设备的“微型版”；另一个是用于服务器端处理的“企业版”。我们主要学习的也是标准版和企业版，而学习企业版则需要首先对标准版有深入的了解。

标准版的1.3和1.4版本对最初的Java2版本做出了某些改进，扩展了标准类库，提高了系统性能。当然，还修改了一些bug。在此期间，Java applet采取了低调姿态，并淡化了客户端的应用，但Java却成为服务器端应用的首选平台。

5.0版是自1.1版以来第一个对Java语言做出重大改进的版本（这一版本原来命名为1.5版，在2004年的JavaOne会议之后，版本数字升为5.0）。在经历多年研究后，这个版本添加了泛型类型（Generic Type）（类似于C++的模板），其挑战性在于添加这一特性并没有对虚拟机做出任何修改。另外，还有几个受C#启发的很有用的语言特性：“for each”循环、自动拆装箱和注解。

版本6（没有后缀0）于2006年年末发布。同样，这个版本没有对语言方面再进行改进。但是，改进了其他性能，并增强了类库。

随着数据中心越来越依赖于商业硬件而不是专用服务器，Sun Microsystems终于沦陷了，于2009年被Oracle收购。Java的开发停滞了很长一段时间。直到2011年Oracle发布了一个新版本，Java7，其中只做了一些简单改进。

2014年，Java8终于发布，在近20年中这个版本有了最大改变。Java8提供了一种“函数式”编程方式，可以很容易的表述并发执行的计算。所有编程语言都需要与时俱进，Java在这方面显示出非凡的能力。下图展示了Java语言以及类库的发展状况。可以看出，应用程序编程接口API的规模发生了惊人的变化。

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **年份** | **语言新特性** | **类与接口的数量** |
| 1.0 | 1996 | 语言本身 | 211 |
| 1.1 | 1997 | 内部类 | 477 |
| 1.2 | 1998 | Strictfp修饰符 | 1524 |
| 1.3 | 2000 | 无 | 1840 |
| 1.4 | 2002 | 断言 | 2723 |
| 5.0 | 2004 | 泛型类、加强for循环、可变参数、自动拆装箱、元数据、枚举、静态导入、注解 | 3279 |
| 6 | 2006 | 无 | 3793 |
| 7 | 2011 | 基于字符串的switch、钻石操作符、二进制字面量、异常处理改进 | 4024 |
| 8 | 2014 | Lambda表达式，包含默认方法的接口、流和日期/时间库 | 4240 |

### 1.2.2 Java语言与常见的误解与概念名词

Java是一种通过解释方式来执行的语言，其语法规则和C++类似。Java非常适于企业网络和Internet环境，并且已成为Internet中最具影响力、最受欢迎的编程语言之一。与目前常用的C++相比，Java语言简洁的多，而且提高了可靠性，除去了最大的程序错误根源，此外他还有较高的安全性，可以说它是有史以来最为卓越的编程语言。

Java语言编写的程序既是编译型的，又是解释型的（通常我们说其是解释型语言）。程序代码经过编译之后转换为一种称为Java字节码的中间语言，Java虚拟机（JVM）将对字节码进行解释和运行。编译只进行一次，而解释在每次运行程序时都会进行。编译后的字节码采用一种针对JVM优化过的机器码形式保存，虚拟机将字节码解释为机器码，然后在计算机上面运行。Java语言程序代码的编译和运行过程如图1.1所示。

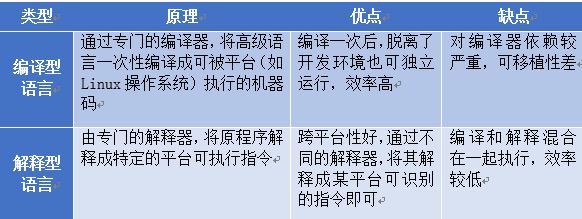
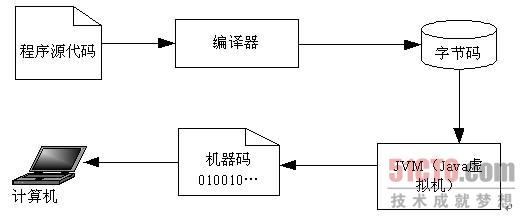


图1.1 Java程序的编译和运行过程

#### 1.2.2.1 关于Java的一些误解与概念

关于Java我们在研究它的过程中难免会有一些模糊的地方，本节我们挑选几个进行详细说明。

1. Java是HTML的扩展：Java是一种程序设计语言；HTML是一种描述网页结构的方式。除了用于在网页上放置Java applet的HTML扩展之外，两者没有任何共同之处。
2. 使用XML，所以不需要Java：Java 是一种程序设计语言；XML是一种描述数据的方式。可以使用任何一种程序设计语言除了XML数据，而Java API对XML除了提供了很好地支持。此外，许多重要的第三方XML工具采用Java编写，关于XML我们在后面再单独叙述。
3. Java是一种非常容易学习的程序设计语言：像Java这种功能强大的语言大都不太容易学习。首先，必须将编写玩具式程序的轻松和开发实际项目的艰难区分开。如同上面看到的Java8有4000多个类与接口，但幸运的是，我们并不需要知道它们中的每一个，然而，要想Java解决实际问题，还是需要了解不少内容的。
4. Java将成为适用于所有平台的通用性编程语言：从理论上讲，这是完全可能的。但在实际中，在某些领域其他语言有更出色的表现，比如Objecttive C和后来的Swift在iOS设备上就有着无可替代的地位。浏览器中的处理几乎完全由JavaScript掌控。Windows程序通常都用C++或C#编写。Java在服务器端编程和跨平台客户端应用领域则很有优势。
5. Java只不过是另外一种程序设计语言：Java是一种很好的程序设计语言，很多程序设计人员喜欢Java胜过了C、C++或C#。有上百种好的程序设计语言没有广泛的流行，而带有明显缺陷的语言，如：C++和Visual Basic却大行其道。这是为什么呢？

程序设计语言的成功更多的取决于其支撑系统的能力，而不是优美的语法。人们主要关注：是否提供了易于实现某些功能的易用、便捷和标准的库？是否有开发工具提供商能建立强大的编程和调试环境？语言和工具集是否能够与其他计算基础架构整合在一起？Java的成功源于其类库能够让人们轻松地完成原本有一定难度的事情。例如：联网web应用和并发。Java减少了指针错误，这是一个额外的好处，因此使用Java编程的效率更高。但这些并不是Java成功的全部原因。其所提供的平台系统才是它成功的关键。

1. Java是专用的，应该避免使用：最初创建Java时，Sun为销售者和最终用户提供了免费许可。尽管Sun对Java拥有最终的控制权，不过在语言版本的不断发展和辛苦的设计过程中还涉及很多其他公司。虚拟机和类库的源代码可以免费获得，不过仅限于查看，而不能修改和再发布。Java是“闭源的，不过可以很好的使用”。

这种状况在2007年发生了戏剧性的变化，Sun声称Java未来的版本将在General Public Licence（GPL 通用公共许可证）下提供。Linux使用的是同一个开放源代码许可。Oracle一直致力于保持Java开源。只是一点美中不足-----专利。根据GPL，任何人都可以得到专利许可，允许其修改者修改Java，不过仅限于桌面和服务器平台。如果你想要在嵌入式系统中使用Java，就需要另外一个不同的许可，这很可能需要付费。不过，这些专利在未来十年就会到期，那时Java就会完全免费了。

1. Java是解释型的，因此对于关键的应用程序速度太慢了：早期的Java是解释型的。现在Java虚拟机使用了即时编译器，因此采用Java编写的“热点”代码其运行速度与C++相差无几，有些情况下甚至更快。

对于Java桌面应用速度慢，人们已经抱怨很多年了。但是，今天的计算机速度远比人们发出抱怨的时候快了很多。一个较慢的Java程序与几年前相当快的C++程序相比还要更快一些。

1. 所有的Java程序都是在网页中运行的：所有的Java applet都是在网页浏览器中运行的。这也恰恰是applet的定义，记忆中在浏览器中运行的Java程序。然而，大多数Java程序是运行在web浏览器之外的独立应用程序。实际上，很多Java程序都在web服务器上运行并生成用于网页的代码。
2. Java程序是主要的安全风险：对于早期的Java，有过关于安全系统失效的报道，曾经一度引起公众哗然。研究人员将这视为一种挑战，即努力找出Java的漏洞，对applet安全模型的强度和复杂度发起挑战。随后，人们很快就解决了引发问题的所有技术因素。后来又发现了更严重的漏洞，而Sun以及后来的Oracle反应却过于迟缓。浏览器制造商则有些反应过度，他们甚至默认禁用了Java。客观地来讲，可以想想针对Windows可执行文件和Word宏有数百万种病毒攻击，并造成了巨大的伤害，不过奇怪的是却很少有人批评被攻击平台的脆弱。

有些系统管理员甚至在公司浏览器中禁用了Java，而同时却允许用户下载可执行文件和Word文档，实际上，这些带来的风险远甚于使用Java。尽管距离Java诞生已经20年之久，与其他常用的平台相比，Java还是安全得多。

1. JavaScript是Java的简易版：JavaScript是一种在网页中使用的脚本语言，它是由Netspace发明的，原来的名字叫做LiveScript。JavaScript的语法类似Java，除此之外，两者无任何关系。当然，名字有些相像。JavaScript的一个子集已经标准化为ECMA-262.与Java applet相比，JavaScript更紧密地与浏览器集成在一起。特别是JavaScript程序可以修改正在显示的文档，而applet只能在有限的区域内控制外观。
2. 使用Java可以廉价的Internet设备取代桌面计算机：当Java刚刚发布的时候，一些人打赌：肯定会有这样的好事发生。一些公司已经生产出Java网络计算机的原型，不过用户还不打算放弃功能强大而便利的桌面计算机，而去使用没有本地存储而且功能有限的网络设备。当然，如今世界已经发生改变，对于大多数最终用户，常用的平台往往是手机或平板电脑。这些涉笔大多使用安卓平台，这是Java的衍生产物。学习Java编程会对Android编程有很大的帮助。
3. 什么是强类型语言，Java属于什么类型：如果按照强弱来定义类型的话，那么语言类型可以分为三种“强类型“、“弱类型”、”无类型”；其中**无类型不检查甚至不区分指令和数据；弱类型的检查很弱，仅能够严格的区分指令和数据；强类型则严格的在编译期进行检查**。强类型语言在设有强制类型转换前不允许两种不同类型的变量相互操作。弱类型的语言的数据没有明显的类型，它能随着环境的不同自动变换类型；而强类型则没有这样的规定，不同类型间的操作有严格定义，只有相同类型的才能够进行操作。

它们的异同点在于：（1）不管是强类型还是弱类型，变量都有两个属性；类型和值，即弱类型的语言变量同样有类型。不管是哪一种编程语言，其中使用的变量都既有类型，又有值。强类型的变量类型，必须在源代码中明确定义称为“变量声明”，弱类型的变量类型则是无需声明的，由解释器解释。但是，这并不意味着弱类型的变量就没有类型这一概念了，举例来说，PHP的gettype就是返回该类型的“当前”的类型。（2）**强类型的变量类型是不能改变的，弱类型变量是随需改变的，这是强弱的真正含义。强类型的变量一经声明，就只能存储这种类型的值，其他的值必须通过转换之后才能赋给该变量，有编译器自动理解的转换，也有由程序员明确指定的轻质转换，弱类型的变量类型则是随着需要不断转换。**（3）强制转换并没有改变变量类型。强类型语言有个“强制转换”，这改变的是变量的值的类型，以便进行复制，而没有改变变量的类型。变量的类型是不允许改变的。

**强类型语言的优点在于：编译时刻能检查错误的类型匹配，以提高程序的安全性；可以根据对象类型优化相应运算，以提高目标代码的质量。减少运行时刻的开销。**

**从上面的叙述可以很容易知道Java是一种强类型语言。**

1. 即时编译器可以实现代码的内联，那么什么是内联呢？在计算机科学中，**内联函数（有时称为在线函数或编译时期展开函数）是一种编程语言结构，用来建议编译器对一些特殊函数进行内联扩展（有时称作在线扩展）；也就是说建议编译器将指定的函数体插入并取代每一处调用该函数的地方（上下文），从而节省了每次调用函数带来的额外时间开销。**但在选择使用内联函数时，必须在程序占用空间和程序执行效率之间进行权衡，因为过多的比较复杂的函数进行内联扩展将带来很大的存储资源开支。另外还需要特别注意的是对递归函数的内联扩展可能引起部分编译器的无穷编译。
2. 动态类型语言与静态类型语言：**动态类型语言就是类型的检查是在运行时做的，而静态类型语言的类型判断是在运行前判断（如编译阶段）**，比如C#，Java就是静态类型语言，静态类型语言为了达到多态回采用一些类型鉴别手段，如集成、接口，而动态类型语言却不需要，所以一般动态类型语言都会采用dynamic typing，常出现于脚本语言中，需要明确说明一点，那就是，是不是动态类型语言与这门语言是不是类型安全得完全不相干，不要将它们联系在一起。

优缺点：静态类型语言的主要优点在于其结构非常规范，便于调试，方便类型安全；缺点是为此需要写更多的类型相关的代码，导致不便于阅读，不清晰明了。动态类型语言的优点在于方便阅读，不需要写非常多的类型相关的代码，缺点自然就是你不方便阅读，命名不规范时会造成读不懂，不利于阅读等。顺便说一下，现在有这样一种趋势，那就是合并动态类型与静态类型在一种语言中，这样可以在必要的时候取长补短，Boo是就是这样一个例子。

1. 类型安全代码指访问被授权可以访问的内存位置。例如，类型安全代码不能从其他对象的私有字段读取值。它只从定义完善的允许方式访问类型才能读取。类型安全的代码具备良好定义的数据类型。
2. Java与C#：Java成功的推出后不久，微软就发布了一个叫做J++的产品，它与Java有几乎相同的编程语言以及虚拟机。现在，微软不再支持J++，取而代之的是另一种名为C#的语言。C#与Java有很多相似之处，然而使用的确实完全不同的虚拟机。
3. 什么是java的平台无关性？答：Java源文件被编译成字节码的形式，无论在什么系统环境下，只要有java虚拟机就能运行这个字节码文件。也就是一处编写，处处运行。这就是java的跨平台性。需要注意的是编译后的中间码可以跨平台，但是JVM却不是跨平台的。特定平台有自己对应的JVM。

### 1.2.3 Java的应用领域

借助Java，程序开发人员可以自由地使用现有的硬件和软件系统平台。这是因为Java是独立于平台的，它还可以应用于计算机之外的领域，Java程序可以在便携式计算机、电视、电话、手机和其他的大量电子设备上运行。Java的用途不胜枚举，他拥有无可比拟的能力，其节省的时间和费用也十分可观。Java的应用领域主要有以下方面：

* 桌面应用系统开发。
* 嵌入式系统开发。
* 电子商务应用。
* 企业级应用开发。
* 交互式系统开发。
* 多媒体系统开发。
* 分布式系统开发。
* Web应用系统开发。

Java无处不在，它已经拥有几百万个用户，其发展速度要快于在它之前的任何一种计算机语言。Java能够给企业和最终用户带来数不尽的好处。Oracle公司董事长和首席执行官Larru Ellison说过:“Java正在进入企业、家庭、和学校。它正在像Internet本身一样成为普遍存在的技术。”

如果仔细观察，就会发现Java就在我们身边，如我们经常使用的Java开发工具Eclipse、NetBeans、JBuilder等,另外还有RSSOwl、Limewire、Azureus、CyberDuck、OpenOffice等优秀软件都是使用Java编写的。此外，各手机厂商都为自己的产品提供了Java技术的支持，各种手机上的Java程序和游戏已经数不胜数。

### 1.2.4 Java的版本

自从Sun公司推出Java以来，就力图使之无所不能。Java发展至今，按应用范围分为三个版本，即Java SE、Java EE、Java ME，也就是Java ONE（Open Net Environment）体系。本节将分别介绍这3个Java版本。

1. Java SE（Standard Edition）

Java SE是Java的标准版，主要用于桌面应用程序的开发，同时也是Java的基础，它包含Java语言基础、JDBC（Java数据库连接性操作）、I/O（输入输出）、网络通信、多线程等技术。Java SE的结构如图1.2所示。

1. Java EE（Enterprise Edition）

Java EE是Java的企业版，主要用于开发企业级分布式的网络程序，如电子商务网站和ERP（企业资源规划）系统，其核心为EJB（企业Java组件模型）。Java EE的结构如图1.3所示。

1. Java ME（Micro Edition）

Java ME主要应用于嵌入式系统开发，如掌上电脑、手机等移动通信电子设备，现在大部分手机厂商所生产的手机都支持Java技术。Java ME的结构图如图1.4所示。

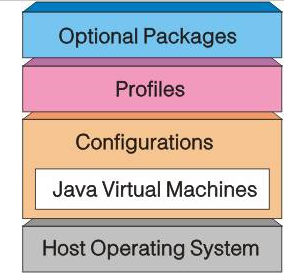
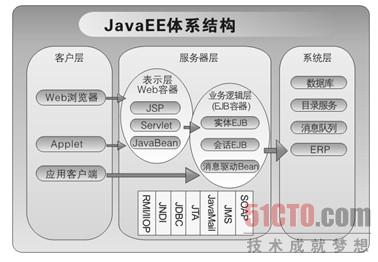
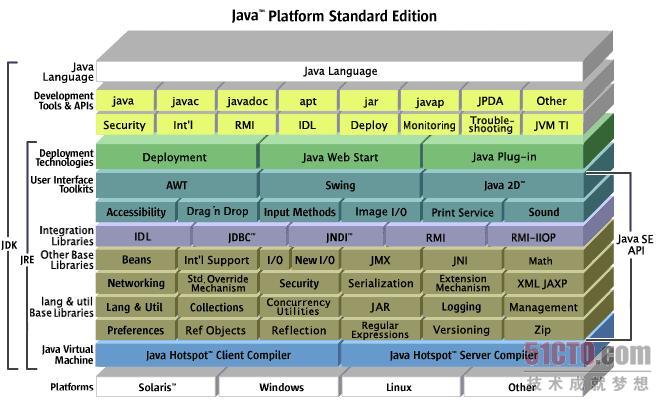


图1.2 Java SE的结构 图1.3 Java EE的结构图 1.4 Java ME的结构

Java SE、Java EE、Java ME又叫J2SE、J2EE、J2ME，区别在于[J2EE](https://www.baidu.com/s?wd=J2EE&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuWndnHnzujIBPjNBPjT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHTzrj63n10d" \t "http://zhidao.baidu.com/_blank)是在java1.2是沿用下来的名字，从Java1.5开始更名为JavaEE 5.0。相应的J2SE-->JavaSE、[J2ME](https://www.baidu.com/s?wd=J2ME&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuWndnHnzujIBPjNBPjT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHTzrj63n10d" \t "http://zhidao.baidu.com/_blank)-->JavaME现在是JavaEE 6.0只是名字的变换，并没有什么大的改变。要说有改变，那也就是版本的升级。

### 1.2.5 怎么学好Java

如何学好Java语言，是所有初学者需要共同面对的问题，其实每种语言的学习方法都大同小异，初学者需要注意的主要有以下几点：

* 明确自己的学习目标和大的方向，选择并锁定一门语言，按照自己的学习方向努力学习，认真研究。
* **初学者不要看太多的书，先找本相对基础的书系统的学习。**很多程序开发人员工作了很久也只是熟悉部分基础而已，并没有系统地学习Java语言。
* **了解设计模**式。开发程序必须编写程序代码，这些代码必须具有高度的可读性，这样编写的程序才能调试、维护和升级的价值。学习一些设计模式，能够更好的把握项目的整体结构。
* **不要死记语法。**在刚接触一门语言，特别是Java语言时，掌握好基本语法，并大概了解一些功能即可。借助开发工具（如Eclipse或NetBeans）的代码辅助功能，完成代码的录入，这样可以快速进入学习状态。
* **多实践，多思考，多请教。**仅读懂书本中的内容和技术是不行的，必须动手编写程序代码，并运行程序、分析运行结构，从而对学习内容有个整体的认识和肯定。用自己的方式思考问题、编写代码来提高编程思想。平时多请教老师或经理，和其他人多沟通技术问题，提高自己的技术和见识。
* 不要急躁。**遇到技术问题，必须冷静对待，不要让自己思维混乱，保持清醒的头脑才能分析和解决各种问题**。可以尝试用歌和散步等方式来放松自己。
* **遇到问题，首先尝试自己解决，这样可以提高自己的程序调试能力，并对常见问题有一定的了解，明白出错的原因，甚至举一反三，解决其他关联的错误问题。**
* **多查阅资料。**可以经常到Internet上搜索相关资料或解决问题的方法，网络上已经摘录了很多人遇到的问题和不同的解决方法，分析这些解决问题的方法，找到最适合自己的方法。
* **多阅读别人的源代码。**不但要看懂别人的程序代码，还要分析编程者的编程思想和设计模式，并融为己用。

### 1.2.6 Java API文档

API的全称是 Application Programming Interface，即应用程序编程接口。Java API文档是Java程序开发不可缺少的编程词典，并记录了Java语言中海量的API，主要包括类的继承结构、成员变量和成员方法、构造方法、静态成员的详细说明和描述信息。可以在Java网站http://java.sun.com中找到最新版本的JDK文档（即API文档），Java API文档的页面效果如图1.5所示。API是我们熟悉编程最重要的一个工具，且对于各种语言的学习API都是很重要的。同时API也是构成Java平台的重要元素之一（Java编程语言也即语法、文件格式、虚拟机、API）

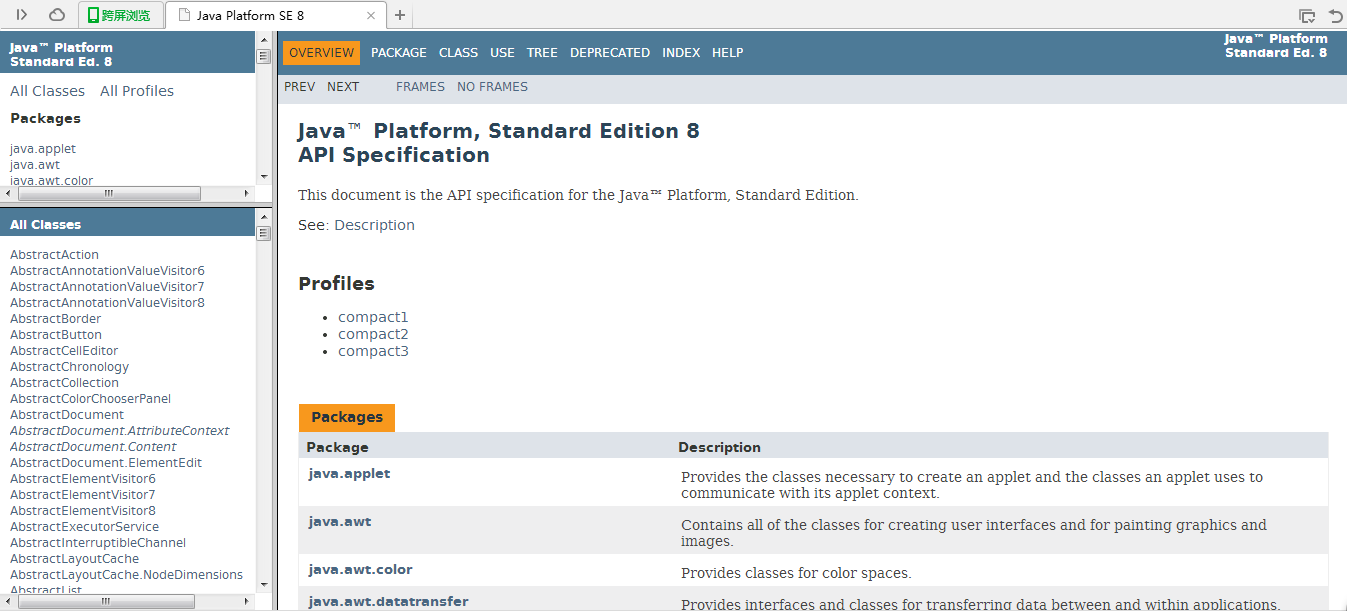


图1.5 Java API文档页面

## 1.3 Java语言的特性

Java语言的作者们编写了具有广泛影响的Java白皮书，里面详尽地介绍了他们的设计目标以及实现成果，还用简短的篇幅介绍了Java语言的特性。下面将对这些特性进行扼要的介绍。

### 1.3.1 简单

Java语言的语法简单明了，容易掌握，并且是纯面向对象的语言。Java语言的简单性主要体现在以下几个方面：

* 语法规则和C++类似。从某种意义上讲，Java语言是由C++语言演变而来的，所以C++程序设计人员可以很容易地掌握Java语言的语法。
* Java语言对C++进行了简化和提高。例如，Java使用接口取代了多重继承，并取消了指针，因为指针和多重继承通常使程序变得复杂。Java语言还通过实现垃圾自动收集，大大简化了程序设计人员的资源释放管理工作。
* Java提供了丰富的类库和API文档以及第三方开发包，另外还有大量的基于Java的开源项目，JDK（Java开发者工具箱）已经开放源代码，读者可以通过分析项目的源代码，从而提高自己的编程水平。

### 1.3.2 面向对象

面向对象是Java语言的基础，也是Java语言的重要特性，它本身就是一种纯面向对象的程序设计语言。Java提倡万物皆对象，语法中不能在类外面定义单独的数据和函数，也就是说，Java语言最“外部”的数据类型是对象，所有的元素都要通过类和对象来访问。

简单来讲，面向对象设计是一种程序设计技术。它将重点放在数据（即对象）和对象的接口上。用木匠打一个比方，一个“面向对象的”木匠始终关注的是所制作的椅子，第二位才是所使用的工具；一个“非面向对象的”木匠首先考虑的是所有的工具。在本质上，Java的面向对象能力与C++是一样的。与C++主要不同点在于多重继承，在Java中，取而代之的是更简单的接口概念。与C++相比，Java提供了更丰富的运行时自省功能（关于自省将会在后面详细介绍）。

面向对象是相对于面向过程而言的，面向过程着重于功能的实现过程，面向对象则是将功能实现封装进一个类，通过类的一个实例对象调用方法实现功能，侧重点在于能够实现功能的对象。

### 1.3.3 分布性

Java的分布性包括操作分布和数据分布，其中操作分布是指在多个不同的主机上布置相关操作，而数据分布是将数据分别存放在多个不同的主机上，这些主机是网络中的不同成员。Java可以凭借URL（统一资源定位符）对象访问网络对象，访问方式与访问本地系统相同。

### 1.3.4 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。Java编译器通过生成与特定的计算机体系结构无关的字节码指令来实现这一特性。精心设计的字节码不仅可以很容易地在任何机器上解释执行，而且还可以动态的翻译成本地机器代码。

当时，为“虚拟机”生成代码并不是一个新思路。诸如Lisp、Smalltalk和Pascal等编程语言多年前就已经采用了这种技术。当然，解释虚拟机指令肯定会比全速运行机器指令慢很多。然而，虚拟机有一个选项，可以将执行最频繁的字节码序列翻译成机器码，这一过程被称为即时编译。

Java虚拟机还有一些其他的优点。它可以检测指令序列的行为，从而增强其安全性。

### 1.3.5 可移植性

Java程序具有与体系结构无关的特性，可以方便的移植到网络上的不同计算机中。同时，Java的类库中也实现了针对不同平台的接口，使这些类库可以移植。可移植与Java最初的设计目的（机顶盒）有很大的关系，尽管后来Java是在互联网领域大放异彩。

与C和C++不同，Java规范中没有“**依赖具体实现**”（其实就是指Java提倡的面向接口编程）的地方。基本数据类型的大小以及有关运算都作了明确的说明。例如，Java中的int永远为32位的整数，而在C/C++中，int可能是16位整数、32位整数，也可能是编译器供应商指定的其他大小。唯一的限制只是int类型的大小不能低于short int，并且不能高于long int。**在Java中，数据类型具有固定的大小，这消除了代码移植时令人头疼的主要问题。二进制数据以固定的格式进行存储和传输，消除了字节顺序的困扰。字符串是用标准的Unicode格式存储的。**

作为系统组成部分的类库，定义了可移植的接口。例如，有一个抽象的Window类，并给出了在UNIX、Windows和Macintosh环境下的不同实现。选择Window类作为例子可能并不太合适。凡是尝试过的人都知道，要编写一个在UNIX、Windows和Macintosh等十多种不同风格的UNIX上看起来都不错的程序有多么的困难。Java1.0就尝试着做了这么一个壮举，发布了一个将常用的用户界面元素映射到不同平台上的简单工具包。遗憾的是，花费了大量的心血，却构建了一个在各个平台上都难以让人接受的库。原先的用户界面工具包已经重写，而且后来又再次重写，不过跨平台的可移植性仍然是个问题。

不过，除了与用户界面有关的部分外，所有其他Java库都能很好地支持平台独立性。你可以处理文件、正则表达式、XML、日期和时间、数据库、网络连接、线程等，而不用操心底层操作系统。不仅程序是可移植的，Java API往往也比原生API（即Native API）质量更高。

### 1.3.6 解释型

运行Java程序需要解释器。任何移植了Java解释器的计算机或其他设备都可以对Java字节码进行解释执行。字节码独立于平台，它本身携带了许多编译时的信息，使得链接过程更加简单，开发过程更加迅速，更具探索性。

### 1.3.7 安全性

Java语言删除了类似C语言中的指针和内存释放等语法，有效的避免了非法操作内存。Java程序代码要经过代码校验、指针校验等很多测试步骤才能够运行，所以未经过允许的Java程序不可能出现损害系统平台的行为，而且使用Java可以编写防病毒和防修改的系统。

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攻击，其中包括：

* 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。

原先，Java对下载代码的态度是“尽管来吧！”。不可信代码在一个沙箱环境中执行，在这里它不会影响主系统。用户可以确信不会发生不好的事情，因为Java代码不管来自哪里，都不能脱离沙箱。

不过Java的安全模型很复杂。Java开发工具包（JDK Java Development Kit）的第一版发布之后不久，普林斯顿大学的一些安全专家就发现一些小bug会允许不可信的代码攻击主系统。

最初，安全bug可以快速修复。遗憾的是，经过一段时间之后，黑客已经很擅长找出安全体系结构视线中的小漏洞。Sun以及之后的Oracle为修复bug度过了一段很是艰难的日子。

遭遇多次高调攻击之后，浏览器开发商和Oracle都越来越谨慎。Java浏览器插件不再信任任何远程代码，除非代码有数字签名而且用户同意执行这个代码。

现在看来，尽管Java安全模型没有原先设定的那么成功，但Java在那个时代确实相当超前。微软提供了一种与之竞争的代码传输机制，其安全性完全依赖于数字签名。显然这是不够的，因为微软自身产品的任何用户都可以证实，知名开发商的程序确实会崩溃并对系统产生危害。

### 1.3.8 健壮性

Java程序的设计目标之一，是编写多方面的、可靠的应用程序，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。

Java编译器能够检测许多在其他语言中仅在运行时才能够检测出来的问题。至于第二点，对于曾经话费几个小时来检查由于指针bug而引起的冲突的人来说，一定会很喜欢Java这一点的。

### 1.3.9 多线程

多线程可以带来更好地交互响应和实时行为。

如今，我们非常关注并发性，因为摩尔定律行将完结。我们不再追求更快的处理器，而是着眼于获得更多的处理器，而且要让他们一直保持工作。不过，可以看到，大多数 编程语言对于这个问题并没有显示出足够的重视。

Java在当时很超前。它是第一个支持并发程序设计的主流语言。从白皮书中可以看到，他的出发点稍有些不同。当时 ，多核处理器还很神秘，而web编程才刚刚起步，处理器要花很长时间等待服务器响应，需要并发程序设计来确保用户界面不会“冻住”。

并发程序设计绝非易事，不过Java在这方面表现很出色，可以很好地管理这个工作。

摩尔定律：该定律是由Intel的创始人之一戈登.摩尔提出来的。其内容为：当价格不变时，集成电路上可容纳的元器件的数目，约每隔18-24个月便会增加一倍，性能也将提升一倍，换言之，每一美元所能买到的电脑性能将每隔18-24个月翻一倍以上，这一定律揭示了信息技术的进步。

尽管这种趋势已经持续了超过半个世纪，摩尔定律仍应该被认为是观测或推测，而不是一个物理或自然法。预计摩尔定律将持续到至少2015年或2020年。然而2010年国际半导体技术发展路线图的更新增长已经放缓到2013年底，之后的时间里晶体管数目密度预计只会每3年翻一番。

### 1.3.10 高性能

尽管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。

使用Java的头几年，许多用户不同意这样的看法：**性能就是“适用性更强”**。然而，现在的即时编译器已经非常出色，以至于成了传统编译器的竞争对手。在某些情况下，甚至超越了传统编译器，原因是它们含有更多的可用信息。例如，即时编译器可以监控经常执行哪些代码并优化这些代码以提高速度。更为复杂的优化是消除函数调用（即“内联”）。即时编译器知道哪些类已经加载。基于当前加载的类集，如果特定的函数不会被覆盖，就可以使用内联。必要时，还可以撤销优化。

### 1.3.11 动态

从各种角度看，Java与C或C++相比更加具有动态性，它在很多方面比C和C++更能适应发展的环境，可以动态调整库中方法和增加变量，而客户端却不需要任何更改。在Java中进行动态调整是非常简单和直接的。

当需要将某些代码添加到正在运行的程序中时，动态性是一个非常重要的特性。一个很好地例子是：从Internet下载代码，然后在浏览器上运行。如果使用C或C++，这确实难度很大，不过Java设计者很清楚动态语言可以很容易地实现运行程序的演进。最终，他们将这一特性引入这个主流程序设计语言中。

## 1.4 Java环境的搭建

工欲善其事必先利其器。在学习Java语言语法之前，必须了解并搭建好它所需要的开发环境。要编译和执行Java程序，JDK（Java Developers Kits）是必备的。下面将具体认识下JDK和JRE，并进行介绍下载安装JDK和配置环境变量的方法。

### 1.4.1 Java的跨平台的原理实现

前文已经提到过Java的跨平台了。Java的跨平台是指：通过Java语言编写的应用程序在不同的系统平台上都可以运行。

原理是：只要在需要运行java应用程序的操作系统上，先安装一个Java虚拟机(JVM Java Virtual Machine)即可。由JVM来负责Java程序在该系统中的解释运行。简单来说就是如图1.1所示那样程序代码经过编译之后转换为一种称为**Java字节码的中间语言**，**Java虚拟机（JVM）将对字节码进行解释和运行。编译只进行一次，而解释在每次运行程序时都会进行。编译后的字节码采用一种针对JVM优化过的机器码形式保存，虚拟机将字节码解释为机器码，然后在计算机上面运行。**如下图1.6所示。

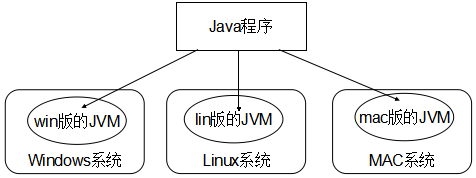
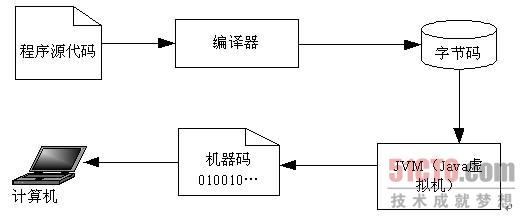


图1.6 Java的跨平台性的原理

因为有了JVM，所以同一个Java程序在三个不同的操作系统中都可以执行。这样就实现了Java程序的跨平台性。也称为Java具有良好的可移植性。

Java语言是跨平台的，而JVM不是跨平台的。

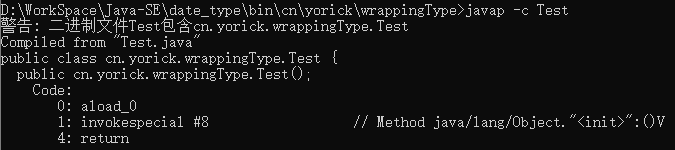
### 1.4.2 JDK

**jdk**是 [Java](http://baike.baidu.com/subview/29/12654100.htm" \t "http://baike.baidu.com/_blank) 语言的[软件开发工具包](http://baike.baidu.com/view/592963.htm" \t "http://baike.baidu.com/_blank)，是提供给Java开发人员使用的，其中包含了java的开发工具，也包括了运行环境JRE。所以安装了JDK，就不用再单独安装JRE了。**JDK包含的基本组件包括：**

* javac – 编译器，将源程序转成字节码



* jar – 打包工具，将相关的类文件打包成一个文件
* javadoc – 文档生成器，从源码注释中提取文档
* jdb – debugger，查错工具
* java – 运行编译后的java程序（.class后缀的）
* 
* appletviewer：小程序浏览器，一种执行[HTML文件](http://baike.baidu.com/view/394827.htm" \t "http://baike.baidu.com/_blank)上的Java小程序的Java浏览器。
* Javah：产生可以调用Java过程的C过程，或建立能被Java程序调用的C过程的头文件。
* Javap：Java反汇编器，显示编译类文件中的可访问功能和数据，同时显示[字节代码](http://baike.baidu.com/view/185293.htm" \t "http://baike.baidu.com/_blank)含义。



* Jconsole: Java进行系统调试和监控的工具

**为什么JDK中包含一个JRE呢？**

其一，开发完的程序，总需要运行一下看看效果。

其二，也是最重要的，JDK中的开发工具其实都是java语言编写的应用程序，为了方便使用才打包成exe文件，如果没有JRE，那么这些工具是运行不了的。

### 1.4.3 JRE

JRE（Java Runtime Environment，Java运行环境），运行JAVA程序所必须的环境的集合，包含JVM标准实现及Java核心类库。如果想要运行一个开发好的Java程序，计算机中只需要安装JRE即可。**简单而言：使用JDK开发完成的java程序，交给JRE去运行。**

### 1.4.4 JDK下载

Java的JDK又称Java SE（以前称J2SE），是Sun公司的产品，由于Sun公司已经被Oracle公司收购，因此JDK可以在Oracle公司的官方网站http://www.oracle.com/index.html下载。

|  |
| --- |
| 在Java 5 出版之后，J2SE、J2EE和J2ME正式更名，将名字中的2去掉，分别更改为Java SE、Java EE和Java ME |

下面以JDK 最新版本为例介绍下载安装JDK的方法，具体步骤如下：

* 打开IE浏览器，输入网址“http://www.oracle.com/index.html”，浏览Oracle官网，将光标移动当Downloads菜单项上,将显示下载列表下拉菜单,单击Java for Developers 超链接,如图1.7所示。

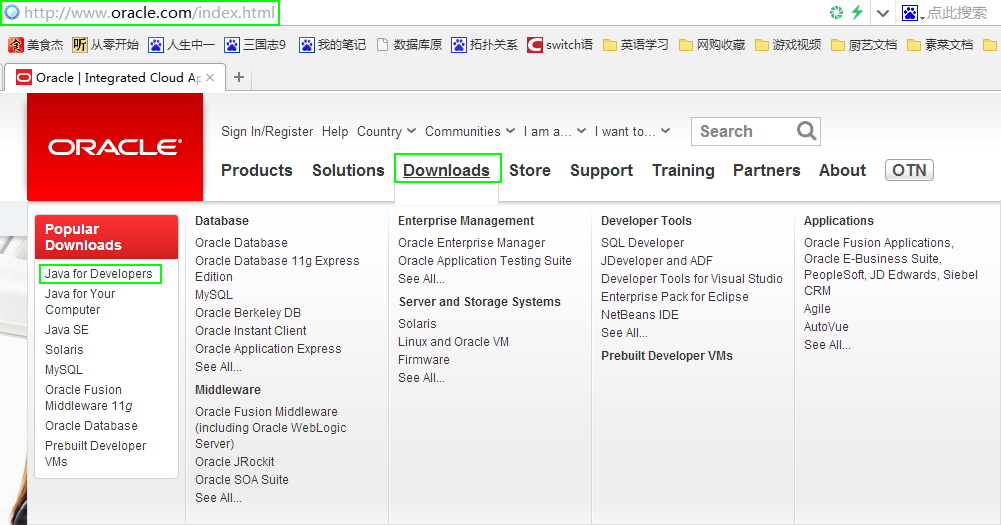


图1.7 Oracle主页 图1.8 JDK的下载页

* 将跳转到JDK的下载页面，在该页面中单击最新版本JDK的超链接，即如图1.8所示的Download按钮。
* 在进入的新页面中，需要先选中同意协议中的单选按钮，这时将显示如图1.9所示的页面,否则单击要下载的链接时不能进行下载。

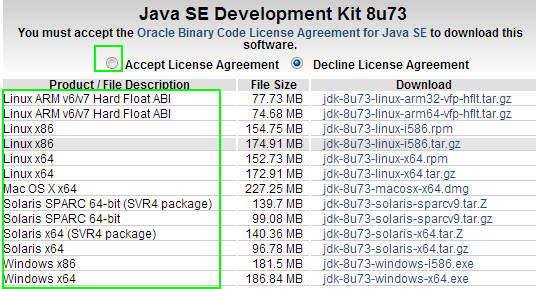


图1.9 JDK最新版本的下载页面

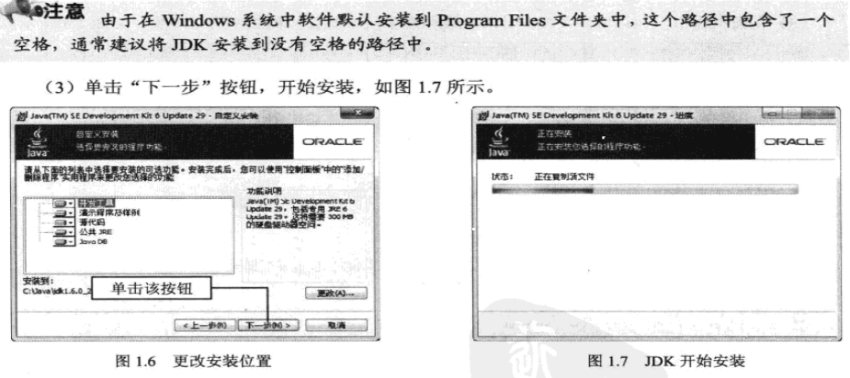
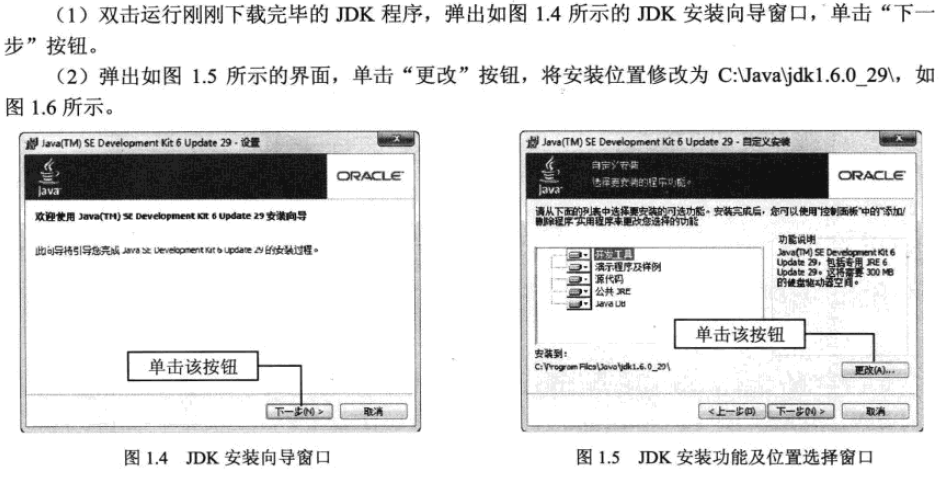
|  |
| --- |
| 下载时要选择适合自己操作系统平台的的安装文件，如Windows系统平台是无法安装Linux系统平台的安装文件的。如果我们需要一些旧版本的JDK则可以在进入JDK的下载链接页面后选择旧版本下载，单击进入历史页面，我们以JDK 7 为例如图1.10所示。      图1.10 历史版本的JDK下载 |

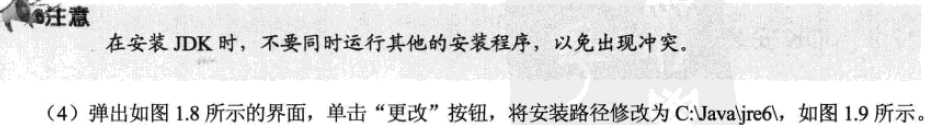
### 1.4.5 Windows系统的JDK环境

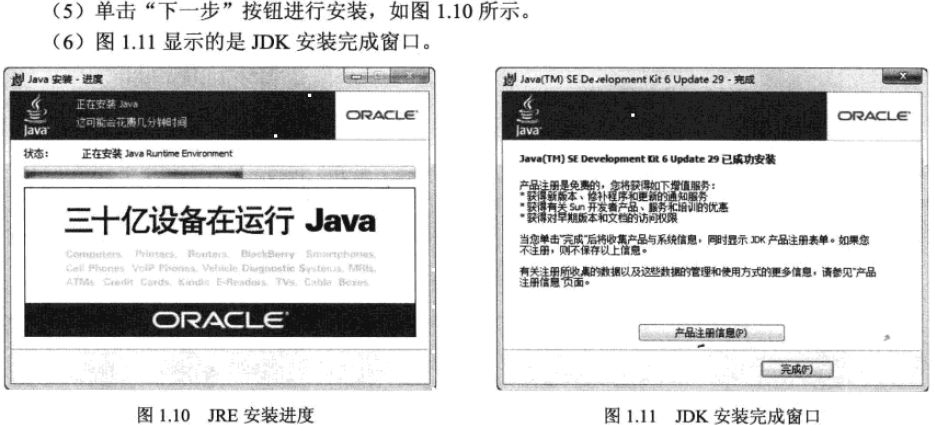
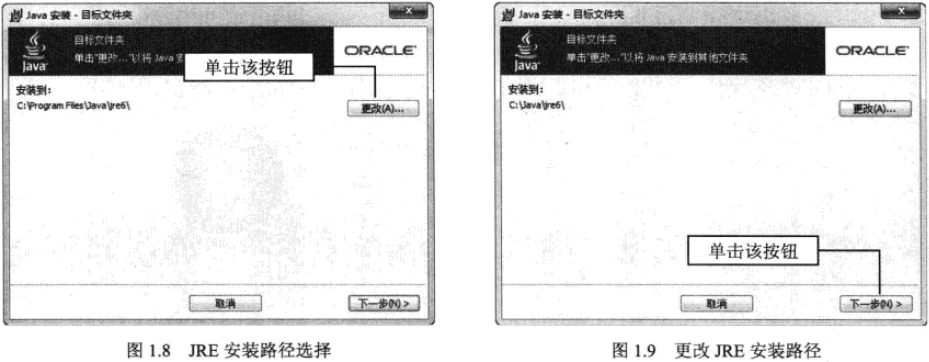
我们日常的开发工作大部分都是在Windows下同下完成的，所以我们这里以Windows为例进行JDK的安装介绍，在其他平台上流程也是大同小异的。下面开始JDK的安装。

#### 1.4.5.1 JDK的安装

我们以上文下载的JDK 7为例进行安装介绍，步骤如下：







Notes：JavaFX是由甲骨文宣布推出的，是一款为企业业务应用提供的先进Java用户界面（UI）平台，它能帮助开发人员无缝的实现与本地java功能及Web技术动态能力的混合与匹配。

#### 1.4.5.2 配置JDK

在安装完JDK后需要对环境变量进行配置，具体步骤如下所示。

##### 1.4.5.2.1 path环境变量配置的作用

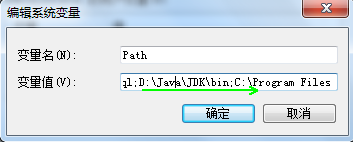
首先要理解什么是path（环境变量）?path（环境变量）是dos以前的内部命令，windows继续沿用至今。用作运行某个命令的时候，本地查找不到某个命令或文件，会到这个声明的目录中去查找。因为java必须经过编译才能被JVM解释运行,编译和执行就要通过常用命令比如:编译javac等，而这些常用的编译命令就存放在jdk的bin目录。而window运行java编译命令的时候，本地查找不到某个命令或文件，会到path中去查找，所以不可能在每个java文件所在目录放置jdk,故把jdk放置path环境变量中,这样每个java文件都可以通过path中设定的jdk目录找到编译命令进行编译，一般设定java的时候为了在任何目录下都可以运行bin文件夹下的命令。就将java的bin目录声明到path中。编译后的class类就要通过jre运行环境进行运行实现。这只是在文本编辑java时候需要这样,设置环境变量等。如果用开发工具Myeclipse就不需要设置,myeclipse自带jdk,就不需要环境变量，而eclipse启动时候会要求path,方法可以设置path,也可以不设置,不设置方法是:  
在eclipse.exe目录里面有一个eclipse.ini文件进行配置，比如  
**-vm  
C:\Java\jre1.6.0\bin\javaw.exe  
我的是  
-vm  
C:\Program Files (x86)\Java\jdk1.6.0\_21\bin\javaw.exe**  
进行设置虚拟机，可以把jdk放在你的eclipse文件夹中,配置它的目录,就不需要依靠本地jdk运行，然后就可以启动了。  
  
那么运行中发生什么呢？比如:在D:\test目录下有HelloWorld.java(第一个程序输出HelloWorld)

* D:\test>:javac HelloWorld.java //javac编译HelloWorld.java文件  
  首先window找HelloWorld.java文件所在目录中是否有javac命令,有就进行编译,没有找path,按照系统中定义的path目录寻找javac命令,找到进行编译,找不到就报不是外部内部命令错误
* D:\test>java HelloWorld //编译通过java HelloWorld运行编译后的文件  
  首先window找HelloWorld.class文件所在目录中是否有java命令,有就进行运行,没有找path,按照系统中定义的path目录寻找java命令,找到进行运行,找不到就报不是外部内部命令错误
* HelloWorld //输出结果

当然我们也可以将程序写入bin目录下，然后在bin目录下测试执行，那么相像一下，若干天后你的bin目录是不是就爆炸了呢？这是我们极其不建议的做法。

##### 1.4.5.2.2 path环境变量配置方式

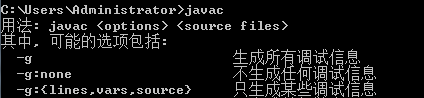
* 通过配置path环境变量，将javac指令所在目录也就是JDK安装目录下的bin目录配置到path变量下，即可使javac指令在任意目录下运行
* win7,win8系统：右键点击桌面计算机→选择属性→选择高级系统设置→选择高级选 项卡→点击环境变量→下方系统变量中查找path→双击path
* XP系统：右键点击桌面计算机→选择属性→选择高级选项卡→点击环境变量→下方系统变量中查找path→双击path将jdk安装目录下的bin目录添加到最左边并添加分号



* path环境变量的参照形配置方式

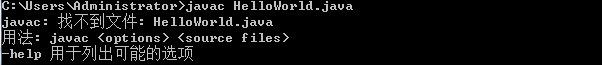
创建新的变量名称：JAVA\_HOME，为JAVA\_HOME添加变量值：JDK安装目录，将path环境变量中JDK目录修改，%JAVA\_HOME%\bin;，path环境变量具有先后顺序

我们配置成功后发现javac和java可以到处运行



##### 1.4.5.2.3 classpath环境变量配置方式

我们配置成功path环境变量后发现java和javac可以到处运行，那么此时我们已不必将源代码放入bin目录下了，但是我们遇到这么一个问题就是，源文件的运行必须在源文件的目录下编译



那么如果我们要摆脱这个限制做到可以再任意目录下编译任何想要编译的文件，那么我们就用到了classpath 类路径，**classpath就是告诉java.exe在哪些目录查找class文件了**，**设置Classpath的目的，在于告诉Java执行环境，在哪些目录下可以找到您所要执行的Java程序所需要的类或者包。**CLASSPATH它是javac编译器的一个环境变量。它的作用与import、package关键字有关。当你写下improt java.util.\*时，编译器面对import关键字时，就知道你要引入java.util这个package中的类；但是编译器如何知道你把这个package放在哪里了呢？所以你首先得告诉编译器这个package的所在位置；如何告诉它呢？就是设置CLASSPATH啦，如果java.util这个package在c:\jdk\ 目录下，你得把c:\jdk\这个路径设置到CLASSPATH中去！当编译器面对import java.util.\*这个语句时，它先会查找CLASSPATH所指定的目录，并检视子目录java\util是否存在，然后找出名称吻合的已编译文件（.class文件）。如果没有找到就会报错！

 CLASSPATH有点像c\c++编译器中的INCLUDE路径的设置哦，是不是？当c\c++编译器遇到include <iostream>这样的语句，它是如何运作的？哦，其实道理都差不多！搜索INCLUDE路径，检视文件！

当你自己开发一个package时，然后想要用这个package中的类；自然，你也得把这个package所在的目录设置到CLASSPATH中去！

CLASSPATH的设定，对JAVA的初学者而言是一件棘手的事。所以Sun让JAVA2的JDK更聪明一些。你会发现，在你安装之后，即使完全没有设定CLASSPATH，你仍然能够编译基本的JAVA程序（java.long包下的工具类），并且加以执行。classpath环境变量配置方式

* 创建新的变量名称：classpath
* 值设定为指定的还有class文件的目录，多个目录间使用分号(;)分割

它的作用是：使classpath目录中的.class文件可以在任意目录运行

* 技巧：通常将配置的目录最前面添加.配置，即便当前目录，使.class文件搜索时首先搜索当前目录，然后根据目录配置的顺序依次查找，找到后即运行，因此classpath目录中的配置存在先后顺序。通常情况下我们对classpath的配置如下

**CLASSPATH=.;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar**

补充说明：dt.jar和tools.jar位于：{Java\_Home}/lib/下，而rt.jar位于：{Java\_Home}/jre/lib/下,其中：rt.jar是JAVA基础类库，也就是你在java doc里面看到的所有的类的class文件，dt.jar是关于运行环境的类库，tools.jar是工具类库,编译和运行需要的都是toos.jar里面的类分别是sun.tools.java.\*; sun.tols.javac.\*;

  在Classpath设置这几个变量，是为了方便在程序中 import；Web系统都用到tool.jar。

1. rt.jar  
    rt.jar 默认就在Root Classloader的加载路径里面的，而在Claspath配置该变量是不需要的；同时jre/lib目录下的

    其他jar:jce.jar、jsse.jar、charsets.jar、resources.jar都在Root Classloader中

2. tools.jar

    tools.jar 是系统用来编译一个类的时候用到的，即执行javac的时候用到

    javac XXX.java

    实际上就是运行

    java -Calsspath=%JAVA\_HOME%\lib\tools.jar xx.xxx.Main XXX.java

    javac就是对上面命令的封装 所以tools.jar 也不用加到classpath里面

3. dt.jar  
    dt.jar是关于运行环境的类库,主要是swing的包   在用到swing时最好加上。

##### 1.4.5.2.4 path和classpath的区别

path环境变量里面记录的是可执行性文件，如.exe文件，对可执行文件先在当前路径去找，如果没找到就去path环境变量中配置的路径去找，classpath环境变量里记录的是java类的源代码文件所在的目录。

## 1.5 第一个Java程序“Hello World”

对Java的各种概念有了解，且了解了Java环境的安装配置后，我们开始开篇的最后一个课题，那就是编写第一个Java Demo。

### 1.5.1 工具介绍

编写Java程序可以使用很多种工具

* notepad(微软操作系统自带)
* Notepad++
* Eclipse
* MyEclipse

我们在初始学习阶段先通过Notepad++来学习，这样我们可以更直观的看到程序运行的流程。

### 1.5.2 HelloWorld

源代码

class HelloWorld{

public static void main(String[] args){

System.out.print("Hello World;");

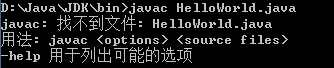
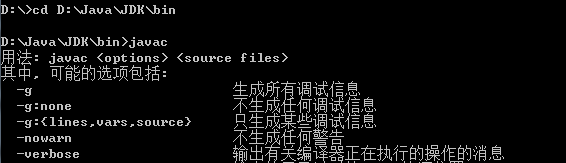
System.out.println("Hello World;");

}

}

将源代码存放在K:\WorkSpace\JavaJC\Chapter 1

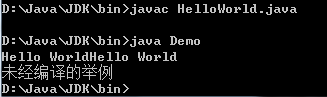
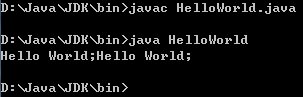
此时我们打开命令行，定义到JDK的目录下，此时我们编译程序会发现



原因在于javac.exe它只会编译bin目录下的文件（编译当前路径下的文件），而HelloWorld.java不在当前路径bin目录下,我们将需要编译的文件放在当前bin目录下，重新编译发现提示编码错误错误，这是我们可以解决错误提示：编码 GBK 的不可映射字符，使用命令：javac -encoding utf-8 helloword.java，编译成功



运行我们编译成功的class，如下左图。在这里我们要注意当我们更改源文件时，我们在运行可执行文件.class时，如果没有编译，那么更改是无效的，简单举例：当我们将HelloWorld.java中再加入一句输入语句System.out.print(“未经编译的举例”)；此时我们不编译，直接运行可执行文件，结果如下中图



输出结果是上一次我们编译后的呈现，我们重新编译，重新执行，这次变化才会显示出来，如上有图

### 1.5.3程序解析

class Demo{ /\*首先定义一个类 class 类名 这个类名将是未来我们编译后的字节码文件名

在类定义后加上一对大括号{}，类体写在这对大括号中\*/

/\*在大括号中间添加一个主(main)方法/函数 public static void main(String [] args){ } main是给类定义的方法，当类作为包中的主类时，必须要有main方法，main方法是程序的入口，当程序执行时，优先调用它。然后在main方法中再调用别的类和方法，就完成了程序。\*/

public static void main(String[] args){

/\*在主方法的大括号中间添加两行输出语句 System.out.print(“hello world”); System.out.println(“hello world”);Ln的区别在于是一个回车 有Ln代表有一个回车\*/

System.out.print("Hello World");

System.out.println("Hello World");

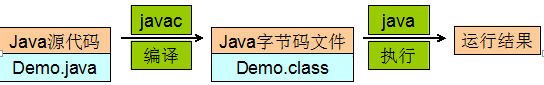
}

}

当我们不输入main方法时，我们的编译仍可进行，但执行可执行文件时会报错误，这是由于JVM需要一个入口，而main函数就是这个JVM运行的入口。



### 1.5.4运行与工作原理



### 1.5.5常见错误

* 文件扩展名隐藏导致编译失败
* class写错 Class claas…
* 类名格式有问题，暂时全部使用英文
* 类名后面的大括号匹配不正确
* main方法格式错误 public static void main(String [] args){ }
* main方法大括号缺失
* 打印语句拼写错误 System.out.println(“”);
* 引号使用错误，使用成中文全角引号

如果编译后无法生成可执行文件一般可以按以下步骤进行排查，找到问题所在：

* 1.先检查jdk环境是否配置成功

做法：DOS下输入：java -version,看是否有jdk版本的信息；java，看是否有相关命令；javac，看是否有这个命令；验证这3个命令，会对jdk环境配置有个直观的认识，如果没有正常显示相关信息，就要去配置jdk环境，具体步骤可参考上面JDK配置。

* 2.检查路径是否正确

做法：DOS下输入：dir,查看当前路径下是否真的有这个HelloWorld.java文件

* 3.通过命令 javac -d 你想生成class文件的目录 HelloWorld.java 来查看是否有class文件

做法：DOS下输入：javac -d C:\ HelloWorld.java  ，运行后，看是否在C盘下有HelloWorld.class文件

好了作为Java的开篇之述，进行到这里就已经到了尾声了，在本篇中我们着重介绍的是Java开发的准备工作，本篇之后我们才算是真正今日Java的世界，编程式意见枯燥却有好玩的事情，坚持不懈方有始终！！！