


第 1 章

初识 Java

( 视频讲解：20 分钟)

Java 是一种可以编写跨平台的、面向对象的程序设计语言。本章将向读者简单介绍 Java 语言和相关特性、不同的版本以及学好 Java 语言的方法等。初识 Java 所需要掌握的内容，其主要目的是让读者对 Java 语言有一个整体的了解，然后再慢慢地学习具体内容，最后达到完全掌握 Java 语言的目的。

通过阅读本章，您可以：

- » 了解 Java 语言的历史
- » 了解 Java 语言的现状
- » 了解 Java 语言的面向对象特性
- » 掌握不同平台的 JDK 环境搭建
- » 掌握 Java 程序的编写方法



1.1 Java 语言概述

 视频讲解：光盘\TM\lx\1\Java 语言概述.exe

1.1.1 Java 语言的历史

Java 语言是 Sun 公司于 1990 年开发的，当时 Green 项目小组的研究人员正在致力于为未来的智能设备开发一种新的编程语言。由于该小组的成员 James Gosling 对 C++ 语言在执行过程中的表现非常不满，就把自己封闭在办公室里编写了一种新的语言，并将其命名为 Oak（Oak 就是 Java 语言的前身），这个名字源于 Gosling 办公室窗外的一棵橡树（Oak）。此时的 Oak 已经具备安全性、网络通信、面向对象、多线程等特性，是一款相当优秀的程序语言。后来，在注册 Oak 商标时，发现它已经被另一家公司注册，所以不得不改名。要取什么名字呢？工程师们边喝咖啡边讨论着，看看手上的咖啡，联想到印度尼西亚有一个盛产咖啡的岛屿（中文名叫爪哇），于是将其改名为 Java。

随着 Internet 的迅速发展，Web 应用日益广泛，Java 语言也得到了迅速发展。1994 年，Gosling 用 Java 语言开发了一个实时性较高、可靠、安全、有交互功能的新型 Web 浏览器，它不依赖于任何硬件平台和软件平台。该浏览器被命名为 HotJava，并于 1995 年在业界发表，引起了巨大的轰动，Java 语言的地位随之得到了肯定。1995 年 5 月 23 日，JDK（Java Development Kit）1.0a2 版本正式对外发布。此后，Java 语言的发展异常迅速。2009 年 4 月 20 日，Sun 公司被 Oracle 公司收购。

1.1.2 Java 语言的现状

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。由于 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、手机和其他设备上运行。Java 的用途数不胜数，它拥有无可比拟的能力，使用它所节省的时间和费用十分可观。

如果仔细观察就会发现，Java 就在我们身边。使用 Java 语言编写的常见开源软件包括 NetBeans 和 Eclipse 集成开发环境、JBoss 和 GlassFish 应用服务器；商业软件包括永中 Office、合金战士 Chrome、WebSphere 和 Oracle Database 11g。此外，各手机厂商都为自己的产品提供了 Java 技术的支持，手机上的 Java 程序和游戏已经不胜枚举。

为了满足不同开发人员的需求，Java 开发分成了以下 3 个方向。

- ☒ Java SE：主要用于桌面程序的开发。它是学习 Java EE 和 Java ME 的基础，也是本书的重点内容。
- ☒ Java EE：主要用于网页程序的开发。随着互联网的发展，越来越多的企业使用 Java 语言来开发自己的官方网站，其中不乏一些世界 500 强。
- ☒ Java ME：主要用于嵌入式系统程序的开发。

1.1.3 Java 语言的面向对象特性

Java 语言的作者们编写了具有广泛影响的 Java 白皮书，里面详细介绍了他们的设计目标以及实现

成果。此外，还用简短的篇幅介绍了 Java 语言的特性。下面将对其进行简要介绍。

1. 简单

Java 语言的语法简单明了，容易掌握，而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点：

- ☑ 语法规则和 C++ 类似。从某种意义上讲，Java 语言是由 C 和 C++ 语言转变而来，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，现在 JDK 也开放源代码了，读者可以通过分析项目的源代码，提高自己的编程水平。

2. 面向对象

Java 语言本身是一种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变量和方法，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。

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 对象访问网络对象，访问方式与访问本地系统相同。

4. 可移植性

Java 程序具有与体系结构无关的特性，从而使 Java 程序可以方便地移植到网络的不同计算机中。同时，Java 的类库中也实现了针对不同平台的接口，使这些类库也可以移植。

5. 解释性

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用 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信息，使得连接过程更加简单，开发过程也就更加迅速，更具探索性。

6. 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，从而有效地避免了非法操作内存。Java 程序代码要经过代码校验、指针校验等很多的测试步骤才能够运行，所以未经允许的 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。

7. 健壮性

Java 的设计目标之一是编写多方面可靠的应用程序，Java 将检查程序在编译和运行时的错误，以及消除错误。类型检查能帮助用户检查出许多在开发早期出现的错误。同时，很多集成开发工具 IDE（如 Eclipse、NetBeans）的出现使编译和运行 Java 程序更加容易。

8. 多线程

多线程机制能够使应用程序在同一时间并行执行多项任务，而且相应的同步机制可以保证不同线程能够正确地共享数据。使用多线程，可以带来更好的交互能力和实时行为。

9. 高性能

Java 编译后的字节码是在解释器中运行的，所以它的速度比多数交互式应用程序提高了很多。另外，字节码可以在程序运行时被翻译成特定平台的机器指令，从而进一步提高运行速度。

10. 动态

Java 在很多方面比 C 和 C++ 更能够适应发展的环境，可以动态调整库中方法和变量的增加，而客户端却不需要作任何更改。在 Java 中动态调整是非常简单、直接的。

1.2 JDK 的下载和安装

视频讲解：光盘\TM\1\JDK 的下载和安装.exe

开发 Java 程序必须安装 JDK（Java Development Kit）开发环境，它包含演示程序和样例、Java 公共 API 类的源代码、Java 运行环境、编译调试等工具。本节将介绍如何下载和安装 JDK。

1.2.1 JDK 下载

由于 Sun 公司已经被 Oracle 收购，因此 JDK 可以在 Oracle 公司的官方网站(<http://www.oracle.com/index.html>)下载。下面以目前最新版本 JDK 6 Update 29 为例介绍下载 JDK 的方法，具体步骤如下：

(1) 打开浏览器，进入 Oracle 官方主页，如图 1.1 所示。



图 1.1 Oracle 官方主页

(2) 选择 Downloads 选项卡, 选择 Java for Developers, 在跳转的页面中滚动到如图 1.2 所示位置, 单击 JDK 下方的 Download 按钮。

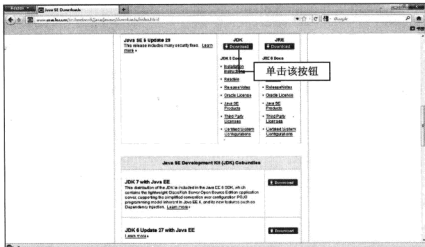


图 1.2 Java 开发资源下载页面

(3) 在新页面中, 同意协议并根据计算机硬件和系统选择适当的版本进行下载, 如图 1.3 所示。

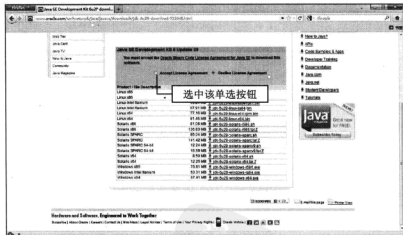


图 1.3 JDK 下载页面

1.2.2 JDK 安装

下载完适合自己系统的 JDK 版本后, 就可以进行安装了。这里以 Windows 系统为例, 讲解 JDK

的安装过程，具体步骤如下：

(1) 双击运行刚刚下载完毕的 JDK 程序，弹出如图 1.4 所示的 JDK 安装向导窗口，单击“下一步”按钮。

(2) 弹出如图 1.5 所示的界面，单击“更改”按钮，将安装位置修改为 C:\Java\jdk1.6.0_29\，如图 1.6 所示。

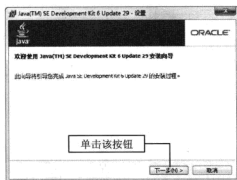


图 1.4 JDK 安装向导窗口

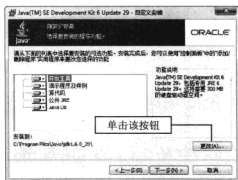


图 1.5 JDK 安装功能及位置选择窗口



注意

由于在 Windows 系统中软件默认安装到 Program Files 文件夹中，这个路径中包含了一个空格，通常建议将 JDK 安装到没有空格的路径中。

(3) 单击“下一步”按钮，开始安装，如图 1.7 所示。

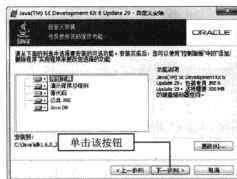


图 1.6 更改安装位置

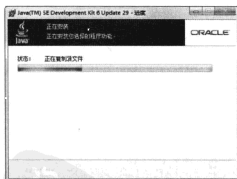


图 1.7 JDK 开始安装



注意

在安装 JDK 时，不要同时运行其他的安装程序，以免出现冲突。

(4) 弹出如图 1.8 所示的界面，单击“更改”按钮，将安装路径修改为 C:\Java\jre6\，如图 1.9 所示。

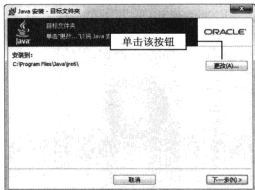


图 1.8 JRE 安装路径选择

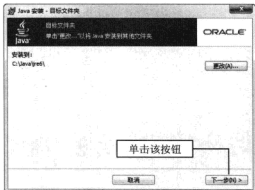


图 1.9 更改 JRE 安装路径

(5) 单击“下一步”按钮进行安装,如图 1.10 所示。

(6) 图 1.11 显示的是 JDK 安装完成窗口。



图 1.10 JRE 安装进度

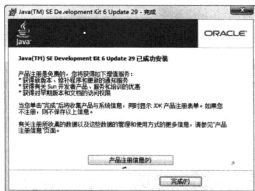


图 1.11 JDK 安装完成窗口

1.2.3 配置 JDK

在安装完 JDK 之后,需要对环境变量进行配置,具体步骤如下:



说明

如果使用集成开发工具,如 Eclipse、NetBeans 等,可以省略此步骤。

(1) 在 Windows 7 系统中,同时按住 Win 键和 Pause 键打开“系统属性”窗口,如图 1.12 所示。选择“高级系统设置”选项。

(2) 弹出如图 1.13 所示的对话框,单击“环境变量”按钮。



图 1.12 系统基本信息窗口

(3) 弹出如图 1.14 所示的“环境变量”对话框，单击“新建”按钮，新建系统变量。

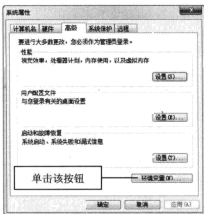


图 1.13 “系统属性”对话框



图 1.14 “环境变量”对话框

(4) 弹出“新建系统变量”对话框，在“变量名”文本框中输入“JAVA_HOME”，在“变量值”文本框中输入 JDK 的安装路径“C:\Java\jdk1.7.0_24\”，如图 1.15 所示。单击“确定”按钮，完成环境变量 JAVA_HOME 的配置。

(5) 在系统变量中查找 Path 变量，如果不存在，则新建系统变量 Path；否则选中该变量，单击“编辑”按钮，打开“编辑系统变量”对话框，如图 1.16 所示。在“变量值”文本框的起始位置添加“%JAVA_HOME%\bin;”。



图 1.15 “新建系统变量”对话框



图 1.16 “编辑系统变量”对话框

说明

在 Windows 系统中，环境变量需要使用英文的分号进行分隔；在 Linux 系统中，环境变量需要使用英文的冒号进行分隔。要注意全角和半角的区别。

1.2.4 测试开发环境

JDK 程序的安装和配置完成后，需要测试 JDK 是否能够在计算机上运行。具体测试步骤如下：

同时按住 Win 键和 R 键，打开“运行”窗口。输入“cmd”并单击“运行”按钮，将进入到 DOS 环境中。在命令提示符后面直接输入“javac”，按 Enter 键，系统会输出 javac 的帮助信息，如图 1.17 所示。这说明已经成功配置了 JDK，否则需要仔细检查上面步骤的配置是否正确。

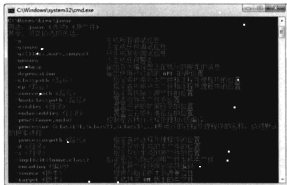


图 1.17 测试 JDK 安装的 DOS 窗口

1.3 第一个 Java 程序

视频讲解：光盘\TM\1\1\第一个 Java 程序.exe

在完成 JDK 的下载和安装后，就可以开始编写 Java 程序了。下面将编写第一个 Java 程序，其用途是在 DOS 控制台上显示“我能学好 Java 语言！”。通过本节内容，读者可以学习开发 Java 程序的流程。

1. 编写源代码

使用 Java 编程的第一步是编写源代码，这里需要使用到文本编辑器。目前有各种各样的文本编辑

器，如“记事本”工具、Office 等。这里使用 Windows 系统自带的“记事本”工具。选择“开始”/“所有程序”/“附件”/“记事本”命令，然后输入如下内容：

```
public class MyApp {
    public static void main(String[] args) {
        System.out.println("我能学好 Java 语言！");
    }
}
```

按 Ctrl+S 键保存源代码，弹出如图 1.18 所示的窗口。

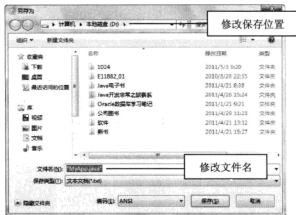


图 1.18 保存窗口

将文件保存到 D 盘，文件名使用 MyApp.java。

注意

默认“记事本”软件会为文件增加扩展名.txt，而 Java 程序的源代码扩展名要使用.java。所以在修改文件名时需要使用双引号将文件名括起来，这样就不会增加新的扩展名。

说明

文件名需要与源代码中公共类的名字相同，这里公共类是 MyApp，因此文件名也是 MyApp。

注意

Java 语言是区分大小写的，因此文件名必须是 MyApp，而不能是 Myapp，或者其他。

2. 编译源代码

选择“开始”/“所有程序”/“附件”/“命令提示符”命令，将路径切换到 D 盘，并使用 dir 命令查看 D 盘中的文件，其运行效果如图 1.19 所示。

接着运行 `javac MyApp.java` 命令，然后使用 `dir` 命令查看 D 盘中的文件，其运行效果如图 1.20 所示。



图 1.19 编译前的 D 盘文件列表

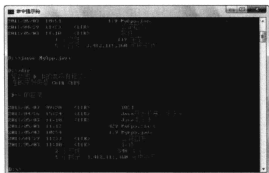


图 1.20 编译后的 D 盘文件列表

读者可以发现，在 D 盘中出现了一个名为 `MyApp`、类型为 `class` 的文件。

说明

`javac` 是 JDK 中提供的编译源代码的命令，由于已经配置了环境变量，因此可以直接使用。

注意

`javac` 命令和 `MyApp.java` 之间存在一个空格。

3. 运行 class 文件

输入“`java MyApp`”来运行 `class` 文件，其运行效果如图 1.21 所示。

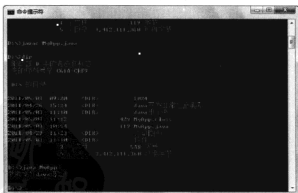


图 1.21 运行 class 文件后的窗口

读者可以看到，在 DOS 控制台上输出了“我能学好 Java 语言!”。

**注意**

使用 `java` 命令时，不需要输入文件的扩展名。而使用 `javac` 命令时，需要输入文件的扩展名。

1.4 本章小结

本章首先简单介绍了 Java 语言的历史、现状及各种特性，接着介绍了 JDK 的安装和配置，这是进行 Java 程序开发的第一步，也是本章的重点。最后介绍了如何开发第一个 Java 应用程序，读者可以从中学学习到 Java 开发的一般流程是编写源代码、编译源代码和运行 class 文件。

