

第 1 章

Java 语言概述

内容提要

- (1) 认识 Java 并了解其特点。
- (2) 掌握 Java 程序的运行机制。
- (3) 安装配置 Java 开发环境。
- (4) 编写一个简单的 Java 程序。
- (5) Java 控制台输入/输出。

Java 是一种可以编写跨平台应用程序的面向对象程序设计语言,本章介绍 Java 平台的分类、Java 的特性、运行机制、开发环境的安装和配置,以及一个简单的 Java 程序示例。

1.1 Java 简介

Java 是由 Sun 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台(即 Java SE、Java EE、Java ME)的总称。Java 技术具有卓越的通用性、高效性、平台移植性和安全性,广泛应用于 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网,同时拥有规模巨大的开发者专业社群。在全球云计算和移动互联网的产业环境下,Java 更具备显著优势和广阔前景。2009 年 4 月 20 日,Oracle(甲骨文)公司收购了 Sun 公司,取得 Java 的版权。

Java 提供了三类开发平台,分别用于开发三类不同的程序。

- (1) Java ME(Java Micro Edition),主要应用于嵌入式系统开发,如手机和 PDA 的编程。
- (2) Java SE(Java Standard Edition),针对桌面计算机应用开发,主要有 Java Applet(网页中内嵌的小程序)及 Java Application(应用程序)。
- (3) Jakarta EE,这个版本以前称为 Java EE(Java Enterprise Edition),主要用于分布式的网络程序的开发,如电子商务网站和 ERP 系统。它包括 JDBC、CORBA、EJB、Java Servlets、JSP 以及对 XML 的支持。

注: 2017 年 8 月,Oracle(甲骨文)决定将 Java EE 移交给开源组织,最后由 Eclipse 基金会接手并将 Java EE 更名为 Jakarta EE(雅加达)。

其中,Java SE 是最核心的部分,Java ME 和 Java EE 是以 Java SE 为基础的。本书介绍的就是 Java SE 部分。

1.2 Java 语言的特性

Java 语言有许多很鲜明的特性,下面介绍其中最主要的特性。

(1) 简单性。Java 语言的语法与 C 语言和 C++ 语言很接近,使得大多数程序员很容易学习和使用 Java。另外,Java 丢弃了 C++ 中很少使用的、很难理解的、令人迷惑的那些特性,如操作符重载和多继承。Java 语言不使用指针,并提供了自动的对象垃圾收集,使得程序员不必为内存管理而担忧。

(2) 面向对象。Java 语言提供类、接口和继承,支持类之间的单继承,支持接口之间的多继承,并支持类与接口之间的实现机制。Java 语言全面支持动态绑定,而 C++ 语言只对虚函数使用动态绑定。

(3) 健壮性。Java 的强类型机制、异常处理、垃圾自动收集等是 Java 程序健壮性的重要保证。对指针的丢弃是 Java 的明智选择。Java 的安全检查机制使得 Java 更具健壮性。

(4) 安全性。Java 通常被用在网络环境中,为此,Java 提供了一个安全机制以防恶意代码的攻击。除了 Java 语言具有的许多安全特性以外,Java 对通过网络下载类具有一个安全防范机制(ClassLoader 类),如分配不同的命名空间以防止替代本地的同名类、进行字节代码检查,并提供安全管理机制(SecurityManager 类)让 Java 应用设置安全哨兵。

(5) 体系结构中立。Java 程序(扩展名为 .java 的文件)在 Java 平台上被编译为体系结构中立的字节码格式(扩展名为 .class 的文件),然后可以在实现 Java 平台的任何系统中运行。这种途径适合于异构的网络环境和软件的分发。

(6) 可移植性。可移植性来源于体系结构中立性,另外,Java 还严格规定了各个基本数据类型的长度。Java 系统本身也具有很强的可移植性,Java 编译器是用 Java 实现的,Java 的运行环境是用 ANSI C 实现的。

(7) 解释型。如前所述,Java 程序在 Java 平台上被编译为字节码格式,然后可以在实现 Java 平台的任何系统中运行。在运行时,Java 平台中的 Java 解释器对这些字节码进行解释执行,执行过程中需要的类在链接阶段被载入运行环境中。

(8) 高性能。与那些解释型的高级脚本语言相比,Java 的确是高性能的。事实上,Java 的运行速度随着 JIT(Just-In-Time)编译器技术的发展越来越接近于 C++。

(9) 多线程。在 Java 语言中,线程是一种特殊的对象,它由 Thread 类或其子类来创建。Java 语言支持多个线程的同时执行,并提供多线程间的同步机制。

(10) 动态性。Java 语言的设计目标之一是适应动态变化的环境。Java 程序需要的类能够动态地被载入运行环境,也可以通过网络来载入所需的类。这也有利于软件的升级。另外,Java 中的类有一个运行时的表示,能进行运行时的类型检查。

总之,Java 语言的优良特性使得 Java 应用具有无比的健壮性和可靠性,这也减少了应用系统的维护费用。Java 的“编译一次,到处运行”的特性使得它能够提供一个随处可

用的开放结构和在多平台之间传递信息的低成本方式。特别是 Java 企业应用编程接口 (Java Enterprise API) 为企业计算及电子商务应用系统提供了有关技术和丰富的类库。

1.3 Java 程序的运行机制

计算机高级语言主要有编译型和解释型两种。Java 语言属于解释型语言,但又不是纯解释型语言,因为 Java 源程序要经过编译生成一种十分接近机器码的称为字节码的目标代码,然后才可以在解释器(称为 Java 虚拟机,简称 JVM)上解释执行,如图 1-1 所示。

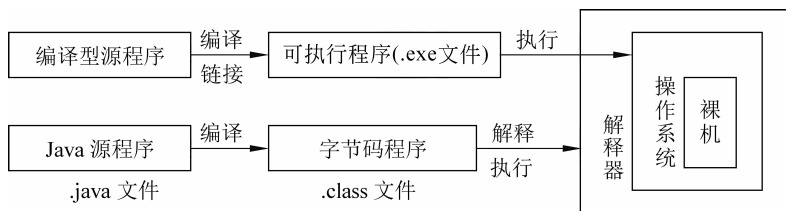


图 1-1 Java 和编译型语言的对比

Java 语言的一个非常重要的特点就是与平台无关,而使用 Java 虚拟机是实现这一特点的关键。一般用高级语言编写的程序是针对特定操作系统的,只能运行在特定的操作系统中。例如,在 Windows 操作系统平台编写的程序只能运行在使用 Windows 操作系统的计算机上,在使用 Linux 或 UNIX 等操作系统的计算机上无法直接运行,反之亦然。

而 Java 语言使用 Java 虚拟机屏蔽了与具体平台相关的信息,使得 Java 语言编译程序只须生成在 Java 虚拟机上运行的目标代码(字节码),就可以在多种平台上不加修改地运行。Java 虚拟机在执行字节码时,把字节码解释成具体平台上的机器指令,这一过程如图 1-2 所示。

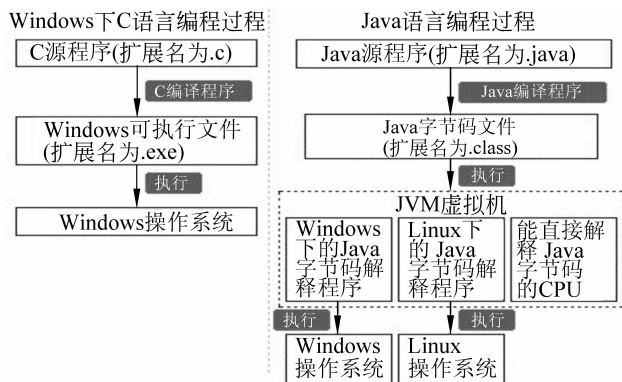


图 1-2 Java 跨平台原理

Java 程序和编译型程序一样也要经过编写、编译、运行三个步骤。编写是指在 Java 开发环境中进行程序代码的输入,最终形成扩展名为 .java 的 Java 源文件。编译是指使用 Java 编译器对源文件进行错误排查并生成扩展名为 .class 的字节码文件,这不像 C 语

言那样最终生成可执行文件。运行是指使用 Java 解释器将字节码文件翻译成机器代码,执行并显示结果。

字节码文件是一种和任何具体机器环境及操作系统环境无关的中间代码,它是一种二进制文件,是 Java 源文件由 Java 编译器编译后生成的目标代码文件。编程人员和计算机都无法直接读懂字节码文件,它必须由专用的 Java 解释器来解释执行,因此,Java 是一种在编译基础上进行解释运行的语言。

Java 解释器负责将字节码文件翻译成具体硬件环境和操作系统平台下的机器代码,以便执行。因此,Java 程序不能直接运行在现有的操作系统平台上,它必须运行在 Java 虚拟机上。

Java 虚拟机(JVM)是运行 Java 程序的软件环境,Java 解释器就是 Java 虚拟机的一部分。在运行 Java 程序时,首先会启动 JVM,然后由它来负责解释执行 Java 的字节码,并且 Java 字节码只能运行于 JVM 之上。这样利用 JVM 就可以把 Java 字节码程序和具体的硬件平台以及操作系统环境分隔开,只要在不同的计算机上安装了针对特定平台的 JVM,Java 程序就可以运行,而不用考虑当前具体的硬件平台及操作系统环境,也不用考虑字节码文件是在何种平台上生成的。JVM 把这种不同软硬件平台的具体差别隐藏起来,实现了真正的二进制代码级的跨平台移植。JVM 是 Java 平台无关的基础,Java 的跨平台特性正是通过在 JVM 中运行 Java 程序实现的。

Java 语言这种“一次编写,到处运行(write once,run anywhere)”的特性有效地解决了目前大多数高级程序设计语言需要针对不同系统来编译产生不同机器代码的问题,即硬件环境和操作平台的异构问题,大大降低了程序开发、维护 and 管理的开销。

注意: Java 程序通过 JVM 可以跨平台运行,但 JVM 是不跨平台的。也就是说,不同操作系统上的 JVM 是不同的,Windows 平台上的 JVM 不能用在 Linux 上,反之亦然。

1.4 Java 开发工具 JDK

Java 语言的开发环境可以使用 JDK(Java Development Kit,Java 开发工具箱),也可以使用集成开发环境,如 Eclipse。其中,JDK 的使用和配置是初学者应该掌握的基础知识,应该在学会了配置和使用 JDK 之后,再去使用更方便、功能更强大的集成开发环境。下面首先介绍 JDK 的配置和使用,然后介绍广为流行的集成开发环境 Eclipse 的使用。

1.4.1 JDK 的下载和安装

JDK 中主要包括以下内容。

(1) JVM(Java Virtual Machine,Java 虚拟机),其主要作用是进行 Java 程序的运行和维护。

(2) Java API(Java 应用程序编程接口),其主要作用是为编程人员提供已经写好的功能,便于快速开发。

(3) Java 编译器(javac.exe)、Java 运行时解释器(java.exe)、Java 文档化工具(javadoc.exe)及其他工具和资源。

要获取最新版的 JDK,可以到甲骨文公司官方网站下载,JDK 的下载地址是: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>,将 jdk-11.0.1_windows-x64_bin.exe(编者在编写本书时的最新版本)下载到本地,然后进行安装。JDK 默认安装在 C:\Program Files\Java\jdk-11.0.1 文件夹中,该文件夹中主要有以下内容。

(1) bin 文件夹:存放 JDK 的可执行文件,包括编译器、解释器等。系统的 PATH 环境变量应该包含此项。

(2) lib 文件夹:存放 JDK 的类库文件。lib 文件夹中有一个 Src.zip 文件,是包含 JDK 源代码的压缩包。

1.4.2 JDK 的环境变量配置

右击桌面上的“计算机”图标,在弹出的快捷菜单中选择“属性”命令(见图 1-3),弹出“系统属性”对话框。选择“高级”选项卡,如图 1-4 所示。单击“环境变量”按钮,打开“环境变量”对话框,如图 1-5 所示。



图 1-3 选择“属性”命令



图 1-4 “系统属性”对话框

(1) 设置 JAVA_HOME 变量。在系统变量(或者用户变量)中新增一个 JAVA_HOME 变量,变量值为 C:\Program Files\Java\jdk-11.0.1(JDK 的安装路径),如图 1-6 所示。

(2) 编辑 Path 变量。在“系统变量”下选中 Path 变量,单击“编辑”按钮,在“变量值”的最后先使用“;”将之前的路径分隔开,然后添加 %JAVA_HOME%/bin,如图 1-7 所示。

注意:在 Windows 7 和 Windows 10 系统中插入新变量的方法不太一样,但是只要最终输入相符即可。

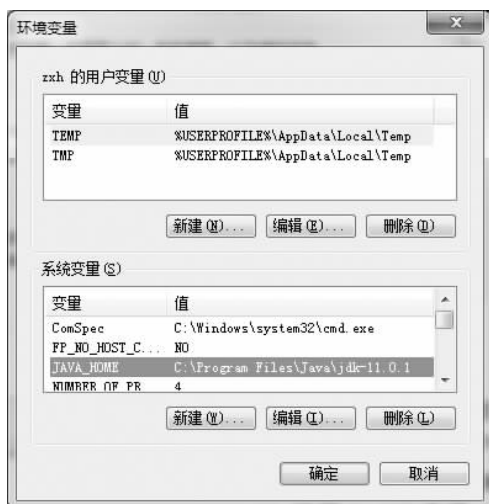


图 1-5 “环境变量”对话框



图 1-6 设置 JAVA_HOME 变量

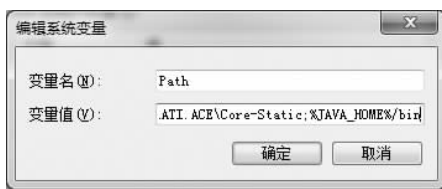


图 1-7 设置 Path 变量

(3) 测试环境变量配置是否成功。单击“开始”菜单，在“搜索程序和文件”输入框中输入 cmd，然后单击搜索出的 cmd.exe。在弹出的命令行窗口中执行 javac 命令(编译 Java 源程序的命令)，如果出现如图 1-8 所示的内容，则表示 JDK 环境变量配置成功。

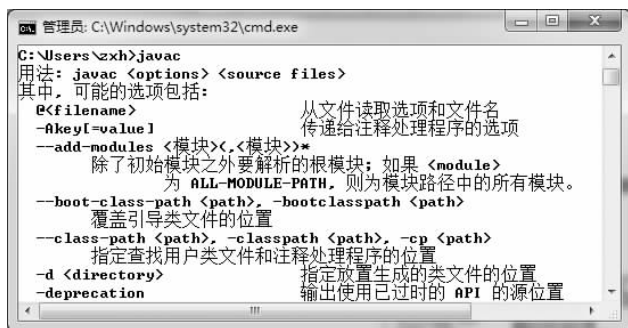


图 1-8 JDK 环境变量配置成功

提示：① 设置 JAVA_HOME 变量的目的是方便引用，比如，JDK 安装在 C:\Program Files\Java\jdk-11.0.1 文件夹中，设置 JAVA_HOME 为该路径后，要使用这个路径时，只须输入 %JAVA_HOME% 即可，避免每次引用都输入很长的路径字符串；② 设置 Path 变量的目的是在任何路径下都可以用 java 命令来执行程序。

1.4.3 使用 JDK 和文本编辑器编写 Java 程序

Java 程序分为两种类型：一种是 Java Application 程序,也就是 Java 应用程序,以 main()方法为程序入口;另一种是 Java Applet 程序,也称为 Java 小程序。本书主要讲解 Java Application 程序,Java Applet 程序主要应用于网页编程上,现在基本已经很少使用了,所以本书不再介绍。下面以输出“Hello World!”的程序作为第一个 Java 程序示例。开发步骤如下。

1. 编辑 Java 源程序

打开 Windows 的记事本程序(或其他任何纯文本编辑器,注意不要使用 Word 等带格式的编辑器),在记事本中输入源程序代码,保存为 HelloWorld.java 文件,保存到 D:\MyJavaFile 文件夹中。代码如下:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

注意: ①用记事本编写的源程序的文件名必须和类名完全相同(包括大小写);
②Java 严格区分字母的大小写,注意 String 和 System 的第一个字母是大写的。

程序说明如下。

(1) public 和 class 是 Java 语言中的关键字,public 表示公开的,class 是定义类的关键字,Java 程序就是由若干类(还有接口等)构成的。关于类的定义以后会有详细的介绍,HelloWorld 是我们自己起的类的名字。

(2) public static void main(String[] args)是程序的主方法,格式是固定的,是程序的入口,程序从主方法的第 1 行开始执行。

(3) System.out.println("Hello World!");是输出语句,要输出的文本内容放在圆括号里的一对双引号中。System.out 代表标准输出设备对象,默认是显示器,println()是 System.out 对象的行输出方法,输出内容后换行(与它不同的还有一个 print()方法输出内容后不换行)。

2. 编译 Java 源程序

在命令行窗口中输入如图 1-9 所示的命令即可。

说明: 第 1 行 D:是切换到磁盘 D;第 2 行 cd MyJavaFile 是进入 MyJavaFile 文件夹,其中,cd 命令是进入文件夹的 DOS 命令;第 3 行 javac HelloWorld.java 中,javac 是 JDK 的编译命令,HelloWorld.java 是 Java 源程序文件名;第 4 行出现命令提示符,表示编译成功。编译成功后会在 D:\MyJavaFile 目录下生成一个 HelloWorld.class 类文件,也就是字节码文件,它是可以在虚拟机上运行的二进制文件。

3. 运行程序

输入 java HelloWorld 命令运行程序,结果如图 1-9 所示。



图 1-9 HelloWorld 程序运行结果

1.5 Eclipse 集成开发环境

1.5.1 Eclipse 简介

Eclipse 是一个基于 Java、开放源码、可扩展的应用开发平台,它为编程人员提供了很好的 Java 集成开发环境。Eclipse 附带了一个标准的插件集,包括 Java 开发工具(Java Development Kit,JDK)。Eclipse 最初是由 IBM 公司开发的用于替代商业软件 Visual Age for Java 的下一代 IDE 开发环境,2001 年 11 月贡献给开源社区,现在它由非营利的软件供应商联盟 Eclipse 基金会(Eclipse Foundation)管理。

可以从 <http://www.eclipse.org/downloads/eclipse-packages/> 免费下载 Eclipse 软件,下载时选择 Eclipse IDE for Java Developers,再根据你的操作系统选择 32 位或 64 位版本。

Eclipse 软件包下载完成后,直接进行解压缩即可。双击 Eclipse 解压缩文件夹下的 eclipse.exe 文件就可以启动 Eclipse。在弹出的设置工作空间(Workspace)的对话框中设置工作空间的位置为 D:\MyJavaFile,如图 1-10 所示。这样,之后在 Eclipse 中建立的项目就会放到 D:\MyJavaFile 文件夹中。

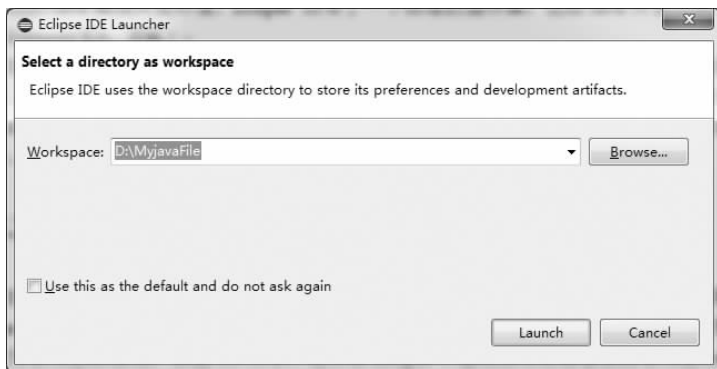


图 1-10 设置工作空间

单击 Launch 按钮,进入欢迎界面,如图 1-11 所示。直接单击窗口中 Welcome 标签页的×按钮,关闭欢迎界面,即可进入开发窗口。



图 1-11 欢迎界面

1.5.2 使用 Eclipse 开发 Java 程序

使用 Eclipse 开发 Java 程序的一般步骤如下。

(1) 启动 Eclipse 后,选择 File→New→Java Project 命令,弹出新建 Java 项目的对话框,如图 1-12 所示。在 Project name (项目名称)文本框中输入项目的名称,这里输入

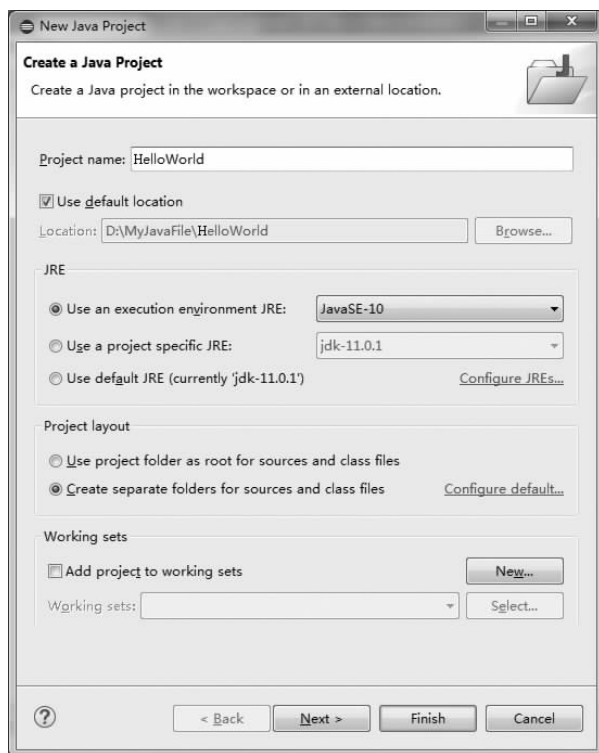


图 1-12 新建 Java 项目对话框

HelloWorld。注意下方项目的保存位置,Eclipse 默认会把项目保存在之前设置的工作空间 D:\MyJavaFile 文件夹下的 HelloWorld 文件夹中。

(2) 单击 Next 按钮,弹出配置 Java 的对话框,清除 Create module-info.java file 复选框,如图 1-13 所示。然后单击 Finish 按钮,完成项目的创建。



图 1-13 配置 Java 的对话框

(3) 单击左侧窗格中 HelloWorld 项目左边的三角形图标,然后在出现的 src 上右击,在弹出的快捷菜单中选择 New→Class 命令,如图 1-14 所示,弹出新建 Java 类的对话框。在 Name(名称)文本框中输入 HelloWorld,选中 public static void main(String [] args) 复选框。然后单击 Finish 按钮,如图 1-15 所示。

(4) 在程序窗格中的 HelloWorld.java 文件中输入程序代码,如图 1-16 所示。

(5) 单击工具栏中的运行按钮 (或右击左侧窗格中的项目名 HelloWorld,在弹出菜单中选择 Run As→Java Application 命令),程序输出会显示在控制台窗格中,如图 1-16 所示。

Eclipse 默认是自动编译的,所以不用自己编译源程序。如果源程序中有语法错误,错误处会出现波浪线(同时错误语句所在行的行号左面会出现×图标)。例如,将 System