

普通高等教育“十二五”规划教材

Java语言 程序设计

JAVA YUYAN CHENGXU SHEJI

孙莉娜 主编

清华大学出版社

普通高等职业教育“十二五”规划教材

Java语言 程序设计

JAVA YUYAN CHENGXU SHEJI

孙莉娜 主 编
胡国柱 谭营军 赵亚娟 副主编

清华大学出版社
北 京

内 容 简 介

本书针对初学和自学读者的特点,以通俗易懂的语言介绍 Java 开发技术。全书内容分为 4 个学习领域 12 个学习任务,全面讲解 Java 语言开发技术的各个知识点,包括 Java 语言的发展、环境配置、基本语法、程序流程控制、字符串处理、数组、面向对象、异常处理、包和接口、图形界面设计、多线程编程、输入/输出、网络编程、数据库等内容。每个学习任务通过项目训练贯穿各个知识点,以便读者更好地体会 Java 语言编程开发方法与应用技巧。

本书内容结构安排合理,实例简明易懂,适合作为高等职业院校教材使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。
版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 语言程序设计/孙莉娜主编. --北京:清华大学出版社,2015
(普通高等职业教育“十二五”规划教材)
ISBN 978-7-302-39119-7

I. ①J… II. ①孙… III. ①JAVA 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 017683 号

责任编辑:刘志彬
封面设计:汉风唐韵
责任校对:王凤芝
责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:三河市海新印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.25 字 数:431 千字

版 次:2015 年 2 月第 1 版 印 次:2015 年 2 月第 1 次印刷

印 数:1~3000

定 价:39.80 元

产品编号:063123-01

Java 是 Sun 公司推出的能够跨平台、可移植性高的一种面向对象的编程语言,也是目前安全先进、特征丰富、功能强大的计算机程序设计语言之一。Java 自从面世以来,一直以易学易用、功能强大的特点而得到广泛应用。Java 程序可以运行在任何一个系统平台上,甚至在手持电话、商务助理等电子产品中都可以运行,真正做到了跨平台使用。利用 Java 语言可以编写桌面应用程序、Web 应用程序、分布式系统以及嵌入式系统应用程序等,这使得其成为应用最广泛的开发语言之一。

从 Java 程序开发初学者晋级到编程高手通常需要经历 3 个阶段。本书的内容就是按照这一规律精心编写的,书中的内容分为 4 个部分。

第 1 部分:Java 语言概述。此部分包含了 Java 基础、Java IDE 开发工具、Java 语法基础、流程控制语句和数组等内容。书中结合大量的图标、举例、技巧等使读者快速掌握 Java 开发工具和 Java 语言,为以后进行 Java 编程打下坚实的基础。

第 2 部分:面向对象编程。此部分介绍了有关 Java 面向对象的知识,包括面向对象、类的继承与多态特征、包与接口应用、异常处理等。书中使用大量的实例和图解,详细讲解了面向对象的程序设计方法和面向对象的各种特征。通过此部分的学习,读者应该能够完全掌握面向对象的继承、封装、多态特性以及方法的重写和重载技术等。

第 3 部分:图形界面设计。此部分介绍了使用 Java 语言进行图形界面编程的各种关键技术,包括 Swing 程序设计基础、GUI 事件处理、使用线程实现多任务等。使用 Swing 技术开发桌面应用程序,增加了程序的交互性,使读者更容易学习和理解。

第 4 部分:编程技术应用。此部分介绍了编程常用类、数据的输入输出处理、数据库编程、网络程序设计等。学习完此部分内容后,读者能够开发数据库、网络等领域的小型应用程序。

每部分内容都引入学习任务,这些任务是由作者精心挑选的、涵盖了各个知识点的项目。通过项目,读者可以巩固前面所学的知识和技术,积累项目开发经验。

【本书特点】

1. 通俗易懂,而且对图示、代码几乎都加了注释,以帮助读者降低理解难度,快速上手。

2. 提供大量的实战模块、实战案例、实战练习,以增强读者动手能力,激发学习兴趣,读者可仔细研究这些模块,并亲自动手调试。

3. 本书同时提供电子教案和课件,为教师授课和学生学习提供方便。

本书由孙莉娜(辽宁机电职业技术学院)、胡国柱(辽宁机电职业技术学院)、赵亚娟(永城职业学院)、谭营军(河南职业技术学院)等共同编写。孙莉娜负责编写学习领域2的各个任务,学习领域4的任务4.3,以及附录1、附录2;胡国柱负责编写学习领域1的各个任务;谭营军负责编写学习领域3的各个任务;赵亚娟负责编写学习领域4中的任务4.1和任务4.2。

由于作者水平有限,书中难免会有纰漏之处,敬请广大读者批评指正。

编者

Contents 目 录

学习领域 1:Java 语言概述

任务 1.1 安装 Java 开发环境	3
1.1.1 任务内容	3
1.1.2 相关知识	3
1.1.3 任务实施	13
1.1.4 技能提高	37
任务 1.2 Java 基础程序设计	39
1.2.1 任务内容	39
1.2.2 相关知识	39
1.2.3 任务实施	59
1.2.4 技能提高	60
任务 1.3 数组和字符串程序设计	61
1.3.1 任务内容	61
1.3.2 相关知识	61
1.3.3 任务实施	73
1.3.4 技能提高	75

学习领域 2:面向对象编程

任务 2.1 面向对象编程技术初步	81
2.1.1 任务内容	81
2.1.2 相关知识	81
2.1.3 任务实施	98
2.1.4 技能提高	102
任务 2.2 面向对象编程技术进阶	104
2.2.1 任务内容	104
2.2.2 相关知识	105

2.2.3	任务实施	122
2.2.4	技能提高	128
任务 2.3	异常处理	132
2.3.1	任务内容	133
2.3.2	相关知识	133
2.3.3	任务实施	142
2.3.4	技能提高	143

学习领域 3:图形界面设计

任务 3.1	Swing 程序设计	149
3.1.1	任务内容	149
3.1.2	相关知识	149
3.1.3	任务实施	177
3.1.4	技能提高	179
任务 3.2	事件处理	185
3.2.1	任务内容	185
3.2.2	相关知识	185
3.2.3	任务实施	192
3.2.4	技能提高	194
任务 3.3	多线程	198
3.3.1	任务内容	198
3.3.2	相关知识	198
3.3.3	任务实施	204
3.3.4	技能提高	207

学习领域 4:编程技术应用

任务 4.1	输入输出处理	217
4.1.1	任务内容	217
4.1.2	相关知识	217
4.1.3	任务实施	234
4.1.4	技能提高	236
任务 4.2	数据库编程	237

4.2.1 任务内容.....	238
4.2.2 相关知识.....	238
4.2.3 任务实施.....	252
4.2.4 技能提高.....	257
任务 4.3 网络编程	262
4.3.1 任务内容.....	263
4.3.2 相关知识.....	263
4.3.3 任务实施.....	278
4.3.4 技能提高.....	282
附录 1 使用 Javadoc 工具制作开发文档	285
附录 2 Java 编程风格简述	293
参考文献	299

Java语言概述

任务 1.1 安装 Java 开发环境

本部分介绍 Java 语言的发展历程、技术特点和 Java 体系结构等,使读者对 Java 有一个基本的认识。随后配置 Java 开发环境,使初学者学会能够以较高的效率开发 Java 应用程序。

► 1.1.1 任务内容

Java 开发环境的搭建:安装 JDK 1.7 和集成开发环境 Eclipse,开发执行第一个 Hello World 程序。执行过程如下。

- (1)JDK 的安装与配置。
- (2)Java 开发环境的配置。
- (3)IDE 集成开发环境 Eclipse 的安装与配置。
- (4)开发执行 Java Application 应用程序和 Java Applet 小应用程序。

► 1.1.2 相关知识

| 知识点一:Java 简介 |

1991 年,Sun 公司的 James Gosling 等人为了解决消费类电子产品的微处理器计算问题,开发出一种名为“Oak”(中文译为“橡树”)的与平台无关的语言,它就是 Java 语言的前身。它用于控制嵌入在有线电视交换盒、PDA(Personal Digital Assistant,个人数字助理)、家用电器等的微处理器。

1993 年交互式电视和 PDA 市场用量开始滑坡,而 Internet 正处于发展时期,因此 Sun 公司将目标市场转向 Internet 应用程序。

1994 年 Sun 公司将 Oak 语言更名为 Java(Java 译为“爪哇”,是印度尼西亚一个盛产咖啡的岛屿),并于 1995 年正式推出它的第一个版本。Internet 的迅猛发展与 WWW(万维网)应用的快速增长,为 Java 的发展带来了契机。Java 语言优秀的跨平台特性使之非常适合于 Internet 编程,最初用 Java 语言编写的 Hot Java 浏览器和应用 Web 页面的 Applet 程序,使 Java 语言成为 Internet 上最受欢迎的开发语言。Sun 公司采取了“开放式”的合作政策,采用了颁发使用许可证的方式来允许各家公司把 Java 虚拟机(JVM)嵌入自己的操作系统或应用软件中,这吸引了大批公司加入到 Sun 联盟,如 IBM、HP、Netscape、Novell、Oracle、Apple 等公司;并且开发平台的源代码完全开放,这使得开发人员很容易只使用一种 Java 语言来实现网络各平台之间的开发、编程和应用,这也是 Java 语言得以迅猛发展的一个主要原因。现在全球有近 70% 的公司使用 Java 语言开发自己的计算机软件系统。

1995 年,Sun 公司发布了 Java 的第一个版本 Alpha 1.0a2 版本,开发出 Hot Java 浏览器。1996 年,Sun 公司发布了 Java 的第一个开发包 JDK v1.0。1997 年,Sun 公司发布了

Java 开发包 JDK v1.1。1998 年, Sun 公司发布了 Java 开发包 JDK v1.2(称为 Java 2)。

1999 年, Sun 公司重新组织了 Java 平台的集成方法, 并将企业级应用平台作为公司今后发展的方向。现在的 Java 开发平台的编程构架一共有三种: J2SE、J2EE、J2ME。

J2SE (Java 2 Platform, Standard Edition), 即 Java 2 平台标准版, 包含构成 Java 语言核心的类。它是用于工作站、PC 的开发平台, 也是三个编程构架中最基本的一个构架, J2EE 和 J2ME 就是在 J2SE 的基础上发展、转化而来的。

J2EE (Java 2 Platform, Enterprise Edition), 即 Java 2 平台企业版。J2EE 应用于可扩展的、基于分布式计算的企业级开发平台, 如 Intranet(企业内部网), 有业界大量的其他软件技术融入 J2EE 构架中(如微软的 XML 技术), 因此它具有更高的可扩展性、灵活性和集成性。

J2ME (Java 2 Platform, Micro Edition), 即 Java 2 平台微型版。J2ME 用于嵌入式开发, 为消费类电子产品软件开发服务, 如手机系统或手机游戏软件的开发。

三种版本使用的类库也不尽相同, 本书内容的大部分类都来自于 J2SE 构架。

| 知识点二: Java 语言的特点 |

1. 简单性与分布式

Java 作为一种高级编程语言, 在语法规则上和 C++ 类似, C++ 程序员会比较容易地掌握 Java 编程技术。Java 摒弃了 C++ 容易引起错误的内容, 如指针操作和内存管理, 使程序设计变得简单、高效。

Java 是面向网络的编程语言, 它提供了基于网络协议(如 TCP/IP)的类库。使用这些类, Java 应用程序可以很容易地访问网络上的资源。Java 应用程序可通过一个特定的 URL 对象打开并访问网络资源, 就像访问本地文件系统那样简单。

2. 纯面向对象

面向对象编程技术具有很多优点, 比如通过对象的封装, 减少了对数据非法操作的风险, 使数据更加安全; 通过类的继承, 实现了代码的重用, 提高了编程效率等。Java 语言的编程主要集中在类、接口的描述和对象引用方面。面向对象编程技术适合开发大型的、复杂的应用程序, 且程序代码易于理解和维护, 是编程发展的一个趋势。

3. 健壮性与安全性

Java 语言在编译和运行时具有健壮性, 可以消除程序错误带来的影响。Java 语言提供了较完备的异常处理机制, 在编译和运行程序时, 系统对代码进行逐级检查, 指出可能产生错误的地方, 要求必须对可能存在错误的代码进行必要的处理, 以消除因产生错误而造成系统崩溃的情况。它提供自动垃圾收集功能来进行内存管理, 防止出现程序员在管理内存时容易产生的错误, 这些措施都保证了 Java 系统运行的可靠性。

作为网络编程语言, 安全是至关重要的。一方面, 在语言功能上, 由于 Java 不支持指

针,消除了指针操作带来的安全隐患;另一方面,Java 具有完备的安全结构和策略,代码在编译和运行过程中,被逐级检查,可以防止恶意程序和病毒的攻击,如编译器会查找出代码错误,编译之后生成字节码,通过字节码校验器,病毒代码将无所遁形,因此也有人称 Java 语言为“永远不会感染病毒的语言”;在加载类的时候,还会通过类加载器进行校验。

4. 平台独立与可移植性

互联网是由各种各样的计算机平台构成的,如果要保证应用程序在网络中任何计算机上都能正常运行,必须使程序具有平台无关性,即软件本身不受计算机硬件和操作系统的限制。Java 是一种“与平台无关”的编程语言,Java 的源文件是与平台无关的纯文本,而 Java 源文件通过编译后生成的类文件(即字节码文件)通过 Java 虚拟机(JVM)可以在不同的平台上运行,与具体机器指令无关。Java 的基本数据类型在设计上不依赖于具体硬件,为程序的移植提供了方便。

5. 解释执行

Java 是一种先编译后解释执行的编程语言,Java 源程序经过编译后生成被称作字节码(Byte Code)的二进制文件,JVM 的解释器解释执行字节码文件。解释器在执行字节码文件时,能对代码进行安全检查,以保证没有被修改的代码才能执行,提高了系统的安全性。另外,JVM 由 Sun 公司特别制作并在网上实时更新,它的运行效率远高于一般的解释性语言的解释器。

6. 多线程与动态性

多线程机制使程序代码能够并行执行,充分发挥了 CPU 的运行效率。程序设计者可以用不同的线程完成不同的子功能,极大地扩展了 Java 语言的功能。支持多线程机制是现今网络开发语言的基本特性之一。

Java 在设计上力求适合不断发展的环境。在类库中可以自由地加入新的方法和实例而不会影响用户程序的执行。Java 通过接口来支持多重继承,使之比严格的类继承具有更灵活的方式且易于扩展。Java 的类库是开放的,所有的程序员可以根据需要自行定义类库。

| 知识点三:Java 平台特性 |

1. 了解 Java 的跨平台特性

什么是平台呢?简单的理解是计算机软件系统与计算机硬件的结合体。比如 IBM PC Windows 机、Applet 公司的 Mac OS 等。我们知道,不同种类的计算机有不同的机器语言(内码),为一种平台编写的代码不能在另一种平台上运行,这是因为它们的内码不同。

编程语言分为三类:机器语言、汇编语言、高级语言。

Java 语言是一种“先编译后解释”的高级语言,它的地位等同于 C、C++ 或 Visual Basic

等语言。从功能上来看,Java 也可以实现 C、C++ 或 Visual Basic 等语言的大部分功能,如控制台(Console)程序和 GUI(Graphics User Interface,图形用户界面)程序,只不过侧重点有所不同。Java 语言是基于 Web 开发的一种高级语言,它的“强项”在网络上!

下面我们介绍一下 Java 的编译和运行过程,如图 1-1 所示。

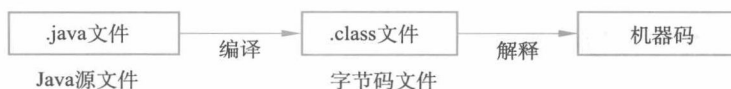


图 1-1 Java 编译与运行过程

Java 语言通过编译器在本地将源程序(扩展名为 .java)文件编译成字节码文件(扩展名为 .class),可以通过复制或通过网络传送到目的平台,然后通过目的平台的解释器(也可能是浏览器的解释器)来解释执行。

那么 Java 又是如何完成跨平台的呢?如图 1-2 所示,Java 在运行过程的中间环节引入了解释器来帮助它完成跨平台。

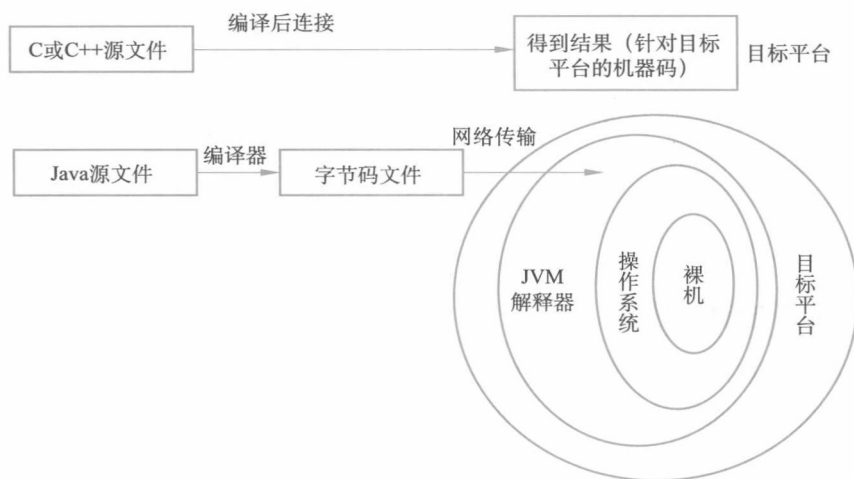


图 1-2 Java 语言的运行过程

下面介绍一个非常重要的概念——Java 虚拟机(Java Virtual Machine,JVM)。

JVM 是在计算机内部模拟运行的假想的计算机。它可以是硬件或软件(通常为软件)。它有自己独立的指令集系统(相当于计算机的 CPU、内存和寄存器等设备)。JVM 负责将 Java 字节码程序翻译成机器码,然后由计算机执行。

JVM 的主要功能为:加载 .class 文件,管理内存,执行垃圾收集。

在计算机执行 Java 程序时,需要 JVM 和核心类库的支持。Java 采用的方法是:在操作系统和硬件平台上模拟一种抽象化的计算机系统运行时环境(Java Runtime Environment,JRE),而 JRE 包含了 JVM 和运行程序所需系统核心类库。JVM 和 JRE 是随着 JDK(Java Development Kit,Java 开发工具包)的安装而建立起来的。

对 Java 语言而言,它的源文件和字节码文件(中间码文件)都是与平台无关的,它们可

以通过网络传输到任何一个网络平台中并可以被识别。然后通过目标平台本地的 JVM 解释执行。但要注意:JVM 是与平台相关的。因为字节码是通过网络传输到目标计算机平台上再通过 JVM 运行的,而不同种类的计算机有不同的内码,从这里就可以推断出,每一个特定平台上应该有一个特定的 JVM,即 JVM 是与平台相关的。

2. Java 程序的运行时环境

前面介绍了 Java 虚拟机(JVM)的概念,JVM 的核心是解释器。而程序运行时需要的是 JRE,可以简单地把 JRE 理解成工作在操作系统之上的一个小型操作系统,它包含了运行在其上的 JVM 以及本地平台的核心类库,如图 1-3 所示。

下面详细描述 JRE 中各部件的作用。

(1) 类加载器(Class Loader):用来加载 Class 文件的部件,同时针对跨网络的类,进行安全性检查。

(2) 字节码校验器(Byte Code Verifier):基于代码规范,对语法语义、对象类型转换和权限安全性访问进行检查。

(3) 解释器(Interpreter):JVM 的核心部件,把字节码指令映射到本地平台的库和指令上,使之得以执行。

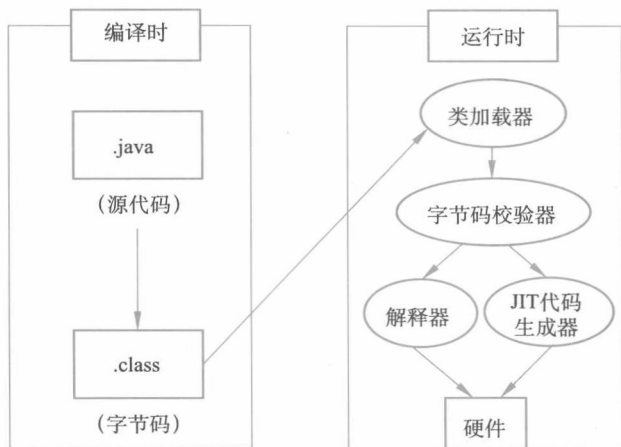


图 1-3 Java 运行时环境

(4) JIT 代码生成器(Just In Time):即时代码生成器(即时代码编译器)是另外一种解释执行字节码的方法。通常的解释器是逐行解释和运行,而编译器是对代码做全体编译后再连接执行,因此解释型语言的执行效率一般都低于编译型语言。而为了提高运行效率,Java 提供了 JIT 运行方式,它可以一次性地解释完所有代码,再去运行机器码,而且曾经解释过的代码会进入缓存,如果下次再调用这部分代码,就从缓存中取出,这样就极大地提高了 Java 的运行效率。因为这种解释运行的方式类似于编译器,因此也称其为“JIT 即时编译器”。JIT 就类似于引擎对于一辆赛车的意义,是 JRE 的核心部件。

(5) API 类库:实现标准 Java 平台 API 的代码。

(6) 硬件本地平台接口:提供对底层系统平台资源库的调用接口。

3. 垃圾收集器

许多计算机语言都有在程序运行时动态分配存储空间的功能,当这部分内存空间不再使用的时候,程序应停止分配内存空间并回收它们。但是回收内存空间却不是件容易的事情,C和C++语言通常需要程序员自行编写代码回收动态内存空间,这增加了程序员的负担,还会因为代码不健壮造成系统问题。

Java 语言提供了一个自动进行内存管理的工具,就是垃圾收集器(Garbage Collector)。它是一个系统级的线程,专用于对内存进行跟踪和回收。但因为垃圾收集器是一个优先级比较低的后台线程(Daemon Thread),所以它只在系统有空闲的时候才会回收垃圾内存,而且也无法判断垃圾收集器何时回收内存,以及要运行多长时间,这一切都是自动完成的。这使得程序在运行时会出现不连贯的现象,在一定程度上降低了代码的运行效率,但这个代价还是值得付出的。

知识点四:JDK 目录结构

1. 什么是 JDK

JDK(Java Development Kit)即 Java 开发工具包,是有助于程序员开发 Java 程序的工具包,其中包括类库、编译器、调试器、Java 运行时环境(JRE)。

Sun 公司为各种主流平台(如 Windows、Solaris、Macintosh 等)制作了 JDK,可以从网址 <http://Java.sun.com/products/> 下载 JDK。如下载的文件:j2sdk-1_6_0-beta-windows-i586.exe,表示此 JDK 为 Java 2 标准版,版本号为 1.6(内部版本号为 6.0),beta 表示测试版,适用于 Windows 系统的计算机。

2. JDK 的目录结构

下载并安装完 JDK 后,假设安装的目录为:C:\Program Files\Java\j2sdk1.6.0。在 \Java\j2sdk1.6.0 目录下有以下文件及子目录。

根目录包括版权、许可和 Readme 文件,还有构成 Java 核心 API 的所有类文件的归档文件 src.jar。

bin 目录是 JDK 开发工具的可执行文件,包括编译器、解释器、调试器等。

demo 目录中有丰富的演示程序源代码。

include 目录支持 Java 本地机接口(JNI)和 Java 虚拟机调试程序接口的 C 语言头文件。

JRE(Java Runtime Environment),即 Java 运行时环境,包含 JVM、运行时的类包和 Java 链接启动器,但是不包含编译器和调试器。jre 目录包含的部分子目录及文件如下。

lib/jaw.jar:提供了 Netscape 的 JavaScript 和 Security 类。

lib/charsets.jar:字符转换类。

lib/rt.jar:Java 基本类库(JFC)。

lib/ext/: 包含扩展的 jar 文件。

bin/keytool: 密码认证和管理工具。

另外, \jre\bin 目录下, 还包括 Java Web Start(JWS) 的运行程序 javaws.exe。

lib 目录包含开发工具使用类的归档文件。其中有:

tools.jar: 包含支持 JDK 的工具和实用程序的非核心类。

dt.jar: 是 Swing 组件类 BeanInfo 文件的 DesignTime 归档。

sample 目录中有一些 JNLP(Java 网络装载协议) 应用的样例程序。JNLP 能使远程客户访问和运行那些位于本地机器的应用程序。JNLP 应用程序的优点是它可以在网络上实现自安装和自更新。

| 知识点五: Java 程序开发 |

Java 的基础应用中包含两种类型的应用, 一种是 Java Application, 称为 Java 应用程序; 另一种是 Java Applet, 称为 Java 小应用程序。

Java Application 以 main() 方法作为程序入口, 由 Java 解释器解释执行, 用以实现控制台或 GUI 方面的应用。

Java Applet 没有 main() 方法, 但是有 init() 和 paint() 等方法, 由浏览器解释执行, 主要用于在网页上显示动画等功能。对此本书只做简单介绍, 因为 Applet 已基本不再使用。

1. Java Application 应用程序开发

【例 1.1】 编写一个在屏幕上显示“Hello World!”字符界面的应用程序。

```
public class HelloApp{                                // 定义公共类 HelloApp
    public static void main(String[] args){           // 应用程序入口即 main() 方法
        System.out.println("Hello World!");          // 输出 Hello World!
    }
}
```

例 1.1 是最简单的 Java 应用程序, 它给出了 Java 应用程序的基本框架。

注意以下几点内容。

(1) 类是构成 Java 程序的主体, class 是类的说明符号, 类中包含了实现具体操作的方法。

(2) 使用“//”声明的部分是 Java 的注释, 它有助于程序的阅读, 在编译时不会被编译。

(3) 每个应用程序中必须包含主方法 main(), 主方法是程序的入口, 读者可先记住声明格式, 不要改变关键字顺序。

(4) System.out.println() 方法起到输出作用, 直接输出括号内的内容。

(5) 读者在编写以上程序时要注意字母大小写, Java 语言中严格区分大小写。