

普通高等教育“十三五”规划教材

Java 程序设计教程

Java CHENGXU SHEJI JIAOCHENG

何受倩 曾 昊 主编

陈方昕 陆晓东 邹 月 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

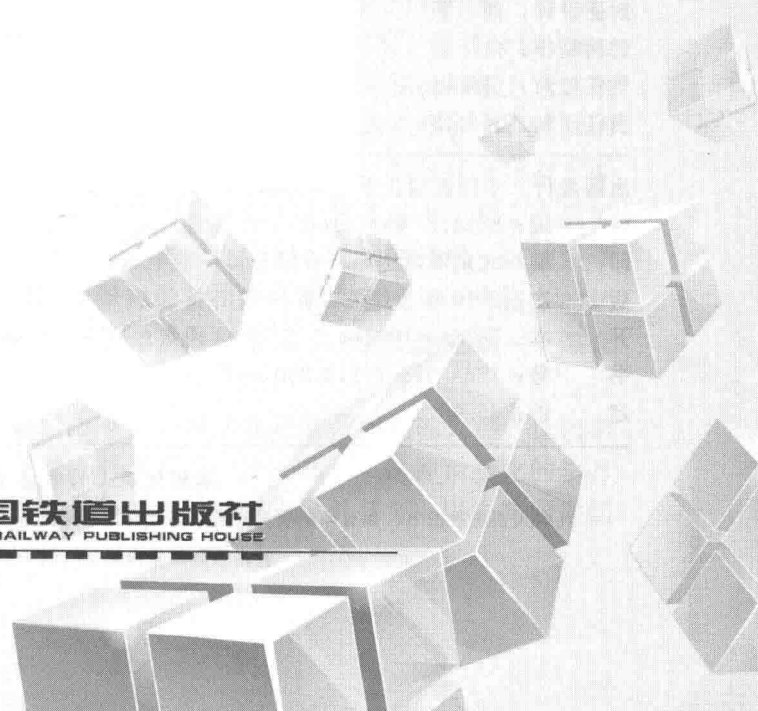
普通高等教育“十三五”规划教材

Java 程序设计教程

何受倩 曾 昊 主 编

陈方昕 陆晓东 邹 月 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

A decorative graphic at the bottom of the page featuring several 3D cubes of varying sizes and orientations, some of which are stacked or arranged in a way that suggests a larger structure or a complex geometric design. The cubes are rendered in a light gray color with black outlines, giving them a three-dimensional appearance.

内 容 简 介

本书以项目为驱动,将项目分解成多个任务,以任务描述的形式引入问题进而解决问题。全书共分15个项目,主要内容包括:Java概述及开发环境搭建,Java语言编程基础,数组与方法,Teacher类与对象的使用,类的继承与多态,抽象类、接口和包,异常捕获,Java中I/O的应用,图形用户界面编程,多线程,Java网络编程,用Java集合来实现学生信息的管理,使用JDBC实现超市进销存管理,API帮助文档的使用及MyEclipse调试入门。

本书由易到难,循序渐进,内容全面,知识点详尽,适合作为普通高等院校计算机类专业的基础教材,也可作为使用Java语言相关工作人员及初学者的自学参考书。

图书在版编目(CIP)数据

Java 程序设计教程 / 何受倩, 曾昊主编. —北京:
中国铁道出版社, 2016.8
普通高等教育“十三五”规划教材
ISBN 978-7-113-22016-7
I. ①J… II. ①何… ②曾… III. ①JAVA 语言—程序
设计—高等学校—教材 IV. ①TP312.8
中国版本图书馆 CIP 数据核字 (2016) 第 203650 号

书 名: Java 程序设计教程
作 者: 何受倩 曾 昊 主编

策 划: 王占清 韩从付
责任编辑: 周 欣 周海燕
编辑助理: 祝和谊
封面设计: 刘 颖
封面制作: 白 雪
责任校对: 汤淑梅
责任印制: 郭向伟

读者热线: (010) 63550836

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街8号)
网 址: <http://www.51eds.com>
印 刷: 北京明恒达印务有限公司
版 次: 2016年9月第1版 2016年9月第1次印刷
开 本: 787mm×1092mm 1/16 印张: 19.25 字数: 456千
书 号: ISBN 978-7-113-22016-7
定 价: 45.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659



前

言



Java 语言是 TIOBE 编程语言排行榜长期排名第一位的编程语言，是目前最流行的语言之一，它在网络程序设计和应用领域已经取得了巨大的成功，同时也被广泛应用于电子商务、手机和嵌入式芯片领域。由于 Java 语言的开放性和跨平台分布式特性，使全球数以万计的 Java 开发公司可以得到相互兼容的产品。

Java 继承了 C++ 语言面向对象技术的核心，同时封装了 C 语言中容易引起错误的指针、以接口取代多重继承等特性，降低了程序员出错的风险，增加了垃圾回收器功能，用于回收不再被引用的对象所占据的内存空间，使得程序员不用再为内存管理而担忧。它通过将源代码编译成二进制字节码，然后依赖各种不同平台上的虚拟机来解释执行字节码，从而实现了“一次编译、到处运行”的跨平台特性。所以，尽管它在桌面应用程序的开发方面略显不尽如人意，但是在网络应用和移动嵌入应用方面，Java 平台可驾驭从智能卡、小型消费类器件到大型数据中心的各种应用。

Java 是一个纯粹的面向对象的程序设计语言，良好地体现了面向对象的设计思想，因此，使用 Java 作为基础教学语言的方式也越来越得到重视。

关于本书的说明

目前市面上有很多 Java 教材，我们也选用了部分教材内容作为参考，但发现部分内容过于理论化、不够吸引人，案例与案例之间比较松散，联系不紧密。编者从事了多年的 Java 程序教学，积累了一定的教学经验，觉得有必要写一本从易到难、循序渐进，既符合认知规律，又有方法论的教材。

本书是以项目为驱动的，并将一个项目分解成多个任务，以任务描述的形式引入问题，围绕解决任务描述中引入的问题而展开的。要解决问题，首先要具备一定的理论知识，这就是【必备知识】部分，知识本着够用的原则，精益求精，避免长篇大论。如有些相关的补充知识，在【知识链接】中阐述。介绍完要用到的相关知识点后，给出解决任务的【解题思路】，解题思路按照解决问题的步骤有条理地叙述。有了解题的思路后就可以编写程序代码，【任务透析】给出了源程序代码，部分任务还有【课堂提问】和【现场演练】环节。同时，每个项目后面还配有【思考练习】和【上机实训】并配有参考答案，上机实训中对实训目的、实训内容都有明确的要求，以加强学生对知识点的掌握和强化编程能力。采用这样层层紧扣的环节来完成每一个任务的学习，当学习完几个任务后，项目所要求掌握的知识点都掌握了，也就能完成一个综合性较强的小项目。

本书采用以项目为驱动、问题分解的方法，以到达简化复杂问题的目的，让初学者更易学习和掌握，相关联的几个任务完成后堆积成一个小项目，这种系统化的学习效果是比较理想的。同时，我们尽可能选择一些比较有代表性的任务，以提高同学们的学习兴趣。

本书共有 15 个项目，其中，项目一主要介绍 Java 开发平台的搭建；项目二、三是 Java 语言基础，主要介绍 Java 基本语法、程序等基本结构以及数组和方法；项目四、五、六主要介绍面向对象三大特征——继承、多态、封装以及抽象类与接口中的应用；项目七到项目十一是 Java 高级应用部分，主要有异常、Java I/O、图形用户界面编程、多线程和网络编程；项目十二、十三是类集与数据库编程，使用 JDBC 实现超市进销存管理；项目十四、十五介绍 API 帮助文档的使用和 MyEclipse 调试技巧。大多数教材并没有介绍 API 文档的使用和调试技巧，而事实上，掌握 API 帮助文档的使用和调试技巧对学生编程能力的提高是非常有帮助的，同时也能提高他们的自学能力和自我解决问题的能力。所谓“授之以鱼不如授之以渔”，在信息科技发展如此迅速的今天，软件的更新换代是非常快的，学习方法和自学能力尤为重要。

本书的重点是面向对象开发技术。软件开发人员除了要有良好的面向对象的程序设计思想，同时，还要养成规范的代码编写习惯，熟练掌握常用的编程工具，具有团队合作精神等。

本书由何受倩、曾昊任主编，陈方昕、陆晓东、邹月任副主编。其中：项目一、五、十、十一、十四由何受倩编写；项目二由黄静编写，项目三由邹月编写；项目四由陆晓东、韩娜、王丽艳编写；项目六、八、九由陈方昕编写；项目七、十三由曾昊编写；项目十二由严梅编写；项目十五由符志强、陆晓东和广州为学教育科技有限公司的黄勇工程师编写。本书的统稿定稿工作由钱英军、何受倩、谷灵康完成。

本书作为普通高等教育“十三五”规划教材，不仅适合作为高等院校、各类职业技术学院和各种 Java 技术培训班的教材，也适合没有任何编程经验的初学者使用。

致谢

在本书顺利出版之际，感谢我教过的所有学生，教他们学习 Java 的经历对于本书内容的选择和组织有很多帮助，感谢广东科贸职业学院信息工程系的孙继红、张雷、王磊、曾海峰老师提出的建议和给予的协助。另外，还要感谢广州光大教育科技有限公司的谭福民 Java 工程师，他对本书的撰写提了许多宝贵的意见。此外，本书还参考了许多作者的书籍和资料，在此一并表示深深的感谢。

意见反馈

尽管我们做了很大努力，但很难避免教材会有一些错漏，欢迎各界专家和读者朋友提出宝贵意见，我们将不胜感激。在阅读本书过程中，如发现任何问题或有不认同之处，欢迎与我们联系，联系邮箱：Lfbird 2000@126.com。

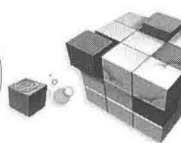
编 者

2016 年 5 月



目

录



项目一 Java 概述及开发环境搭建 1

任务一 认识 Java 1

必备知识 1

1. Java 的发展历史 1
2. Java 语言的特点 2
3. Java 程序的运行机制和 Java 虚拟机 3

任务二 Java 开发环境搭建 4

必备知识 4

1. Java 平台简介 4
2. JDK 的安装与配置 5

任务三 编写并运行第一个

Java 程序 7

必备知识 7

1. 什么是 Java 源程序 7
2. Java 中类的定义格式 7
3. 如何编译和运行一个 Java 程序 8

任务四 Java 与其他语言的比较 13

必备知识 13

1. Java 与 C# 比较 13
2. Java 与 PHP 比较 13

思考练习 13

上机实训（一） 14

项目二 Java 语言编程基础 15

任务一 数据类型与常量、变量 15

必备知识 15

1. 认识标识符 15
2. 认识关键字 16
3. 认识数据类型 16
4. 认识常量 16
5. 认识变量 18

任务二 运算符、表达式与语句 23

必备知识 23

1. 运算符 23
2. 运算符的优先级 27
3. 表达式 28
4. 语句 28

任务三 程序控制结构 31

必备知识 31

1. 程序的结构设计 31

2. 顺序结构 32

3. 选择结构 32

4. 循环结构 37

5. 循环跳转语句 43

思考练习 48

上机实训（二） 50

项目三 数组与方法 52

任务一 数组的定义及使用 52

必备知识 52

1. 一维数组的声明及使用 52
2. 二维数组的定义及使用 55
3. 字符数组 56
4. 字符串 57

任务二 方法的定义及使用 63

必备知识 63

1. 方法的定义 63
2. 方法的调用 63
3. 参数传递 65

任务三 数组的引用传递 70

必备知识 70

1. 数组参数 70
2. 示例——一维数组求最大值 70
3. 示例——二维数组作形参 71

思考练习 73

上机实训（三） 75

项目四 Teacher 类与对象的 创建与使用 77

任务一 Teacher 类的创建 77

必备知识 77

1. 面向对象与面向过程的区别 77
2. 类与对象的关系 77
3. 类的定义格式 78
4. 成员变量和局部变量的区别 78

任务二 调用构造方法创建

Teacher 类对象 80

必备知识 80

1. 对象的创建格式 80



2. 构造方法.....81	任务二 接口与接口的实现.....120
3. 对象的使用.....81	必备知识.....120
4. 对象的引用传递.....81	1. 什么是接口.....120
任务三 使用 setter() 和 getter()	2. 接口如何定义.....120
方法访问被封装属性.....85	3. 接口怎样继承.....121
必备知识.....85	4. 接口怎样实现.....121
1. 封装的概念.....85	任务三 包与访问权限.....124
2. 封装的目的.....85	必备知识.....124
3. setter() 和 getter() 方法.....85	1. 什么是包.....124
4. 访问控制权限.....85	2. 如何定义包.....124
思考练习.....89	3. 怎样设定权限.....124
上机实训(四).....93	思考练习.....127
项目五 类的继承与多态.....94	上机实训(六).....130
任务一 类的继承.....94	项目七 异常捕获.....131
必备知识.....94	任务一 除法计算工具——异常
1. 继承的概念.....94	捕获及处理.....132
2. 继承的作用.....94	必备知识.....133
3. 继承的实现.....95	1. 异常的概念.....133
任务二 进一步学习继承.....101	2. 异常处理的语法格式.....134
必备知识.....101	任务二 进制转换工具
1. 子类实例化过程.....101	——异常抛出.....137
2. 使用 setter 和 getter 方法	必备知识.....137
访问父类的私有属性.....101	1. 什么叫抛出异常.....137
任务三 类的多态.....104	2. 抛出异常的语法格式.....137
必备知识.....104	任务三 进制转换工具
1. 多态的概念.....104	——声明异常.....139
2. 多态的作用.....104	必备知识.....139
3. 多态的实现.....104	1. 什么叫声明异常.....139
任务四 进一步学习多态.....107	2. 声明异常的语法格式.....139
必备知识.....107	任务四 圆的计算工具
1. Java 多态存在的三个	——自定义异常.....141
必要条件.....109	必备知识.....141
2. 对象多态的两种类型.....109	1. 自定义异常.....141
3. 对象转型格式.....109	2. 如何自定义异常.....141
思考练习.....112	思考练习.....143
上机实训(五).....114	上机实训(七).....144
项目六 抽象类、接口和包.....116	项目八 Java 中 I/O 的应用.....146
任务一 抽象类和抽象方法.....116	任务一 标准输入/输出.....146
必备知识.....116	必备知识.....146
1. 什么是抽象方法.....116	1. 什么是输入/输出.....146
2. 抽象方法的声明.....117	2. System 类对 I/O 的支持.....146
3. 什么是抽象类.....117	
4. 抽象类的定义格式.....117	

任务二 File 类	149
必备知识	149
1. 流的分类	149
2. File 类	149
任务三 I/O 流的分类	152
必备知识	152
1. 字符流	152
2. 字节流	153
3. 节点流和过滤流	154
任务四 常用 I/O 流的应用	154
必备知识	154
1. Java.io 包中的常用的类	154
2. FileReader 类	155
3. FileWriter 类	155
思考练习	158
上机实训 (八)	159
项目九 图形用户界面编程	160
任务一 框架的设计	160
必备知识	160
1. 什么是图形界面程序设计	160
2. GUI 使用的组件	160
3. 容器类、组件类与辅助类	161
4. 框架 JFrame	162
任务二 按钮和文本区的设计布局	164
必备知识	164
1. 文本框	164
2. 几种常见的布局管理器	164
3. GridLayout 布局	165
任务三 事件响应及界面程序设计	167
必备知识	168
1. 什么是事件响应	168
2. 如何定义事件响应	168
思考练习	173
上机实训 (九)	175
项目十 多线程	176
任务一 用 Thread 类实现多线程	176
必备知识	176
1. 多线程的概念	176
2. 线程的生命周期	176
3. 实现多线程的两种方法	177
4. 继承 Thread 类来实现多线程	177

任务二 用 Runnable 接口实现多线程	179
必备知识	179
1. 利用接口 Runnable 来实现多线程	179
2. 线程的几个重要方法	180
任务三 线程的同步与死锁	182
必备知识	182
1. 多线程资源共享安全问题	182
2. 线程的同步操作	183
思考练习	187
上机实训 (十)	188
项目十一 Java 网络编程	190
任务一 网络编程 InetAddress 类的应用	190
必备知识	190
1. 计算机网络的几个术语	190
2. 网络编程概述	190
3. 网络通信方式	191
4. java.net 包	191
任务二 TCP 网络编程	192
必备知识	193
1. TCP 网络编程	193
2. TCP 客户端编程步骤	193
3. TCP 服务器端编程步骤	193
任务三 UDP 网络编程	197
必备知识	197
1. UDP 网络编程	197
2. UDP 客户端编程步骤	198
3. UDP 服务器端编程步骤	199
任务四 多线程与网络编程的综合应用	202
必备知识	202
1. 客户端和服务端程序功能划分	202
2. 多线程与网络编程的应用	202
思考练习	208
上机实训 (十一)	209
项目十二 用 Java 集合来实现学生信息的管理	211
任务一 集合类的选择	211
必备知识	211
1. Java 集合概述	211



2. 集合与数组的区别	211	2. PreparedStatement 对象的 常用方法	261
3. Java 集合体系结构	212	思考练习	263
任务二 采用 List 派生集合 管理有序数据	213	上机实训 (十三)	263
必备知识	214	项目十四 API 帮助文档的使用	265
1. List 接口简介	214	任务一 格式化日期 SimpleDate Format 类的使用	265
2. List 接口重要 API	214	必备知识	265
3. 常用派生类	215	1. Java API 中有哪些包	265
任务三 采用 Set 派生集合 管理无序数据	218	2. API 的结构及使用	266
必备知识	218	任务二 网络编程类 ServerSocket 和 Socket 的使用	269
1. Set 接口简介	218	必备知识	269
2. Set 接口重要 API	219	1. ServerSocket 类的使用	269
3. 常用派生类 HashSet 类	219	2. Socket 类的使用	270
任务四 采用 Map 派生集合 管理映射关系的数据	222	3. 输入/输出流的使用	270
必备知识	222	任务三 数据库编程类的使用	272
1. Map 接口简介	222	必备知识	273
2. Map 接口重要 API	223	1. Java.sql 包中的主要接口	273
3. 常用派生类 HashMap	223	2. 两种常用的数据库访问技术	273
思考练习	227	3. 数据库相关类的使用	273
上机实训 (十二)	228	小结	277
项目十三 使用 JDBC 实现超市 进销存管理	231	项目十五 MyEclipse 调试入门	278
任务一 JDBC 编程环境的搭建	231	任务一 调试阶乘程序代码	278
必备知识	231	必备知识	278
1. JDBC 简介	231	1. 断点的概念	278
2. JDBC 驱动程序	231	2. 断点的分类	279
3. DriverManager 类和 Connection 类	232	任务二 求水仙花数问题	285
4. 常用的数据库介绍	232	必备知识	285
5. 常用的 SQL 语句举例	234	1. 水仙花数的定义	285
任务二 采用 Statement 完成数据 库的增、删、改、查	240	2. 求水仙花数的算法	285
必备知识	240	任务三 分解质因数问题	289
1. Statement 类	240	必备知识	289
2. ResultSet 结果集	241	1. 分解质因数	289
3. JDBC 编程一般步骤	242	2. 分解质因数的算法	289
任务三 采用 PreparedStatement 完成数据库的增删 改查	261	小结	292
必备知识	261	附录 思考练习参考答案	293
1. PreparedStatement 简介	261	参考文献	300

Java 概述及开发环境搭建 <<<

项目描述

下载并配置 JDK 开发工具，利用记事本编写第一个 Java 程序，程序运行输出“Hello, Java!”。

项目分解

本项目可分解为以下几个任务：

- 认识 Java；
- Java 开发环境搭建；
- 编写并运行第一个 Java 程序；
- Java 与其他语言的比较。

任务一 认识 Java

任务描述

了解 Java 的发展历史；理解 Java 语言的特点以及 Java 程序的运行机制和 Java 虚拟机。

必备知识

1. Java 的发展历史

Java 是 SUN 公司开发出来的一套编程语言，主设计者是 James Gosling。它最早来源于一个叫 Green 的项目，起初打算采用 C++ 进行开发，可是发现 C++ 不能胜任此项工作。由于 C++ 在内存管理是可直接访问地址的，会使系统出现一些问题，所以 SUN 公司的工程师在 C++ 的基础之上开发了一个新的平台，称为 Oak（Java 的前身）。然后，他们又对 Oak 进行了小规模改造，就这样，Java 在 1995 年诞生了。

Java 的发展历程如下：

1996 年 1 月，第一个 JDK——JDK1.0 诞生。



1999 年 6 月, SUN 公司发布 Java 的三个版本: 标准版 (J2EE)、企业版 (J2EE) 和微型版 (J2ME)。

J2SE: 整个 Java 技术的核心和基础, 它是 J2ME 和 J2EE 编程的基础, 也是本书主要介绍的内容。

J2EE: Java 技术中应用最广泛的部分, 提供了企业应用开发的完整解决方案。

J2ME: 主要用于控制移动设备和信息家电等邮箱存储的设备。

2002 年 2 月, SUN 公司发布了 JDK 历史上较为成熟的版本——JDK 1.4, 此时由于 Compaq、Fujitsu、SAS、Symbian、IBM 等公司的参与, 使 JDK 1.4 成为当时发展最快的一个 JDK 版本。到 JDK 1.4 为止, 用户已经可以使用 Java 实现大多数的应用。

2004 年 10 月, SUN 公司发布了万众期待的 JDK 1.5, 同时 SUN 将 JDK 1.5 改名为 Java SE 5.0, J2EE、J2ME 也相应地改名为 Java EE 和 Java ME。JDK 1.5 增加了诸如泛型、增强的 for 语句、可变形参、注释、自动拆箱和装箱等功能。同时, 推出了 EJB 3.0 规范和 MVC 框架规范。

2005 年 6 月, JavaOne 大会召开, SUN 公司公开 Java SE 6。此时, Java 的各种版本已经更名, 取消其中的数字“2”: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

2006 年 12 月, SUN 公司发布 JRE 6.0。

2009 年 4 月 20 日, 甲骨文 74 亿美元收购 SUN, 取得 Java 的版权。

2011 年 7 月, 甲骨文公司发布 Java 7 的正式版。

2. Java 语言的特点

Java 语言的流行在于 Java 语言本身的面向对象、简单、可移植性、安全性、多线程等特点。Java 语言的结构与编写方式与 C++ 语言很类似, 因此学习 Java 语言, 不仅要了解 Java 语言独有的编程特点, 同时还要有程序设计基础和面向对象的概念。

(1) 简单性。Java 语言与 C++ 类似, 如果用户了解 C++ 和面向对象的概念, 就可以很快编写出 Java 程序; 此外, Java 摒弃了 C++ 中如头文件、指针变量、结构、运算符重载、多重继承等复杂特性, 它只提供了基本的方法, 减少了编程的复杂性。

(2) 面向对象。所谓面向对象是指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。传统的过程式编程语言是以过程为中心、以算法为驱动, 而面向对象编程语言则是以对象为中心、以消息为驱动。过程式编程语言用公式表示为: 程序=算法+数据; 面向对象编程语言用公式表示为: 程序=对象+消息。

面向对象的三个特征: 封装、多态性和继承。现实世界中的对象均有属性和行为, 属性表示对象的数据, 行为表示对象的方法。

(3) 可移植性。Java 严格规定了各种基本数据类型的长度。Java 语言经编译后生成与计算机硬件结构无关的字节代码, 这些字节代码被定义为不依赖任何硬件平台和操作系统。Java 系统本身 also 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。Java 程序经过一次编译后可移植到别的系统上解释执行, 如 MS-DOS、Windows、UNIX 等任何平台上运行, 具有很强的可移植性。



(4) Java 语言是解释型的。如前所述, Java 程序在 Java 平台上被编译为字节码格式, 然后可以在实现这个 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连接阶段被载入运行环境中。

(5) 交互式特性。Java 是面向对象的网络编程语言, 由于它支持 TCP/IP 协议, 使得用户可以通过浏览器访问到 Internet 上的各种动态对象。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, 方便实现 Java 的分布式特性。

(6) 多线程机制。Java 语言支持多线程机制, 多线程机制使得 Java 程序能够并行处理多项任务。多线程机制可以很容易地实现网络上的交互式操作。C 和 C++ 采用单线程体系结构, 而 Java 支持多个线程的同时执行, 并提供多线程之间的同步机制。

(7) 动态的内存管理机制。Java 语言采用了自动垃圾回收机制进行内存的管理。在 C++ 语言中, 程序员在编写程序时要及时释放不用的内存单元, 一旦内存管理不当, 就可能会造成内存泄漏、程序运行故障问题。在 Java 系统中包括了一个自动垃圾回收程序, 它可以自动、安全地回收不再使用的内存块。这样, 编程人员就无须担心内存的管理问题, 从而使 Java 程序的编写变得简单, 同时也减少了内存管理方面出错的可能性。

(8) Java 是高性能的。与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) 可靠性和安全性。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。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。Java 虽然源于 C++, 但它消除了许多 C++ 的不可靠因素, 可以防止许多编程错误。首先, Java 是强类型的语言, 要求用显式的方法声明, 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。

上述几种机制结合起来, 使得 Java 成为安全的编程语言。

3. Java 程序的运行机制和 Java 虚拟机

(1) Java 程序的运行机制。Java 语言具有解释性语言和编译性语言的特征。编译性语言是指: 使用专门的编译器, 针对不同的平台将源码一次性翻译成该平台能够执行的机器码, 此过程称为编译。编译型语言一次性翻译成机器码的好处是, 程序可以脱离开发环境独立运行, 运行效率高。但其缺点是因为被翻译成特定平台的机器码, 所以无法移植到其他平台上; 如果要移植需要在其他平台上重新编译。

解释型语言是指: 使用专门的解释器, 对源程序进行逐行解释成特定平台的机器码并立即执行的语言, 不会进行整体性的编译和连接处理, 每次执行时都要进行一次编译。其特点是移植性好, 但程序执行效率低。



(2) Java 虚拟机 (JVM)。在 Java 中的所有程序都是在 JVM (Java 虚拟机) 上运行的。Java 虚拟机有其完善的硬件架构, 如处理器、堆栈、寄存器等, 还有相应的指令系统。JVM 屏蔽了与具体操作系统相关的信息, 使得 Java 程序只需要产生在 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, 实际上最终还是把字节码解释成具体平台上的指令执行。

从图 1.1 中可以得知, Java 程序并不是直接交由操作系统处理, 而是经过一系列转换, 最后经过 Java 虚拟机的解释器解释后再交由操作系统的。正是这一套机制使得 Java 程序可以不依赖操作系统, 它有自己的处理机制, 可以根据不同系统编译、解释出适合于特定系统的代码, 从而实现了跨平台功能。

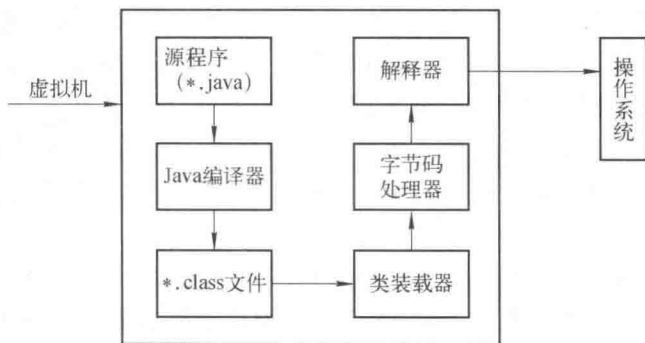


图 1.1 Java 虚拟机机制



任务二 Java 开发环境搭建



任务描述

搭建 Java 开发环境。



必备知识

1. Java 平台简介

根据应用领域不同, Java 提供了以下三个平台。

Java SE (Java Standard Edition), Java 的标准版, 它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序, 用于主应用程序和数据库开发。

Java EE (Java Enterprise Edition), Java 的企业版, 它提供了企业开发的各种技术, 主要用于企业级开发, 现在用得最多的也就是这个平台。

Java ME (Java Micro Edition), Java 的微型版, 这个版本主要用于嵌入式和移动平台的开发。

2. JDK 的安装与配置

(1) JDK 的安装。下面将要搭建 Java 的开发平台，这是学习 Java 语言的基础，只有搭建了 Java 的平台开发环境，才能调试和运行编写的 Java 程序。可以到甲骨文的官方网站（<http://www.oracle.com/technetwork/java/javase/downloads/index.html>）免费下载一个 JDK (Java Development Kit, Java 开发工具)，选择适合所用操作系统的一款。例如：选择 jdk-7u7-windows-i586.exe 安装文件。下载完成后，按照安装向导安装就可以了。JDK 的安装向导如图 1.2 所示。

单击“下一步”按钮，按照对话框提示逐步安装，这里选择默认的安装路径，也可以更改 JDK 安装路径。系统一并安装 JRE，安装完成后，出现如图 1.3 所示的界面。



图 1.2 JDK 的安装向导

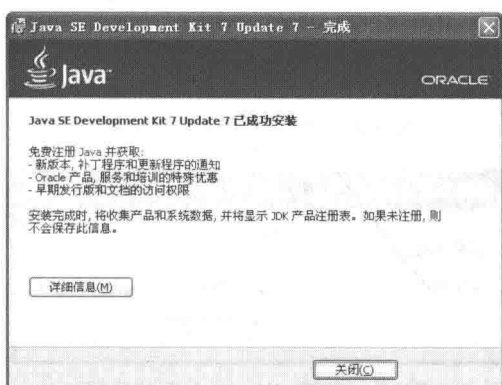


图 1.3 JDK 安装完成

在安装好 JDK 以后，配置系统环境变量前，在命令行窗口中输入“javac”，会出现“‘javac’不是内部或外部命令，也不是可运行程序或批处理文件”的提示，此时表示 javac 编译命令尚不可用。

(2) JDK 的配置。JDK 安装完成后，要在系统属性对话框中对环境变量进行配置。为了说明几个 JDK 环境变量的作用，这里先给出环境变量的定义。

环境变量一般是指在操作系统中用来指定操作系统运行环境的一些参数，比如临时文件夹位置和系统文件夹位置等。类似于 DOS 的默认路径，当运行某些程序时除了在当前文件夹中查找外，还会到设置的默认路径中去查找。

具体配置步骤如下：

① 右击“我的电脑”，选择“属性”命令，在弹出的“系统属性”对话框中选择“高级”选项卡，如图 1.4 所示，单击“环境变量”按钮，出现图 1.5 所示的“环境变量”对话框。单击“新建”按钮，新建一个名为 ClassPath 的系统变量，变量值输入“.;C:\Program Files\Java\jdk1.7.0_07”，如图 1.6 所示。

② 在系统变量中，找到并编辑 Path 变量，在末尾添加路径 C:\Program Files\Java\jdk1.7.0_07\bin，如图 1.7 所示。

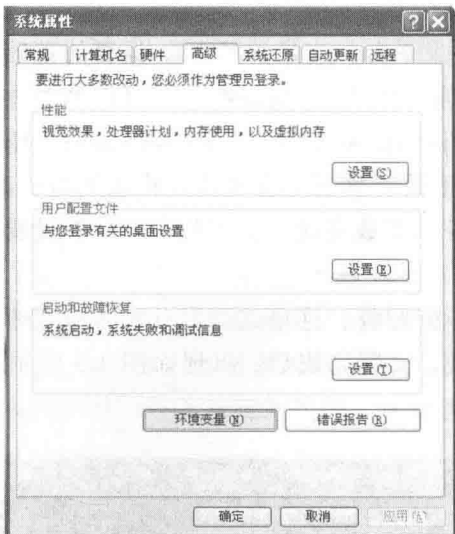


图 1.4 “系统属性”对话框



图 1.5 “环境变量”对话框



图 1.6 新建 ClassPath 系统变量



图 1.7 编辑 Path 系统变量

配置完 JDK 环境后，可以在命令行窗口中用 `javac`（回车）和 `java`（回车）命令来测试是否配置成功。若出现图 1.8 和图 1.9 所示界面，表示配置成功。若配置不成功，修正系统变量的内容后，需重新打开命令行窗口再测试。

图 1.8 输入命令 `javac` 后的屏幕显示

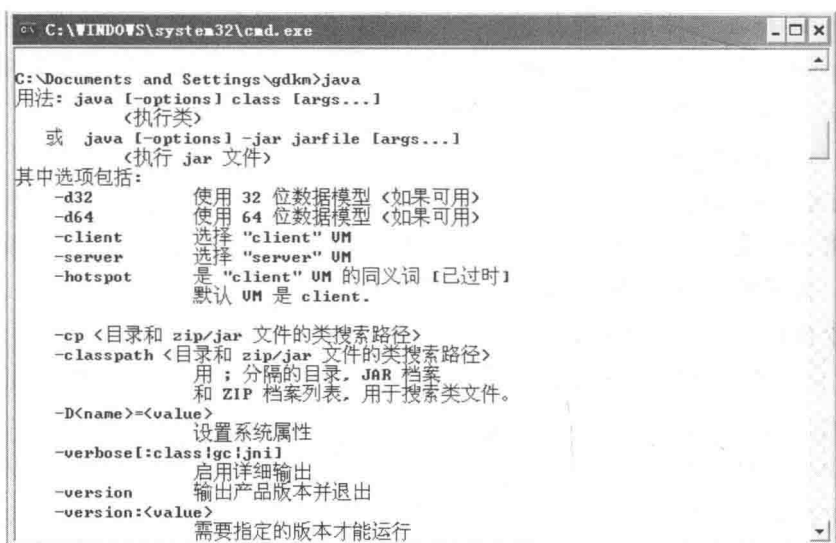


图 1.9 输入命令 java 后的屏幕显示



任务三 编写并运行第一个 Java 程序



任务描述

编写第一个 Java 程序, 运行该程序, 使得在控制台上输出信息 “Hello! Welcome to Java!”



必备知识

1. 什么是 Java 源程序

所谓源程序是指程序员从键盘上输入的还没有经过编译、解释等处理的最原始的程序代码。例如, 用 C 语言编写的源程序扩展名为 .c, 用 C++ 编写的源程序扩展名为 .cpp, 用 Java 语言编写的源程序扩展名为 .java。

2. Java 中类的定义格式

Java 中类的定义格式:

```
class 类名 {
    类的属性;
    类的方法() {
    }
}
```

(1) 在 Java 中, 类是用关键字 class 定义的, class 后面是类名, 类名是用户自定义的, 只要满足一定的命名规则即可。

(2) 类体是由类名后的一对 {} 花括号括起来的内容, 类的两个组成部分是类的属性和类的方法, 属性和方法可以有多个。属性用于描述类的静态特征, 而方法用于描述类的动态特征。



(3) 一个 Java 文件是可以由多个类组成的, 但只能有一个 public 类, 如果某一个类是用 public class 去声明的, 则这个类称为主类, 主类名要和 Java 源文件名一致, 而且主方法 main() 只能定义在主类中。有关更多类的知识可参考项目四。

3. 如何编译和运行一个 Java 程序

(1) 方法一: 用记事本写 Java 源程序, 利用命令行方式进行编译、运行 Java 程序。

我们将 Java 源代码写在记事本上, 并命名为 Hello, 同时把扩展名 .txt 改成 .java, 然后保存。为了便于后面讲解, 这里把它保存在 D 盘 jdemo (d:\jdemo) 目录下。然后打开命令窗口, 可以直接输入 cmd 命令, 也可以利用“开始”菜单, 将当前目录切换到 Hello.java 源文件所在的目录下。这里用 cd 命令进行切换, 具体操作如下:

```
d: (Enter)
cd jdemo (Enter)
```

用盘符带冒号 (如 D: 命令), 可以切换到 D 盘下; cd 命令带文件名可以在同盘下切换, 切回上级目录使用 cd .. 命令。

用 javac Hello.java 命令, 对 Hello.java 源文件进行编译, 如果没有语法错误, 编译器 (Compiler) 将会生成一个扩展名为 .class 的字节码文件 (Hello.class); 如果编译时有语法错误, 编译器会报错, 当然也不会生成字节码文件。此时需要修改源代码, 然后再重新编译, 直到能通过编译。

用 java Hello 命令, 执行 Hello 程序, 并显示结果, 输出 “Hello! Welcome to Java!”, 如图 1.10 所示。

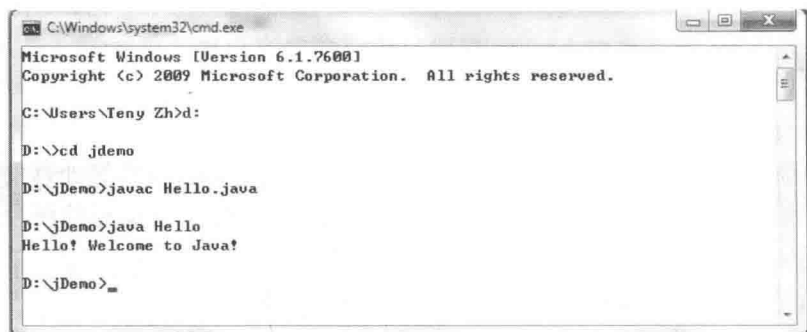


图 1.10 用 javac 命令编译和 java 命令运行 Hello 程序后的结果

(2) 方法二: 还可以用 Java IDE (Integrated Development Environment, 集成开发环境) 编写、编译并执行 Java 程序。

这里用的是 MyEclipse 8.5, IDE 便于用户开发和调试程序。创建 Java 程序的步骤如下:

① 选择工作空间。打开 MyEclipse 并将工作空间目录设为 D:\jdemo, 如图 1.11 所示。工作空间是项目文件的存放目录。这里也可以使用默认的工作空间, 此时项目文件存放在默认工作空间目录下。