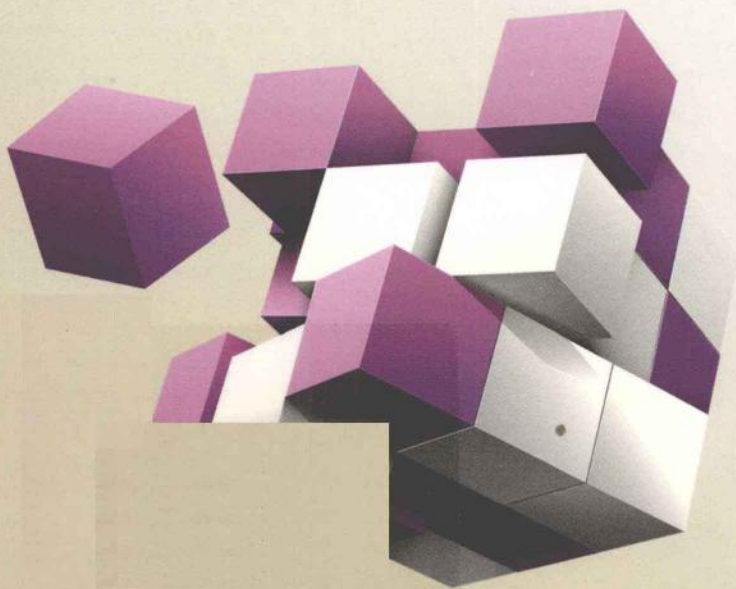


Java Programming

Java程序设计

- 唐振明 主 编
- 王晓华 周清平 董傲霜 编 著



- ★ 详细示例源代码
- ★ 完整PPT讲义
- ★ 教材同步视频教程

中软国际卓越人才培养系列丛书

Java 程序设计

唐振明 主编

王晓华 周清平 董傲霜 编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

古人云,“万事开头难”,熟练掌握 Java 语言是 Java 应用开发的第一个台阶,能够深入理解 Java 程序设计,是学习 Java 系列技术的必要前提。本教材结合作者多年开发和教学经验,从入门者的学习特点出发,循序渐进地带领读者走进 Java 世界,使得“开头并不难”。本教材主要分五个部分,包括 Java 语言的类、类之间的关系、异常处理、常用 API、其他特性总结。书中内容打破了传统的堆砌知识点的方式,而是从解决问题的角度诠释面向对象程序设计,涵盖了企业实际工作中常用的技能与知识点,简单明了,易读易懂。

本教材适合 Java 语言初学者以及 Java 程序员阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java 程序设计 / 唐振明主编; 王晓华, 周清平, 董傲霜编著. —北京: 电子工业出版社, 2011.9

(中软国际卓越人才培养系列丛书)

ISBN 978-7-121-14624-4

I. ①J… II. ①唐… ②王… ③周… ④董… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 190516 号

策划编辑: 程超群

责任编辑: 陈 虹 文字编辑: 刘少轩

印 刷: 涿州市京南印刷厂

装 订: 涿州市桃园装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 15.5 字数: 384 千字

印 次: 2011 年 9 月第 1 次印刷

印 数: 4 000 册 定价: 28.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

序

当我翻阅了“中软国际卓越人才培养系列丛书”后，不禁为这套丛书的立意与创新之处感到欣喜。教育部“卓越工程师教育培养计划”有三个主要特征：一是行业企业深度参与培养过程；二是学校按通用标准和行业标准培养工程人才；三是强化培养学生的工程能力和创新能力。这套丛书紧紧围绕“卓越计划”的要求展开，以企业人才需求为前提，同时又充分考虑了高校教育的特点，能让企业有效参与高校培养过程，是一套为“卓越计划”量身打造的丛书。

丛书的设计理念紧扣中软国际 ETC 的“5R”理念，即真实的企业环境、真实的项目经理、真实的项目案例、真实的工作压力、真实的就业机会，切实地将企业真实需求展现给读者。丛书中的知识点力求精简、准确、实用，显然是编著者经过反复推敲并精心设计的成果。丛书对企业用之甚少的知识点，都进行了弱化，用较少篇幅讲解，而对于企业关注的知识点，都使用非常详尽的内容进行学习。这样的设计对初学者尤其是在校学生非常必要，能够节省很多学习时间，在有限的时间内学习到企业关注的技能，而不是花费很多精力去钻研并不实用的内容。

丛书非常强调“快速入门”这一法宝，能够对某门技术“快速入门”永远是激发学习兴趣的关键。丛书设计了很多“快速入门”章节，使用详尽丰富的图示以及代码示例，保证读者只要根据丛书的指导进行操作，就能够尽快构建出相关技术的实例。

丛书非常注重实际操作，很多知识点都是从提出问题引出，从而在解决这个问题过程中讲解相关的技能。丛书中没有大篇幅的理论描述，尽力用最通俗、最简练的语言讲解每一个问题，而不是“故作高深”地使用很多新名词。

非常值得一提的是，丛书配备了对应的 PPT 讲义，并将 PPT 讲义显示到了相应章节，这种形式令人耳目一新。首先能起到提纲挈领的作用，帮助读者快速了解每个章节的主要内容，掌握完整的知识体系。另外，这种方式非常适合在高校教学中使用，能够完全与教材同步，方便学生课后复习及课前预习，可以有效提高教学效果。

这套丛书是中软国际多年行业经验的积累和沉淀，也是众多编者智慧与汗水的结晶，一定能在校企合作的道路上发挥积极长远的作用。

国家示范性软件学院建设工作办公室副主任
北京交通大学软件学院院长



前 言

Java 语言是学习 Java 开发技术的第一步，学好 Java 语言，打好基础，对于使用 JavaEE 进行企业级编程非常重要。大多数企业的面试官，都非常关注应聘者对 Java 语言的掌握及理解程度。所以，Java 编程语言堪称 Java 学习之路上的奠基石，其重要性不言而喻。

Java 语言的相关知识点非常繁多，这往往给初学 Java 语言者造成很多困扰。如果面面俱到去研究所有知识点，时间和精力都要非常充沛才可以；而如果只是蜻蜓点水般去了解 Java，在实际工作中又无法使用。这本教材力求简洁明了，从繁多知识点中提取企业常用的技能，总结初学者必须掌握的内容，使得读者能够在有限时间内，掌握更多的能在实际工作中使用的编程技术。

本书编著者在实际工作中发现，很多 Java 初学者甚至有开发经验的 Java 程序员，对 Java 语言的掌握都不够系统，没有一个完整的知识体系，而仅仅掌握一些片面的内容。本书配套了 PPT 讲义，并把讲义内容显示到每个章节的对应位置，这样有助于读者能够快速了解每个章节的主要内容，建立起 Java 的完整知识体系，对整体掌握 Java 程序设计起到非常好的作用。本书也非常适合作为高校 Java 程序设计相关课程的教材使用，书中的 PPT 可以单独下载，这样能够保证讲师授课思路和顺序完全与教材对应，取得良好的教学效果。附录部分提供了企业关注的技能点，并从企业的角度给予了解析，能够帮助读者进一步整理书中内容，掌握企业需要的技能。

教材相关的 PPT、源代码及视频均可到 <http://www.hxedu.com.cn> 下载。

本书分成了五个部分，每一部分都专注一个大的主题，而每个大的主题都是前后关联，相辅相成的关系。读者按照这五个部分进行学习，不仅能循序渐进掌握 Java 编程语言的核心知识点，而且能真正建立面向对象的编程思想。

第一部分：Java 语言的类。既然 Java 应用都是由 Java 类组成，所以首先需要了解 Java 类的相关知识。如 Java 的类由哪些部分组成，以及这些组成元素所涉及的知识点。本部分学习结束后，读者将对一个 Java 类的各个组成部分都有深入理解。

第二部分：类之间的关系。第一部分已经对一个独立的 Java 类本身有了深入理解。然而，一个 Java 应用中不可能只有一个类，一定是由多个类组成。既然有多个类，类与类之间就一定存在着各种关系。本部分将详细讲解类与类之间的各种关系，如关联、依赖、继承、实现。

第三部分：Java 异常处理。学习完前两部分后，读者对于封装、继承、多态的概念已经掌握。异常处理是保证程序鲁棒性的一个有效方法，本部分主要学习 Java 的异常处理机制。

第四部分：核心 API。学习完前三部分后，对于 Java 语言的核心概念已经掌握。本部分主要关注 Java 语言核心 API 的使用。包括集合、输入输出、GUI、线程等。本部分学习结束后，读者能够熟练使用常用的 API 进行编程。

第五部分：其他特性总结。本部分集中讨论一些特性，如泛型、枚举、可变参数、Annotation 等，客观讨论每种特性的适用场景。

在编写本书的过程中，得到了很多领导、同事以及朋友的帮助。感谢中软国际的所有领导以及 CTO 办公室的所有同事，是你们的帮助、鼓励以及支持才有这本书的问世。感谢编辑程超群，如果没有你的辛苦工作以及真诚建议，这本书的出版将不会这么顺利。

由于编者水平有限，也由于时间仓促，书中一定存在一些不尽人意的地方，甚至会有一些错误。如果您发现了任何内容方面的问题，烦请一定通知我们 (wangxh@chinasofti.com)，我们争取尽快勘误。

目 录

第一部分 Java 语言的类	(1)
第 1 章 Java 语言概述	(2)
1.1 Java 语言特点	(2)
1.2 面向对象的相关概念	(3)
1.3 第一个 Java 类	(4)
1.3.1 环境搭建	(4)
1.3.2 编译运行 Java 类	(6)
1.4 本章小结	(7)
第 2 章 Java 类的组成	(8)
2.1 类	(8)
2.2 属性	(9)
2.3 方法	(9)
2.3.1 方法声明形式	(9)
2.3.2 方法重载	(11)
2.4 构造方法	(12)
2.5 块	(15)
2.6 内部类	(16)
2.7 本章小结	(17)
第 3 章 访问权限修饰符	(18)
3.1 包	(18)
3.2 权限访问修饰符	(19)
3.3 封装性	(19)
3.4 本章小结	(20)
第 4 章 数据类型	(21)
4.1 基本数据类型	(21)
4.2 引用类型	(22)
4.3 字符串类型	(22)
4.4 包装器类型	(24)
4.5 值传递	(25)
4.6 本章小结	(26)
第 5 章 高级类特性	(27)
5.1 static	(27)
5.2 final	(30)
5.3 Java 类中的变量总结	(32)
5.4 本章小结	(33)

第6章 操作符、流程控制	(34)
6.1 操作符	(34)
6.2 流程控制	(38)
6.2.1 流程控制概述	(38)
6.2.2 使用 break/continue 语句控制循环	(42)
6.2.3 “中断”语句比较	(44)
6.3 本章小结	(44)
第7章 第一部分自我测试	(45)
第二部分 类之间的关系	(46)
第8章 关联关系	(47)
8.1 关联关系的表示	(47)
8.2 关联关系的方向	(49)
8.3 关联关系的多重性	(50)
8.4 数组	(50)
8.4.1 数组的基本概念	(50)
8.4.2 数组的创建及长度	(51)
8.4.3 数组元素的访问	(52)
8.4.4 数组的赋值	(53)
8.4.5 多维数组	(54)
8.4.6 复制数组	(55)
8.4.7 Arrays 类	(56)
8.5 本章小结	(58)
第9章 依赖关系	(59)
9.1 Java 语言中依赖关系的表示	(59)
9.2 依赖与关联的区别	(59)
9.3 本章小结	(60)
第10章 继承关系	(61)
10.1 Java 语言的继承	(61)
10.2 继承中构造方法的调用	(62)
10.3 方法覆盖	(65)
10.4 抽象类	(68)
10.5 抽象方法	(69)
10.6 多态	(71)
10.7 this 与 super 关键字	(73)
10.8 final 在继承中的使用	(74)
10.9 static/abstract/final 总结	(75)
10.10 关联和继承	(76)
10.11 Object 类	(77)
10.12 本章小结	(80)

第 11 章 实现关系	(81)
11.1 接口定义	(81)
11.2 类与接口的关系: 实现	(82)
11.3 接口的作用	(84)
11.4 接口的语法细节	(85)
11.5 Comparable 接口	(87)
11.6 本章小结	(90)
第 12 章 第二部分自我测试	(91)
第三部分 异常处理	(92)
第 13 章 异常处理	(93)
13.1 什么是异常	(93)
13.2 Java 标准异常类型	(94)
13.3 如何处理异常	(96)
13.4 如何抛出异常	(101)
13.5 为何要抛出异常	(102)
13.6 抛出异常后如何处理	(103)
13.7 自定义异常类	(105)
13.8 throw 与 throws 总结	(107)
13.9 try/finally 的作用	(108)
13.10 重新抛出异常	(109)
13.11 本章小结	(110)
第 14 章 第三部分自我检测	(111)
第四部分 核心 API 的使用	(112)
第 15 章 集合框架	(113)
15.1 泛型快速入门	(113)
15.2 Java 集合框架概述	(115)
15.3 Iterator 接口	(116)
15.4 Collection 及其子接口	(116)
15.5 List 的实现类	(117)
15.6 Set 的实现类	(119)
15.7 Map	(123)
15.8 Collections 类	(127)
15.9 集合与数组之间的转换	(128)
15.10 本章小结	(130)
第 16 章 输入/输出系统	(131)
16.1 File 类	(131)
16.2 文件过滤器	(133)
16.3 IO 流的分类	(135)
16.4 如何使用 IO 流	(136)

16.5 本章小结	(139)
第 17 章 GUI 编程	(140)
17.1 GUI 编程概述	(140)
17.2 Swing 中的组件	(141)
17.3 Swing 中的布局	(142)
17.4 Swing 中的事件处理	(145)
17.5 使用内部类进行事件处理	(146)
17.6 使用匿名内部类进行事件处理	(148)
17.7 GUI 中的并发任务	(149)
17.8 本章小结	(150)
第 18 章 多线程编程	(151)
18.1 与线程有关的 API	(151)
18.1.1 Thread 类	(151)
18.1.2 Runnable 接口	(152)
18.2 创建线程的方法	(152)
18.3 线程同步	(156)
18.4 线程通信	(158)
18.5 Thread 类的方法	(162)
18.6 线程死锁	(163)
18.7 守护线程	(165)
18.8 本章小结	(166)
第 19 章 Java 网络编程	(167)
19.1 TCP/IP 概述	(167)
19.2 使用 Socket 基于 TCP 编程	(168)
19.3 使用 Datagram 进行基于 UDP 的编程	(172)
19.4 本章小结	(175)
第 20 章 其他常用 API	(176)
20.1 日期	(176)
20.2 国际化	(180)
20.3 格式化	(182)
20.4 大数据类型	(185)
20.5 反射	(186)
20.6 本章小结	(189)
第 21 章 第四部分自我检测	(190)
第五部分 其他特性总结	(191)
第 22 章 泛型	(192)
22.1 泛型介绍	(192)
22.2 定义简单泛型类	(193)
22.3 泛型与继承	(194)

22.4	通配符	(195)
22.5	泛型方法	(196)
22.6	本章小结	(196)
第 23 章	枚举	(197)
23.1	为什么使用枚举	(197)
23.2	枚举的创建	(199)
23.3	Enum 类.....	(200)
23.4	遍历 enum 的值	(201)
23.5	enum 的细节问题	(201)
23.6	本章小结	(203)
第 24 章	其他特性	(204)
24.1	增强 for 循环	(204)
24.2	自动装箱、拆箱	(205)
24.3	静态导入	(206)
24.4	可变参数	(207)
24.5	Annotation	(208)
24.6	本章小结	(210)
第 25 章	第五部分自我检测	(211)
附录 A	企业关注的技能	(212)

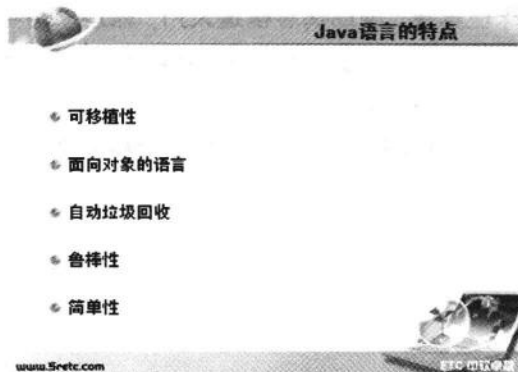
第一部分 Java 语言的类

要学习 Java 程序设计，首先要熟悉 Java 应用的基本组成单位。作为一门面向对象的语言，可以说 Java 应用都是由若干个类(class)组成的。无论一个多么简单或者多么复杂的 Java 应用，都是由若干个 Java 类组成，所以先学习 Java 类的相关知识非常必要。本部分先介绍 Java 语言的基本特征以及环境搭建，帮助读者快速入门，然后学习一个 Java 类的 5 种主要组成元素，帮助读者对一个 Java 类有较完整的了解。在后面的几章将介绍 Java 类中的一些重要且通用的知识点，如数据类型、权限修饰符、static、final、操作符、流程控制等。学习本部分后，读者能熟练掌握并理解一个 Java 类的相关语法，也能初步了解面向对象的思想。

第 1 章 Java 语言概述

本章主要介绍 Java 语言的特点、开发环境的安装以及如何编译运行一个简单的 Java 类，通过学习本章，读者将能快速上手，揭开 Java 语言的面纱。

1.1 Java 语言特点



Java 语言是 Sun 公司（目前已被 Oracle 公司收购）于 1995 年正式推出的面向对象的编程语言。秉承“简单明了”的编写原则，对于其发展历史本书不再赘述，下面就从了解 Java 语言的主要特点开始，走进 Java 的世界。Java 语言有很多优秀的特点，本节中只列出几种重要的特征，总结如下：

1. 可移植性

可移植性，又称跨平台性、平台无关性、或者“一次编写、处处运行”，可以说是 Java 语言最为闪光的特点，在任何场合介绍 Java 语言，都少不了介绍这一特征。可移植性的意思是开发 Java 应用，无须为不同平台开发不同的程序，只要开发一次，就可以在任何平台运行。要理解可移植性，首先需要了解 Java 程序的运行过程：Java 程序的源文件都被存储为以 .java 为后缀的文本文件，要运行 Java 程序，首先需要对源文件进行编译（compile）；编译后将生成类文件（.class 文件）。类文件是字节码文件，不是机器码文件，可谓“半成品”。CPU 执行类文件前需要将类文件解释生成符合当前平台规范的字节码文件。负责将字节码解释成机器码的是 JVM（Java 虚拟机）中的解释器。而 SUN 公司为主流平台都提供了相应版本的 JVM。JVM 负责根据平台特征，将编译生成的字节码解释成符合当前平台规范的机器码，从而实现了只要一次编写，就能在不同平台上运行的效果。可以说，如果要运行 Java 程序，首先必须安装 JVM，因为只有 JVM 的解释器能将编译生成的字节码成功解释为符合当前平台规范的机器码，才能保证在当前平台上正常运行 Java 应用。



什么是 JVM？即 Java 虚拟机，是一个想象中的机器，在实际的计算机上通过软件模拟来实现。Java 虚拟机有自己想象中的硬件，如处理器、堆栈、寄存器等，还具有相应的指令系统。只要在计算机上安装了 JDK（Java 开发工具包），该计算机就拥有了虚拟机。

2. 面向对象

Java 是一门面向对象 (Object Oriented) 的语言, 也就是说, Java 程序都是以对象作为基本组成单元的。面向对象语言都有三大特征: 封装、继承、多态。这三大特征将在后面的章节中深入学习。要了解面向对象的思想, 首先需要理解一系列相关的概念, 参见 1.2 节。

3. 自动垃圾回收

众所周知, 内存的有效使用对于程序设计是非常重要的。对于不再被使用的数据, 就应该及时释放其占据的内存, 以提高内存使用效率, 这个过程被称为“垃圾回收”。Java 语言的垃圾回收采用后台线程自动完成, 不需要程序员通过代码完成, 称为“自动垃圾回收”。后台线程将跟踪并检查内存使用情况, 对于不再被引用的数据, 将自动进行内存释放。然而, 值得注意的是, 虽然 Java 语言有自动垃圾回收机制, 但是并不能保证不出现内存泄露的问题。如果源代码的结构或算法等有问题, 也同样可能发生内存泄露。

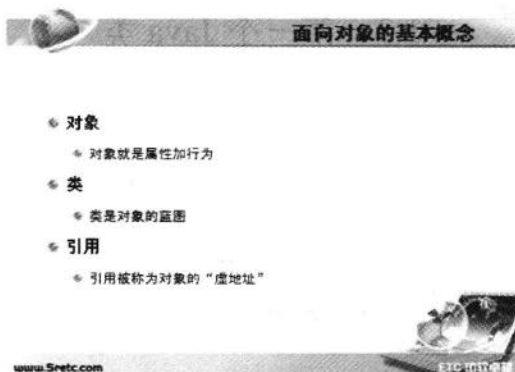
4. 鲁棒性

鲁棒性又称健壮性 (Robustness)。Java 在编译和运行时, 都要对可能出现的问题进行检查, 以消除错误的产生。Java 提供自动垃圾回收机制来进行内存管理, 防止程序员在管理内存时容易产生错误; 通过异常处理机制, 帮助程序员正确地处理异常, 以防止系统的崩溃。异常处理的相关知识参见本书第三部分。

5. 简单性

Java 语言取消了指针, 内存管理通过后台线程自动进行。不再使用 goto 语句, 不支持多继承 (继承的相关知识参见本书第二部分)。这些特性都保证了 Java 的简单性。

1.2 面向对象的相关概念



Java 语言是一门面向对象的语言, 所以在开始编写 Java 程序前, 有必要先理解与面向对象相关的概念。

本节将学习与面向对象有关的几个重要概念, 包括对象、类、引用。

1. 对象

对象(object)这个词在很多地方都会出现,到底什么是对象?可以说“万物皆对象”,这句话虽然看似玄妙,实则是对象非常准确的描述。可以说,现实世界中任何一个存在的事物,都可以称为对象,桌子、鼠标、人、苹果……都是对象。想象一下,如果一个外星人问地球人“什么是汽车?”,地球人将如何描述清楚?可以说:汽车有4个轮子,有刹车系统,可以前进、后退……如果问“什么是近视眼镜?”,可能说:眼镜有两个镜片,能够让近视的人看清物体……总之,对于任何一个对象的描述,其实都是从对象的**属性**和**行为**出发,属性就是对象的数据,如汽车的四个轮子、刹车系统,行为是对象能做的事情,如汽车能前进、后退,即操作。那么可以理解为,对象就是属性加行为。也可以说,对象封装了属性和行为。

Java 语言是面向对象的语言,即使用 Java 语言开发应用,从需求到设计,再到编程实现,都以对象为驱动。一个真正的面向对象的应用是严格反映现实世界的,即现实的业务系统中存在的实际对象,计算机应用系统中都有相应的语言对象。反之也成立,即现实业务系统中不存在的对象,计算机应用系统中也不应该存在。

2. 类

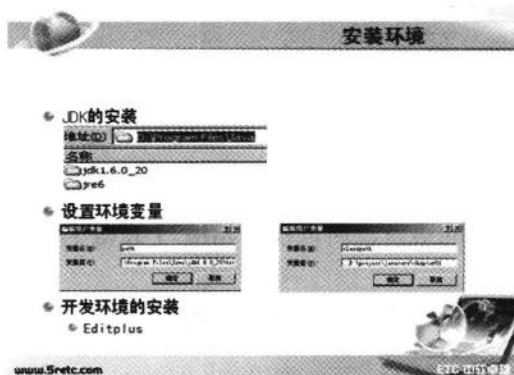
类(class)是对象的蓝图,对象都是通过类创建出来的,也称为实例化。Java 应用都是由若干个类组成,而在运行过程中,都是通过类实例化出的若干个对象在相互协作,从而实现业务逻辑。

3. 引用

Java 中取消了指针,却有一个概念非常类似指针,即引用(reference)。如 `String s=“hello”`,声明创建了一个字符串类型的对象,其中 `s` 即引用。Java 语言中没有明确的指针定义,实质上用来创建对象的每一个 `new` 语句返回的都是一个指针的引用,只不过在大多时候 Java 中不用关心如何操作这个“指针”。引用也可以被称为“虚地址”,如果两个对象的引用相同,那么在物理内存上就认为是一个对象。反之,如果两个对象的引用不同,那么在物理内存上就认为是两个对象。对对象的属性、方法的调用,都是通过引用进行的。如 `s.length()`,调用了对象 `s` 的 `length()` 方法。

1.3 第一个 Java 类

1.3.1 环境搭建



通过前面两节的学习，已经了解了 Java 语言基本知识，接下来写一个最简单的 Java 类，并成功编译、运行。工欲善其事，必先利其器。本节将从环境安装开始，完成最简单的 Java 类。

1. 安装 JDK

要编译运行 Java 程序，必须先安装相应版本的 JDK (Java 开发工具包)。按照默认步骤进行安装即可，成功安装后，相关文件将存放在 Program Files 目录下，如图 1-1 所示。

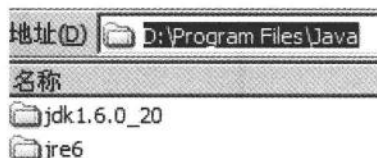


图 1-1 JDK 安装后的目录

要使用 JDK 工具编译 Java 源代码，需要先设置环境变量。首先设置 path 变量，指定 JDK 的 bin 目录所在位置，如图 1-2 所示。

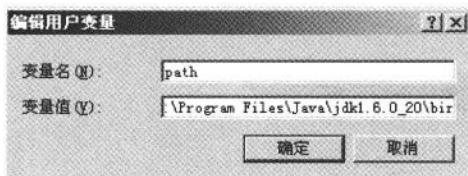


图 1-2 设置 path 环境变量

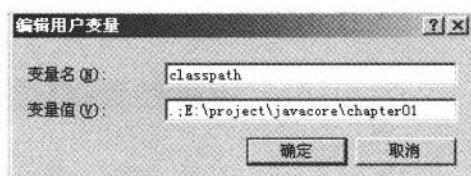


图 1-3 设置 classpath 环境变量

2. 安装开发工具

Java 源代码可以在任意文本编辑器中编写，在开发企业级应用过程中，都会采用 IDE (集成开发环境) 工具以提高效率，如 Eclipse 等。在学习 Java 语言的初期，笔者建议使用最简单的编辑器即可，不必使用 IDE 工具，以强化对语言基础的理解和掌握，如 Editplus 编辑器，其工作界面如图 1-4 所示。

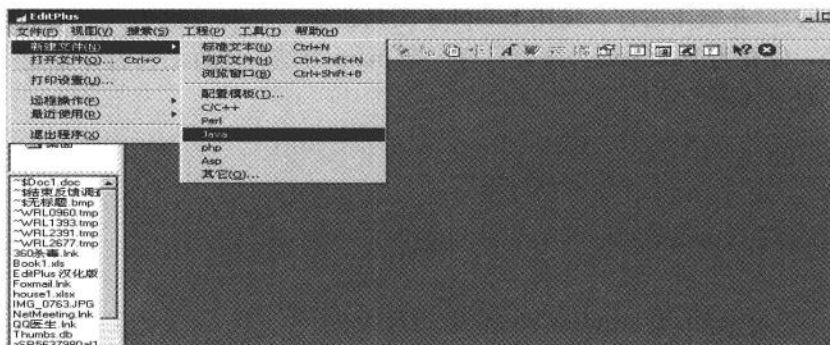


图 1-4 Editplus 编辑器工作界面

当对 Java 语言有了基本了解后就可以使用 IDE 工具，目前使用较多的是 Eclipse。其工作界面如图 1-5 所示。

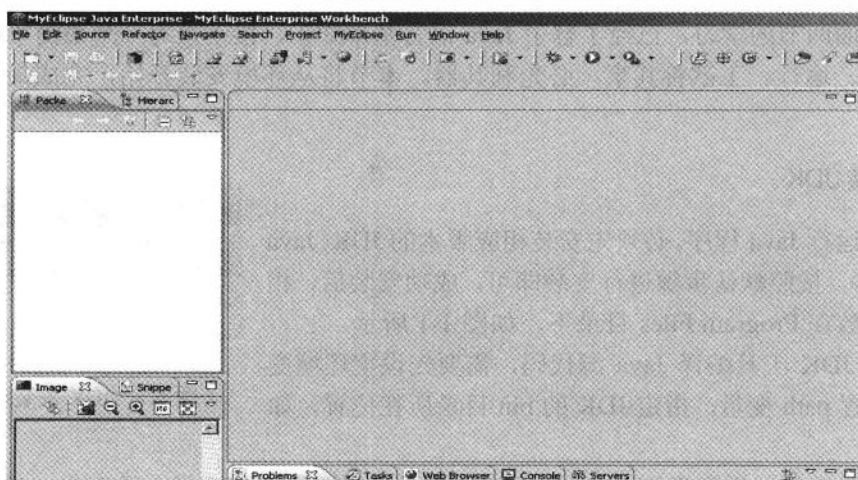
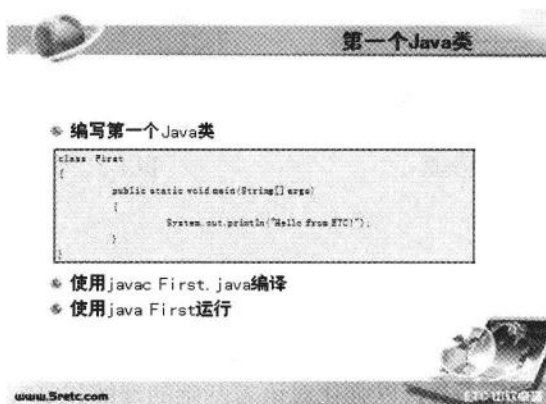


图 1-5 Eclipse 工作界面

1.3.2 编译运行 Java 类



通过上面章节学习，已经成功安装了开发运行环境，并设置了必要的环境变量，本节将编写第一个 Java 类，并编译运行。使用 Editplus 工具新建一个 Java 文件，填写相应代码，完成第一个 Java 类，保存为 First.java。该类只有一个 main 方法，运行后将打印输出字符串“Hello from ETC !”如下代码所示：

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

(1) 编译 First.java。

运行 cmd 命令, 转到 First.java 文件目录下, 使用 javac 命令编译 First.java, 如图 1-6 所示。

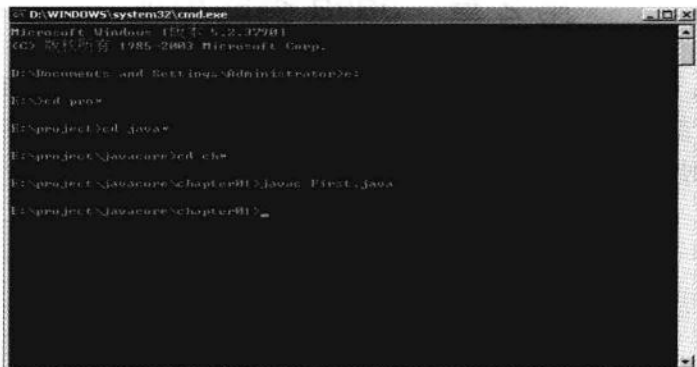


图 1-6 编译 First.java

编译成功后, 会生成 First.class 文件, 即类文件。

(2) 运行 First.class。

使用 Java 命令, 可以运行 Java 类, 这里是 First 类, 如图 1-7 所示。

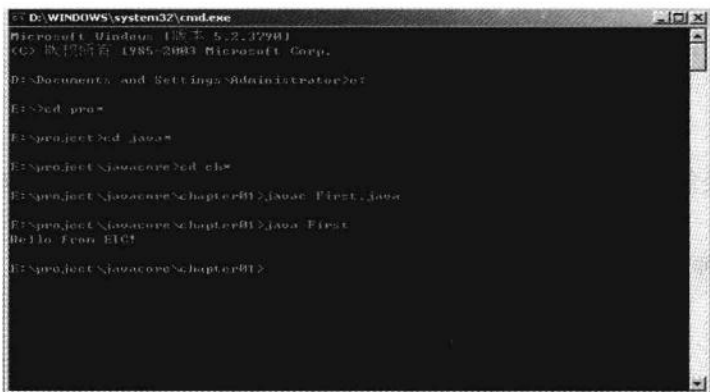


图 1-7 运行 First 类



为什么会打印出“Hello from ETC”? 运行一个 Java 类, 都会默认执行其中的 main 方法, main 方法必须按照规范声明, 即 public static void main(String[] args), 否则将会出现运行错误。该类中只有一个符合规范的 main 方法, 所以运行后打印输出 Hello from ETC。

1.4 本章小结

本章主要起到快速入门的作用, 帮助读者快速了解 Java 语言的特点, 能够顺利安装设置开发运行环境, 并能成功编译运行第一个 Java 类。学习本章后, 读者将对 Java 语言的主要特点有所了解, 重点理解可移植性、面向对象的特征, 也同时对面向对象的相关概念有初步理解, 为继续深入学习打下基础。