高等院校信息技术应用型特色教材

Java 面向对象程序设计

——基础、设计、实现与应用程序开发(5 .0 版)

邵鹏鸣　编著

清 华 大 学 出 版 社

北　京

内 容 简 介

本书针对 Java 2 平台标准版 5.0,详细介绍了如何使用 Java 语言进行面向对象编程。较早地把有关面向对象的内容贯穿其中,使初学程序开发的读者们能够逐步体会并深刻理解“对象”技术的强大功能;较早地引入事件的处理与 GUI 的使用等,使学生们将逐步掌握 GUI 的创建。全书用两章的篇幅通过一个具体的实际应用程序实例讲述使用 JDBC 与 SQL 访问数据库的编程技术。通过多个详尽的实例分析,使学生尽快地掌握面向对象的编程技巧。此外,书中提供了大量与开发相关的技术要点,同时配合实用、有效的 GUI 应用程序,使读者能够迅速掌握并巩固所学到的知识。本书汲取了很多来自实际编程中的经验,这将为培养良好的编程习惯打下一个坚实的基础。

本书可作为高职高专计算机专业和高等技术型院校进行编程语言和 Java 教学的教材,也可供程序设计与开发人员参考。

图书在版编目(CIP)数据

Java 面向对象程序设计——基础、设计、实现与应用程序开发(5.0 版)/邵鹏鸣编著. —北京:清华大学出版社,2006.9
(高等院校信息技术应用型特色教材)
ISBN 7-302-13680-7

Ⅰ. J…　Ⅱ. 邵…　Ⅲ. JAVA 语言—程序设计—高等学校—教材　Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2006)第 100717 号

出 版 者:清华大学出版社　　**地　　址**:北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**:100084
社 总 机:010-62770175　　**客户服务**:010-62776969
组稿编辑:孟毅新
文稿编辑:易慧珍　孟毅新
印 装 者:北京嘉实印刷有限公司
发 行 者:新华书店总店北京发行所
开　　本:185×260　**印张**:28.25　**字数**:643 千字
版　　次:2006 年 9 月第 1 版　2006 年 9 月第1 次印刷
书　　号:ISBN 7-302-13680-7/TP·8255
印　　数:1～4000
定　　价:35.00 元

前　言

当今，计算机技术发展迅猛，在各行各业的应用广泛。面对日新月异的新技术、新方法，我们必须对现有计算机课程的设置和教学内容进行调整，以适应技术进步与市场变化的需要，使教授的东西是市场上最需要的东西。

与其他语言相比，Java 是一个较新的程序开发语言。目前，Java 在全世界的许多商业环境中得到广泛的应用，是现今发展速度最快的程序开发语言之一。随着软件开发技术的发展，Java 越来越受到人们的青睐，越来越多的企业都选择 Java 及其相关技术来构建它们的系统。因此，Java 不仅是一门学习程序开发的好语言，还给人们提供了一门挑战性的技术。

学习一种先进的语言和一种先进的编程方法将激起学生更大的兴趣。这些知识在他们离开学校，进入一个由信息占据重要地位的世界时可以立即发挥作用。正是这一点激发了他们对知识的学习热情。运用 Java 能够出色地完成任务，所以他们更愿意投入更多的精力和时间。

很多院校都有软件专业的学生，有的学校的学生，在几年的学习时间中，不仅学了 C、C++、Java，而且也学了 VB 或 VB .NET 甚至 C#，同时还学习了 Web 应用程序设计及其他有关软件开发的课程。但据作者的调查，仍然有些院校软件专业的学生在毕业时，很难找到一份对口的工作，他们的工作居然与软件无关，甚至与计算机无关。不是企业不需要软件专业的人才，而是没有所需要的人才。我认识一位学生，学习成绩不错，学习也很勤奋，在毕业前夕还通过了全国计算机技术与软件专业技术资格（水平）考试，获得软件设计师资格。但当我交给他一份实际任务时，他根本无从下手，尽管该任务很简单。这些都说明，我们的教学体系、教学内容、教学模式和方法都存在很多需要改进的地方。

在传统教学方法中，学生在学习某个知识点之前，首先需要学习和熟记一些基本的概念，然后在老师的讲解下学习相关的应用。虽然这种方法有助于学生奠定一定的理论基础，但由于其重理论而轻实践，而且各知识点之间以及各章之间的关联性较弱，所以使得学生在举一反三和知识的横向联系等方面能力较为欠缺，常常无法适应实际工作的需要。

本人也不赞成用“玩具程序示例”说明概念、原理的教学方法。这些例子完全脱离实际，看起来很幼稚，尽管它能够说明一个概念，学生也容易看懂，上课也容易听懂。但是检查学习好坏的标准，不是“知道还是不知道，听懂还是没有听懂，看懂还是没有看懂”，而是“会不会应用”。真实程序示例同样能够说明概念、原理和知识点。只不过是面对复杂的

技术实践体系，需要花大量的时间、精力去设计这样的例程。只要设计得好，真实例程学生同样容易看懂、听懂，并且懂得在实际应用中如何运用这些概念、原理和知识点，起到事半功倍的效果。这就是本人编著本书的目标之一。

作者的目标很明确，即写一本实用、有一定的深度且易读的 Java 程序设计教材。无论读者有无编程经验，要做到开卷有益，重要的是能够激发他们的学习兴趣，能够将他们所学的知识应用于实际。

本书以培育技术应用人才为目标，将基本技能培养和主流技术相结合，以市场对人才的需求为依据，把软件工程的思想融入教材体系中，从应用与工程实践的角度出发进行组织编写，其主要特色如下：

1．新体系、新内容、新手段、新思路。无论是内容体系、编写的思路、教学的模式及理念等都具有一定的新意。

2．先进性及实用性。本书的内容反映最新的实用的程序设计方法及技术，并符合新世纪教学发展的规律，书中讲授的程序设计方法与现代编程方法步调一致，具有很强的实用性。

3．以问题、模型为中心的教学方法和面向实际的应用程序驱动的教学方式。本书完全通过实例来学习，利用大量的完整实例，介绍相关的主题，而不是采用传统的“提出概念—解释概念—举例说明”的方法；以实践为主线，以应用为目标，通过完成实际的应用程序的方式学习程序设计知识。整个学习是由许多小的教学模块和任务组成。每个教学模块和任务首先提出一个来自实际的问题，学生首先可“探试”解决问题的应用程序以了解其工作过程，接着教师介绍解决问题的方法及步骤，并归纳出一般的规律、概念及知识点。随后学生参照教师的方法和步骤来解决问题，最后教师对学生提出一个新的实际问题，学生应用所学的方法和步骤来解决该问题。学生通过 “实践—学习—实践—提高”的过程，不仅可以更快、更深入地理解和掌握课程的内容，而且提高了自己的实践技能和独立解决实际问题的能力。

每一个新概念、知识点的提出都伴随着一个完整的、可实际运行的实用程序及其输入、输出。学生通过完成实际的应用程序，掌握概念和知识点。

4．使学生从一开始就编写有用的程序，这有助于保持读者的学习兴趣及动力。书中的实例不是只有几行代码的小程序，这些实例来源于实际应用程序或其中的一部分，它们与我们的生活相关或是大家感兴趣的内容。本书包含的大量实例及其代码分析与讨论是本书的精髓。学生可先按照所描述的方法和步骤编写代码并运行自己所创建的应用程序，“探试”该应用程序并了解其工作过程。然后通过其代码分析与讨论掌握它所包含的概念、原理和方法。

5．逐步展现主题鲜明的各章。本书将集中重要的主题并进行充分论述，而不是肤浅地罗列许多概念、术语和语法。全书的内容及体系结构使学生感到它既具有易读性又增进知识，具有很强的实用性。

6．面向实际的技术。学习程序设计语言的目的是为了开发程序，本书不只是讲授 Java 的语言要素、语法，而是教会读者如何用 Java 开发程序，书中的每一个面向实际的应

用程序实例都说明了开发过程及包含的知识点。通过学习如何开发程序来掌握语言要素和语法。在第10章,通过一个实际应用程序实例,讲授了三层架构的应用程序设计方法和开发过程。三层应用程序设计是企业应用程序常用的程序设计方法。后续的许多实例都应用了三层应用程序设计,作者通过这种方式,使学生能够熟练应用这种企业常用的程序设计方法和开发过程。

7.面向对象的程序设计。本书自始至终贯穿面向对象编程的思想。我们的目标之一是使学生习惯于现实世界的程序设计,仅仅知道面向对象的概念是不够的,学生们必须能够运用这些知识开发现实世界的程序。教学概念的核心之一是在成为一名对象设计者之前,必须首先成为一名对象用户。换句话说,在有效的设计自己的类之前,必须首先学会使用预定义的类,我们从第一个程序开始就接触如何使用类。学习使用类库的类和编写自己的类相辅相成,互相促进,学习使用类库中的类有助于学习编写自己定义的类,学习自定义类又有助于学会使用类库中的一些类,并加深对类库中的一些类的理解。当今学生必须同时掌握基础语言、面向对象编程和类库。

任何一本真正讲授面向对象方法的教材都必须从对象出发,即所有处理过程都需要依照面向对象的术语来进行讨论。然而,这并不意味着学生看到的第一个程序就涉及多个类和方法。在掌握类和方法的编写之前,应教会学生如何使用它们。本书采用的是一个自然推进的教学方式,目的是最终能够让学生在实践中设计出面向对象的问题解决方案。

8.循序渐进,由浅入深,综合应用。本书在设计时贯穿了一个思想:面向实际,同时使读者容易理解所包含的所有细节。因此在内容安排时,循序渐进,由浅入深,后序章节在讲授新的知识点的同时综合应用了前面所学章节的知识。这样做既温习了前面所学的知识,同时也懂得了如何综合应用前面所学的知识解决问题。书中很多实际应用程序实例,只是在讲授新知识点的同时,应用了前面所学过的知识。

9.GUI及Swing控件。带有图形用户界面的程序总能激发学生们的学习欲望,以这类程序作为讲授面向对象概念的实例会取得很不错的效果。本书中实例采用了图形用户界面来介绍事件的处理与GUI的使用等知识。通过对这些内容的不断学习,学生们将逐步掌握GUI的创建。

10.数据库。数据库应用程序如今对所有企业来说是至关重要的,因此本书用两章的篇幅通过一个具体的实际应用程序实例讨论使用JDBC与SQL访问数据库的编程技术,并应用了三层架构的应用程序设计方法和开发过程。

针对GUI及控件概念的提出,我们采用的是一个循序渐进的过程。因此,在讨论GUI及控件之前,学生已对事件的处理与GUI的使用有了一个基本的认识。因此在学习GUI及控件一章时,引入了访问数据库的一些关键性要素。通过创建数据库应用程序来学习GUI及控件。

11.真实的编程经验。本书汲取了很多来自实际编程中的经验,这将为培养良好的编程习惯打下一个坚实的基础。这些经验会运用到所有实例当中并在探讨中得到进一步强调,学生最终将学会如何解决并实现它们。本书还引入并吸收了许多来自软件工程方

面的基础知识。

由于水平和时间的关系,书中错误在所难免,欢迎专家和读者提出宝贵意见(E-mail:pmshaoa163.com)。本书的源代码可在 http://www.pyp.edu.cn/java 站点下载。如果我们发现书中的内容和代码有不当之处,我们也会在该站点发布勘误信息。要想与本书的作者进行交流,可加入论坛 http://www.pyp.edu.cn/java/form/default.aspx。

邵鹏鸣

2006 年 5 月于广州

目　录

第 1 部分　认识 Java

第 2 部分　Java 程序设计基础

第 3 部分　面向对象程序设计

第 4 部分 图形用户界面和数据库程序设计

第 1 部分

认　识　Java

第 1 章

认　识　Java

本章主要内容

1. 创建、编译、运行 Java 程序。
2. 在命令窗口显示指定信息。在消息对话框中显示指定信息。
3. Java 程序的基本组成。使用 Java 类库中的类。

目标

- 熟悉 Java 开发和运行环境。
- 编写简单的 Java 应用程序。
- 能够使用输出语句。

1.1　第一个简单的 Java 应用程序

实例 1-1　Hello,World

问题描述：创建一个应用程序，在命令窗口中显示“Hello,World”，如图 1.1 所示。

图 1.1　程序运行结果

1）创建代码。

在标准的文本编辑器中创建一个新文件。为了演示的目的，使用了记事本应用程序。创建新文件之后输入如下代码，然后将该代码以文件名 Hello.java 存储在你自己的目录中（如 D:\JavaExamples\ch1），注意保存 Java 源代码文件时，其扩展名必须是 .java。

```
//第一个简单的 Java 应用程序
public class Hello {
    // 该类有一个名为 main 的方法
    public static void main(String[] args)
    {
        System.out.println("Hello,World");
    }
```

```
}
```

2）编译并运行程序。

为了立即检验环境是否已经正确地建立起来了，把代码的分析与讨论放到后面，先编译并运行这个应用程序。

现在，可以使用 SDK 中的 Java 编译器（名为 javac 的程序）对源代码进行编译。编译器读取扩展名为 .java 的源代码文件，并创建一个或多个可被 Java 解释器运行的 .class 文件。

请打开命令行窗口，切换到 Hello .java 所在的文件夹。

如果该文件位于主盘根目录的 JavaExamples\ch1 文件夹中，则可以使用下述命令切换到该文件夹：

```
cd\javaexamples/ch1
```

切换到该文件夹后，可以在命令提示符下输入下述命令来编译 Hello .java：

```
javac Hello .java
```

图 1 .2 显示了用于切换到该文件夹 JavaExamples\ch1 并编译 Hello .java 的 MS-DOS 命令。

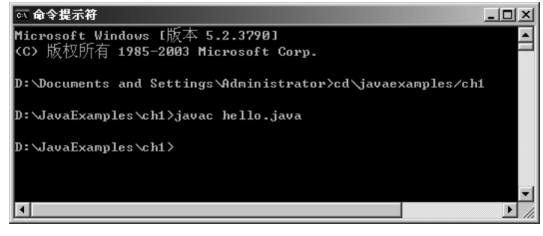

图 1 .2　在命令行窗口中编译 Java 程序

如果程序通过编译，SDK 编译器将不会显示任何消息；如果程序有错误，编译器将指出错误以及导致错误的代码行。

如果该程序编译时没有发生任何错误，将创建一个名为 Hello .class 的类文件，它位于 Hello .java 所在的文件夹中。

类文件中包含将被 Java 解释器执行的 Java 字节码。如果发生错误，请查看源代码文件，确保输入的内容与以上给出的程序完全相同。

创建类文件后，可以使用 Java 解释器来运行它。SDK 中的解释器名为 Java，它也是从命令行运行的。

要运行类文件 Hello，请切换到 Hello .class 所在的文件夹，然后执行下述命令：

```
java Hello
```

你将看到“Hello，World”。

图 1 .3 显示了用于编译、运行该程序的命令以及该程序的输出。

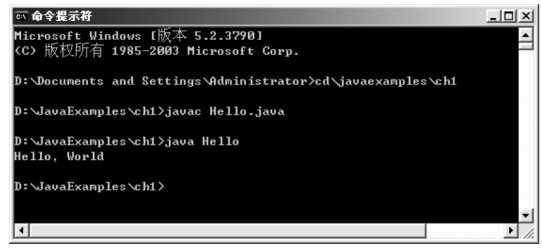

图 1.3 编译和运行 Java 应用程序

代码分析与讨论

1）代码注释。第一行包含注释语句：

```
//第一个简单的 Java 应用程序
```

其中“//”字符将这行的其余内容转换为注释内容。可以将整行作为注释，或者可以在其他语句的结尾追加一个注释，如下所示：

```
System.out.println("Hello,World"); //输出 Hello World
```

程序员在程序中加入注释，用于提高程序的可读性，使程序易于阅读和理解。计算机在执行程序时不会执行注释行。以“//”开始的注释叫“单行注释”，它只对当前行有效。以“/*”开始并以“*/”结束的注释称为多行注释。例如：

```
/* 这是多行注释，
    第一个简单的 Java 应用程序 */
```

2）定义类。Java 的每一个程序包括至少一个自定义类。这些类称作程序员自定义类或用户自定义类。

在 Java 中用关键字 class 引导一个类的定义，其后接着是类的名称（本例中是 Hello）。关键字是 Java 的保留用字，它们用小写字母拼写。习惯上，Java 中所有的类名以一个大写字母开头，并且类名中每一个单词的首字母是大写，类名称为标识符。标识符是遵守下列规则的一系列字符：

① 它们可以为所选择的任意长度。

② 它们可以包括空格之外的任何字符。

③ 它们必须以字母表中的字母、$ 符号或下划线字符（_）作为开头。

5link 不是有效的标识符，因为它是以数字开头；name text 也不是有效的标识符，因为它含有空格字符。

Java 区分大小写。例如 Name 和 name 是不同的标识符。

class 前面的关键字 public 表示该类是可公共访问的，意味着每个人都可使用它。

class Hello 后的左“{”表示一个类的定义，对应的右“}”用来结束类的定义。例如：

```
public class Hello {
    …
}
```

注意： ， 。

3）main 方法。Java 程序必须包含一个名为 main 的方法，而且必须按第 4 行那样定义，main 方法是程序的入口点，程序控制在该方法中开始和结束。方法是用来执行任务，及在任务完成后返回信息。void 关键字表明该方法将执行一个任务，但完成任务后不返回信息。

main 方法在类的内部声明，它必须具有 static 关键字，是静态方法。第 6 章“类与对象”将讨论静态方法。在“Hello，World”示例中，main 方法是 Hello 类的成员。

左“{”开始定义方法的主体内容，对应的右“}”用来结束方法的定义。

main 方法前面的关键字 public 表示该方法是公有的，任何其他程序均可调用该方法。main 方法必须被定义为公有的。

main 方法有一个参数：一个字符数组。它们将在以后介绍。要求读者在刚开始写程序时模仿我们的程序，为让大家使用 Java 的某个特性，而知道该特性的所有细节又不是很重要时，尤其要求大家这么做。最初，所有程序员在学习编程时，都模仿老一辈程序员的编程风格。对每个要求模仿的特性，都会在后面详细讨论。

4）输出。在 main 方法中，语句：

```
System.out.println("Hello,World");
```

使计算机打印双引号之间的字符串。双引号之间的一个或多个字符称为字符串。字符串中的空格符不会被编译器忽略。

System.out 是标准输出对象，它可以提供完成某些任务的方法，其中的一个方法 println 在命令窗口中显示（或打印）传递给它的文本后，自动将光标位置移到下一行。

这里使用的是 System.out 对象，调用它的 println 方法。调用方法使用的是点操作符。对象调用其方法的通用语法是：

```
对象名.方法名(参数);
```

例子中，System.out 对象调用它的 println 方法时，传递给它一个字符串参数，此方法把传递给它的字符串参数显示到命令窗口，然后结束此输出行。这样，每个 println 调用都会在新行上显示输出。

Java 的方法可以没有参数，也可以有一个或多个参数，但即使一个方法没有参数，圆括号也是必需的。例如，System.out 对象还有一个没有参数的 println 方法，它只打印一空行，调用该方法应使用下面的语句：

```
System.out.println();
```

System.out 对象还有另一个 print 方法，它输出的末尾并不加换行符，如，System.out.print("Hello");它将只打印“Hello”，并不换行。下一个输出将紧跟于字符“o”之后。

以下程序使用两条语句产生与上面实例中给出的程序相同的输出。

```
//用两条语句打印一行
public class Hello {
    public static void main(String[] args)
```