

高等教育国家级教学成果二等奖

清华大学计算机基础教育课程系列教材

# Java语言程序设计 案例教程

---

郑 莉 刘兆宏 编著



清华大学出版社

高等教育国家级教学成果二等奖

清华大学计算机基础教育课程系列教材

# Java语言程序设计 案例教程

---

郑 莉 刘兆宏 编著

清华大学出版社  
北京

## 内 容 简 介

本书基于 Java 语言的基本语法设计了 112 个经典语法案例,以综合性和实用性为基准设计了十几个应用案例。通过这些精心设计的案例,深入浅出地介绍了 Java 语言程序设计的特点,循序渐进地介绍了面向对象程序设计的思想和方法。

全书共分为 12 章,内容包括 Java 语言基础知识、类与对象的基本概念、类的方法、类的重用、接口与多态、输入输出流、对象群体的组织、多线程、图形用户界面、JDBC 与数据库访问、Servlet 程序设计、JSP 程序设计。

本书各章安排与《Java 语言程序设计》一致,对于以《Java 语言程序设计》为教材的自学读者和在校学生,可以将本书作为参考书,配合使用。已经初步掌握 Java 语言基础的读者,也可以独立使用本书。讲授 Java 课程的教师可以将本书作为教学参考书,或用作案例课程教材。

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 13501256678 13801310933

## 图书在版编目(CIP)数据

Java 语言程序设计案例教程/郑莉,刘兆宏编著. —北京:清华大学出版社,2007.7  
(清华大学计算机基础教育课程系列教材)

ISBN 978-7-302-15156-2

I. J… II. ①郑… ②刘… 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(2007)第 063675 号

责任编辑:袁勤勇 赵晓宁

责任校对:梁毅

责任印制:何芊

出版发行:清华大学出版社

<http://www.tup.com.cn>

[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

社总机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印 刷 者:北京市清华园胶印厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:19.25 字 数:445 千字

版 次:2007 年 7 月第 1 版 印 次:2007 年 7 月第 1 次印刷

印 数:1~5000

定 价:24.00 元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:026014-01

# 序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国备受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学:

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课件建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色。

(1) 自成体系: 该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,也包括适用于各专业的软、硬件基础知识;既包括基本概念、方法与规范,也包括计算机应用开发的工具与环境。

(2) 内容先进: 该系列教材注重将计算机技术的最新发展适当地引入教学中来,保持了教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

(3) 适应面广：该系列教材照顾了理、工、文等各种类型专业的教学要求。

(4) 立体配套：为适应教学模式、教学方法和手段的改革,该系列教材中多数都配有习题集和实验指导、多媒体电子教案,有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践,凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善,不断更新,为我国高校计算机基础教育作出新的贡献。



---

注：周远清,曾任教育部副部长,原清华大学副校长、计算机专业教授。

# 前言

Java 语言是应用非常广泛的一种面向对象的程序设计语言,主要特点在于跨平台和支持网络程序设计。本书是《Java 语言程序设计》系列教材的组成部分,旨在帮助读者通过实践学习和掌握 Java 语言。

计算机语言是实践性很强的课程,只有通过大量实践才能真正掌握。一般教材限于篇幅,不能提供足够的例题,教师讲课时也因为课时限制,往往不能演示大量案例。这就导致学生在学习了基本概念、基本语法以后,很难动手实践。往往是课听懂了,但是不会写程序。很多案例教程针对的都是具有一定基础的软件开发人员,案例规模较大。而初学者往往既需要阅读、实践一些案例,又暂时没有能力看懂大案例。此书主要是针对刚开始学习 Java 语言的学生,书中的案例规模都很小。对于小的案例学生可以比较容易动手验证、尝试改编,进而巩固课上所学的内容。千里之行,始于足下,如果希望成为高水平的 Java 程序员,那么请从入门这一刻开始,从小程序开始,养成勤于实践、善于思考的学习习惯。这本书的目的就是在初学者的实践之路上助一臂之力。

本书每一章均包含知识点综述、基础案例、应用案例、综合案例。知识点综述部分概要介绍本章涉及的主要知识点,基础案例针对各个语法要点精心设计,帮助读者深入理解相关基础知识和语法要点,每章最后的综合案例则集综合性、实用性于一身,结合本章主要知识点设计而成,帮助读者灵活掌握和使用本章的知识要点。本书的宗旨是,不仅要使读者掌握 Java 语言本身,而且能够对现实世界中较简单的问题和解决方案用 Java 语言进行描述。当然,要能够描述较复杂的问题,还需要学习面向对象的软件工程课等其他课程。

本书各章安排与《Java 语言程序设计》一致,对于以《Java 语言程序设计》为教材的自学读者和在校学生,可以将本书作为参考书,配合使用。已经初步掌握 Java 语言基础的读者,也可以独立使用本书。

讲授 Java 课程的教师可以将本书作为教学参考书,或用作案例课程教材。

本书的源代码可以从清华大学出版社网站下载。读者也可以访问作者郑莉在清华大学讲授的课程,课程位于“清华大学网络学堂”,网址是: <http://learn.tsinghua.edu.cn/>。读者请在“本学期课程”栏目中查找郑莉主讲的“Java 语言程序设计”(如果首页未列出,请单击“更多”),找到后单击课程名,然后以用户名“GUEST”、密码“guest”登录,即可阅读、下载教学资源。在作者不授课的学期,读者可以查找“以往课程”,下载学习资源。

参加本书编写工作的还有：董渊、田荣牌、孟威、张文举、杨兴朋、王绚、邵毓华、孟鸿利等，在此深表感谢。

感谢读者选择使用本书，欢迎您对本书内容提出意见和建议，我们将不胜感激。作者的电子邮件地址：zhengli@mail.tsinghua.edu.cn，来信标题请包含“Java book”。

作者

2006年12月 于清华大学

# 目 录

<b>第 1 章 Java 语言基础知识</b> .....	1
1.1 主要知识点 .....	1
1.1.1 Java 语言与面向对象的程序设计 .....	1
1.1.2 Java 程序概述 .....	2
1.1.3 基本数据类型与表达式 .....	2
1.1.4 数组的概念 .....	4
1.1.5 数组的创建和引用 .....	4
1.2 案例 .....	5
1.2.1 基础案例 .....	5
1.2.2 应用案例 .....	19
<b>第 2 章 类与对象的基本概念</b> .....	21
2.1 主要知识点 .....	21
2.1.1 面向对象的程序设计方法概述 .....	21
2.1.2 类与对象 .....	21
2.1.3 对象初始化和回收 .....	22
2.1.4 UML 简介 .....	23
2.2 案例 .....	23
2.2.1 基础案例 .....	23
2.2.2 应用案例 .....	37
<b>第 3 章 类的方法</b> .....	41
3.1 主要知识点 .....	41
3.1.1 方法的控制流程 .....	41
3.1.2 Java 的异常处理机制 .....	43
3.1.3 方法的重载 .....	43
3.2 案例 .....	43
3.2.1 基础案例 .....	43
3.2.2 应用案例 .....	56
<b>第 4 章 类的重用</b> .....	62
4.1 主要知识点 .....	62



---

4.1.1	类的继承 .....	62
4.1.2	Object 类 .....	63
4.1.3	终结类与终结方法 .....	63
4.1.4	抽象类 .....	63
4.1.5	类的组合 .....	63
4.1.6	包的应用 .....	63
4.2	案例 .....	64
4.2.1	基础案例 .....	64
4.2.2	应用案例 .....	85
<b>第 5 章</b>	<b>接口与多态 .....</b>	<b>92</b>
5.1	主要知识点 .....	92
5.1.1	接口 .....	92
5.1.2	塑型 .....	93
5.1.3	多态的概念 .....	93
5.1.4	多态的应用 .....	94
5.1.5	构造方法与多态 .....	94
5.1.6	内部类 .....	94
5.2	案例 .....	94
5.2.1	基础案例 .....	94
5.2.2	应用案例 .....	114
<b>第 6 章</b>	<b>输入/输出流 .....</b>	<b>120</b>
6.1	主要知识点 .....	120
6.1.1	输入/输出流 .....	120
6.1.2	文件读写 .....	121
6.2	案例 .....	122
6.2.1	基础案例 .....	122
6.2.2	应用案例 .....	138
<b>第 7 章</b>	<b>对象群体的组织 .....</b>	<b>147</b>
7.1	主要知识点 .....	147
7.1.1	对象数组 .....	147
7.1.2	二维数组 .....	148
7.1.3	集合(Collection, Map) .....	148
7.2	案例 .....	149
7.2.1	基础案例 .....	149
7.2.2	应用案例 .....	164

<b>第 8 章 多线程</b> .....	172
8.1 主要知识点 .....	172
8.1.1 多线程编程基础.....	172
8.1.2 线程的生命周期.....	174
8.1.3 线程的优先级.....	174
8.2 案例 .....	174
8.2.1 基础案例.....	174
8.2.2 应用案例.....	189
<b>第 9 章 图形用户界面</b> .....	195
9.1 主要知识点 .....	195
9.1.1 Applet 概述 .....	195
9.1.2 Applet 应用程序接口(API).....	195
9.1.3 绘图.....	195
9.1.4 Swing 基础.....	196
9.1.5 Swing 组件.....	196
9.1.6 其他 Swing 特性 .....	197
9.2 案例 .....	197
9.2.1 基础案例.....	197
9.2.2 应用案例.....	212
<b>第 10 章 JDBC 与数据库访问</b> .....	216
10.1 主要知识点.....	216
10.1.1 数据库基础知识.....	216
10.1.2 通过 JDBC 访问数据库 .....	216
10.2 案例.....	219
10.2.1 基础案例.....	219
10.2.2 应用案例.....	237
<b>第 11 章 Servlet 程序设计</b> .....	242
11.1 主要知识点.....	242
11.1.1 Java 网络程序设计的基本概念 .....	242
11.1.2 Servlet 基础 .....	243
11.1.3 Servlet 的生命周期 .....	243
11.1.4 与客户端交互.....	243
11.1.5 客户端跟踪.....	244
11.1.6 协作与通讯.....	244

---

11.2 案例.....	245
<b>第 12 章 JSP 程序设计</b> .....	<b>265</b>
12.1 主要知识点.....	265
12.1.1 JSP 简介 .....	265
12.1.2 JSP 与 JavaBean .....	266
12.1.3 JSP 标签库 .....	266
12.1.4 JSP 与 Servlet .....	267
12.2 案例.....	267
<b>案例索引</b> .....	<b>294</b>
<b>参考文献</b> .....	<b>296</b>

# 第 1 章

## Java 语言基础知识

Java 语言是一个功能强大的跨平台程序设计语言,是目前应用最为广泛的计算机语言之一。本书将介绍 Java 语言与面向对象的程序设计方法,以及 Java 语言应用的几个专题。本章案例主要演示 Java 程序基本组成、基本数据类型、表达式和类型转换等内容。

### 1.1 主要知识点

#### 1.1.1 Java 语言与面向对象的程序设计

面向对象的编程语言将客观事物看作具有状态和行为的对象,通过抽象找出同一类对象的共同状态(静态特征)和行为(动态特征),构成模型——类。

Java 语言是一个面向对象的程序设计语言。除了面向对象的特点以外,Java 语言还在安全性、平台无关性、支持多线程、内存管理等许多方面具有卓越的优点。

- 面向对象

区别于传统语言,Java 是完全面向对象的语言。Java 语言提供了类的机制,在对象中封装了成员变量和方法,实现了数据的封装和信息隐藏;类提供了一类对象的模型,通过继承和多态,实现了代码的复用。

- 安全性

Java 的内部安全措施保证 Java 程序在 Java 虚拟机规则下操作,防止未授权的程序访问含有专有信息的系统资源或危及客户机的完整性。

- 操作平台无关性

Java 编译器生成与平台无关的字节码指令,只要安装了 Java 运行系统,其程序就可以在任意的处理器上运行。与平台无关的特性使 Java 程序可以方便地移植到不同的机器。

- 多线程

Java 是第一个在语言级提供内置多线程支持的高级语言,这大大简化了多线程程序的编写。

- 内存管理

Java 中所有的对象都是通过动态内存分配建立的,Java 对内存自动进行管理并进行垃圾回收,防止了因程序员失误而导致的内存分配错误,进而更好地利用了系统资源。

Java 类库也称为 Java API(Application Programming Interface)是由大量已经编译

好、经过测试的类组成。

### 1.1.2 Java 程序概述

程序的运行需要一定的硬件和软件环境,这个环境被称为平台(platform)。Java 平台包括 Java 应用程序接口(API)和 Java 虚拟机(Java Virtual Machine,JVM)。

JSP 是 Java Server Pages 的缩写,是 Servlet 技术的扩展,目的是为了简化动态网页的构建和管理。

### 1.1.3 基本数据类型与表达式

#### • 常量与变量

标识符是一个名称,其第一个字符必须是下列字符中的一个:大写字母(A~Z),小写字母(a~z),下划线(\_)或者(\$),后面的字符可以是上述字母或者数字(0~9)中的一个。

在标识符中有一部分被系统定义,用户不能使用,被称为保留字或关键字。关键字列表如下:

abstract	default	if	private	this
boolean	do	implements	protected	throw
break	double	import	public	throws
byte	else	instanceof	return	transient
case	extends	int	short	try
catch	final	interface	static	void
char	finally	long	strictfp	volatile
class	float	native	super	while
const	for	new	switch	
continue	goto	package	synchronized	

变量是由标识符命名的项,它具有类型和作用域,它的值可以被改变。声明一个变量的语法形式如下:

```
数据类型 变量名[=初值][,变量名[=初值]...];
```

在变量声明格式前加上 final 修饰符,就声明了一个常量,常量一旦被初始化就不能被改变。语法形式如下:

```
final 数据类型 变量名[=初值][,变量名[=初值]...];
```

#### • 基本数据类型

基本数据类型是指 Java 固有的数据类型,是编译器本身能理解的,可以分为数值型、布尔型和字符型。

数值型基本数据类型有六种,分别为字节型(byte)、短整型(short)、整型(int)、长整

型(long)、单精度浮点型(float)和双精度浮点型(double)。关于这六种数值类型的说明如表 1-1 所示。

表 1-1 六种数值类型说明表

类型	说明	长度	最小值	最大值
byte	单字节整数	8 位	$-2^7$	$2^7 - 1$
short	短整数	16 位	$-2^{15}$	$2^{15} - 1$
int	整数	32 位	$-2^{31}$	$2^{31} - 1$
long	长整数	64 位	$-2^{63}$	$2^{63} - 1$
float	单精度浮点数	32 位	$-2^{-149}$	$(2 - 2^{-23}) \cdot 2^{127}$
double	双精度浮点数	64 位	$2^{-1074}$	$(2 - 2^{-52}) \cdot 2^{1023}$

#### • 运算符与表达式

运算符也称为操作符,表示对操作数所进行的运算。按照功能,可以把运算符分为算术运算符、赋值运算符、关系运算符、逻辑运算符、位运算符和条件运算符。

算术运算符: + (加)、- (减)、\* (乘)、/ (除)、% (取余)、+ (正号)、- (负号)、++ (自增)、-- (自减),由算术运算符构成的表达式为算术表达式。

赋值运算符: =、+=、-=、\*=、/=、%=、>>=、<<=、>>>=、^=、|=、&=,由赋值运算符构成的表达式为赋值表达式。

关系运算符: <、<=、>、>=、==、!=,由关系运算符构成的表达式为关系表达式。

逻辑运算符: ! (非)、&& (与)、|| (或),由逻辑运算符构成的表达式为逻辑表达式。

位运算符: & (按位与)、| (按位或)、^ (按位异或)、~ (按位取反)、<< (左移运算)、>> (右移运算)、>>> (无符号右移运算符)。

条件运算符:?:实现简单的选择功能。

Java 中运算符的优先次序如表 1-2 所示。

表 1-2 运算符优先级次序表

优先级别	运算符
1	. [] ()
2	++ -- ! ~ instanceof
3	new
4	* / %
5	+ -
6	>> >>> <<
7	< > <= >=

续表

优先级别	运算符
8	== !=
9	&
10	^
11	
12	&&
13	
14	?:
15	= += -= *= /= %= ^=
16	&=  = <<= >>= >>>=

• 表达式中的类型转换

每个表达式都有类型,如果表达式的类型和程序上下文不符,要么产生编译错误,要么发生隐含的类型转换。类型转换可以分成扩展转换(如图 1-1 所示)和窄化转换(如图 1-2 所示)两类。从一种整型扩展转换到另一种整型,或者从 float 转换到 double 时,不损失任何信息。从整型转换到 float 或者 double 将损失精度。窄化转换可能会丢失信息。

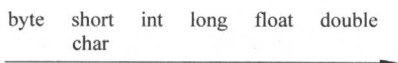


图 1-1 扩展转换

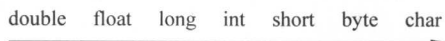


图 1-2 窄化转换

### 1.1.4 数组的概念

数组是由同类型的数据元素构成的一种数据结构。数据元素可以是基本数据类型也可以是引用类型。通过数组名和数组元素下标(或称为索引)可以引用数组中的元素。

每个数组都有一个名为 length 的成员变量,用来表示数组所包含元素的个数,length 只能是正整数或者零。数组创建之后 length 就不能被改变。

### 1.1.5 数组的创建和引用

声明一维数组的语法形式为:

```
数组类型 数组名[];
```

或者:

```
数组类型[] 数组名;
```

包括数组在内,Java 中的所有对象都是在运行时动态创建的,创建新对象的方法之





```

MyFirstProgram myFirstProgram1;           // 声明引用变量
myFirstProgram1 = new MyFirstProgram();    // 创建对象实例
myFirstProgram1.display();                // 调用显示方法
}                                           // 方法 main 定义结束
}                                           // MyFirstProgram 类定义结束

```

程序运行结果如下：

Liutao

### 分析与思考

编写完程序文本后,要将其存储为 MyFirstProgram.java 的文件,并把源文件(.java 文件)编译成字节码文件即类文件(.class),才能由解释器解释执行。

### 【案例 1.2】 数字交换

#### 问题描述

编写一个 Java 程序,将两个变量的值互相交换。

#### 要点解析

由于变量只能保存最新的数据,之前的数据将会丢失,所以要另外定义一个变量来保存先前的数据。

#### 程序与注释

```

class SwapNumbers {
    public static void main(String[] args) {
        int firstNumber = 60;
        int secondNumber = 80;
        int temp;
        System.out.println("数字交换前:");
        System.out.println("firstNumber = " + firstNumber);
        System.out.println("secondNumber = " + secondNumber);
        System.out.println();

        // 开始数字交换
        temp = firstNumber;
        firstNumber = secondNumber;
        secondNumber = temp;

        // 数字交换结束,显示交换结果
        System.out.println("数字交换后:");
        System.out.println("firstNumber = " + firstNumber);
        System.out.println("secondNumber = " + secondNumber);
    } // main 方法结束
} // 类 SwapNumbers 结束

```

程序运行结果如下：