

<<Java程序设计案例教程>>

图书基本信息

书名：<<Java程序设计案例教程>>

13位ISBN编号：9787111325000

10位ISBN编号：7111325001

出版时间：2011-1

出版时间：机械工业出版社

作者：孙一林,彭波

页数：302

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;Java程序设计案例教程&gt;&gt;

## 前言

本书的作者都是从事多年Java语言教学的教育工作者和使用Java开发项目多年的软件工程师，对Java有着深入的理解。

在从事Java语言教学的过程中，他们都或多或少地了解了学生在学习Java语言时遇到的难点以及如何使学生更快、更准确地掌握和使用Java语言。

在此基础上，作者尽量让本书通俗易懂，使其符合循序渐进、由浅入深的学习规则，帮助读者规范、系统地学习Java语言，为后续深入学习Java语言奠定扎实的基础。

计算机语言与人类交流使用的自然语言的功能是一样的，只是计算机语言是人与计算机交流时使用的。

更确切地说，它是人命令计算机做某些事情的语言。

计算机语言也有单词（称为关键字）、词组（称为类库或函数库）、语法规则等，但是其单词是有限的，因为计算机CP[J]中的指令译码器只能理解很少的单词，而语法规则也是规定好的，人们只可以按照语法规则来编写一些语句，否则计算机不会认识。

因此，掌握计算机语言的关键字和语法规则就可以编写计算机认识的可执行语句，而计算机程序就是一些遵循语法规则的语句的逻辑组合，它可以让计算机完成特定的功能。

顾名思义，计算机只能实现计算功能，为什么它能用在各行各业？

因为几乎所有的事物以及事物的改变都可以用“数据”来描述，例如，最常见的阿拉伯数字、ASCII码（描述文字）、PCM码（描述视、音频信息）、表格、图等“广义数据”，处理这些“数据”正是计算机的专长。

因此，编写计算机程序首先需要通过“数据”和对数据的操作来建立一些模型（称之为数学模型），然后计算机执行程序则是针对数据的操作来实现要达到的目的。

目前的面向对象编程的理念就是帮助实现或建立这样的模型，它可以使编写计算机程序变得简单。

掌握面向对象编程思想是编写应用程序的基础，而Java语言就是适用于面向对象编程的语言。

另外，计算机语言除了具有关键字和规定的语法规则外，还提供了大量的类库或函数库，这些类库或函数库则是针对各种类型的“数据”实施处理的。

在编写应用程序时，几乎所有需要处理的数据类型都可以在类库或函数库中找到处理方法，类库或函数库可以说是包罗万象。

总之，要学习计算机语言，在理解了其使用的编程思想后，剩下的就是学习关键字、语法规则和类库或函数库的使用了。

本书分为两个部分：第一部分包含6章，主要讲解Java语言的关键字、由关键字组成的表达式、程序流控制、类定义、接口定义、对象的使用、异常的处理等语句以及Java语言的语法规则；第二部分包含9章，主要是关于Java基础类库JFC的应用。

在JFC中每一个类库都是针对一类数据进行的操作，类库提供了许多数据操作方法，每一类操作都制定了操作机制，了解操作机制是更好地使用类库的前提。

Sun公司提供的J2SDK中类库使用说明文档是最直接的使用类库的学习资料，本书则是通过案例介绍类库的操作机制和使用方法，案例涉及的内容有计算机输入和输出操作、图形用户界面操作、Applet小程序的编写、图形绘制操作、多线程处理、图像动画制作、网络编程、数据库操作等，为读者深入学习各个应用领域中的由Sun公司提供的JavaAPI奠定一些基础。

## <<Java程序设计案例教程>>

### 内容概要

本书阐述了Java语言的关键字、语句、语法规则、类库等的使用方法，其内容包括Java语言开发工具的介绍、面向对象编程的概念、Java基础语句、Java类、类的继承与多态等特性、接口、包、异常、Java基础类库的应用，并通过应用型案例帮助读者掌握Java语言的输入和输出操作、图形用户界面操作、Applet小程序的编写、图形绘制操作、多线程处理、图像动画制作、网络编程、数据库操作等相关类库的使用。

本书可作为大学、大专等院校计算机相关专业的本科或专科生的教材，也可作为学习Java语言的初、中级读者的参考书。

## &lt;&lt;Java程序设计案例教程&gt;&gt;

## 书籍目录

|       |              |                |                               |           |              |                   |              |       |
|-------|--------------|----------------|-------------------------------|-----------|--------------|-------------------|--------------|-------|
| 前言    | 教学建议         | 第一部分           | 基础原理篇                         | 第1章       | Java程序设计基础   | 1.1               | Java程序与运行    | 1     |
| 1.1.1 | Java语言的特征    | 1.1.2          | Java虚拟机-                      | 1.1.3     | Java字节代码     | 1.1.4             |              |       |
|       | Java编译单元     | 1.2            | Java程序的开发                     | 1.2.1     | Java语言程序开发平台 | 1.2.2             |              |       |
|       | J2SDK        | 1.2.3          | 在Windows操作系统中搭建Java程序的开发和运行环境 | 1.2.4     | Java程序的编译和运行 |                   |              |       |
|       | 1.3          | Java语言的标识符和关键字 | 1.3.1                         | Java注释语句  | 1.3.2        |                   |              |       |
|       | Java标识符      | 1.3.3          | Java关键字                       | 1.4       | Java基本数据类型   | 1.4.1             | 数据值的表示       |       |
|       | 法            | 1.4.2          | 变量                            | 1.4.3     | 变量的数据类型转换    | 1.4.4             | 常量           | 1.5   |
|       | Java运算符和表达式  | 1.5.1          | Java运算符                       | 1.5.2     | Java表达式      | 1.5.3             | 表达式中运算符的使用规则 |       |
|       | 1.6          | Java程序流控制语句    | 1.6.1                         | 分支结构语句    | 1.6.2        | 循环结构语句            |              |       |
|       | 1.6.3        | 辅助流控制语句        | 1.6.4                         | 流控制语句应用示例 | 1.7          | 小结                | 1.8          |       |
| 习题    | 第2章          | 面向对象编程         | 2.1                           | 面向对象程序设计  | 2.1.1        | 面向对象的程序设计方法       |              |       |
|       | 2.1.2        | Java面向对象程序设计   | 2.2                           | Java类     | 2.2.1        | Java类的定义          | 2.2.2        |       |
|       | 方法的定义        | 2.2.3          | Java修饰符与其权限                   | 2.3       | Java对象       | 2.3.1             | 对象的创建        |       |
|       | 2.3.2        | 构造方法的使用        | 2.3.3                         | 对象的使用     | 2.3.4        | Java对象的清除         | 2            |       |
|       | 2.3.5        | Java类和对象的关系    | 2.4                           | 数组对象      | 2.4.1        | 一维数组的声明和创建        | 2.4          |       |
|       | 2.2          | 多维数组的声明和创建     | 2.4.3                         | 数组的应用     | 2.5          | 小结                | 2.6          | 习题    |
|       | Java类的继承与多态  | 3.1            | Java类的继承                      | 3.1.1     | 概念和语法        | 3.1.2             | Java类继承关系的测试 |       |
|       | 3.1.3        | 隐藏、覆盖和重载       | 3.1.4                         | 构造方法的重载   | 3.2          | abstract和final修饰符 |              |       |
|       | 3.2.1        | absffact修饰符    | 3.2.2                         | final修饰符  | 3.3          | tllis和super变量     |              |       |
|       | 3.3.1        | tllis变量        | 3.3.2                         | super变量   | 3.4          | Java的多态性          | 3.4.1        | 多态的概念 |
|       | 3.4.2        | 多态的应用          | 3.4.3                         | 构造方法与多态   | 3.5          | 小结                | 3.6          | 习题    |
|       | Java接口和Java包 | 4.1            | Java接口                        | 4.1.1     | 接口的定义        | 4.1.2             | 接口的实现        |       |
|       | .....        | 第二部分           | 案例应用篇                         |           |              |                   |              |       |

## 章节摘录

插图：本章介绍Java异常处理机制的相关知识，包括异常处理语法规则和语句的使用，以及异常的捕获和抛出等。

5.1 Java异常处理机制广义的异常是指应用程序在编译和运行期间出现的错误，在编译时出现的错误会被编译器指出，通过修改代码可以完成纠正，但在运行期间出现的错误将直接影响应用程序的正常运行。

严格地讲，异常是指应用程序在运行期间出现的错误。

5.1.1 异常的类型在程序执行期间产生异常的情况有很多，例如，非法的变量赋值、整数被0除、访问数组下标越界、将要打开的文件不存在、在网络中数据传输被中断、计算机系统资源耗尽等。

应用程序在运行期间发生错误将导致程序的正常执行流程被中断，异常可能造成程序终止运行或失去控制，更为严重的可能会造成计算机系统的瘫痪。

一般应用程序在运行期间是可能会发生错误的，错误大致有两类，一类是程序逻辑错误，逻辑错误一般是人为造成的，其修正还需要人来处理；另一类是程序控制操作时可能发生的错误，这类错误大多数是与某种操作相关联的，即该操作可能会引发某些错误，这类错误通常是已知的或是可预见的。

有些错误是不可控制的，例如，Java虚拟机产生错误；而有些错误是可控制的，例如，整数被0除、访问数组下标越界、要打开的文件不存在等错误是可以处理和预防的，而异常处理就是针对这类可控制的错误而言的。

5.1.2 异常处理机制异常处理是指当程序运行时出现错误后该程序将如何处理错误。

因为程序运行时发生异常总是在应用程序做操作处理期间，因此，在编写操作代码时，都会对每一个可能发生异常的操作代码做一些前期的评估，判断是否有潜在的可能要发生的异常，并在程序运行期间对所有可能出现的异常进行技术处理，确保应用程序安全、可靠地运行。

在传统的程序设计语言中，异常的处理一般是在设计应用程序主体逻辑的同时就要考虑到可能出现的错误，并根据错误情况设计处理异常的逻辑。

通常的做法是根据调用函数返回的错误代码（标志）来了解函数执行是否出现错误，并做出相应的处理，而这种检测和处理错误的方式也会带来一些负面影响，例如，程序代码的逻辑相对复杂（程序主体逻辑和错误处理代码混杂在一起）、错误信息不够准确等。

在面向对象的程序设计语言中，异常的处理被规范化了，它将各种不同的异常进行分类，并提供良好的异常处理接口，制定了异常处理机制，在编写程序代码时将异常处理代码与常规代码分离，当异常发生时，应用程序运行的流程就会发生改变，其程序的控制权将转移到异常处理代码部分，完成异常的处理，或者使应用程序安全地退出运行。

面向对象程序设计的异常处理模式可以使编程人员从编写处理异常逻辑中解脱出来，但同时又可以有的放矢地对异常进行处理。

## <<Java程序设计案例教程>>

### 编辑推荐

《Java程序设计案例教程》：阐述Java面向对象基础编程机制介绍Java基础类库的使用规则重点给出具有应用价值的程序案例详解《Java程序设计案例教程》包含两大部分内容，一部分为、Java语言基础知识，重点介绍Java语言的关键字、由关键字组成的表达式、程序流控制、类定义、接口定义、对象的使用、异常的处理等语句以及Java语言的语法规则、面向对象编程等；另一部分为JavaAPI的应用，重点介绍Java基础类库JFC的使用，通过针对性案例循序渐进地帮助读者学习并掌握Java基础类库的应用，同时可将案例直接移植到实际应用项目中。

《Java程序设计案例教程》特色·全面介绍Java语言自身功能以及通过Java语言实现面向对象编写程序

- 通过各种类型的应用案例介绍Java语言提供的基础类库、JFC的使用。
- 注重基础和实用性。
- 通过对Java语言和案例的介绍，帮助读者由浅入深地掌握Java语言以及Java的基础类库并逐步具备使用、Java语言开发实际应用程序的能力。

<<Java程序设计案例教程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>