

## <<Java程序设计>>

### 图书基本信息

书名：<<Java程序设计>>

13位ISBN编号：9787030344649

10位ISBN编号：7030344642

出版时间：2012-6

出版时间：科学出版社

作者：刘德山，金百东，张建华 主编

页数：330

字数：491000

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>

## <<Java程序设计>>

### 内容概要

《普通高等教育"十二五"重点规划教材·计算机系列·中国科学院教材建设专家委员会"十二五"规划教材:Java程序设计》以通俗易懂的语言、翔实的案例、全新的内容组织来介绍Java语言。

全书分为11章,包括Java语言概述、Java语言基础知识、类与对象、继承与多态、数组和Java的常用类、集合类与泛型、异常处理、多线程、输入输出及文件操作、图形用户界面、网络和数据库编程等内容。

《普通高等教育"十二五"重点规划教材·计算机系列·中国科学院教材建设专家委员会"十二五"规划教材:Java程序设计》内容以应用为核心展开,力求以知识的最小集来实现最大范围的应用。

## <<Java程序设计>>

### 书籍目录

#### 前言

#### 第1章 Java语言概述

##### 1.1 Java语言简介

###### 1.1.1 Java语言的产生

###### 1.1.2 Java语言的版本

###### 1.1.3 Java语言的特点

###### 1.1.4 Java与C++的比较

##### 1.2 Java程序示例

###### 1.2.1 Java Application

###### 1.2.2 Java Applet

##### 1.3 Java程序的开发和运行环境

###### 1.3.1 JDK的安装

###### 1.3.2 设置环境变量

###### 1.3.3 Java开发环境

##### 1.4 Java程序的执行过程

#### 习题

#### 第2章 Java语言基础知识

##### 2.1 数据类型、常量与变量

###### 2.1.1 Java语言的数据类型

###### 2.1.2 标识符、关键字、常量和变量

##### 2.2 运算符和表达式

##### 2.3 流程控制语句

###### 2.3.1 程序的基本流程

###### 2.3.2 分支结构

###### 2.3.3 循环结构

###### 2.3.4 跳转语句

##### 2.4 程序注释与编码规范

###### 2.4.1 程序注释

###### 2.4.2 编码规范

##### 2.5 综合示例

#### 习题

#### 第3章 类与对象

##### 3.1 类概述

##### 3.2 对象概述

##### 3.3 类中变量

###### 3.3.1 变量分类

###### 3.3.2 变量初始化

##### 3.4 类中方法

###### 3.4.1 方法定义

###### 3.4.2 方法的内存分配

###### 3.4.3 构造方法

###### 3.4.4 方法参数

###### 3.4.5 方法重载

##### 3.5 this关键字

##### 3.6 目录访问机制

## &lt;&lt;Java程序设计&gt;&gt;

- 3.6.1 classpath存在的问题
- 3.6.2 package
- 3.6.3 import
- 3.7 访问权限
  - 3.7.1 变量和方法的访问权限
  - 3.7.2 类的访问权限
- 3.8 setor与getor
- 3.9 综合示例
- 习题
- 第4章 继承与多态
  - 4.1 继承的概念
  - 4.2 继承中的权限控制
  - 4.3 子类对象构造过程
  - 4.4 继承中方法重载与重写
    - 4.4.1 重载与重写的含义
    - 4.4.2 重写访问控制约束
  - 4.5 super关键字
  - 4.6 多态
    - 4.6.1 方法多态
    - 4.6.2 逻辑多态思想
    - 4.6.3 逻辑多态实现方法
    - 4.6.4 进一步理解逻辑多态
  - 4.7 instanceof运算符
  - 4.8 抽象类
  - 4.9 接口
    - 4.9.1 接口定义
    - 4.9.2 接口实现
    - 4.9.3 接口派生
  - 4.10 深入理解普通类、抽象类、接口关系
    - 4.10.1 三者关系
    - 4.10.2 一个编程模型
    - 4.10.3 多态在方法参数中的作用
  - 4.11 内部类
    - 4.11.1 内部类定义
    - 4.11.2 内部类的作用
    - 4.11.3 静态内部类
  - 4.12 匿名类
    - 4.12.1 匿名类定义
    - 4.12.2 匿名类作用
  - 4.13 final类和final方法
- 习题
- 第5章 数组与Java的常用类
  - 5.1 数组
    - 5.1.1 一维数组
    - 5.1.2 多维数组
    - 5.1.3 复制数组
  - 5.2 Java基础类库

## &lt;&lt;Java程序设计&gt;&gt;

5.3 Object类

5.4 字符串类

5.4.1 String类

5.4.2 StringBuffer类

5.4.3 StringTokenizer类

5.5 Math类

5.6 包装类

5.7 System类

5.8 日期类

5.8.1 Date类

5.8.2 Calendar类及GregorianCalendar类

5.9 Random类

5.10 综合示例

习题

第6章 集合类与泛型

6.1 集合框架与Collection接口

6.1.1 集合框架

6.1.2 Collection接口

6.2 Set接口及子类实现

6.3 List接口及子类实现

6.3.1 List接口

6.3.2 Vector类与Stack类

6.4 Map接口及子类实现

6.5 集合的遍历与Iterator接口

6.6 泛型

6.6.1 泛型类

6.6.2 泛型接口

6.6.3 泛型在集合框架中的应用

6.7 Collections和Arrays

6.7.1 Collections类

6.7.2 Arrays类

6.8 综合示例

习题

第7章 异常处理

7.1 异常处理概述

7.1.1 异常的概念

7.1.2 异常示例

7.1.3 异常类的层次结构

7.2 异常处理机制

7.2.1 捕获异常

7.2.2 声明异常

7.2.3 抛出异常

7.3 用户自定义异常类

习题

.....

第8章 多线程

第9章 输入输出及文件操作

<<Java程序设计>>

第10章 图形用户界面  
第11章 网络和数据库编程  
参考文献

## 章节摘录

前面各章节的程序在运行时，一个程序只有一条执行线索。

从线程的观点看，这是单线程的程序。

然而现实世界里的很多过程都具有多条线索同时执行的特征。

例如，我们可以一边喝咖啡，一边听音乐：一个网络服务器可能要同时处理几个客户机的请求等。

如果将这些活动映射到计算机中，从线程的观点看，是多线程。

Java语言的多线程机制为模拟这类问题提供了一个很好的环境。

每个Java程序都可以创建多个线程，每个线程可以完成一个相对独立的任务，且能够与其他线程并行执行。

例如，当从网络服务器上下载一个非常耗时的文件时，CPU资源被占用的时间非常少。

如果是单线程环境，一次只能运行一个任务，只有前一个任务完成后，才能开始执行下一个任务，这就浪费了CPU资源。

在Java的多线程环境下，可以有多个线程同时运行，当一个线程暂时不需要CPU时，另一个线程就可以占用CPU资源。

所以，多线程机制有助于充分利用CPU资源，提高程序的运行效率。

本章将详细介绍Java语言的多线程技术，包括线程和进程的概念、创建线程、线程的调度和线程的同步等内容。

8.1 线程的概念      8.1.1 线程和进程      要理解线程，我们首先来了解进程。

进程是程序的动态执行过程，每一个进程都占有独立的内存空间、独立的内部数据、独立的系统资源。

多进程是指在操作系统中能同时运行多个程序。

这些进程之间相互独立，一个进程一般不允许访问其他进程的内存空间，因此，进程间通信非常困难。

例如，在Windows操作系统中，同时运行的“计算器”程序和“IE浏览器”可以理解为两个进程。

线程与进程类似，是一段完成某个特定功能的代码，但线程是比进程更小的执行单位。

与进程不同，同一个类的多个线程是共享一块内存空间和一组系统资源，这样，多个线程间切换时，负担要比进程小得多，正因如此，线程被称为轻量级进程。

概括地说，一个进程在其执行过程中，可以产生多个线程。

每个线程是进程内部单一的一个执行流。

多线程指的是在单个程序中可以同时运行多个不同的线程，执行不同的任务。

在基于线程的多任务处理环境中，线程是最小的处理单位。

一个进程内的多个线程可以共享一块内存空间和一组系统资源，有可能互相影响。

线程的切换比进程的切换负担要小。

.....

<<Java程序设计>>

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>