

## <<Java程序设计之网络编程基础教程>>

### 图书基本信息

书名：<<Java程序设计之网络编程基础教程>>

13位ISBN编号：9787302299417

10位ISBN编号：7302299412

出版时间：2012-12

出版时间：清华大学出版社

作者：李芝兴,杨瑞龙

页数：414

字数：664000

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程序设计之网络编程基础教程>>

### 内容概要

《21世纪高等学校计算机基础实用规划教材：Java程序设计之网络编程基础教程》从基础知识和网络编程两个方面出发，全面介绍java相关的各项技术。

《21世纪高等学校计算机基础实用规划教材：Java程序设计之网络编程基础教程》力求重点突出、层次清晰、立论严谨、语言通俗易懂，各章均提供丰富的示例和练习，并有配套的实验用书《java程序设计之实验及课程设计教程》。

《21世纪高等学校计算机基础实用规划教材：Java程序设计之网络编程基础教程》既可作为高等院校应用型本科（含部分专科、高职类）相关专业程序设计类课程的教材，也适合于编程开发人员培训及广大计算机技术爱好者自学使用。

书籍目录

第1章 java语言概述

1.1 java语言的产生及其特点

1.1.1 java语言发展简史

1.1.2 java虚拟机

1.1.3 java平台是不断扩展的计算平台

1.1.4 java语言的特点

1.1.5 java与c/c++语言的异同

1.2 java运行环境与开发环境

1.2.1 java运行环境

1.2.2 java sdk开发环境

1.2.3 jdk开发环境工具简单介绍

1.3 java程序举例

1.3.1 简单的java应用程序

1.3.2 简单的applet小程序

1.3.3 servlet

1.4 其他集成运行环境

1.5 eclipse开发环境的搭建

习题及思考

第2章 数据类型及其运算

2.1 标识符和关键字

2.1.1 标识符

2.1.2 关键字

2.1.3 注释

2.2 变量和常量

2.2.1 变量

2.2.2 常量

2.3 基本数据类型

2.3.1 布尔类型

2.3.2 字符类型

2.3.3 整数类型

2.3.4 浮点类型

2.3.5 各类型数据间的相互转换

2.4 运算符

2.4.1 算术运算符

2.4.2 赋值运算符

2.4.3 条件运算符

2.4.4 位运算符

2.4.5 关系运算符

2.4.6 逻辑运算符

2.5 表达式

2.6 数组

2.6.1 一维数组

2.6.2 二维数组

2.7 字符串

习题及思考

### 第3章 程序控制语句

#### 3.1 选择语句

##### 3.1.1 if语句

##### 3.1.2 switch语句

#### 3.2 循环语句

##### 3.2.1 while语句

##### 3.2.2 do-while循环

##### 3.2.3 for循环

#### 3.3 跳转语句

##### 3.3.1 使用break语句

##### 3.3.2 使用continue语句

##### 3.3.3 使用return语句

#### 习题及思考

### 第4章 java面向对象程序设计基础

#### 4.1 面向对象的基本概念

##### 4.1.1 面向对象编程的概念

##### 4.1.2 客观事物的抽象

#### 4.2 类的定义

##### 4.2.1 类声明

##### 4.2.2 类体

##### 4.2.3 实例化对象

##### 4.2.4 构造方法说明

##### 4.2.5 对象的清除

##### 4.2.6 方法重载

#### 4.3 类和对象的使用

##### 4.3.1 实例变量和类变量

##### 4.3.2 实例方法和类方法

#### 4.4 包package

##### 4.4.1 package语句

##### 4.4.2 import语句

##### 4.4.3 编译和运行包

#### 4.5 成员变量及方法的访问权限

##### 4.5.1 公共类型 ( public )

##### 4.5.2 保护类型 ( protected )

##### 4.5.3 私有类型 ( private )

#### 4.6 final、this及其他

##### 4.6.1 final关键字

##### 4.6.2 this引用

##### 4.6.3 super关键字

##### 4.6.4 null关键字

##### 4.6.5 java.lang.object类介绍

##### 4.6.6 递归方法

##### 4.6.7 命令行参数的输入

##### 4.6.8 jar文件的使用

#### 习题及思考

### 第5章 面向对象高级程序设计

#### 5.1 继承

- 5.1.1 创建子类
- 5.1.2 成员变量的隐藏和方法的重写
- 5.1.3 super
- 5.1.4 对象的上转型对象
- 5.2 多态性
- 5.3 抽象类和抽象方法
- 5.4 接口
  - 5.4.1 接口声明
  - 5.4.2 使用接口的优点
- 5.5 枚举类型
- 5.6 annotation
  - 5.6.1 jdk内置的基本注释
  - 5.6.2 自定义annotation注释
- 5.7 内部类和匿名类
  - 5.7.1 内部类的定义
  - 5.7.2 内部类特性
  - 5.7.3 匿名类
- 5.8 简单案例
- 习题及思考
- 第6章 字符串处理
  - 6.1 string类
    - 6.1.1 字符串的构造
    - 6.1.2 string类的常用方法
  - 6.2 stringbuffer类
    - 6.2.1 stringbuffer类的构造方法、声明和实例化
    - 6.2.2 stringbuffer类的常用方法
  - 6.3 stringtokenizer类的应用
    - 6.3.1 构造方法
    - 6.3.2 stringtokenizer类的常用方法
  - 6.4 字符串与其他数据类型的转换
    - 6.4.1 其他数据转换为字符串
    - 6.4.2 字符串转化成其他数据
- 习题及思考
- 第7章 java标准类库
  - 7.1 简单类型包装器类
    - 7.1.1 抽象包装器类number
    - 7.1.2 浮点包装器类型
    - 7.1.3 整型包装器类
    - 7.1.4 字符包装器类
    - 7.1.5 布尔包装器类
    - 7.1.6 自动装箱与拆箱
  - 7.2 system类
    - 7.2.1 记录程序执行的时间
    - 7.2.2 复制数组
    - 7.2.3 访问jvm环境属性
  - 7.3 runtime类
  - 7.4 math类

## 7.5 日期时间实用工具类

### 7.5.1 日期类date

### 7.5.2 日历类

### 7.5.3 日期的格式化与解析

## 7.6 java类集合

### 7.6.1 集合接口

### 7.6.2 list接口实现类

### 7.6.3 set接口实现类

### 7.6.4 通过迭代接口访问类集合

### 7.6.5 泛型简介

### 7.6.6 映射接口map

### 7.6.7 map接口实现的类

### 7.6.8 比较器comparator

### 7.6.9 通用类集算法

### 7.6.10 数组类arrays

### 习题及思考

## 第8章 java异常处理

### 8.1 什么是异常

### 8.2 异常处理机制

#### 8.2.1 java的异常处理机制

#### 8.2.2 异常类的类层次

#### 8.2.3 throwable类的常用方法

### 8.3 异常的处理

#### 8.3.1 捕获异常try-catch-finally

#### 8.3.2 声明抛弃异常

#### 8.3.3 抛出异常

### 8.4 创建用户异常类

### 习题及思考

## 第9章 输入输出处理

### 9.1 i/o流的概念和划分

#### 9.1.1 理解数据流

#### 9.1.2 java流的划分

### 9.2 文件的操作

#### 9.2.1 file类

#### 9.2.2 目录

### 9.3 字节流处理

#### 9.3.1 inputstream/outputstream

#### 9.3.2 标准输入输出流

#### 9.3.3 文件字节流

#### 9.3.4 过滤流

#### 9.3.5 随机存取文件

#### 9.3.6 其他字节流

### 9.4 字符流处理

#### 9.4.1 reader/writer

#### 9.4.2 文件字符流

#### 9.4.3 字节流向字符流的转化

#### 9.4.4 scanner及其他字符流

## 9.5 串行化

### 9.5.1 串行化的概念

### 9.5.2 串行化的方法

### 9.5.3 定制串行化

### 习题及思考

## 第10章 线程

### 10.1 线程的概念

### 10.2 线程的创建

#### 10.2.1 采用继承创建线程

#### 10.2.2 通过实现接口创建线程

### 10.3 线程的生命周期及调度

#### 10.3.1 线程生命周期

#### 10.3.2 线程调度和优先级

### 10.4 线程互斥

#### 10.4.1 问题的提出

#### 10.4.2 互斥对象

### 10.5 线程同步

### 10.6 线程通信

### 10.7 线程死锁

### 10.8 线程池

#### 10.8.1 固定尺寸线程池

#### 10.8.2 可变尺寸线程池

### 习题及思考

## 第11章 gui程序设计

### 11.1 jfc简介

### 11.2 swing组件的结构

#### 11.2.1 类层次结构

#### 11.2.2 mvc模式

### 11.3 顶层容器类

#### 11.3.1 jframe

#### 11.3.2 jdialog、jwindow和japplet

### 11.4 布局管理

#### 11.4.1 无布局管理器布局

#### 11.4.2 flowlayout

#### 11.4.3 borderlayout

#### 11.4.4 gridlayout

#### 11.4.5 其他布局管理器

#### 11.4.6 复杂界面布局

### 11.5 事件处理

#### 11.5.1 事件处理模型

#### 11.5.2 事件类

#### 11.5.3 事件监听器

#### 11.5.4 事件适配器

#### 11.5.5 键盘与鼠标事件

### 11.6 swing组件

#### 11.6.1 按键jbutton

#### 11.6.2 文本标签jlabel

- 11.6.3 单行文本框jtextfield
- 11.6.4 文本输入区jtextarea
- 11.6.5 jtable和jtree
- 11.6.6 菜单
- 11.7 图形编程
  - 11.7.1 awt图形api
  - 11.7.2 java2d图形api
  - 11.7.3 图形相关的父类方法
- 11.8 加载和使用多媒体资源
  - 11.8.1 加载图像
  - 11.8.2 播放音频文件
  - 11.8.3 jar文件中多媒体资源的加载
- 习题及思考
- 第12章 java applet
  - 12.1 applet程序设计基础
    - 12.1.1 加载applet
    - 12.1.2 applet的安全性限制
    - 12.1.3 applet的生命周期
  - 12.2 java.applet包
    - 12.2.1 applet类的继承关系
    - 12.2.2 applet类的方法
    - 12.2.3 html和applet
    - 12.2.4 java插件技术
    - 12.2.5 appletviewer
    - 12.2.6 applet的显示
  - 12.3 applet中的多媒体应用
    - 12.3.1 url
    - 12.3.2 加载图像
    - 12.3.3 播放声音
  - 12.4 applet中的事件处理
    - 12.4.1 鼠标事件
    - 12.4.2 键盘响应
  - 12.5 japplet和applet
- 习题及思考
- 第13章 jdbc
  - 13.1 jdbc的结构
    - 13.1.1 jdbc数据库应用模型
    - 13.1.2 jdbc驱动程序
  - 13.2 通过jdbc访问数据库
    - 13.2.1 基本流程
    - 13.2.2 常用的jdbc api
    - 13.2.3 事务
  - 13.3 数据库存取优化
    - 13.3.1 prepared语句
    - 13.3.2 存储过程
  - 13.4 jdbc编程实例
    - 13.4.1 odbc配置

## <<Java程序设计之网络编程基础教程>>

13.4.2 jdbc编程实例

13.4.3 java数据类型和sql数据类型间的关系

习题及思考

第14章 网络通信

14.1 网络通信简介

14.2 url通信

14.2.1 url简介

14.2.2 url类

14.2.3 通过字节流访问www资源

14.2.4 通过urlconnection实现双向通信

14.2.5 使用httpurlconnection

14.3 socket通信

14.3.1 服务器程序

14.3.2 客户机程序

14.3.3 服务多个客户

14.3.4 数据报通信

14.4 远程方法调用

习题及思考

第15章 web服务器端编程

15.1 在tomcat上运行servlet及jsp的简单例子

15.2 servlet介绍

15.2.1 servlet的概念

15.2.2 servlet应用范围和运行环境

15.2.3 servlet常用类、接口和生命周期

15.2.4 servlet应用举例

15.3 jsp介绍

15.3.1 jsp的概念

15.3.2 jsp的运行方式

15.3.3 jsp指令介绍

15.3.4 jsp中的隐藏对象

15.3.5 jsp应用举例

15.4 jsp和servlet协同工作

习题及思考

第16章 轻型框架介绍

16.1 hibernate、struts和spring介绍

16.1.1 hibernate框架介绍

16.1.2 struts框架介绍

16.1.3 spring框架介绍

16.1.4 轻型框架的流行

16.2 hibernate案例实现

16.2.1 hibernate的体系结构

16.2.2 hibernate的文档和软件

16.2.3 hibernate的简单案例

16.3 hibernate synchronizer插件

习题及思考

附录上机实验作业

实验1java程序基础

实验2面向对象程序设计基础

实验3掌握继承、接口、内部类和包

实验4熟练掌握字符串处理技术

实验5掌握日期处理技术

实验6掌握线程编程技术

实验7掌握gui编程技术

实验8掌握简单的网络通信技术

实验9综合试验--网上书店后台设计

参考文献

## 章节摘录

版权页：插图：Java语言为什么这么流行？

其中一个非常重要的原因，就是它是纯粹的面向对象的编程语言（Object Oriented Programming，OOP）。

本章讲述面向对象程序设计的基本概念、特点、类的定义、包、封装以及如何编写简单的面向对象程序等基础知识。

4.1面向对象的基本概念 4.1.1面向对象编程的概念 很多人使用过FORTRAN、BASIC、C等面向过程的程序设计语言，这些语言是按流程化的思想来组织的。

在这些语言的设计思想中，通常将存放基本数据类型的变量作为程序处理对象、以变量的赋值作为程序的基本操作、以变量值的改变作为程序运行的状态。

这种程序设计风格存在着数据抽象简单、信息完全暴露、算法复杂、无法很好地描述客观世界等缺点。

在程序设计过程中，为了实现有限度的代码重用，公共代码被组织成为过程或函数。

当需要代码重用时，调用已经组织好的过程或函数。

在这种应用方式中，如果软件项目庞大，程序的调试和维护将变得异常困难。

面向对象的程序设计思想是将数据以及对于这些数据的操作，封装在一个单独的数据结构中。

这种模式更近似于现实世界，这里所有的对象都同时拥有属性以及与这些属性相关的行为。

对象之间的联系是通过消息实现的，消息是请求对象执行某一处理或回答某些信息的要求。

某个对象在执行相应的处理时，可以通过传递消息请求其他对象完成某些处理工作或回答某些消息。

其他对象在执行所要求的处理活动时，同样可以通过传递消息和另外的对象联系。

所以，一个面向对象程序的执行，就是靠对象间传递消息来完成的。

面向对象程序设计是一种先进的程序设计方法，或者是一种先进的程序设计规范，它使用对象、类、继承、封装、消息等基本概念来进行程序的设计。

从现实世界中客观存在的事物（即对象）出发来构造软件系统，并且在系统构造中尽可能运用人类的自然思维方式。

开发一个软件是为了解决某些问题，这些问题所涉及的业务范围称做该软件的问题域。

其应用领域不仅仅是软件，还有计算机体系结构和人工智能等。

那么，面向对象程序设计有哪些特点呢？

简单地说就是封装、继承、多态三大特点。

1.封装 封装就是把对象的属性和对这些属性的操作封装在一个单独的数据结构中，并尽可能隐蔽对象的内部细节，包含两个含义：把对象的全部属性和对属性的全部操作结合在一起，形成一个不可分割的独立单元（即对象）；信息隐蔽，即尽可能隐蔽对象的内部细节，对外形成一个边界（或者说形成一道屏障），只保留有限的对外接口使之与外部发生联系。

封装的原则在软件上的反映是，要求对象以外的部分不能随意存取对象的内部数据，从而有效地避免了外部错误对它的影响，使软件错误能够局部化，这样可大大减少查错和排错的难度。

2.继承 继承是一种由已有的类创建新类的机制。

利用继承，可以先创建一个拥有共有属性的一般类，根据该一般类再创建具有特殊属性的新类，新类继承一般类的状态和行为，并根据需要增加它自己的新的状态和行为。

由继承而得到的类称为子类，被继承的类称为父类或超类。

Java不支持多重继承，子类只能有一个父类。

在Java语言中，通过继承可利用已有的类，并扩展它的属性和方法。

这个已有的类可以是语言本身提供的、其他程序员编写的或程序员原来编写的。

继承在Java中无所不在。

3.多态 对象的多态是由封装和继承引出的面向对象程序设计语言的另一特征。

主要体现在两个方面：方法重载时实现的静态多态和方法重载时实现的动态多态。

多态性使得同一方法可以有多种形式。

另外，父类中定义的属性或方法被子类继承之后，可以具有不同的数据类型或表现出不同的行为，同一个属性或方法在父类及其各个子类中可以具有不同的语义。

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>