

## <<面向对象程序设计>>

### 图书基本信息

书名：<<面向对象程序设计>>

13位ISBN编号：9787302265221

10位ISBN编号：7302265224

出版时间：2011-10

出版时间：清华大学出版社

作者：李延军 等编著

页数：167

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>

## <<面向对象程序设计>>

### 内容概要

本书强调C++程序设计的基本方法，在内容安排上循序渐进，突出重点。

从C到C++语言的基本数据类型与基本控制结构入手，逐步过渡到函数、类与对象、继承、模板、多态等复杂的C++机制，使读者准确找到编程技术要领，能在较短的时间内掌握面向对象程序设计(OOP)的基本概念和技术方法。

全书采用引入知识点、用例验证知识点、变化实例综合知识点的模式，由浅入深地展开对技术内容的讲述，将前后知识串联力求做到融会贯通。

最后以五子棋游戏和基于MFC的图像处理DIY作为综合实例，在快乐学习的过程中帮助读者理解OOP的重要概念，提高设计、编写、调试程序的能力。

本书结构紧凑，内容丰富，可供高校信息类专业学生使用，也可作为C++语言的培训教材和各类自学读者学习参考。

# <<面向对象程序设计>>

## 书籍目录

### 第1章 概述

#### 1.1 程序设计概述

##### 1.1.1 计算机语言和程序

##### 1.1.2 结构化程序设计

##### 1.1.3 面向对象程序设计

#### 1.2 C++语言简介

##### 1.2.1 C++的起源和特点

##### 1.2.2 开发过程

##### 1.2.3 最简单的程序

#### 1.3 C++程序上机步骤

##### 1.3.1 Visual C++6.0上机指南.

##### 1.3.2 程序调试

#### 本章小结

#### 习题与上机

### 第2章 从C到C++

#### 2.1 C++的数据类型与输入输出

##### 2.1.1 C++的数据类型

##### 2.1.2 输入输出

#### 2.2 表达式和语句

##### 2.2.1 运算符与表达式

##### 2.2.2 流程控制语句

#### 2.3 数组

#### 2.4 指针

##### 2.4.1 指针与数组

##### 2.4.2 指针与函数

#### 2.5 内存的动态分配和释放

#### 本章小结

#### 习题与上机

### 第3章 函数

#### 3.1 函数概述

##### 3.1.1 函数的定义与声明

##### 3.1.2 函数原型

#### 3.2 函数的参数传递

#### 3.3 内联函数和递归函数

#### 3.4 具有默认参数的函数

#### 3.5 函数重载

#### 3.6 作用域和可见性

#### 本章小结

#### 习题与上机

### 第4章 类和对象

#### 4.1 类

##### 4.1.1 结构与类

##### 4.1.2 类的定义

##### 4.1.3 类的成员函数

##### 4.1.4 类成员的访问控制

## <<面向对象程序设计>>

- 4.1.5 类的作用域
- 4.2 对象
  - 4.2.1 对象的创建和撤销
  - 4.2.2 构造函数
  - 4.2.3 复制构造函数
  - 4.2.4 析构函数
- 4.3 this指针
- 本章小结
- 习题与上机
- 第5章 类的其他特性
  - 5.1 静态成员
    - 5.1.1 静态数据成员
    - 5.1.2 静态成员函数
  - 5.2 友元
    - 5.2.1 友元函数
    - 5.2.2 友元类
  - 5.3 常类型
  - 5.4 类成员指针
  - 5.5 类的组合
- 第6章 类的继承和派生
- 第7章 多态性和虚函数
- 第8章 模板
- 第9章 I/O流
- 第10章 异常处理
- 第11章 综合案例

## &lt;&lt;面向对象程序设计&gt;&gt;

## 章节摘录

版权页：插图：2.程序“程序”有两种含义：一是由程序员编写的源代码，其二是可执行的软件。任何语言编制的程序，最后一定都需要语言处理程序转换成机器语言程序才能被计算机执行。语言处理程序按其处理方式不同，可分为解释型程序和编译型程序两大类。

现在人们非常重视的是易维护的代码。

1.1.2 结构化程序设计以前，人们把程序看成是处理数据的一系列过程。

过程或函数定义为一个接一个顺序执行的一组指令。

数据与程序分开存储，编程的主要技巧在于追踪函数如何调用，哪些数据发生了变化。

为了解决其中可能存在的问题，结构化编程应运而生。

当一些任务十分复杂以至无法描述时，可以将它拆分为一系列较小的功能部件，直到这些自完备的子任务小到成为易于理解的程序。

结构化程序设计的主要思想是功能分解并逐步求精。

结构化程序设计成功地处理复杂问题提供了有力的手段。

然而，随着时间的推移，它的一些缺点也越来越突出。

当数据量增大时，数据与处理这些数据的方法之间的分离使程序变得越来越难以理解。

首先，对数据处理能力的需求越强，这种分离所造成的负面影响越显著；其次，结构化程序设计相对于老问题的新方法都要带来额外的开销，即重用性较差。

对于软件工程师们来说，在面向对象程序设计出现之前，一直缺乏具备这种能力的工具。

1.1.3 面向对象程序设计面向对象程序设计（Object-Oriented Programming，OOP）的本质是把数据和处理数据的过程当成一个整体——对象。

所有的OOP语言（包括C++）都具有三个共同的特性，即封装性、多态性和继承性。

1.封装性（Encapsulation）封装是一种机制，它将代码和数据连接起来，并使之安全，以防外界的干扰和误用。

代码和数据用这种方式连接时一个对象（object）就产生了。

## <<面向对象程序设计>>

### 编辑推荐

《面向对象程序设计(C++)实践教程》：教学目标明确，注重理论与实践的结合、教学方法灵活，培养学生自主学习的能力、教学内容先进，反映了计算机学科的最新发展、教学模式完善，提供配套的教学资源解决方案。

<<面向对象程序设计>>

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>