

## <<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：9787505819979

10位ISBN编号：7505819976

出版时间：2000-3

出版时间：经济科学

作者：刘振安

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

《面向对象程序设计》内容简介：在程序设计方法方面，C++既支持传统的面向过程的程序设计方法，也支持新的面向对象的程序设计方法。

因为C++是一种混合语言，所以就使得它保持与C语言兼容，C程序员仅需学习C++语言的特征，就可很快地用C++编写程序。

全书共分9章。

第1章是面向对象及C++基础知识；第2章是类和对象；第3章是构造函数与析构函数；第4章介绍继承和派生类；第5章介绍多态性和虚函数；第6章是进一步使用成员函数；第7章介绍运算符重载及流类库；第8章介绍模板；第9章是进一步掌握面向对象程序设计。

各章均附有与试题题型相对应的习题，典型习题的答案见与《面向对象程序设计》配套的辅导书。

《面向对象程序设计》的对象是自学考试的读者，既应易于自学，也应注重培养独立解决问题的能力，所以除概念清楚，重点突出，容易理解之外，还专门开辟一章“进一步掌握面向对象程序设计”，系统训练思考问题和解决实际问题的能力，使学生能对使用C++进行面向对象编程有一个完整的整体认识，并初步掌握实用程序的编制方法及大程序的设计方法，为课程设计打下基础。

《面向对象程序设计》虽然作为自学考试的教材，但也适合作为大专院校及社会上各种培训班的教材，并可供广大计算机工作者自学之用。

## &lt;&lt;面向对象程序设计&gt;&gt;

## 书籍目录

第1章 面向对象及C++基础知识1.1 面向对象程序设计基础知识1.1.1 面向过程与面向对象1.1.2 面向对象程序设计方法1.1.3 C++语言的来龙去脉1.2 C++基本程序结构1.2.1 新的风格1.2.2 内联函数1.2.3 引用1.2.4 编译指令1.3 改变习惯重新思考习题1第2章 类和对象2.1 定义类2.2 使用类和对象2.3 内联成员函数2.4 成员函数的重载及其缺省参数2.5 this指针2.6 结构和联合2.7 有关类的其它知识2.7.1 类作用域..2.7.2 空类2.7.3 类对象的性质及存取2.7.4 嵌套类2.7.5 类的实例化习题2第3章 构造函数与析构函数3.1 构造函数3.1.1 定义构造函数3.1.2 构造函数和运算符new3.1.3 缺省构造函数3.1.4 复制初始化构造函数3.2 析构函数3.2.1 定义析构函数3.2.2 析构函数和对象数组3.2.3 析构函数和运算符delete3.2.4 缺省析构函数3.3 构造函数类型转换3.4 对象的初始化3.5 对象赋值3.6 对象成员习题3第4章 继承和派生类4.1 继承和派生的基本概念4.2 单一继承4.3 类的保护成员4.4 访问权限和赋值兼容规则4.5 多重继承4.6 构造函数与析构函数调用顺序4.7 两义性及其支配规则4.7.1 两义性和作用域分辨操作符4.7.2 两义性及其支配规则4.8 虚基类习题4第5章 多态性和虚函数5.1 多态性5.1.1 编译时的多态性5.1.2 运行时的多态性5.2 虚函数5.2.1 虚函数的访问权限5.2.2 在成员函数中调用虚函数5.2.3 构造函数和析构函数调用虚函数5.2.4 空的虚函数5.2.5 纯虚函数与抽象类5.2.6 多重继承与虚函数5.3 虚函数的多态性5.4 虚析构函数习题5第6章 进一步使用成员函数6.1 静态成员6.2 友元函数6.3 const对象和volatile对象.....第7章 运算符重载及流类库第8章 模板第9章 进一步掌握面向对象程序设计面向对象程序设计自学考试大纲

## &lt;&lt;面向对象程序设计&gt;&gt;

## 章节摘录

版权页：插图：由此可见，封装要求一个对象应具备明确的功能，并具有接口以便和其它对象相互作用。

同时，对象的内部实现（代码和数据）是受保护的，外界不能访问它们，只有局部于对象的代码才可以访问该对象的内部数据。

对象内部数据结构的不可访问性称为数据隐藏。

封装使得一个对象可以像一个部件一样用在各种程序中，而不用担心对象的功能受到影响。

数据封装是软件工程发展的必然产物。

早期的软件是把数据与程序混在一起，结构化很差，被戏称为“一碗面条”的编程方法。

在这个阶段，程序的可读性和可维护性都很差，于是产生了“软件危机”，为了解除这种危机便提出了结构化程序设计。

在结构化程序设计里，虽然程序被分成不同的模块，以便大大减少不同模块之间的相互调用，但数据仍然是属于整个程序的。

这就又存在着这样一些问题：一方面，程序员在设计每一个模块的时候，都要或多或少地作全局考虑，增加了工作量；另一方面，在某地方对数据的改动，有可能又对整个程序产生难以预料的影响。

随着软件工程的进一步发展，软件越来越大，数据越来越多，这个问题也越来越突出。

数据封装的提出，就是为了解决这一问题。

它一方面使得程序员在设计程序时可以专注于自己的对象，“各人自扫门前雪，莫管他人瓦上霜”，同时也切断了不同模块之间数据的非法使用，减少了出错的可能性。

不同的对象可以调用相同名称的函数，并可导致完全不同的行为的现象称为多态性。

利用多态性，程序中只需进行一般形式的函数调用，函数的实现细节留给接受函数调用的对象。

这大大提高了我们解决复杂问题的能力。

例如，虽然发送一图表对象时所调用的打印函数与发送一正文对象所调用的打印函数其效果肯定是不同的，但我们可以设计一个公共的sendit（）函数名代表发送，而不同对象发送的具体细节则分别由具体对象负责实现。

在计算机科学中，对象是包含现实世界物体特征的抽象实体。

建立和操作对象并不一定非要面向对象的程序设计语言。

基于对象的编程方法早就存在了，不同的语言如FORTRAN、LISP和汇编语言中也使用了这种方法。

所以说对象是程序设计方法的产物，而不是某种语言的产物。

但在C++中，对象具有更大的重要性。

在这种新方法下，程序中操作的任何成分都可以看做对象。

对象是在程序执行时生成和删除的，对象之间互相作用，可以放在一起形成组、集合、数组及表等结构。

有些语言（例如Smalltalk）是完全基于对象的，Smalltalk程序中操作的任何成分都是某种对象。

这种统一的方法赋予语言更多的优势，但同时也带来一些缺点。

主要是因为使用对象有时会增加运行时刻的开销，从而大大降低程序的性能。

C++程序设计语言利用了对象最受欢迎的特征，并尽可能避免了运行时刻的缺点和开销。

在C++语言中，程序操作的大部分（但不是所有的）成分是对象，因此，C++被称为混合式面向对象的程序设计语言。

C++语言是C语言的扩充。

在1980年，贝尔实验室的Bjarne Stroustrup博士及其同事开始对C语言进行改进和扩充，最初被称为“带类的C”，1983年才取名为C++。

以后又经过不断完善和发展，成为目前的C++语言。

## <<面向对象程序设计>>

### 编辑推荐

《面向对象程序设计》：全国高等教育自学考试指定教材·计算机及应用专业(独立本科段)。

<<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>