

<<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：9787302227816

10位ISBN编号：7302227810

出版时间：2010-8

出版时间：清华大学出版社

作者：董正言，张聪 主编

页数：287

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>>

前言

面向对象程序设计（OOP）技术是当前主流的程序设计技术。和传统的面向过程的程序设计技术相比，面向对象程序设计技术具有明显的优势。这种优势主要体现在以下几个方面：（1）传统的面向过程的程序设计方法忽略了数据和操作之间的内在联系，程序中的数据和操作它们的方法分离，而面向对象程序设计技术将程序要处理的数据和处理它们的方法封装在一起，构成一个统一体——对象。

程序中用对象模型来建模现实世界中的事物。这样就使得空间模型的结构和问题空间模型的结构相一致。使用面向对象的方法解决问题的思路更加符合人类一贯的思维方法。

（2）面向对象的封装技术为程序提供了更好的数据安全性。

（3）面向对象的继承技术为程序提供了更好的可重用性。

（4）面向对象的多态调用技术使程序具有更好的可扩展性。

（5）和传统的面向过程的程序设计方法相比，面向对象的程序设计技术更适合开发大型的图形界面应用程序。

目前，常用的面向对象的编程语言有C++、Java、C#等。

C++语言是由C语言发展演变成的一种面向对象的程序设计语言。

它既具备C语言面向过程设计方法的特点，同时又支持面向对象的程序设计方法。

它是目前程序员使用最多的编程语言之一。

本书以C++语言为基础，阐述面向对象的编程原理。

本书的特色是内容全面、语言简洁易懂、重点突出，是一部面向大学本科计算机科学相关专业的入门教材。

为了使读者能够透彻理解面向对象编程的原理和方法，本书使用了大量自编的例程。

全部例程的源代码均使用visual C++6.0编写，并编译通过。

本书由董正言主编，张聪、刘文涛、阮灵等老师也参加了部分章节的编写工作。

由于学识水平和时间的限制，疏漏和不妥之处在所难免，敬请批评指正。

<<面向对象程序设计>>

内容概要

本书以面向对象程序设计技术的本质属性为主线，以C++语言为基础，全面地阐述了面向对象程序设计的基本原理。

全书共分13章，前5章介绍C++语言的基本编程技术，包括C++的数据类型、常用的运算符、结构化控制语句和函数；第6章~第9章介绍面向对象程序设计方法的核心内容，包括类和对象、类的继承和多态；第10章介绍C++的程序结构、编译预处理指令和命名空间等内容；第11章介绍C++语言标准库中的输入/输出流类；第12章以C++语言为基础，介绍面向对象的异常处理机制；第13章简要介绍使用C++语言和面向对象的程序设计方法设计Windows应用程序的方法。

本书语言简洁流畅，通俗易懂，内容全面，重点突出，对核心和重点内容都佐以大量的例证。

本书既可以作为高等院校计算机科学与技术及相关专业“面向对象程序设计”课程的授课教材，也可以作为C++程序开发人员的参考书。

<<面向对象程序设计>>

书籍目录

第1章 绪论 1.1 编程语言的发展 1.2 C++语言简介 1.3 面向对象的程序设计方法 1.4 第一个C++程序 1.5 小结 习题第2章 基本数据类型和运算符 2.1 基本概念 2.1.1 程序实例 2.1.2 C++字符集 2.1.3 C++关键字 2.1.4 标识符 2.1.5 程序注释 2.2 基本数据类型 2.2.1 整数类型 2.2.2 浮点类型 2.2.3 字符类型 2.2.4 布尔类型 2.3 变量和常量 2.3.1 变量 2.3.2 常量 2.4 简单的输入和输出 2.5 C++运算符和表达式 2.5.1 赋值运算符和赋值表达式 2.5.2 算术运算符和算术表达式 2.5.3 关系运算符和关系表达式 2.5.4 逻辑运算符和逻辑表达式 2.5.5 位运算符 2.5.6 逗号运算符和逗号表达式 2.5.7 条件运算符和条件表达式 2.5.8 sizeof运算符 2.5.9 其他运算符 2.6 数据类型转换 2.7 小结 习题第3章 控制语句 3.1 选择结构 3.1.1 if/else选择语句 3.1.2 嵌套的if/else语句 3.1.3 if/else if语句 3.1.4 switch语句 3.2 循环结构 3.2.1 while循环语句 3.2.2 do?while循环语句 3.2.3 for循环语句 3.2.4 嵌套的循环语句 3.3 其他流控制语句 3.3.1 break语句和continue语句 3.3.2 goto语句 3.4 小结 习题第4章 复合数据类型 4.1 数组 4.1.1 数组的定义和初始化 4.1.2 使用数组 4.1.3 字符数组和字符串 4.2 指针和引用 4.2.1 定义和使用指针 4.2.2 指针和数组 4.2.3 数组指针和指针数组 4.2.4 使用操作符new和delete进行动态存储分配 4.2.5 引用的定义和使用 4.3 枚举和结构 4.3.1 枚举 4.3.2 结构 4.4 小结 习题第5章 函数 5.1 定义和调用函数 5.1.1 函数的定义 5.1.2 函数的调用 5.1.3 函数原型 5.2 传递参数 5.2.1 传值传递 5.2.2 引用传递 5.3 局部变量和全局变量 5.4 函数调用的实现 5.5 内联函数 5.6 递归函数 5.7 参数的默认值 5.8 指针函数和函数指针 5.8.1 指针函数 5.8.2 函数指针 5.9 函数重载 5.10 函数模板 5.11 小结 习题第6章 类和对象(上) 6.1 面向对象程序设计概述 6.2 创建类 6.2.1 定义类 6.2.2 类的实现 6.3 创建和使用对象 6.4 类成员的访问控制 6.4.1 类的公有成员 6.4.2 类的私有成员 6.4.3 类的保护成员 6.5 内联的成员函数 6.6 构造函数 6.6.1 定义类的构造函数 6.6.2 默认的构造函数 6.6.3 带默认参数值的构造函数 6.7 拷贝构造函数 6.7.1 定义类的拷贝构造函数 6.7.2 默认的拷贝构造函数与对象的浅拷贝问题 6.8 析构函数 6.9 小结 习题第7章 类和对象(下) 7.1 类的静态成员 7.1.1 静态数据成员 7.1.2 静态函数成员 7.2 对象指针 7.3 动态创建 7.3.1 动态创建对象 7.3.2 动态创建对象数组 7.4 类作用域 7.4.1 类成员具有类作用域 7.4.2 具有类作用域的数据成员被局部变量屏蔽 7.5 this指针 7.6 类的组合 7.7 常对象和类的常成员 7.7.1 常对象 7.7.2 常数据成员 7.7.3 const成员函数 7.8 类模板 7.9 友元 7.9.1 友元函数 7.9.2 友元类 7.10 string类 7.10.1 构造字符串 7.10.2 常用的字符串操作 7.11 小结 习题第8章 类的继承 8.1 基类和派生类 8.2 定义派生类 8.3 继承方式与访问权限 8.4 构造派生类对象 8.4.1 派生类对象的结构 8.4.2 派生类的构造函数 8.5 成员覆盖 8.6 实例学习——图形类家族 8.7 多重继承 8.7.1 多重继承简介 8.7.2 多重继承引发的二义性问题 8.8 虚基类 8.9 对象类型转换 8.10 小结 习题第9章 多态 9.1 什么是多态 9.2 虚函数和动态绑定 9.3 纯虚函数和抽象类 9.4 编译期多态——运算符重载 9.4.1 什么是运算符重载 9.4.2 用类的成员函数实现运算符重载 9.4.3 用类的友元函数实现运算符重载 9.4.4 重载赋值运算符= 9.5 运行时类型识别(RTTI) 9.5.1 dynamic_cast操作符 9.5.2 typeid操作符和type_info类 9.6 小结 习题第10章 程序结构、预处理和命名空间 10.1 多文件结构的源程序 10.2 文件间的信息共享 10.2.1 头文件 10.2.2 关键字extern 10.2.3 使用关键字static避免同名冲突 10.2.4 函数的声明 10.2.5 类的声明 10.3 预处理 10.3.1 #include指令 10.3.2 #define指令 10.3.3 条件预处理指令 10.3.4 使用条件预处理指令避免重复包含 10.4 命名空间 10.5 小结 习题第11章 输入和输出 11.1 什么是输入/输出流 11.2 输入/输出流类 11.2.1 预定义的流对象 11.2.2 插入运算符和提取运算符 11.2.3 格式化标志和设置格式化标志的函数 11.2.4 I/O格式操作符 11.2.5 控制输入/输出格式的函数 11.2.6 常用的I/O函数 11.3 磁盘文件的输入/输出 11.3.1 打开文件 11.3.2 数据的存储格式和文件的打开模式 11.3.3 文件输入/输出 11.3.4 文件指针 11.4 小结 习题第12章 异常处理 12.1 抛出异常 12.2 捕获和处理异常 12.3 异常的传递途径 12.4 异常类 12.5 小结 习题第13章 Windows编程基础

<<面向对象程序设计>>

13.1 事件驱动机制和Windows SDK编程 13.2 MFC 13.2.1 一个简单的MFC应用程序 13.2.2 MFC简介 13.2.3 MFC程序结构分析 13.3 消息映射机制 13.3.1 Windows消息 13.3.2 MFC消息映射机制 13.4 Windows程序实例 13.5 小结 习题附录A参考文献

<<面向对象程序设计>>

章节摘录

在上面的语句中，定义变量b的同时，就给它赋了初值，这叫做变量初始化。

如上所述，每个变量都有数据类型。

例如，例2.1 程序中的变量是整数类型（int）。

变量的值存放在存储器中。

根据存储地点的不同，变量分为不同的存储类型。

C++变量的存储类型分自动型和持续型两大类。

自动型变量的特点是：变量的存储空间在程序运行时被自动分配。

当包含该变量的函数开始运行时，系统自动为该变量分配存储空间；当包含该变量的函数模块运行结束时，其存储空间被系统回收，该变量随之被自动销毁。

自动型变量包括以下两种。

auto型：这种类型的变量被存放在称为“栈”的内存空间中。

定义这种变量时，关键字auto可以省略；程序中在函数或模块内部声明的变量，如果不做其他声明，它们都属于这种存储类型的变量。

例如，例2.1 程序中的变量。

register型：称为寄存器型变量。

这种类型的变量被直接存放在计算机中央处理器（CPU）的寄存器中，特点是变量的访问速度较快。

定义寄存器变量时，要使用关键字register。

例如，registerinti。

持续性变量的特点是：变量从被定义时开始，系统为其分配存储空间；在程序的个运行过程中一直存在，直到程序运行结束才被销毁。

持续型变量包括以下两种。

全局变量：是程序中在任何函数之外声明的变量。

存放在程序的静态存储区中。

程序开始执行时，系统给全局变量分配存储空间，程序执行完毕并释放存储空间。

在程序执行期间，全局变量占据固定的存储单元，而不是由系统动态分配。

例如，例2.1 程序中的变量j就是一个全局变量。

<<面向对象程序设计>>

编辑推荐

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>