

<<C++面向对象程序设计>>

图书基本信息

书名：<<C++面向对象程序设计>>

13位ISBN编号：9787560624440

10位ISBN编号：7560624448

出版时间：2010-9

出版时间：西安电子科技大学出版社

作者：李兰 编

页数：352

字数：535000

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>

<<C++面向对象程序设计>>

前言

C++语言为面向对象技术提供了全面支持，是目前应用较广的一种优秀的高级程序设计语言，也是一个可编写高质量的用户自定义类型库的工具。

C++保留了对传统的结构化程序设计方法的支持，同时又增加了对面向对象程序设计方法的完全支持，但后者是其主要特色和应用。

C++语言是一种具有代表性的面向对象的程序设计语言，它具有丰富的数据类型和各种运算功能，带有庞大的函数库和类库，既支持面向过程的程序设计，又支持面向对象的程序设计。

目前，国内外许多高等院校计算机专业和相关专业都开设了“面向对象程序设计”课程，其目的是让学生掌握面向对象程序设计的概念和方法，深刻理解面向对象程序设计的本质，并用面向对象技术来开发软件。

本书力求在内容、编排顺序和教学方法上有所创新和突破，以帮助学生快速理解与程序设计相关的基本概念，掌握程序设计语言的基本知识，建立程序设计的基本思想，并通过实际动手领会和掌握C++的精髓，最终获得面向对象C++程序设计的真实本领，为今后的进一步学习和开发应用打下坚实的基础。

基于这些理念，本书对C++语言的基本概念、原理和方法的叙述由浅入深，条理分明、循序渐进；以“概念-语法-举例”的形式进行讲解，并指出了学生常犯的错误和容易混淆的概念；各章均配有大量的习题，以帮助学生深入理解面向对象语言和C++的性质。

<<C++面向对象程序设计>>

内容概要

面向对象程序设计是目前流行的软件开发方法。

本书根据“面向对象程序设计”课程的基本教学要求，针对面向对象的本质和特性，系统地讲解了面向对象程序设计的基本理论和基本方法，阐述了用C++语言实现面向对象基本特性的关键技术。

本书的内容主要包括：面向对象程序设计概述、C++语言基础、封装性、继承性、运算符重载、多态性、模板和STL、输入/输出流、异常处理等。

本书可作为高等院校计算机及相关专业“C++面向对象程序设计”课程的教材，也可作为从事计算机开发和应用的工程技术人员的参考书。

同时，也适合初学程序设计或有一定编程实践基础、希望突破编程难点的读者作为自学教材。

<<C++面向对象程序设计>>

书籍目录

第1章 面向对象程序设计概述

1.1 计算机程序设计语言的发

1.1.1 程序设计语言概述

1.1.2 机器语言与汇编语言

1.1.3 高级语言

1.1.4 面向对象语言

1.2 程序设计方法

1.2.1 结构化程序设计方法

1.2.2 面向对象程序设计方法

1.3 面向对象程序设计的基本特点

1.3.1 抽象

1.3.2 封装

1.3.3 消息

1.3.4 继承

1.3.5 多态

1.4 简单的C++程序

本章小结

习题1

第2章 C++语言基础

2.1 C++字符集和关键字

2.1.1 字符集

2.1.2 标识符

2.1.3 关键字

2.1.4 其他标识

2.2 基本数据类型和表达式

2.2.1 C++的基本数据类型

2.2.2 常量

2.2.3 变量

2.2.4 表达式

2.3 运算符与表达式

2.3.1 算术运算符与算术表达式

2.3.2 关系运算与逻辑运算

2.3.3 赋值运算符与赋值表达式

2.3.4 条件运算符与逗号表达式

2.3.5 表达式的副作用和表达式语句

2.4 C++程序的基本控制结构

2.4.1 程序的结构与控制

2.4.2 顺序结构程序设计

2.4.3 选择结构程序设计

2.4.4 循环结构程序设计

2.5 动态内存分配

2.5.1 动态内存

2.5.2 和运算符

2.6 常见编程错误

本章小结

<<C++面向对象程序设计>>

习题2

第3章 函数

3.1 函数的定义和声明

3.1.1 函数的定义

3.1.2 函数的声明

3.1.3 函数值和函数类型

3.2 函数的调用与参数传递

3.2.1 函数的调用

3.2.2 函数调用时的参数传递

3.2.3 函数的嵌套调用和递归调用

3.3 内联函数

3.4 函数重载

3.4.1 函数重载的定义

3.4.2 函数重载的绑定

3.5 带默认形参值的函数

3.6 作用域与生存期

3.6.1 标识符的作用域

3.6.2 局部变量与全局变量

3.6.3 动态变量与静态变量

3.6.4 变量的存储类型

3.6.5 生存期

3.6.6 名字空间

3.7 多文件结构

3.8 常见编程错误

本章小结

习题3

第4章 类与对象

4.1 类和对象

4.1.1 类与抽象数据类型

4.1.2 类的声明和定义

4.1.3 类的函数成员的实现

4.1.4 类和对象

4.1.5 类的访问属性

4.2 构造函数与析构函数

4.2.1 构造函数

4.2.2 缺省构造函数

4.2.3 拷贝构造函数

4.2.4 转换构造函数

4.2.5 析构函数

4.3 常成员

4.3.1 const修饰符

4.3.2 常数据成员

4.3.3 常函数成员

4.4 指向对象的指针

4.4.1 对象指针

4.4.2 this指针

4.5 静态成员与友元

<<C++面向对象程序设计>>

4.5.1 静态数据成员与静态函数成员

4.5.2 友元函数与友元类

4.6 常见编程错误

本章小结

习题

第5章 继承

5.1 继承与派生

5.1.1 继承的概念

5.1.2 派生类的声明

5.2 派生类的访问控制

5.2.1 公有继承

5.2.2 私有继承

5.2.3 保护继承

5.3 派生类的构造函数与析构函数

5.3.1 派生类的构造函数

5.3.2 派生类构造函数调用规

5.3.3 派生类的析构函数

5.4 多继承

5.4.1 多继承概念

5.4.2 多继承中的二义性问题及其解决

5.4.3 多继承中构造函数和析构函数的调用顺序

5.5 虚基类

5.5.1 多继承派生的基类拷贝

5.5.2 虚基类的定义

5.5.3 虚基类的构造与析构

5.6 赋值兼容规

5.7 程序举例

5.8 常见编程错误

本章小结

习题5

第6章 多态与虚函数

6.1 静态联编和动态联编

6.1.1 静态联编

6.1.2 动态联编

6.2 虚函数

6.2.1 虚函数的定义和使用

6.2.2 虚函数的特性

6.3 纯虚函数和抽象类

6.3.1 纯虚函数

6.3.2 抽象类

6.3.3 抽象类的应用

6.4 运算符重载

6.4.1 运算符重载的规则

6.4.2 运算符重载为成员函数

6.4.3 运算符重载为友元函数

6.5 实例分析

6.5.1 问题提出

<<C++面向对象程序设计>>

6.5.2 类设计

6.5.3 程序代码设计

6.6 常见编程错误

本章小结

习题6

第7章 模板

7.1 模板的概念

7.2 函数模板与模板函数

7.2.1 函数模板的声明

7.2.2 函数模板

7.2.3 模板函数

7.2.4 重载函数模板

7.3 类模板与模板类

7.3.1 类模板的定义

7.3.2 类模板的使用

7.3.3 类模板的友元

7.3.4 类模板与静态成员

7.4 标准模板库

7.4.1 容器

7.4.2 迭代器

7.4.3 算法

7.4.4 适配器

7.5 常见编程错误

本章小结

习题7

第8章 输入/输出流

8.1 流以及流类库结构

8.1.1 流的概念

8.1.2 流类库

8.2 非格式化的输入和输出

8.3 格式化的输入和输出

8.3.1 ios类中定义的格式控制标志开

8.3.2 操作符

8.3.3 格式化输入和输出的简单应用

8.4 文件的输入和输出

8.4.1 文件与流

8.4.2 文件的打开和关闭

8.4.3 读/写文本文件

8.5 常见编程错误

本章小结

习题8

第9章 异常处理

9.1 异常的概念

9.2 异常处理机制及意义

9.3 标准异常

9.4 异常的捕获

9.4.1 try块

<<C++面向对象程序设计>>

9.4.2 throw表达式

9.4.3 异常处理器

9.4.4 异常规格说

9.4.5 捕获所有类型的异常

9.4.6 未捕获的异常

9.5 异常处理中的构造与析构

9.5.1 在构造函数中抛出异常

9.5.2 不要在析构函数中抛出异常

9.6 开销

9.7 常见编程错误

本章小结

习题9

附录 标准字符码表

附录 C++程序错误提示中英文对照表

参考文献

<<C++面向对象程序设计>>

章节摘录

1. 机器语言 最基本的计算机语言是机器语言。

机器语言（也称为第一代语言）使用二进制位来表示程序指令。

计算机的硬件系统可以直接识别和执行的二进制指令（机器指令）的集合称为这种计算机的机器语言。

每种计算机都有自己的机器语言，并能直接执行用机器语言所编写的程序。

虽然绝大多数计算机完成的功能基本相近，但不同计算机的设计者都可能会采用不同的二进制代码集来表示程序指令。

所以，不同种类的计算机使用的计算机语言并不一定相同，但现代计算机都是以二进制代码的形式来存储和处理数据的。

在早期的计算机应用中，程序都是用机器语言编写的，程序员需要记住各种操作的机器语言指令。

同时，为了存取数据，程序员还必须记住所有数据在内存中的存储地址。

例如，字母A表示为1010，数字9表示为1001。

由于机器语言的指令码可能有多种形式，还要考虑运算中的进位、借位和符号溢出等各种情况，因此更增加了程序员的记忆负担。

这种需要记住大量的编码指令来编写程序的方法不仅难以实现，而且非常容易出错。

直接使用机器语言来编写程序是一种相当复杂的手工劳动，它要求使用者熟悉计算机的有关细节，一般的工程技术人员是难以掌握的。

2. 汇编语言 汇编语言（也称为第二代语言）的出现简化了编程人员的工作。

汇编语言将机器指令映射为一些可以被人们读懂的助记符，如ADD、MOV等，它实际上是与机器语言相对应的语言。

汇编语言的主要特征是可以利用助记符来表示每一条机器指令。

汇编语言同机器语言相比，并没有本质的区别，只不过是把机器指令用助记符号代替。

由于汇编语言比机器语言容易记忆，因此其编程效率比机器语言前进了一大步，而且改进了程序的可读性和可维护性。

直到今天，仍然有人用汇编语言进行编程。

但汇编语言程序的大部分语句还是和机器指令一一对应的，与机器的相关性仍然很强。

.....

<<C++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>