

## <<C++高级语言程序设计>>

### 图书基本信息

书名：<<C++高级语言程序设计>>

13位ISBN编号：9787115246325

10位ISBN编号：7115246327

出版时间：2011-2

出版时间：人民邮电出版社

作者：徐惠民

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++高级语言程序设计>>

### 内容概要

本书将C++语言作为大学生学习程序设计的入门语言。

全书共9章，第1章、第3章、第5章讲解程序的结构和组织，第2章、第4章、第6章讨论数据类型和数据的组织，第7章、第8章、第9章讲述面向对象的程序设计。

本书的编写目的是为学生打好程序设计的基础，因此，特别注意在介绍基本概念和基本方法的同时，重视良好编程习惯的培养。

另外，专业术语配有英文。

本书内容全面，重点突出，深入浅出，同时配有大量习题，适合作为大学各专业的程序设计课程入门教材，也可以作为程序设计基础的培训教材和自?参考书。

## &lt;&lt;C++高级语言程序设计&gt;&gt;

## 书籍目录

第1章 C++初步 11.1 程序设计语言的发展 11.2 面向过程的程序设计 21.3 面向对象程序设计  
 31.4 C++的诞生 41.5 程序开发过程 51.6 最简单的C++程序 61.7 内存的抽象表示和使用  
 8本章小结 9习题和思考题 10第2章 基本数据类型与表达式 112.1 C++的词法记号和标识符  
 112.1.1 字符集 112.1.2 关键字 112.1.3 标识符 122.1.4 分隔符 122.1.5 空白 122.2 基本  
 数据类型 122.3 变量和常量 152.3.1 ?量 152.3.2 常量 172.4 运算符和表达式 202.4.1 表达  
 式 202.4.2 语句和块 202.4.3 运算符 202.5 基本输入/输出 292.5.1 标准输入流 292.5.2 标  
 准输出流 302.5.3 I/O流的格式控制 312.5.4 文件的输入和输出 342.6 使用string类型处理字符串  
 362.6.1 string对象的定义和初始化 362.6.2 string对象的操作 36本章小结 38习题和思考题 38  
 第3章 C++控制语句 423.1 算法的基本控制结构 423.1.1 程序的3种控制结构 423.1.2 算法及  
 其表示 433.2 if选择语句 453.2.1 没有else分支的形式 453.2.2 双分支形式 463.2.3 多分支形式  
 473.2.4 if语句的嵌套 483.3 switch选择语句 503.4 循环语句 543.4.1 while循环语句 553.4.2  
 do-while循环语句 573.4.3 for循环语句 583.4.4 break语句和continue语句 613.5 循环嵌套  
 623.5.1 循环结构嵌套 623.5.2 循环结构和选择结构的相互嵌套 633.6 应用举例 65本章小结  
 69习题和思考题 69第4章 数组及其他自定义类型 724.1 数组 724.1.1 数组定义及初始化  
 724.1.2 访问数组元素 754.1.3 字符数组 774.1.4 多维数组 794.1.5 数组应用举例 844.1.6  
 数组小结 874.2 枚举类型 874.2.1 枚举类型定义 884.2.2 枚举变量定义及使用 884.2.3 应用  
 举例 894.3 结构类型 894.3.1 结构类型的定义和初始化 904.3.2 结构数组 924.4 联合类型  
 94本章小结 96习题和思考题 97第5章 函数 1005.1 函数概述 1005.1.1 自定义函数概述  
 1005.1.2 库?数概述 1015.2 函数定义及使用 1025.2.1 函数的定义 1025.2.2 函数原型  
 1045.2.3 return语句 1065.2.4 函数调用方式 1085.3 函数调用的执行机制和参数传递方式  
 1095.3.1 函数调用的执行机制 1105.3.2 函数的参数传递方式 1115.4 递归函数和递归调用  
 1175.4.1 嵌套调用 1175.4.2 递归函数和递归调用 1185.5 内联函数 1215.6 重载函数 1215.7  
 默认参数值的函数 1225.8 全局变量与局部变量 1245.8.1 局部变量 1245.8.2 全局变量  
 1245.8.3 作用域 1255.9 变量的存储类型和生存期 1265.9.1 变量的存储类型 1275.9.2 生存期  
 1295.9.3 多文件结构 1295.10 编译预处理 1305.11 结构化程序设计举例 133本章小结 136习  
 题和思考题 136第6章 指针和引用 1406.1 指针的概念 1406.1.1 指针和指针变量 1406.1.2 指  
 针变量的声明和初始化 1416.2 指针的运算 1426.2.1 指针的赋值运算 1426.2.2 指针的间接引用  
 运算 1436.2.3 指针的算术运算 1436.2.4 指针的关系运算和逻辑运? 1446.2.5 void类型指针  
 1446.2.6 指针类型转换 1466.3 指针访问动态内存 1466.3.1 动态内存的申请和释放 1476.3.2  
 动态数组空间的申请和释放 1476.3.3 内存泄漏和指针悬挂 1476.4 指向结构体的指针 1486.5 引  
 用概念 1496.5.1 引用的声明 1496.5.2 引用的使用 1506.6 指针和引用作为函数的参数 1506.6.1  
 指针作为函数参数 1506.6.2 引用作为函数参数 1526.6.3 常指针和常引用 1536.6.4 指针的指  
 针作为参数 1556.7 指针和引用作为函数的返?值 1576.7.1 指针函数 1576.7.2 引用作为函数的返  
 回值 1586.8 指针和字符串 1596.8.1 字符串处理的两种方式 1596.8.2 字符串操作函数 1606.9  
 通过指针访问数组 1616.9.1 通过指针访问一维数组 1616.9.2 指针数组 163本章小结 164习题  
 和思考题 165第7章 类与对象 1677.1 类和对象的定义 1677.1.1 基本概念 1677.1.2 类的声明  
 1687.1.3 类的实现 1707.1.4 对象的定义和使用 1717.1.5 类的作用域与可见性 1727.2 对象的  
 使用 1747.2.1 对象指针 1747.2.2 this指针 1757.2.3 对象数组 1767.2.4 对象作为普通函数的参  
 数与返回值 1777.3 构造函数 1807.4 析构函数 1847.5 类的静态成员 1867.5.1 静态数据成员  
 1867.5.2 静态成员函数 1887.6 类成员的保护和使用 1887.6.1 类的封装性 1887.6.2 友元  
 1897.6.3 常对象和常成员 1927.7 运算符重载 1947.7.1 运算符重载的使用及其限制 1947.7.2  
 运算符重载的定义 1957.8 类的组合 1997.9 应用举例 2017.10 面向对象分析和设计 2067.10.1  
 软件工程 2067.10.2 面向对象分析 2067.10.3 面向对象设计 2077.10.4 面向对象的意义 207  
 本章小结 208习题和思考题 209第8章 继承与派生 2138.1 继承的概念 2138.2 定义基类和派生  
 类 2148.2.1 简单的继承和派生 2148.2.2 定义派生类 2168.2.3 访问控制和继承的关系 2168.2.4

## &lt;&lt;C++高级语言程序设计&gt;&gt;

同名覆盖 2188.3 构造函数和析构函数 2198.3.1 基类只有无参构造函数 2198.3.2 派生类构造函数 2198.3.3 包含内嵌对象的派生类构造函数 2228.3.4 析构函数 2308.4 转换与继承 2328.4.1 派生类到基类的转换 2328.4.2 基类到派生类不存在转换 235本章小结 235习题和思考题 235

第9章 多态 2399.1 多态的概念 2399.1.1 面向对象程序设计中的多态 2399.1.2 多态的实现——联编 2409.2 重载、覆盖与静态联编 2409.2.1 重载与静态联编 2409.2.2 覆盖与静态联编 2419.3 虚函数与运行时多态 2479.3.1 虚函数 2489.3.2 虚析构函数 2519.4 纯虚函数与抽象类 2549.5 模板 2559.5.1 函数模板 2559.5.2 函数模板使用中的问题 2589.5.3 重载函数模板 2609.5.4 类模板 261本章小结 264习题和思考题 265附录 常用C++标准类库 268附1 I/O流类 268附1.1 标准输出流对象 269附1.2 标准输入流对象 270附1.3 文件输出流 270附1.4 文件输入流 271附2 string类 272附2.1 string对象的初始化 272附2.2 string对象的基本操作 272附3 vector类 274参考文献 276

<<C++高级语言程序设计>>

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>