

## <<C++语言从入门到精通>>

### 图书基本信息

书名：<<C++语言从入门到精通>>

13位ISBN编号：9787111325468

10位ISBN编号：711132546X

出版时间：1970-1

出版时间：机械工业出版社

作者：陈国建，等 编

页数：417

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>

## &lt;&lt;C++语言从入门到精通&gt;&gt;

## 前言

C++是一种广泛用于大型软件开发的高级语言，它具有很高的复杂性和解决问题的能力，读者能从中学到很多优秀的软件设计思想。

C++在几乎所有的计算环境中都非常普及，而且几乎可以用于所有的应用程序。

C++从C中继承了过程化编程的高效性，并集成了面向对象编程的功能。

C++在其标准库中提供了大量的功能，有许多商业C++库支持数量众多的操作系统环境 and 专业应用程序。

但是因为C++的内容太多，所以并不十分容易掌握。

本书详述了C++语言的各个方面。

为了方便广大读者学习，笔者结合自己多年的C++编程经验，由浅入深地介绍了C++的开发基础、C++的开发过程、面向对象程序设计的思想以及泛型编程技术等。

本书的各章节都附有实例代码，以使读者能够采用理论结合实践的方式巩固知识点。

学完本书之后，读者将对C++编程有一个全面且深入的了解。

本书的特点 1. 内容丰富，知识全面 全书共分五篇19章，采用从易到难、循序渐进的方式进行讲解。

内容几乎涉及了C++程序开发的各个方面。

2. 循序渐进，由浅入深 为了方便读者学习，本书首先让读者了解C++的发展史，以及C++的开发环境和工具。

读者可以在掌握开发环境和工具的基础上，逐渐学习C++的基础知识、过程开发、面向对象程序开发以及C++的其他重要知识。

读者可以边学习，边动手，以更快地掌握C++编程的各种知识。

3. 格式统一，讲解规范 书中每个例程都采用了分步骤实现的方法，这使得读者可以很清晰地知道每个技术的具体实现步骤，从而提高学习效率。

4. 对比讲解，利于理解 由于C++编程的相关技术较多，很多读者无所适从，无从下手。

本书通过细致的代码分析，帮助读者层层理解，逐渐掌握C++程序设计的精髓。

5. 案例精讲，深入剖析 根据笔者多年的项目开发经验，所有C++编程开发的例子都对应于所讲的每一个知识点，每一个代码都有代码剖析，有助于读者对C++编程进行深入学习。

## <<C++语言从入门到精通>>

### 内容概要

《C++语言从入门到精通（视频实战版）》是一本全面介绍C++的书籍，由浅入深、循序渐进，可使读者熟练掌握C++，并能开发C++应用程序。

《C++语言从入门到精通（视频实战版）》分为五篇，第一篇是C++基础篇，介绍了C++程序设计入门、C++变量与运算符以及C++选择和循环语句等基础知识。

第二篇讨论了C++面向过程编程的一些知识点，讨论了数组与字符串、指针与内存、如何自定义数据类型（结构体）和函数的基本知识。

第三篇是C++面向对象编程，重点在于类和对象、运算符重载、类的继承和多态与虚函数等。

第四篇讨论了C++泛型编程，包括C++模板和标准模板库（STL）。

第五篇介绍了C++输入/输出处理和编码规范，讨论了输入/输出、文件操作、命名空间、异常和错误处理的基本知识。

《C++语言从入门到精通（视频实战版）》语言简洁，内容精练，重点突出，实例丰富，适合于C++初学者、具有一定C语言或者C++语言基础的中级学习者、学习C++的大中专院校学生阅读，也可作为高等院校C++教材参考书。

## 书籍目录

丛书序前言第一篇 c++基础第1章 c++程序设计入门 1.1 面向对象的程序概述 1.1.1 面向过程的程序设计 1.1.2 面向对象的程序设计 1.2 c++概述 1.2.1 c++的发展历程 1.2.2 c++的特点 1.3 visual c++开发环境 1.3.1 visual c++ 6.0简介 1.3.2 visual c++开发环境概述 1.4 c、c++与visual c++ 1.4.1 c与c++ 1.4.2 c++与visual c++ 1.5 第一个c++程序 1.5.1 c++程序的结构 1.5.2 创建源程序 1.5.3 编译、链接和运行程序 1.6 小结 1.7 习题 第2章 变量与运算符 2.1 变量与基本数据类型 2.1.1 什么是变量 2.1.2 如何定义变量 2.1.3 整型 2.1.4 浮点型 2.1.5 字符类型 2.1.6 布尔类型 2.2 常量 2.2.1 什么是常量 2.2.2 数值常量-整型常量和实型常量 2.2.3 字符型常量 2.2.4 字符串常量 2.3 算术运算符与算术表达式 2.3.1 基本算术运算符 2.3.2 关系运算符 2.3.3 逻辑运算符 2.3.4 位运算符 2.3.5 sizeof运算符 2.3.6 强制类型转换 2.3.7 运算符的优先级与结合性 2.3.8 算术表达式 2.3.9 逻辑表达式 2.3.10 赋值表达式 2.3.11 逗号表达式 2.4 小结 2.5 习题 第3章 选择与循环 3.1 选择结构和if语句 3.1.1 if语句的3种形式 3.1.2 if语句的嵌套 3.1.3 switch选择语句 3.2 循环结构和循环语句 3.2.1 while循环结构 3.2.2 for循环结构 3.2.3 do-while循环结构 3.2.4 循环的嵌套 3.2.5 几种循环的比较 3.2.6 循环语句的效率 3.2.7 break、continue和goto语句 3.3 小结 3.4 习题 第二篇 c++过程开发第4章 数组与字符串 4.1 数组的概念 4.2 一维数组的定义和引用 4.2.1 一维数组的定义 4.2.2 一维数组元素的引用 4.2.3 一维数组的初始化 4.2.4 一维数组与指针 4.3 二维数组的定义和引用 4.3.1 二维数组的定义 4.3.2 二维数组元素的引用 4.3.3 二维数组的初始化 4.3.4 多维数组的使用 4.4 指针与数组 4.4.1 指针与数组的关系 4.4.2 指向数组元素的指针 4.4.3 指针与一维数组 4.4.4 指针与多维数组 4.4.5 指针与字符数组 4.5 指针数组与数组指针 4.5.1 指针数组 4.5.2 数组指针 4.6 动态分配二维数组 4.6.1 已知二维数组的第二维进行动态分配 4.6.2 已知二维数组的第一维进行动态分配 4.6.3 已知二维数组的第一维并一次分配内存 4.6.4 二维数组的两维都未知的情况下进行动态分配 4.6.5 二维数组的两维都未知的情况下一次分配内存 4.7 字符数组与字符串 4.7.1 字符数组 4.7.2 字符串及其使用 4.7.3 字符串处理函数 4.8 小结 4.9 习题 第5章 指针与内存 5.1 指针的概念 5.2 变量与指针 5.2.1 定义指针变量 5.2.2 引用指针变量 5.2.3 引用指针变量与指针的指针 5.2.4 指针作函数参数 5.3 数组与指针 5.3.1 指向数组元素的指针 5.3.2 用指针变量作函数参数接收数组地址 5.3.3 多维数组与指针 5.4 函数与指针 5.4.1 函数指针与指针函数 5.4.2 用指向函数的指针作函数参数 5.5 动态内存分配 5.5.1 使用new动态分配内存 5.5.2 使用new申请动态数组 5.6 小结 5.7 习题 第6章 自定义数据类型 6.1 结构体类型 6.1.1 结构体概述 6.1.2 结构体类型变量的定义方法及其初始化 6.1.3 结构体变量的引用 6.1.4 结构体数组 6.1.5 指向结构体变量的指针 6.1.6 结构体类型数据作为函数参数 6.2 共用体 6.2.1 共用体的语法形式 6.2.2 定义共用体变量 6.2.3 共用体变量的引用 6.3 枚举类型 6.4 链表 6.4.1 链表的结构 6.4.2 创建链表 6.4.3 链表与数组的区别 6.4.4 删除整个链表 6.4.5 在链表中插入结点 6.5 小结 6.6 习题 第7章 函数 7.1 函数概述 7.2 函数定义 7.2.1 无参函数 7.2.2 有参函数 7.2.3 函数头与函数体 7.2.4 函数返回值 7.2.5 函数声明 7.3 函数参数 7.3.1 形参和实参 7.3.2 值传递 7.3.3 引用传递 7.3.4 指针传递 7.3.5 有缺省参数的函数 7.3.6 内联函数 7.4 函数的嵌套调用 7.4.1 函数的相互嵌套调用 7.4.2 函数的递归调用 7.5 函数的重载 7.5.1 什么是函数重载 7.5.2 如何实现函数重载 7.6 函数变量 7.6.1 局部变量 7.6.2 全局变量 7.6.3 静态局部变量 7.6.4 静态全局变量 7.7 小结 7.8 习题 第8章 c++过程开发的进一步讨论 8.1 c/c++重要的关键字 8.1.1 const关键字 8.1.2 static关键字 8.2 内存问题 8.2.1 内存泄漏 8.2.2 野指针 8.2.3 试图修改常量 8.2.4 用错sizeof 8.3 带参的主函数 8.4 小结 8.5 习题 第三篇 c++面向对象开发第9章 类和对象 9.1 面向对象程序设计概述 9.1.1 什么是面向对象的程序设计 9.1.2 面向对象程序设计的特点 9.1.3 面向对象程序设计的优点 9.1.4 类的概念 9.2 类的声明与定义 9.2.1 类声明和类定义 9.2.2 类和结构体 9.3 类的成员变量 9.3.1 如何定义类的成员变量 9.3.2 类成员变量的赋值 9.4 类的成员函数 9.4.1 成员函数的声明和实现 9.4.2 带默认形参值的成员函数 9.4.3 内联成员函数 9.5 const成员变量与成员函数 9.5.1 const成员变量 9.5.2 const成员函数 9.6 静态成员变量与成员函数 9.6.1 静态成员变量 9.6.2 静态成员函数 9.7 小结 9.8 习题 第10章 类和对象的进一步讨论 10.1 类的构造函数 10.1.1 构造函数的概念 10.1.2 含有参数的构造函数 10.1.3 构造函数的重载 10.1.4 构造函数允许按参数默认方式调用 10.1.5 用参数初始化表对数据成员初始化 10.2 类的析构函数 10.2.1 析构函数的概念 10.2.2 析构函数的作用 10.2.3 构造函数和析构函数的调用顺序 10.3 复制构造函数 10.3.1 复制构造函数的概念与作用 10.3.2 如何定义复制构造函数 10.3.3 浅复制与深复制 10.3.4 复制构造函数的调用 10.4 类作用域和对象生存期 10.5 友元 10.5.1 友元函数 10.5.2 友元成员 10.5.3 友元类 10.6 对

象指针 10.6.1 指向对象的指针 10.6.2 this指针 10.7 对象的动态建立和释放 10.7.1 对象的创建 10.7.2 对象的释放 10.7.3 malloc/free与new/delete的区别 10.8 小结 10.9 习题 第11章 运算符重载 11.1 运算符重载概要 11.2 运算符重载 11.2.1 单目运算符重载 11.2.2 双目运算符重载 11.2.3 运算符“[]”的重载 11.2.4 运算符重载为友元函数 11.3 小结 11.4 习题 第12章 类的继承 12.1 继承与派生的概念 12.2 派生类的声明方式 12.3 派生类成员的访问权限 12.3.1 公有继承 12.3.2 私有继承 12.3.3 保护继承 12.4 派生类的构造函数和析构函数 12.4.1 派生类的构造函数 12.4.2 派生类的析构函数 12.5 虚基类 12.5.1 虚基类的概念 12.5.2 共同基类带来的二义性 12.6 基类与派生类的相互转换 12.7 小结 12.8 习题 第13章 多态与虚函数 13.1 多态与虚函数简介 13.1.1 静态联编 13.1.2 动态联编 13.1.3 虚函数概念 13.1.4 虚函数的声明和定义 13.1.5 关于虚函数的几点说明 13.1.6 虚函数的原理 13.2 虚函数的访问 13.2.1 对象名访问 13.2.2 基指针访问 13.2.3 引用访问 13.2.4 类内访问 13.2.5 在构造函数或析构函数中访问 13.3 纯虚函数与抽象类 13.3.1 纯虚函数的声明和定义 13.3.2 抽象类 13.3.3 另一种抽象类：类中只定义了protected型的构造函数 13.3.4 虚析构函数 13.4 虚函数引入的二义性 13.4.1 多基派生的二义性 13.4.2 虚继承 13.5 重载、覆盖与隐藏 13.6 小结 13.7 习题 第四篇 c++泛型编程第14章 模板 14.1 模板的概念 14.2 函数模板 14.2.1 函数模板的一般定义 14.2.2 显式具体化 14.2.3 优先级与执行顺序 14.3 类模板 14.3.1 类模板的一般定义 14.3.2 模板类的继承 14.3.3 模板类的静态数据成员 14.4 小结 14.5 习题 第15章 标准模板库 (stl) 15.1 stl概述 15.1.1 算法简介 15.1.2 容器简介 15.1.3 迭代器简介 15.1.4 适配器简介 15.2 容器 15.2.1 向量 15.2.2 列表 15.3 迭代器 15.3.1 输入和输出迭代器 15.3.2 前向迭代器、双向迭代器和随机访问迭代器 15.3.3 插入迭代器 15.3.4 混合迭代器函数 15.3.5 迭代器适配器 15.4 算法和函数对象 15.4.1 stl通用算法调用形式 15.4.2 不可变序列算法 15.4.3 可变序列算法 15.4.4 函数对象 15.5 小结 15.6 习题 第五篇 输入/输出处理和编程规范第16章 输入/输出 16.1 输入/输出概述 16.1.1 流 16.1.2 缓冲区 16.1.3 3种输入/输出流 16.1.4 重定向 16.2 c语言i/o库函数 16.2.1 标准输出函数printf 16.2.2 标准输入函数scanf 16.2.3 sprintf和sscanf函数 16.2.4 c语言的文件访问机制 16.3 流类库 16.3.1 流类库更安全、更高效 16.3.2 流类库层次 16.4 c++输入/输出流 16.4.1 标准输出流对象cout 16.4.2 其他输出流对象 16.4.3 标准输入流对象cin 16.4.4 使用格式控制符控制输出格式 16.4.5 用流对象的成员函数控制输出格式 16.4.6 用流成员函数put输出字符 16.4.7 用于字符输入的流成员函数 16.4.8 istream类的其他成员函数 16.5 重载“[[”和“]]” 16.6 c++文件输入/输出流 16.6.1 文件的输出 16.6.2 文件的输入 16.7 小结 16.8 习题 第17章 文件操作 17.1 文件的概述 17.2 文件的打开与关闭 17.2.1 文件的打开 17.2.2 文件的关闭 17.3 文件的读写 17.3.1 c++文件的读写方法 17.3.2 文本文件的读写 17.3.3 文件的定位 17.3.4 二进制文件的读写 17.4 小结 17.5 习题 第18章 命名空间 18.1 什么是命名空间 18.1.1 命名空间范例 18.1.2 定义命名空间 18.2 实体的作用域与可见域 18.2.1 局部作用域 18.2.2 全局作用域和域操作符 18.2.3 作用域嵌套及可见性 18.3 使用命名空间 18.3.1 作用域解析运算符 18.3.2 using指令 18.3.3 using指令与using声明的比较 18.4 小结 18.5 习题 第19章 异常和错误处理 19.1 编码时的防错 19.1.1 使用断言 19.1.2 调用exit、abort、\_exit和atexit函数 19.1.3 返回错误标志 19.2 异常机制 19.2.1 异常的抛出 19.2.2 异常的捕获 19.2.3 异常的处理 19.2.4 terminate函数和set\_terminate函数 19.2.5 unexpected函数与set\_unexpected函数 19.2.6 标准异常 19.3 关于异常处理的进一步讨论 19.3.1 异常规范 19.3.2 堆栈解退 19.3.3 构造函数、析构函数与异常处理 19.4 auto\_ptr类 19.4.1 使用auto\_ptr类 19.4.2 关于auto\_ptr的使用说明 19.5 小结 19.6 习题

## &lt;&lt;C++语言从入门到精通&gt;&gt;

## 章节摘录

机指令集。

使用二进制指令集开发程序是件很头痛的事，需要记住大量的二进制数。

为了便于记忆，人们将二进制数用字母组合代替，以字符串关键字代替二进制机器码的编程语言称为汇编语言，汇编语言是低级语言，虽然汇编语言比机器码容易记忆，但仍然具有可读性差的缺点，大量的跳转指令和地址值很难让程序员在很短的时间内理解程序的意思，于是编程语言进入了高级语言时代。

第一个高级语言是美国尤尼法克公司在1952年研制成功的Short Code，但被广泛使用的高级语言是FORTRAN，它是由美国科学家巴克斯设计并在IBM公司的计算机上实现的，但FORTRAN语言和Alg0160主要应用于科学和工程计算，随后出现了Pascal和C语言。

C语言是在其他语言基础上发展起来的。

首先是Richard Martin开发一种高级语言BCPL，随后Ken Thompson使用BCPL语言对其进行了简化，形成一门新的语言——B语言。

但B语言没有类型的概念，Dennis Ritchie对B语言进行了研究和改进，在B语言基础上添加了结构和类型，并将这个改进后的语言命名为c语言，寓意很简单，因为在英文字母表中，字母c是字母B的下一个字母，这预示着语言的发展。

而C++语言就是从C语言发展过来的。

Stroustrup经过钻研在c语言中加入类的概念，C++最初的名字是c with Class，到1983年12月由Rick Mascitti建议改名为c Plus Plus，即c++。

首先提出类的概念的语言是Simula，它具有很高的灵活性，但无法胜任比较大型的程序。

此后在Simula语言基础上发展的Smalltalk语言才是真正的面向对象语言，但Smalltalk-80不支持多继承

。

.....

<<C++语言从入门到精通>>

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>