

<<我的第一本C++书>>

图书基本信息

书名：<<我的第一本C++书>>

13位ISBN编号：9787560969954

10位ISBN编号：756096995X

出版时间：2011-5-5

出版时间：华中科技大学出版社

作者：陈良乔

页数：376

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>

## <<我的第一本C++书>>

### 内容概要

每一个渴望学习C++语言的初学者都希望能够有一本全面的，实用的，同时又是生动有趣的入门书。

这本入门书可以告诉这些初学者，整个C++世界有哪些内容构成，应该按照怎样的路径进行学习探索，用C++进行程序设计应该具有怎样的设计思想，同时，又如何在实践中运用这些学到的知识。

可以说，这本书基本上囊括了每个C++初学者都应该了解的所有知识，从C++的基础知识到面向对象思想，从设计模式到STL标准模板库，从Windows系统下的开发到Linux/嵌入式系统下的开发，再到最后的项目管理知识。

当一位C++初学者完成本书的学习后，他基本上浏览了C++的整个世界，可以独立地以C++进行应用开发了。

## <<我的第一本C++书>>

### 作者简介

陈良乔

程序员、自由撰稿人，毕业于西安交通大学。

敏而好学，乐于分享，连续六年获得微软MVP称号。

因撰写一系列Visual Studio

2010技术文章和Windows 7开发技术文章，2010年又获“MSDN原创之星”称号。

参与微软中国Visual

C++深度体验圆桌会议，终获得Visual

Studio贡献水晶奖杯。

平素喜好新鲜事物，但决非新潮技术的追捧者，至今还陷在设计模式、面向对象方法之中，不能自拔

。

好舞文弄墨，传播心得，著有博客：<http://imcc.blogbus.com?>

## &lt;&lt;我的第一本C++书&gt;&gt;

## 书籍目录

## 第1篇 叩开C++世界的大门

## 第1章 C++世界地图

- 1.1 C++是什么
- 1.2 C++的前世今生
  - 1.2.1 从B到C
  - 1.2.2 从C到C++
  - 1.2.3 从C++到.NET Framework的CLI
  - 1.2.4 最新标准C++0x让C++重新焕发活力
  - 1.2.5 C++和C#不得不说的那点儿事儿
  - 1.2.6 C++世界的五大子语言
- 1.3 C++世界版图
  - 1.3.1 Windows系统下的C++开发
  - 1.3.2 Linux?统下的C++开发
  - 1.3.3 嵌入式系统下的C++开发
- 1.4 如何学好C++
  - 1.4.1 将自然语言转换为C++程序设计语言
  - 1.4.2 “多读多写”是学好C++的不二法门
  - 1.4.3 和Google做朋友

## 第2章 与C++第一次亲密接触

- 2.1 一个C++程序的自白
  - 2.1.1 用Visual Studio创建C++程序
  - 2.1.2 以手工方式创建C++程序
  - 2.1.3 C++程序=预编译指令+程序代码+注释
  - 2.1.4 编译器和链接器
  - 2.1.5 C++程序的执行过程
  - 2.1.6 程序的两大任务：描述?据与处理数据
- 2.2 基本输入/输出流
  - 2.2.1 标准的输入和输出对象
  - 2.2.2 输出格式控制
  - 2.2.3 读/写文件
- 2.3 最常用的开发环境Visual Studio
  - 2.3.1 Visual C++的常用菜单
  - 2.3.2 Visual C++的常用视图
- 2.4 C++世界旅行必备的物品
  - 2.4.1 编程助手Visual Assist
  - 2.4.2 代码配置管理工具Visual Source Safe
  - 2.4.3 CodeProject和CodeGuru
  - 2.4.4 C++百科全书MSDN

## 第2篇 欢迎来到C++世界

## 第3章 C++世界众生相

- 3.1 C++中的数据类型
- 3.2 变量和常量
  - 3.2.1 声明变量
  - 3.2.2 给变量取个好名字
  - 3.2.3 变量初始化

## &lt;&lt;我的第一本C++书&gt;&gt;

- 3.2.4 常量
- 3.2.5 用宏与const关键字定义常量
- 3.3 数值类型
  - 3.3.1 整型数值类型
  - 3.3.2 浮点型数值类型
- 3.4 布尔类型
- 3.5 字符串类型
  - 3.5.1 字符类型
  - 3.5.2 字符串类型
- 3.6 数组
  - 3.6.1 数组的声明与初始化
  - 3.6.2 数组的使用
- 3.7 枚举类型
- 3.8 用结构体类型描述复杂的事物
  - 3.8.1 结构体的定义
  - 3.8.2 结构体的使用
- 3.9 指向内存位置的指针
  - 3.9.1 指针就是表示内存地址的数据类型
  - 3.9.2 指针变量的定义
  - 3.9.3 指针的赋值和使用
- 第4章 将语句编织成程序
  - 4.1 用运算符对数据进行运算
    - 4.1.1 用表达式表达设计意图
    - 4.1.2 算术运算符
    - 4.1.3 赋值操作符
    - 4.1.4 关系运算符
    - 4.1.5 逻辑运算符
    - 4.1.6 运算符之间的优先顺序
    - 4.1.7 将表达式组织成语句
  - 4.2 条件选择语句
    - 4.2.1 if语句
    - 4.2.2 并列选择的switch语句
  - 4.3 循环控制语句
    - 4.3.1 while循环
    - 4.3.2 do...while循环
    - 4.3.3 for循环
    - 4.3.4 循环控制：break和continue
  - 4.4 从语句到程序
    - 4.4.1 程序是控制语句串联起来的语句
    - 4.4.2 豪华工资统计程序
- 第5章 用函数封装程序功能
  - 5.1 函数就是一个大“箱子”
    - 5.1.1 函数的声明和定义
    - 5.1.2 函数调用机制
    - 5.1.3 函数的声明与函数调用
    - 5.1.4 函数参数的传递
    - 5.1.5 函数的返回值

## &lt;&lt;我的第一本C++书&gt;&gt;

## 5.2 内联函数

## 5.2.1 用体积换速度的内联函数

## 5.2.2 内联函数的使用规则

## 5.3 重载函数

## 5.3.1 重载函数的声明

## 5.3.2 重载函数的解析

## 5.4 函数设计的基本规则

## 5.4.1 函数声明的设计规则

## 5.4.2 函数体的设计规则

## 第6章 当C++爱上面向对象

## 6.1 从结构化设计到面向对象程序设计

## 6.1.1 “自顶向下，逐步求精”的结构化程序设计

## 6.1.2 面向对象程序设计

## 6.1.3 面向对象的三座基石：封装、继承与多态

## 6.2 类：当C++爱上面向对象

## 6.2.1 类的声明和定义

## 6.2.2 使用类创建对象

## 6.2.3 构造函数和析构函数

## 6.2.4 拷贝构造函数

## 6.2.5 操作符重载

## 6.2.6 类成员的访问控制

## 6.2.7 在友元中访问类的隐藏信息

## 6.3 类如何面向对象

## 6.3.1 用类机制实现封装

## 6.3.2 用基类和派生类实现继承

## 6.3.3 用虚函数实现多态

## 6.4 实战面向对象：工资管理系统

## 6.4.1 从问题描述中发现对象

## 6.4.2 分析对象的属性和行为

## 6.4.3 实现类的属性和行为

## 6.5 高手是这样炼成的

## 6.5.1 C++类对象的内存模型

## 6.5.2 指向自身的this指针

## 第7章 C++世界的奇人异事

## 7.1 一切指针都是纸老虎：彻底理解指针

## 7.1.1 指针的运算

## 7.1.2 灵活的void类型和void类型指针

## 7.1.3 指向指针的指针

## 7.1.4 指针在函数中的应用

## 7.1.5 引用

## 7.2 程序中的异常处理

## 7.2.1 异常处理

## 7.2.2 异常的函数接口声明

## 7.2.3 合理使用异常处理

## 7.3 编写更复杂的C++程序

## 7.3.1 源文件和头文件

## 7.3.2 名字空间

## &lt;&lt;我的第一本C++书&gt;&gt;

7.3.3 作用域与可见性

7.3.4 编译预处理

7.4 高手是这样炼成的

7.4.1 用宏定义化繁为简

7.4.2 用typedef定义类型的别名

7.4.3 用const保护数据

第3篇 攀登C++世界的高峰

第8章 用STL优雅你的程序

8.1 跟STL做朋友

8.1.1 算法 + 容器 + 迭代器 = STL

8.1.2 在程序中使用STL

8.1.3 STL到底好在哪里

8.2 用模板实现通用算法

8.2.1 函数模板

8.2.2 类模板

8.2.3 模板的实例化

8.2.4 用模板实现通用算法

第9章 STL中的容器管理数据

9.1 容器就是STL中的瓶瓶罐罐

9.1.1 操作容器中的数据元素

9.1.2 使用迭代器访问容器中的数据元素

9.1.3 容器的使用说明书

9.1.4 如何选择合适的容器

9.2 vector容器是数组的最佳替代者

9.2.1 创建并初始化vector对象

9.2.2 vector容器的操作

9.2.3 访问vector容器中的数据

9.3 可以保存键值对的map容器

9.3.1 创建并初始化map容器

9.3.2 将数据保存到map容器中

9.3.3 根据键找到对应的值

第10章 用STL中的通用算法处理数据

10.1 STL算法中的“四大帮派”

10.2 容器元素的查找与遍历

10.2.1 用for\_each()算法遍历容器中的数据元素

10.2.2 用find()和find\_if()算法实现线性查找

10.3 容器元素的复制与变换

10.3.1 复制容器元素：copy()算法

10.3.2 合并容器元素：merge()算法

10.3.3 变换容器元素：transform函数

10.4 容器元素的排序

10.4.1 使用sort()算法对容器中的数据进行排序

10.4.2 对排序的规则进行自定义

10.5 实战STL算法

10.5.1 “算法”老师带来的一堂别开生面的体育课

10.5.2 删除容器中的冗余元素

第11章 函数指针、函数对象与Lambda表达式

## &lt;&lt;我的第一本C++书&gt;&gt;

## 11.1 函数指针

## 11.1.1 函数指针的声明与赋值

## 11.1.2 用函数指针调用函数

## 11.1.3 用函数指针实现回调函数

## 11.1.4 将函数指针应用到STL算法中

## 11.2 函数对象

## 11.2.1 定义一个函数对象

## 11.2.2 利用函数对象记住状态数据

## 11.3 用Lambda表达式编写更简单的函数

## 11.3.1 最简单直接的函数表达形式：Lambda表达式

## 11.3.2 Lambda表达式的语法规则

## 11.3.3 Lambda表达式的复用

## 第12章 C++世界的几件新鲜事

## 12.1 用右值引用榨干C++的性能

## 12.1.1 什么是右值

## 12.1.2 右值引用在函数返回值上的应用

## 12.1.3 STL算法中被浪费的右值

## 12.1.4 右值引用如何提高性能

## 12.2 智能指针shared\_ptr

## 12.2.1 C++的内存管理

## 12.2.2 用聪明的shared\_ptr解决内存管理问题

## 12.2.3 智能指针的应用场景

## 12.2.4 shared\_ptr的使用

## 12.2.5 shared\_ptr与标准库容器

## 12.2.6 对shared\_ptr进行自定义

## 12.3 用PPL进行多线程开发

## 12.3.1 多核给程序设计带来的挑战

## 12.3.2 PPL带来免费的午餐

## 12.3.3 PPL中的并行算法

## 12.3.4 PPL中的并行任务

## 12.3.5 PPL中的并行对象和并行容器

## 12.3.6 PPL之外的另一种选择：OpenMP

## 第13章 找工作就靠它了

## 13.1 打好基础

## 13.1.1 基本概念

## 13.1.2 函数

## 13.1.3 面向对象思想

## 13.1.4 类与对象

## 13.1.5 STL

## 13.2 积累经验

## 13.3 考查智力

## 附录A 接下来该读什么书

## A.1 开山鼻祖：《C++程序设计语言》

## A.2 初学者必看：《C++ Primer中文版（第4版）》

## A.3 百科全书：《代码大全，第2版》

## A.4 内功秘籍：《Effective C++（Third Edition）》

## A.5 经验很重要：《C++编程规范》



<<我的第一本C++书>>

后记

<<我的第一本C++书>>

编辑推荐

探索C++世界的旅游地图    生动有趣又不乏实用性    掌握C++世界的运作原理    学习C++  
世界“居民”的思考方式    自由地穿梭于现实世界和C++世界

<<我的第一本C++书>>

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>