第一图书网, tushu007.com

<<C++程序设计语言>>

图书基本信息

书名：<<C++程序设计语言>>

13位ISBN编号：9787111298854

10位ISBN编号：7111298853

出版时间：2010-3

出版时间：机械工业出版社

作者：斯特朗斯特鲁普

页数：905

译者：裘宗燕

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

前言

Bjarne Stroustrup的《The C++ Programming Language》是有关C++语言的第一部著作。
毫无疑问，它是关于C++语言及其程序设计的最重要著作，在此领域中的地位是无可替代的。
《The C++ Programming Language》一书伴随着C++语言的发展演化而不断进步，经过第1版（1985年）、第2版（1991年），第3版（1998年），本书的英文原书是《The C++ Programming Language》第3版经过补充和修订后的“特别版（2000）”（对应于国内引进的影印本）。
对于这个中译本，我想说的第一句话就是“来得太晚了”。
要学习C++语言和程序设计，要将C++应用于程序设计实践，本书自然是必读之书。
这个“特别版”以标准化的C++语言为基础，讨论了C++的各种语言特征和有效使用这一语言的程序设计技术。
书中也用了大量的篇幅，在标准库以及一般软件开发的环境下，讨论了使用C++语言编程和组织程序的许多高级技术。
本书内容覆盖了C++语言及其程序设计的各个方面，其技术深度与广度是举世公认的。
然而，作者讨论的并不仅是C++语言及其程序设计。
本书的讨论远远超出这一范围，第四部分用了大量的篇幅去讨论软件开发过程及其问题。
即使是在介绍C++语言及其程序设计的具体问题时，作者也常在程序结构、设计和软件开发的大环境下，提出自己的许多认识。
作者有很强的计算机科学与技术基础，在系统软件开发方面极富经验，他所提出的观点和意见值得每个在这个领域中工作的人的重视。
当然，重视并不是盲从。
在Stroustrup的两本关于C++的重要著作（本书和《C++语言的设计与演化》（已由机械工业出版社出版））中，都有这样一句话使我印象深刻：希望读者带着一种健康的怀疑态度。
看来这是作者深深铭刻在心的一种思想，也特别值得国内每个从事信息技术，或者努力向这个方向发展的人注意。
从来就没有什么救世主，Stroustrup不是在传道，他只是在总结和论述自己在这个领域中工作的经验。
请不要将本书中的东西作为教条，那也一定是本书作者所深恶痛绝的。
许多人说本书比较难读，这种说法有一定道理。
真正理解本书的一般性内容需要花一些时间，融会贯通则更需要下功夫。
理解本书的内容不仅需要去读它，还需要去实践。
问题是，花这个时间值吗？
作者在讨论C++语言的设计时提出了一个观点：你从C++语言中的收获大致与在学习实践这个语言的过程中所付出的努力成正比；而且C++是一个可以伴随你成长的语言。
同样的话也适用于本书。
如果你致力于将自己发展成一个职业的程序员，或者要在计算机方面的技术领域中长期工作下去，我认为，你从本书中的收获大致也会与你所花的时间成正比，这本书也是一本能够伴随你成长的书。

内容概要

本书是在C++语言和程序设计领域具有深远影响、畅销不衰的著作，由C++语言的设计者编写，对C++语言进行了最全面、最权威的论述，覆盖标准C++以及由C++所支持的关键性编程技术和设计技术。
本书英文原版一经面世，即引起业内人士的高度评价和热烈欢迎，先后被翻译成德、希、匈、西、荷、法、日、俄、中、韩等近20种语言，数以百万计的程序员从中获益，是无可取代的C++经典力作。

在本书英文原版面世10年后的今天，特别奉上十周年中文纪念版，希望众多具有丰富实战经验的C++开发人员能够温故而知新，印证学习心得，了解更加本质的C++知识，让获得的理论应用得更加灵活，也期望新的C++程序员从中认识到这本书的价值所在，从更高的起点出发，书写更加精彩的程序设计人生。

作者简介：Bjarne Stroustrup，英国剑桥大学计算机科学博士，C++语言的设计者和最初的实现者，也是《C++程序设计原理与实践》和《C++语言的设计和演化》的作者。
他现在是德州农工大学计算机科学首席教授，同时不审AT&T贝尔实验室特别成员。
1993年，由于在C++领域的重大贡献，他获得了ACM的Grace Murray Hopper大奖并成为ACM院士；2008年，他又获得了Dr.Dobb's杂志的程序设计杰出奖。
在进入学术界之前，他在AT&T贝尔实验室工作。
他是ISO C++标准委员会的创始人之一。

书籍目录

出版者的话
专家指导委员会
中文版序
译者序
序
第2版序
第1版序
导论
第1章 致读者
1.1 本书的结构
1.1.1 例子和参考
1.1.2 练习
1.1.3 有关实现的注记
1.2 学习C++
1.3 C++ 的设计
1.3.1 效率和结构
1.3.2 哲学注记
1.4 历史注记
1.5 C++ 的使用
1.6 C和C++
1.6.1 给C程序员的建议
1.6.2 给C++程序员的建议
1.7 有关在C++里编程的思考
1.8 忠告
1.9 参考文献
第2章 C++概览
2.1 为什么是C++
2.2 程序设计范型
2.3 过程式程序设计
2.3.1 变量和算术
2.3.2 检测和循环
2.3.3 指针和数组
2.4 模块程序设计
2.4.1 分别编译
2.4.2 异常处理
2.5 数据抽象
2.5.1 定义类型的模块
2.5.2 用户定义类型
2.5.3 具体类型
2.5.4 抽象类型
2.5.5 虚函数
2.6 面向对象的程序设计
2.6.1 具体类型的问题
2.6.2 类层次结构
2.7 通用型程序设计

2.7.1 容器
2.7.2 通用型算法
2.8 附言
2.9 忠告
第3章 标准库概览
3.1 引言
3.2 Hello, world！
3.3 标准库名字空间
3.4 输出
3.5 字符串
3.5.1 C风格的字符串
3.6 输入
3.7 容器
3.7.1 向量—vector
3.7.2 范围检查
3.7.3 表—list
3.7.4 映射—map
3.7.5 标准容器
3.8 算法
3.8.1 迭代器的使用
3.8.2 迭代器类型
3.8.3 迭代器和I/O
3.8.4 遍历和谓词
3.8.5 使用成员函数的算法
3.8.6 标准库算法
3.9 数学
3.9.1 复数
3.9.2 向量算术
3.9.3 基本数值支持
3.10 标准库功能
3.11 忠告
第一部分 基本功能
第4章 类型和声明
4.1 类型
4.1.1 基本类型
4.2 布尔量
4.3 字符类型
4.3.1 字符文字量
4.4 整数类型
4.4.1 整数文字量
4.5 浮点类型
4.5.1 浮点文字量
4.6 大小
4.7 void
4.8 枚举
4.9 声明

4.9.1 声明的结构
4.9.2 声明多个名字
4.9.3 名字
4.9.4 作用域
4.9.5 初始化
4.9.6 对象和左值
4.9.7 typedef
4.10 忠告
4.11 练习
第5章 指针、数组和结构
5.1 指针
5.1.1 零
5.2 数组
5.2.1 数组初始化
5.2.2 字符串文字量
5.3 到数组的指针
5.3.1 在数组里漫游
5.4 常量
5.4.1 指针和常量
5.5 引用
5.6 指向void的指针
5.7 结构
5.7.1 类型等价
5.8 忠告
5.9 练习
第6章 表达式和语句
6.1 一个桌面计算器
6.1.1 分析器
6.1.2 输入函数
6.1.3 低级输入
6.1.4 错误处理
6.1.5 驱动程序
6.1.6 头文件
6.1.7 命令行参数
6.1.8 有关风格的注记
6.2 运算符概览
6.2.1 结果
6.2.2 求值顺序
6.2.3 运算符优先级
6.2.4 按位逻辑运算符
6.2.5 增量和减量
6.2.6 自由存储
6.2.7 显式类型转换
6.2.8 构造函数
6.3 语句概览
6.3.1 声明作为语句
6.3.2 选择语句

6.3.3 迭代语句
6.3.4 goto
6.4 注释和缩进编排
6.5 忠告
6.6 练习
第7章 函数
7.1 函数声明
7.1.1 函数定义
7.1.2 静态变量
7.2 参数传递
7.2.1 数组参数
7.3 返回值
7.4 重载函数名
7.4.1 重载和返回类型
7.4.2 重载与作用域
7.4.3 手工的歧义性解析
7.4.4 多参数的解析
7.5 默认参数
7.6 未确定数目的参数
7.7 指向函数的指针
7.8 宏
7.8.1 条件编译
7.9 忠告
7.10 练习
第8章 名字空间和异常
8.1 模块化和界面
8.2 名字空间
8.2.1 带限定词的名字
8.2.2 使用声明
8.2.3 使用指令
8.2.4 多重界面
8.2.5 避免名字冲突
8.2.6 名字查找
8.2.7 名字空间别名
8.2.8 名字空间组合
8.2.9 名字空间和老代码
8.3 异常
8.3.1 抛出和捕捉
8.3.2 异常的辨识
8.3.3 在计算器中的异常
8.4 忠告
8.5 练习
第9章 源文件和程序
9.1 分别编译
9.2 连接
9.2.1 头文件
9.2.2 标准库头文件

9.2.3 单一定义规则
9.2.4 与非C++代码的连接
9.2.5 连接与指向函数的指针
9.3 使用头文件
9.3.1 单一头文件
9.3.2 多个头文件
9.3.3 包含保护符
9.4 程序
9.4.1 非局部变量的初始化
9.5 忠告
9.6 练习
第二部分 抽象机制
第10章 类
10.1 引言
10.2 类
10.2.1 成员函数
10.2.2 访问控制
10.2.3 构造函数
10.2.4 静态成员
10.2.5 类对象的复制
10.2.6 常量成员函数
10.2.7 自引用
10.2.8 结构和类
10.2.9 在类内部的函数定义
10.3 高效的用户定义类型
10.3.1 成员函数
10.3.2 协助函数
10.3.3 重载的运算符
10.3.4 具体类型的意义
10.4 对象
10.4.1 析构函数
10.4.2 默认构造函数
10.4.3 构造和析构
10.4.4 局部变量
10.4.5 自由存储
10.4.6 类对象作为成员
10.4.7 数组
10.4.8 局部静态存储
10.4.9 非局部存储
10.4.10 临时对象
10.4.11 对象的放置
10.4.12 联合
10.5 忠告
10.6 练习
第11章 运算符重载
11.1 引言
11.2 运算符函数

11.2.1 二元和一元运算符
11.2.2 运算符的预定义意义
11.2.3 运算符和用户定义类型
11.2.4 名字空间里的运算符
11.3 一个复数类型
11.3.1 成员运算符和非成员运算符
11.3.2 混合模式算术
11.3.3 初始化
11.3.4 复制
11.3.5 构造函数和转换
11.3.6 文字量
11.3.7 另一些成员函数
11.3.8 协助函数
11.4 转换运算符
11.4.1 歧义性
11.5 友元
11.5.1 友元的寻找
11.5.2 友元和成员
11.6 大型对象
11.7 基本运算符
11.7.1 显式构造函数
11.8 下标
11.9 函数调用
11.10 间接
11.11 增量和减量
11.12 一个字符串类
11.13 忠告
11.14 练习
第12章 派生类
12.1 引言
12.2 派生类
12.2.1 成员函数
12.2.2 构造函数和析构函数
12.2.3 复制
12.2.4 类层次结构
12.2.5 类型域
12.2.6 虚函数
12.3 抽象类
12.4 类层次结构的设计
12.4.1 一个传统的层次结构
12.4.2 抽象类
12.4.3 其他实现方式
12.4.4 对象创建的局部化
12.5 类层次结构和抽象类
12.6 忠告
12.7 练习
第13章 模板

13.1 引言
13.2 一个简单的String模板
13.2.1 定义一个模板
13.2.2 模板实例化
13.2.3 模板参数
13.2.4 类型等价
13.2.5 类型检查
13.3 函数模板
13.3.1 函数模板的参数
13.3.2 函数模板的重载
13.4 用模板参数描述策略
13.4.1 默认模板参数
13.5 专门化
13.5.1 专门化的顺序
13.5.2 模板函数的专门化
13.6 派生和模板
13.6.1 参数化和继承
13.6.2 成员模板
13.6.3 继承关系
13.7 源代码组织
13.8 忠告
13.9 练习
第14章 异常处理
14.1 错误处理
14.1.1 关于异常的其他观点
14.2 异常的结组
14.2.1 派生的异常
14.2.2 多个异常的组合
14.3 捕捉异常
14.3.1 重新抛出
14.3.2 捕捉所有异常
14.4 资源管理
14.4.1 构造函数和析构函数的使用
14.4.2 auto_ptr
14.4.3 告诫
14.4.4 异常和new
14.4.5 资源耗尽
14.4.6 构造函数里的异常
14.4.7 析构函数里的异常
14.5 不是错误的异常
14.6 异常的描述
14.6.1 对异常描述的检查
14.6.2 未预期的异常
14.6.3 异常的映射
14.7 未捕捉的异常
14.8 异常和效率
14.9 处理错误的其他方式

14.10 标准异常
14.11 忠告
14.12 练习
第15章 类层次结构
15.1 引言和概述
15.2 多重继承
15.2.1 歧义性解析
15.2.2 继承和使用声明
15.2.3 重复的基类
15.2.4 虚基类
15.2.5 使用多重继承
15.3 访问控制
15.3.1 保护成员
15.3.2 对基类的访问
15.4 运行时类型信息
15.4.1 dynamic_cast
15.4.2 在类层次结构中漫游
15.4.3 类对象的构造与析构
15.4.4 typeid和扩展的类型信息
15.4.5 RTTI的使用和误用
15.5 指向成员的指针
15.5.1 基类和派生类
15.6 自由存储
15.6.1 数组分配
15.6.2 虚构造函数
15.7 忠告
15.8 练习
第三部分 标 准 库
第16章 库组织和容器
16.1 标准库的设计
16.1.1 设计约束
16.1.2 标准库组织
16.1.3 语言支持
16.2 容器设计
16.2.1 专门化的容器和迭代器
16.2.2 有基类的容器
16.2.3 STL容器
16.3 向量
16.3.1 类型
16.3.2 迭代器
16.3.3 元素访问
16.3.4 构造函数
16.3.5 堆栈操作
16.3.6 表操作
16.3.7 元素定位
16.3.8 大小和容量
16.3.9 其他成员函数

16.3.10 协助函数
16.3.11 vector[bool]
16.4 忠告
16.5 练习
第17章 标准容器
17.1 标准容器
17.1.1 操作综述
17.1.2 容器综述
17.1.3 表示
17.1.4 对元素的要求
17.2 序列
17.2.1 向量—vector
17.2.2 表—list
17.2.3 双端队列—deque
17.3 序列适配器 1
17.3.1 堆栈—stack 1
17.3.2 队列—queue
17.3.3 优先队列—priority_queue
17.4 关联容器
17.4.1 映射—map
17.4.2 多重映射—multimap
17.4.3 集合—set
17.4.4 多重集合—multiset
17.5 拟容器
17.5.1 串—string
17.5.2 值向量—valarray
17.5.3 位集合—bitset
17.5.4 内部数组
17.6 定义新容器
17.6.1 散列映射—hash_map
17.6.2 表示和构造
17.6.3 其他散列关联容器
17.7 忠告
17.8 练习
第18章 算法和函数对象
18.1 引言
18.2 标准库算法综述
18.3 序列和容器
18.3.1 输入序列
18.4 函数对象
18.4.1 函数对象的基类
18.4.2 谓词
18.4.3 算术函数对象
18.4.4 约束器、适配器和否定器
18.5 非修改性序列算法
18.5.1 对每个做—for_each
18.5.2 查找族函数

18.5.3 计数
18.5.4 相等和不匹配
18.5.5 搜索
18.6 修改性序列算法
18.6.1 复制
18.6.2 变换
18.6.3 惟一化
18.6.4 取代
18.6.5 删除
18.6.6 填充和生成
18.6.7 反转和旋转
18.6.8 交换
18.7 排序的序列
18.7.1 排序
18.7.2 二分检索
18.7.3 归并
18.7.4 划分
18.7.5 序列上的集合运算
18.8 堆
18.9 最小和最大
18.10 排列
18.11 C风格算法
18.12 忠告
18.13 练习
第19章 迭代器和分配器
19.1 引言
19.2 迭代器和序列
19.2.1 迭代器的操作
19.2.2 迭代器特征类—iterator_traits
19.2.3 迭代器类别
19.2.4 插入器
19.2.5 反向迭代器
19.2.6 流迭代器
19.3 带检查迭代器
19.3.1 异常、容器和算法
19.4 分配器
19.4.1 标准分配器
19.4.2 一个用户定义分配器
19.4.3 广义的分配器
19.4.4 未初始化的存储
19.4.5 动态存储
19.4.6 C风格的分配
19.5 忠告
19.6 练习
第20章 串
20.1 引言
20.2 字符

20.2.1 字符特征类—char_traits
20.3 基础串类—basic_string
20.3.1 类型
20.3.2 迭代器
20.3.3 元素访问
20.3.4 构造函数
20.3.5 错误
20.3.6 赋值
20.3.7 到C风格字符串的转换
20.3.8 比较
20.3.9 插入
20.3.10 拼接
20.3.11 查找
20.3.12 替换
20.3.13 子串
20.3.14 大小和容量
20.3.15 I/O操作
20.3.16 交换
20.4 C标准库
20.4.1 C风格字符串
20.4.2 字符分类
20.5 忠告
20.6 练习
第21章 流
21.1 引言
21.2 输出
21.2.1 输出流
21.2.2 内部类型的输出
21.2.3 用户定义类型的输出
21.3 输入
21.3.1 输入流
21.3.2 内部类型的输入
21.3.3 流状态
21.3.4 字符的输入
21.3.5 用户定义类型的输入
21.3.6 异常
21.3.7 流的联结
21.3.8 哨位
21.4 格式化
21.4.1 格式状态
21.4.2 整数输出
21.4.3 浮点数输出
21.4.4 输出域
21.4.5 域的调整
21.4.6 操控符
21.5 文件流与字符串流
21.5.1 文件流

21.5.2 流的关闭
21.5.3 字符串流
21.6 缓冲
21.6.1 输出流和缓冲区
21.6.2 输入流和缓冲区
21.6.3 流和缓冲区
21.6.4 流缓冲区
21.7 现场
21.7.1 流回调
21.8 C输入/输出
21.9 忠告
21.10 练习
第22章 数值
22.1 引言
22.2 数值的限制
22.2.1 表示限制的宏
22.3 标准数学函数
22.4 向量算术
22.4.1 valarray的构造
22.4.2 valarray的下标和赋值
22.4.3 成员操作
22.4.4 非成员函数
22.4.5 切割
22.4.6 切割数组—slice_array
22.4.7 临时量、复制和循环
22.4.8 广义切割
22.4.9 屏蔽
22.4.10 间接数组—indirect_array
22.5 复数算术
22.6 通用数值算法
22.6.1 累积—accumulate
22.6.2 内积—inner_product
22.6.3 增量变化
22.7 随机数
22.8 忠告
22.9 练习
第四部分 用C++ 做设计
第23章 开发和设计
23.1 概述
23.2 引言
23.3 目的与手段
23.4 开发过程
23.4.1 开发循环
23.4.2 设计目标
23.4.3 设计步骤
23.4.4 试验和分析
23.4.5 测试

23.4.6 软件维护
23.4.7 效率
23.5 管理
23.5.1 重用
23.5.2 规模
23.5.3 个人
23.5.4 混成设计
23.6 带标注的参考文献
23.7 忠告
第24章 设计和编程
24.1 概述
24.2 设计和程序设计语言
24.2.1 忽视类
24.2.2 忽视继承
24.2.3 忽视静态类型检查
24.2.4 忽视程序设计
24.2.5 排他性地使用类层次结构
24.3 类
24.3.1 类表示什么
24.3.2 类层次结构
24.3.3 包容关系
24.3.4 包容和继承
24.3.5 使用关系
24.3.6 编入程序里的关系
24.3.7 类内的关系
24.4 组件
24.4.1 模板
24.4.2 界面和实现
24.4.3 肥大的界面
24.5 忠告
第25章 类的作用
25.1 类的种类
25.2 具体类型
25.2.1 具体类型的重用
25.3 抽象类型
25.4 结点
25.4.1 修改界面
25.5 动作
25.6 界面类
25.6.1 调整界面
25.7 句柄类
25.7.1 句柄上的操作
25.8 应用框架
25.9 忠告
25.10 练习
附录和索引
附录A 语法

附录B 兼容性
附录C 技术细节
附录D 现场
附录E 标准库的异常时安全性
索引

章节摘录

插图：c++基本类型的某些方面是由实现确定的，例如int的大小（C.2节）。
我总要指出这种依赖性，并常常提出应该避免它们，或者通过某些方式尽可能减小其影响的建议。
为什么需要为这些东西操心呢？
在各种各样的系统中或使用多种编译器去编程序的人们很注意这些事情，因为如果他们不这样做，那他们就会被迫去花时间寻找和纠正很隐蔽的错误。
那些自称不关心移植性的人也确实常常像自己所说的那样去做，因为他们只使用一种系统，并认为自己能承担起如下看法：“ 这个语言不过就是我的编译器所实现的那种东西 ”。
但是，这实际上是一种狭隘和短视的观点。
如果你的程序是成功的，那么它就很可能需要移植，而这时某些人就必须去寻找并纠正那些依赖于实现的特征了。
此外，程序经常需要为了同一个系统而用其他编译器来编译，甚至你最喜爱的编译器的未来版本在做某些事情时，也可能采用与目前的版本不同的方式。
在写程序时，理解程序对实现的依赖性的影响并予以限制，比以后再去踩这个泥潭要容易得多。
限制依赖于实现的语言特征的影响并不太难，限制依赖于系统的库功能的影响就困难得多了。
在所有可能之处都使用标准库的功能是一个办法。
提供了多种整数类型、多种无符号类型、多种浮点类型的原因就是希望使程序员能够利用各种硬件特性。
在许多机器上，不同种类的基础类型之间，在对存储的需求、存储访问时间和计算速度方面存在着明显的差异。
如果你了解一台机器，那么就很容易做出选择，例如，选择对某个变量所适用的整数类型。
而写出真正可移植的低级代码就要困难得多。

编辑推荐

《C++程序设计语言(特别版·十周年中文纪念版)》：十周年纪念版体味C++语言的精妙与魅力享受与大师的心灵对话1979年，Biarne Stroustrup博士开始进行一项现在看来具有深远意义的工作一在C语言的基础上实现了内建支持面向对象程序设计方法的C with Classes，这就是震铄当代、让全世界数百万程序员如痴如狂的C++语言的前身。
1998年，ANSI／ISOC++标准建立，C++的标准化标志着Stroustrup博士俪注多年心血的伟大构想终于实现。
2000年，Stroustrup博士推出其经典著作The C++ Programming Language的特别版，这位C++之父将其对C++语言要义的理解、对编程精髓的把握、致C++程序员的箴言融会在这本书中，使《C++程序设计语言(特别版·十周年中文纪念版)》自面世以来便成为C++编程领域最重要的著作，对全世界C++程序员产生了广泛而深刻的影响。
十年后，让我们重温经典，体味C++语言的精妙与魅力，享受与大师的心灵对话……C++之父的经典之作第1版1985年，第2版1991年，第3版1997年，特别版2000年，让经典无限延伸……《C++程序设计语言(特别版·十周年中文纪念版)》是在C++语言和程序设计领域具有深远影响、畅销不衰的著作，由C++语言之父Bjarne Stroustrup撰写，对C++语言进行了最全面、最权威的论述，覆盖标准C++以及由C++所支持的关键性编程技术和设计技术。
《C++程序设计语言(特别版·十周年中文纪念版)》英文原版一经面世，即引起业内人士的高度评价和热烈欢迎，先后被翻译成德、希、匈、西、荷、法、日、俄、中、韩等近20种语言，数以百万计的程序员从中获益，是拥有最多读者、使用最广泛的C++著作。
在《C++程序设计语言(特别版·十周年中文纪念版)》英文原版面世10年后的今天，特别奉上十周年中文纪念版，希望众多具有丰富实战经验的C++开发人员能够温故而知新，印证学习心得，了解更加本质的C++知识，让获得的理论应用得更加灵活，也期望新的C++程序员从中认识到这本书的价值所在，从更高的起点出发，书写更加精彩的程序设计人生。

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