

<<C++编程思想>>

图书基本信息

书名：<<C++编程思想>>

13位ISBN编号：9787111071167

10位ISBN编号：7111071166

出版时间：2000-01

出版时间：机械工业出版社

作者：埃克尔(美)

译者：刘宗田/等

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++编程思想&gt;&gt;

## 书籍目录

- 目录
- 译者序
- 前言
- 第1章 对象的演化
  - 1.1 基本概念
    - 1.1.1 对象：特性十行为
    - 1.1.2 继承：类型关系
    - 1.1.3 多态性
    - 1.1.4 操作概念：OOP程序像什么
  - 1.2 为什么C++会成功
    - 1.2.1 较好的C
    - 1.2.2 采用渐进的学习方式
    - 1.2.3 运行效率
    - 1.2.4 系统更容易表达和理解
    - 1.2.5 “库”使你事半功倍
    - 1.2.6 错误处理
    - 1.2.7 大程序设计
  - 1.3 方法学介绍
    - 1.3.1 复杂性
    - 1.3.2 内部原则
    - 1.3.3 外部原则
    - 1.3.4 对象设计的五个阶段
    - 1.3.5 方法承诺什么
    - 1.3.6 方法应当提供什么
  - 1.4 起草：最小的方法
    - 1.4.1 前提
    - 1.4.2 高概念
    - 1.4.3 论述（treatment）
    - 1.4.4 结构化
    - 1.4.5 开发
    - 1.4.6 重写
    - 1.4.7 逻辑
  - 1.5 其他方法
    - 1.5.1 Booch
    - 1.5.2 责任驱动的设计（RDD）
    - 1.5.3 对象建模技术（OMT）
  - 1.6 为向OOP转变而采取的策略
    - 1.6.1 逐步进入OOP
    - 1.6.2 管理障碍
  - 1.7 小结
- 第2章 数据抽象
  - 2.1 声明与定义
  - 2.2 一个袖珍C库
  - 2.3 放在一起：项目创建工具
  - 2.4 什么是非正常

## &lt;&lt;C++编程思想&gt;&gt;

- 2.5基本对象
- 2.6什么是对象
- 2.7抽象数据类型
- 2.8对象细节
- 2.9头文件形式
- 2.10嵌套结构
- 2.11小结
- 2.12练习
- 第3章 隐藏实现
- 3.1设置限制
- 3.2C++的存取控制
- 3.3友元
- 3.3.1嵌套友元
- 3.3.2它是纯的吗
- 3.4对象布局
- 3.5类
- 3.5.1用存取控制来修改stash
- 3.5.2用存取控制来修改stack
- 3.6句柄类 ( handleclasses )
- 3.6.1可见的实现部分
- 3.6.2减少重复编译
- 3.7小结
- 3.8练习
- 第4章 初始化与清除
- 4.1用构造函数确保初始化
- 4.2用析构函数确保清除
- 4.3清除定义块
- 4.3.1for循环
- 4.3.2空间分配
- 4.4含有构造函数和析构函数的stash
- 4.5含有构造函数和析构函数的stack
- 4.6集合初始化
- 4.7缺省构造函数
- 4.8小结
- 4.9练习
- 第5章 函数重载与缺省参数
- 5.1范围分解
- 5.1.1用返回值重载
- 5.1.2安全类型连接
- 5.2重载的例子
- 5.3缺省参数
- 5.4小结
- 5.5练习
- 第6章 输入输出流介绍
- 6.1为什么要用输入输出流
- 6.2解决输入输出流问题
- 6.2.1预先了解操作符重载

## &lt;&lt;C++编程思想&gt;&gt;

- 6.2.2插入符与提取符
- 6.2.3通常用法
- 6.2.4面向行的输入
- 6.3文件输入输出流
- 6.4输入输出流缓冲
- 6.5在输入输出流中查找
- 6.6strstreams
  - 6.6.1为用户分配的存储
  - 6.6.2自动存储分配
- 6.7输出流格式化
  - 6.7.1内部格式化数据
  - 6.7.2例子
- 6.8格式化操纵算子
- 6.9建立操纵算子
- 6.10输入输出流实例
  - 6.10.1代码生成
  - 6.10.2一个简单的数据记录
- 6.11小结
- 6.12练习
- 第7章 常量
  - 7.1值替代
    - 7.1.1头文件里的const
    - 7.1.2const的安全性
    - 7.1.3集合
    - 7.1.4与C语言的区别
  - 7.2指针
    - 7.2.1指向const的指针
    - 7.2.2const指针
    - 7.2.3赋值和类型检查
  - 7.3函数参数和返回值
    - 7.3.1传递const值
    - 7.3.2返回const值
    - 7.3.3传递和返回地址
  - 7.4类
    - 7.4.1类里的const和enum
    - 7.4.2编译期间类里的常量
    - 7.4.3const对象和成员函数
    - 7.4.4只读存储能力
  - 7.5可变的 ( volatile )
  - 7.6小结
  - 7.7练习
- 第8章 内联函数
  - 8.1预处理器的缺陷
  - 8.2内联函数
    - 8.2.1类内部的内联函数
    - 8.2.2存取函数
  - 8.3内联函数和编译器

## &lt;&lt;C++编程思想&gt;&gt;

- 8.3.1 局限性
- 8.3.2 赋值顺序
- 8.3.3 在构造函数和析构函数里隐藏行为
- 8.4 减少混乱
- 8.5 预处理器的特点
- 8.6 改进的错误检查
- 8.7 小结
- 8.8 练习
- 第9章 命名控制
  - 9.1 来自C语言中的静态成员
    - 9.1.1 函数内部的静态变量
    - 9.1.2 控制连接
    - 9.1.3 其他的存储类型指定符
  - 9.2 名字空间
    - 9.2.1 产生一个名字空间
    - 9.2.2 使用名字空间
  - 9.3 C++ 中的静态成员
    - 9.3.1 定义静态数据成员的存储
    - 9.3.2 嵌套类和局部类
    - 9.3.3 静态成员函数
  - 9.4 静态初始化的依赖因素
  - 9.5 转换连接指定
  - 9.6 小结
  - 9.7 练习
- 第10章 引用和拷贝构造函数
  - 10.1 C++ 中的指针
  - 10.2 C++ 中的引用
    - 10.2.1 函数中的引用
    - 10.2.2 参数传递准则
  - 10.3 拷贝构造函数
    - 10.3.1 传值方式传递和返回
    - 10.3.2 拷贝构造函数
    - 10.3.3 缺省拷贝构造函数
    - 10.3.4 拷贝构造函数方法的选择
  - 10.4 指向成员的指针 (简称成员指针)
  - 10.5 小结
  - 10.6 练习
- 第11章 运算符重载
  - 11.1 警告和确信
  - 11.2 语法
  - 11.3 可重载的运算符
    - 11.3.1 一元运算符
    - 11.3.2 二元运算符
    - 11.3.3 参数和返回值
    - 11.3.4 与众不同的运算符
    - 11.3.5 不能重载的运算符
  - 11.4 非成员运算符

## &lt;&lt;C++编程思想&gt;&gt;

- 11.5重载赋值符
- 11.6自动类型转换
  - 11.6.1构造函数转换
  - 11.6.2运算符转换
  - 11.6.3一个理想的例子：strings
  - 11.6.4自动类型转换的缺陷
- 11.7小结
- 11.8练习
- 第12章 动态对象创建
  - 12.1对象创建
    - 12.1.1C从堆中获取存储单元的方法
    - 12.1.2运算符new
    - 12.1.3运算符delete
    - 12.1.4一个简单的例子
    - 12.1.5内存管理的开销
  - 12.2重新设计前面的例子
    - 12.2.1仅从堆中创建string类
    - 12.2.2stash指针
    - 12.2.3stack例子
  - 12.3用于数组的new和delete
  - 12.4用完内存
  - 12.5重载new和delete
    - 12.5.1重载全局new和delete
    - 12.5.2为一个类重载new和delete
    - 12.5.3为数组重载new和delete
    - 12.5.4构造函数调用
    - 12.5.5对象放置
  - 12.6小结
  - 12.7练习
- 第13章 继承和组合
  - 13.1组合语法
  - 13.2继承语法
  - 13.3构造函数的初始化表达式表
    - 13.3.1成员对象初始化
    - 13.3.2在初始化表达式表中的内置类型
  - 13.4组合和继承的联合
    - 13.4.1构造函数和析构函数的次序
    - 13.4.2名字隐藏
    - 13.4.3非自动继承的函数
  - 13.5组合与继承的选择
    - 13.5.1子类型设置
    - 13.5.2专门化
    - 13.5.3私有继承
  - 13.6保护
  - 13.7多重继承
  - 13.8渐增式开发
  - 13.9向上映射

## &lt;&lt;C++编程思想&gt;&gt;

- 13.9.1为什么“向上映射”
- 13.9.2组合与继承
- 13.9.3指针和引用的向上映射
- 13.9.4危机
- 13.10小结
- 13.11练习
- 第14章 多态和虚函数
- 14.1向上映射
- 14.2问题
- 14.3虚函数
- 14.4C++如何实现晚捆绑
- 14.4.1存放类型信息
- 14.4.2对虚函数作图
- 14.4.3撩开面纱
- 14.4.4安装vpointer
- 14.4.5对象是不同的
- 14.5为什么需要虚函数
- 14.6抽象基类和纯虚函数
- 14.7继承和VTABLE
- 14.8虚函数和构造函数
- 14.8.1构造函数调用次序
- 14.8.2虚函数在构造函数中的行为
- 14.9析构函数和虚拟析构函数
- 14.10小结
- 14.11练习
- 第15章 模板和包容器类
- 15.1包容器和循环子
- 15.2模板综述
- 15.2.1C方法
- 15.2.2Smalltalk方法
- 15.2.3模板方法
- 15.3模板的语法
- 15.3.1非内联函数定义
- 15.3.2栈模板 ( thestackasatemplate )
- 15.3.3模板中的常量
- 15.4stash&stack模板
- 15.4.1所有权问题
- 15.4.2stash模板
- 15.4.3stack模板
- 15.5字符串和整型
- 15.5.1栈上的字符串
- 15.5.2整型
- 15.6向量
- 15.6.1“无穷”向量
- 15.6.2集合
- 15.6.3关联数组
- 15.7模板和继承

## &lt;&lt;C++编程思想&gt;&gt;

- 15.7.1设计和效率
- 15.7.2防止模板膨胀
- 15.8多态性和包容器
- 15.9包容器类型
- 15.10函数模板
  - 15.10.1存储分配系统
  - 15.10.2为tstack提供函数
  - 15.10.3成员函数模板
- 15.11控制实例
- 15.12小结
- 15.13练习
- 第16章 多重继承
  - 16.1概述
  - 16.2子对象重叠
  - 16.3向上映射的二义性
  - 16.4虚基类
    - 16.4.1 “最晚辈派生”类和虚基初始化
    - 16.4.2使用缺省构造函数向虚基“警告”
  - 16.5开销
  - 16.6向上映射
  - 16.7避免MI
  - 16.8修复接口
  - 16.9小结
  - 16.10练习
- 第17章 异常处理
  - 17.1C语言的出错处理
  - 17.2抛出异常
  - 17.3异常捕获
    - 17.3.1try块
    - 17.3.2异常处理器
    - 17.3.3异常规格说明
    - 17.3.4更好的异常规格说明
    - 17.3.5捕获所有异常
    - 17.3.6异常的重新抛出
    - 17.3.7未被捕获的异常
  - 17.4清除
  - 17.5构造函数
  - 17.6异常匹配
  - 17.7标准异常
  - 17.8含有异常的程序设计
    - 17.8.1何时避免异常
    - 17.8.2异常的典型使用
  - 17.9开销
  - 17.10小结
  - 17.11练习
- 第18章 运行时类型识别
  - 18.1例子     shape



<<C++编程思想>>

- 18.2什么是RTTI
- 18.3语法细节
  - 18.3.1对于内部类型的typeid ( )
  - 18.3.2产生合适的类型名字
  - 18.3.3非多态类型
  - 18.3.4映射到中间级
  - 18.3.5void指针
  - 18.3.6用模板来使用RTTI
- 18.4引用
- 18.5多重继承
- 18.6合理使用RTTI
- 18.7RTTI的机制及花费
- 18.8创建我们自己的RTTI
- 18.9新的映射语法
  - 18.9.1staticcast
  - 18.9.2constcast
  - 18.9.3reinterpret\_\_cast
- 18.10小结
- 18.11练习
- 附录A 其他性能
- 附录B 编程准则
- 附录C 模拟虚构造函数

#### 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>