第一图书网, tushu007.com

<<应用软件设计教程>>

图书基本信息

书名：<<应用软件设计教程>>

13位ISBN编号：9787504357915

10位ISBN编号：750435791X

出版时间：2009-6

出版时间：中国广播电视出版社

作者：徐品，李绍彬，蓝善祯 编著

页数：261

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

内容概要

“软件设计”是学习如何设计一个软件，是“软件工程’’中的重要一环。
但是要说清楚如何设计一个软件却不是一件很容易的事。
在应用软件方面软件的类型实在太多了．按开发规模分类，有个人、3人左右的小团队、5-10人的中等团队、企业式团队（几十人）等；按网络方式分类，有单机、C／S（客户机／服务器）、B／S（浏览器／服务器）等类型；按项目来源分类，则有自创自用型、科研型、内部使用型、商业型等；按适用范围可分为独家订做、通用型、行业型，等等。
由于各种类型软件的要求不同，实现的目标不同，对软件开发的质量不同，当然，编写方式也是不同的。
软件设计课程通常会告诉你编制软件所要遵循的原则，软件开发需要经历哪些工程。
但不能期望学完了软件设计，什么软件都能设计了。
应该说，通常软件设计这个课程属于方法论范畴，而不是如同电路设计、程序语言设计那样让你学会一项技能。
　　很多读者期望通过软件设计课程的学习，就能学会如何去设计一个软件，但事实上并不那么理想。
现在，很多软件设计的论著都致力于对软件开发过程的总结，力求在方法论上找出软件设计的规律，用于指导学习者进行软件设计。
这方面最重要的成果应该是UML（统一建模语言）。
这些抽象的法则和设计方法无疑是非常重要的，在本书中也有专门的介绍。
但如果对一个没有太多的软件开发经验的大学生或研究生，尤其是非计算机专业的学生来说，这些理论实在是太抽象了，以至于很多学完软件设计的学生感觉没有实质性的收获。
这些书本上的软件设计的知识也许要等到从事软件开发两三年以后才有体会，如果那时还记得这些知识的话。
　　但是，要求学习者一定要等到积累了一定经验以后再来学习软件设计的方法是不现实的。
根据作者多年的软件开发经验，并通过几年的教学实践，我们认为，对于没有足够编程经验的读者来说，软件设计课程也是可以学习的；但在学习抽象的软件设计方法之前还是要学一些基本的软件设计技巧，让学习者积累一定的经验，然后再去理解抽象的方法论。
　　本书的主要对象是学过C++语言但没有太多开发经验的学生，特别是非计算机专业的学生。
可作为大学四年级或新入学的研究生学习软件设计的教材。
　　本书分四个部分，共11章。
本书采用以实例为主，力求将抽象的设计方法融入到具体程序实现中，让读者从实例中学习软件设计的方法。
本书所采用的程序语言是C++，开发工具是VC++，软件设计实例是单机运行的绘图软件。
　　第一部分是“程序设计基础知识”，是由第1、2两章组成。
主要是C++语言和数据结构的知识提要，供那些C++基础不太好，或学得不够深入的学生复习和深入学习用。
其中着重介绍运算符重载、多态性、模板、线性表等概念。
如果对这方面很熟的学生，则可以跳过或粗略浏览一遍即可。
　　第二部分是“MFC编程技术”，是由第3、4、5三章组成。
主要介绍MFC程序调试方法、基本原理及基本的开发技术。
使读者对MFC程序有一个比较深入的了解。
MFC程序结构提供了一个很好的软件设计范本，我们在了解它的编程技术的同时，也会对MFC程序结构有一个比较深刻的印象，这对软件设计的学习是非常有好处的。
　　第三部分是“软件开发实例”，是由第6、7、8三章组成。
这部分将引导读者开发一个绘图软件系统。
该软件虽然很小，但其中包含的数据结构、数据的管理、程序流程等对软件设计的学习者都非常具有

借鉴作用。
　　第四部分是“软件工程与软件设计”，是由第9、10、11三章组成。
这部分将结合前面程序的例子，介绍软件工程的概念和软件设计的方法。
后面还着重介绍了UML统一建模语言。
最后还结合开发实例给出了UML的设计方法。
从一个具体的开发实例中获得软件设计的思路，从而能够理解软件设计的理论。
这就是本书所要追求的目标。

书籍目录

第一部分　程序设计基础知识　第1章　C++语言提要　1.1 概述　1.2 类与对象　1.2.1 类的构造函数　1.2.2 拷贝构造函数被调用的场合　1.2.3 带有指针变量的类　1.2.4 关于类的继承问题　1.3 变量与函数　1.3.1 指针与引用　1.3.2 静态变量　1.3.3 函数参数　1.3.4 const的用法　1.4　运算符重载　1.4.1 将运算符理解为函数　1.4.2 不同类之间的运算　1.4.3 用友元定义的运算符　1.4.4 其他运算符的定义　1.5 多态性　1.5.1 一个多态性的例子　1.5.2 动态绑定原理　1.5.3 关于虚函数的进一步探讨　1.6 模板　1.6.1 用模板定义的函数　1.6.2 用模板定义的类　1.6.3 非类型模板参数　思考题　习题　第2章　数据结构提要　2.1 概述　2.2 顺序存储方式：数组　2.2.1 二维数组的物理结构　2.2.2 关于数组的类　2.3 链式存储方式：链表　2.3.1 链表的物理结构　2.3.2 尾部添加新结点　2.3.3 删除结点　2.3.4 插入新结点　2.4 线性表特例：栈　2.4.1 栈的基本概念　2.4.2 用链表方式实现栈的操作　2.5 线性表特例：循环队列　2.5.1 队列的基本概念　2.5.2 循环队列的实现　思考题　习题第二部分　MFC编程技术　第3章　VC++简介　3.1 概述　3.2　MFC基本知识　3.2.1 MFC的数据类型的表示　3.2.2 匈牙利变量命名法　3.2.3 几种常用的工具类　3.3 MFC应用程序框架　3.3.1 与应用程序有关的层次结构　3.3.2 层次结构中的主要类介绍　3.3.3 简单应用程序举例　3.4 程序调试的方法　3.4.1 代码跟踪与断言　3.4.2 Assert Valid与Dump　思考题　习题　第4章　MFC程序的工作原理　4.1 消息处理机制　1.1.1 MFC程序入口　4.1.2 消息与消息循环　4.1.3 消息的传递　4.1.4 消息映射　4.2　运行期识别　4.2.1 运行期——RUNTIME_CLASS　4.2.2 动态创建——DYNCREATE　4.2.3 类型识别——Is Kind of　4.3 串行化　4.3.1 文件的数据读取方式　4.3.2 C Archive的数据读取与写入方式　4.3.3 串行化函数　思考题　习题　第5章　MFC程序的开发技术　5.1 动态链接库与工程管理　5.1.1 动态链接库的基本概念　5.1.2 一个简单的DLL　5.1.3　DLL的调用方式　5.1.4 建立有动态库的工程　5.2 消息发送与接收　5.2.1 消息的发送与接收　5.2.2 自定义消息块　5.2.3 与其他应用程序通信　5.3 串行化文件的保存与读取　5.3.1 让类支持串行化　5.3.2　DOC中的串行化　5.3.3 串行化到文件　5.4 注册表　5.4.1 注册表的基本操作　5.4.2 设置文件关联　5.5 异常处理　5.5.1 异常处理　5.5.2 自己设计的异常处理　思考题　习题第三部分　软件开发实例　第6章　图像法绘图　6.1 CDC类与绘图　6.1.1 绘图类　6.1.2 绘图设备类　6.1.3 用CDC绘制简单图形　6.2 简单的图像法绘图　6.2.1 加入一个绘图菜单　6.2.2 简单的图像法绘图实现　6.2.3 彩色绘图　6.3 图形绘制的橡皮条算法　6.3.1 “ 橡皮条 ”的基本原理　6.3.2 如何在程序中实现橡皮条　6.4 OnDraw（）与图像保持　6.4.1 视图类的OnDraw函数　6.4.2 图像保持　6.5 图像法的撤销与重复（UNDO／REDO）　6.6　类似MSPAINT的界面设计　6.6.1 状态栏　6.6.2 绘图工具栏　6.6.3 颜色工具栏　思考题　习题　第7章　简单的矢量法绘图软件设计　7.1 图形元素类　7.1.1 图形元素基类　7.1.2 直线类　7.1.3 矩形类　7.1.4 椭圆类　7.2 实现矢量绘图　7.2.1 LBUTFONDOWN的消息响应函数　7.2.2 MOUSEMOVE的消息响应函数　7.2.3 LBUTTONUP的消息响应函数　7.3 矢量绘图系统的管理　7.3.1 利用MFC链表管理图形元素对象　7.3.2 系统的管理类　7.4 矢量法的撤销与重复（UNDO／REDO）　7.4.1 设计Redo链表　7.4.2 菜单中响应撤销与重复的消息　7.5 动态库与程序的模块化　7.5.1 建立动态库工程Shape Dll　7.5.2 在My Draw中使用动态库　思考题　习题　第8章　较完善的矢量法绘图软件设计　8.1 串行化与文件读写　8.1.1 如何使类可串行化　8.1.2 在DOC中实现串行化　8.2 图元拾取技术　8.2.1 选中图元　8.2.2 移动图元　8.2.3 修改界面程序，实现选中和移动效果　8.2.4 拉伸图元　8.3 图元编组技术　8.3.1 子图类的组织　8.3.2 图形元素管理类编组功能　8.3.3 弹出式菜单　8.4 操作链表　8.4.1 操作基类的组织　8.4.2 移动、拉伸等操作类的组织　8.4.3 修改图元管理类的Undo／Redo　8.5 总结　思考题　习题第四部分　软件工程与软件设计　第9章　软件工程的基本知识　9.1 概述　9.1.1 软件工程的诞生　9.1.2 软件的开发过程与建模　9.2 可行性研究与需求分析　9.2.1 可行性研究　9.2.2 需求分析　9.3 概要设计　9.3.1 概要设计概述　9.3.2 概要设计方法　9.3.3 概要设计过程　9.4 详细设计与编码　9.4.1 详细设计　9.4.2 编码实现　9.5 软件测试　9.5.1

软件测试原则　9.5.2 软件测试的基本方法　9.5.3 软件测试的复杂性与经济性　9.6 模块的耦合与内聚　9.6.1 耦合　9.6.2 内聚　9.6.3 划分模块的准则　9.7 程序的正确性与健壮性　9.7.1 软件的正确性　9.7.2 软件的健壮性　思考题　习题　第10章　统一建模语言UML　10.1 概述　10.1.1 UML的主要特点　10.1.2 UML在现代软件工程中的重要作用　10.2 UML的主要内容　10.2.1 用例图　10.2.2 类图　10.2.3 对象图　10.2.4 状态图　10.2.5 顺序图　10.2.6 协作图　10.2.7 活动图　10.2.8 构件图　10.2.9 部署图　思考题　习题　第11章　画笔程序设计　11.1 开发背景　11.2 理解需求　11.2.1 绘制功能　11.2.2 图形编辑功能　11.2.3 保存功能　11.3 分析与设计　11.3.1 用例分析　11.3.2 领域分析　11.3.3 业务过程分析（活动图）　11.3.4 交互分析　11.3.5 概要设计　11.3.6 详细设计　11.3.7 用户界面设计　11.4 编码与实现　11.5　测试与部署　11.6 小结参考文献

章节摘录

第1章C++语言提要　1.1概述　让我们简单回顾一下计算机语言的发展历程。
第一个阶段是汇编语言，这个面向机器的语言。
汇编语言解决了用便于理解的缩写字母来代替二进制的机器码的问题。
汇编的每一条语句对应机器的一条指令，便于执行但不便于编写和阅读。
第二阶段是面向过程的语言。
这时候所解决的问题是将指令过程函数化，以及运算过程高级化（采用人类容易理解的四则运算等运算方式，并引进了与具体机器无关的+、-、*、／等运算符）。
这种改进使得程序过程清晰明了，便于阅读和编写。
但当程序过于庞大时，代码的重复率提高，数据和函数得不到有效的保护，代码再利用不方便等缺点也暴露出来，于是出现了第三阶段的面向对象的语言。
面向对象的语言将面向过程语言中杂乱的数据和函数，整齐地包装成一个个自我完备的对象，提高了代码的可读性、程序的健壮性以及代码的可重用性。
时至今日，尽管计算机语言还在不断地发展，面向对象的程序设计方法一直是现代计算机语言的一个基本理念之一。
从20世纪90年代发展起来的面向组件的技术就是基于面向对象技术发展出来的软件设计的新方法。
面向对象语言的最主要的特点是封装、继承和多态性，称为面向对象思想的三基石。
作为面向对象语言的代表，C++语言一方面以完全兼容的方式牵手C语言，另一方面，又全方位地实现了面向对象的思想。
当应用软件发展到一定规模，如果没有面向对象的语言是很难构建程序的。
当回顾C++的全貌，我们应该认识到，从程序设计的角度来看，其实C++与C语言是完全不同的两种语言。
C++更注重设计，而不是过程。
因此，在程序设计思路上，C++语言与C语言有着很大的不同。
比如编写一个图书管理程序，用C语言的思路，针对每一个功能（图书录入、检索、借阅等）编制函数。
如果系统庞大、功能复杂，会导致程序可读性差、代码重复性高、不易修改等问题。
C++按照对象的方式进行管理，整体上可读性好，又具有代码重用、升级方便等特点，可使软件设计人员在更高的层次上管理程序。
本章不打算全面介绍C++的知识，只是用一些实例来复习一下C++的主要概念，并对C++某些难点进行梳理。
对于从未学过C++的读者，我们建议先学一遍C++再看本章。
……

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