

## <<应用程序设计编制>>

### 图书基本信息

书名：<<应用程序设计编制>>

13位ISBN编号：9787801722720

10位ISBN编号：7801722728

出版时间：2004-1

出版时间：兵器工业出版社

作者：全国计算机信息高新技术考试教材编写委员会 编

页数：279

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>

## <<应用程序设计编制>>

### 内容概要

《应用程序设计编制（Visual C++平台）：Visual C++ .NET职业技能培训教程（高级程序员级）》是劳动和社会保障部全国计算机信息高新技术考试中的应用程序设计编制（Visual C++平台）Visual C++ .NET高级程序员级指定教材，由全国计算机信息高新技术考试教材编写委员会组织编写，国家职业技能鉴定专家委员会计算机专业委员会技术部的有关专家、命题组工作人员和一些考评员参加了本书的编写工作。

《应用程序设计编制（Visual C++平台）：Visual C++ .NET职业技能培训教程（高级程序员级）》根据考核的规范标准编写，作为Visual C++ .NET高级教程，主要是针对有一定Visual C++程序设计基础的读者。

本书首先介绍了Visual C++ .NET相对于Visual C++以前版本所具有的一些新特点以及有关安装Visual Studio .NET的基本步骤。

在接下来的章节中，编者依次向读者讲叙了C++重点知识、MFC编程基础、基本事件处理、基于对话框的程序设计、文档视图结构以及网络和数据库编程等知识。

比较同类书籍而言，本书着重介绍读者在学习以及程序设计过程中所需要掌握的一些关键知识以及技巧，这样将有利于提高读者学习的效率，让读者在尽量短的时间内掌握Visual C++ .NET的主要知识。

## 书籍目录

第1章 Visual C++.NET介绍 1.1 Visual C++.NET的新特性 1.1.1集成开发环境 1.1.2添加新类 1.1.3GDI+及相关API 1.1.4Web服务及应用 1.1.5托管代码 1.2 Visual Studio.NET安装 1.2.1系统安装要求 1.2.2系统安装步骤 1.2.3修复产品安装 1.3系统组成 1.3.1NET框架 1.3.2XML Web services 1.3.3语言增强 1.3.4集成开发环境 1.3.5MSDN订阅 1.4一个简单的.NET程序实例 1.5本章小结 第2章 C++的重要性质 2.1数据抽象与类 2.1.1面向对象的概念 2.1.2数据抽象 2.1.3类 2.1.4应用实例 2.2构造函数与析构函数 2.2.1构造函数的定义 2.2.2构造函数的作用 2.2.3说明构造函数的规则 2.2.4析构函数的定义 2.2.5析构函数的作用 2.2.6说明析构函数的规则 2.2.7构造函数和析构函数的调用顺序 2.3重载 2.3.1重载的定义 2.3.2重载的作用 2.3.3类中函数名重载 2.3.4运算符重载 2.4继承 2.4.1继承的概念以及重要性 2.4.2继承的语法以及一个实例 2.5虚函数与多态性 2.5.1虚函数的定义 2.5.2虚函数的作用 2.5.3虚函数的声明 2.5.4类的多态性 2.6友元函数 2.6.1友元函数的声明方式 2.6.2友元函数的作用 2.6.3友元函数应用实例 2.7本章小结 第3章 MFC编程基础 3.1 MFC应用程序框架 3.1.1什么是MFC 3.1.2什么是应用程序框架 3.1.3使用应用程序框架的优点 3.2 MFC的编程领域 3.3使用向导生成应用程序的顺序 3.4一个经典的MFC应用程序 3.5本章小结 第4章 基本事件处理 4.1消息处理中的主要概念 4.1.1 Windows消息 4.1.2消息处理函数 4.1.3映射消息 4.2消息类型 4.2.1分类 4.2.2消息处理方式 4.2.3消息循环 4.3框架调用消息处理程序的方式 4.3.1消息的发送和接收 4.3.2非命令消息到达其处理程序的方式 4.3.3命令传送 4.3.4 OnCmdMsg处理程序 4.3.5重写标准命令传送 4.4框架搜索消息映射的方式 4.4.1查找消息映射 4.4.2派生的消息映射 4.4.3消息映射范围的处理程序 4.5声明消息处理函数 4.5.1标准Windows消息的处理程序 4.5.2命令和控件通知的处理程序 4.5.3处理反射消息 4.6添加MFC消息处理程序 4.6.1将消息映射到函数 4.6.2添加用于任何对话框控件的事件处理程序 4.7在状态栏中显示命令信息的方法 4.8动态更新用户界面对象 4.8.1调用更新处理程序的时机 4.8.2 ON\_UPDATE\_COMMAND\_UI宏 4.8.3 CCmdUI类 4.9本章小结 第5章 基于对话框的应用程序设计 5.1对话框和控件的基本概念 5.1.1对话框的基本概念 5.1.2控件的基本概念 5.2如何使用对话框编辑器创建和编辑对话框 5.2.1创建新对话框 5.2.2创建用户无法在运行时退出的对话框 5.2.3显示或隐藏对话框编辑器工具栏 5.2.4在对话框控件和代码之间切换 5.2.5编辑对话框中的控件 5.2.6自定义控件 5.2.7定义助记键（访问键） 5.2.8指定对话框的位置和大小 5.3对话框的生命周期 5.3.1创建并显示对话框 5.3.2初始化对话框 5.3.3在对话框中处理Windows消息 5.3.4对话框数据交换 5.3.5对话框数据验证 5.3.6关闭对话框 5.3.7销毁对话框 5.4模式对话框设计实例 5.5无模式对话框设计实例 5.6本章小结 第6章 文档和视图 6.1文档和视图结构 6.1.1文档 / 视图的核心是四个关键类 6.1.2文档 / 视图结构的描述 6.1.3文档 / 视图结构的优点 6.1.4“ MFC应用程序向导 ” 创建的文档和视图类 6.1.5文档 / 视图结构的替换选项 6.1.6文档、视图、框架间的关系和消息传送 6.1.7文档、视图、框架之间相互作用 6.2使用文档 6.2.1从CDocument派生文档类 6.2.2用文档数据变量管理数据 6.2.3将数据序列化到文件和从文件序列化数据 6.2.4不使用序列化机制 6.2.5在文档中处理命令 6.3使用视图 6.3.1 MFC中可用的派生视图类 6.3.2在视图中绘图 6.3.3通过视图解释用户输入 6.3.4视图在打印中的作用 6.3.5滚动和缩放视图 6.4单文档界面 6.4.1序列化 6.4.2SDI简介 6.4.3将多个视图添加到单个文档 6.4.4 SDI程序实例 6.5多文档界面 6.5.1多文档类型、视图和框架窗口 6.5.2 MDI程序实例 6.6切分窗口与多视图 6.6.1切分窗口 6.6.2多视选择 6.6.3 CSNitterWnd类介绍 6.6.4演示切分窗口 6.7本章小结 第7章 动态链接库 7.1DLL基础知识 7.1.1DLL概述 7.1.2 DLL的优点 7.1.3 DLL类型 7.2 DLL的导入和导出 7.2.1DLL的导入 7.2.2 DLL的导出 7.2.3 DLL的相互导入 7.3 DLL的链接方法 7.3.1隐式链接 7.3.2显式链接 7.3.3 DLL链接中的几个函数 7.4 DLL的初始化 7.4.1初始化规则DLL 7.4.2初始化扩展DLL 7.4.3初始化非MFC DLL 7.5 DLL的应用 7.5.1DLL的使用 7.5.2 DLL的调试 7.5.3 DLL中使用多线程 1.6动态链接库实例 7.6.1 MFC动态链接库的创建 7.6.2创建一个MFC应用程序 7.7本章小结 第8章 多线程程序设计 8.1进程与线程概述 8.1.1进程与线程的关系 8.1.2多线程编程技术 8.2创建用户界面线程 8.2.1创建用户界面线程 8.2.2启动用户界面线程 8.2.3用户界面线程编程实例分析 8.3创建辅助线程 8.3.1控制函数的实现 8.3.2启动辅助线程 8.4线程的同步 8.4.1线程同步类 8.4.2线程同步实例分析 8.5线程的终止 8.6本章小结 第9章 数据库程序设计 9.1数据库管理的优点 9.2结构化查询语言SQL 9.3 ODBC类数据库编程 9.3.1 ODBC基本概念 9.3.2 ODBC数据库程序设计的实现 9.3.3 MFC ODBC类介绍 9.3.4 MFC ODBC数据库访问 9.3.5 MFC ODBC编程实例 9.4 MFC DAO数据库访问 9.4.1 DAO和ODBC的选择使用 9.4.2使用DAO

## &lt;&lt;应用程序设计编制&gt;&gt;

和ODBC可以访问的数据源 9.4.3 MFC对DAO的封装 9.4.4 MFC的DAO类介绍 9.4.5 DAO类与ODBC类相似之处 9.4.6 DAO的特色 9.4.7 MFC DAO应用程序的编写步骤 9.4.8 DAO实例 9.5 MFC ADO数据库访问 9.5.1一致数据访问技术介绍 9.5.2 ADO对象模型 9.5.3 MFC ADO编程 9.5.4实现对ADO操作 9.5.5 ADO数据绑定 9.6本章小结 第10章 网络编程 10.1TCP / IP网络协议 10.1.1TCP / IP结构 10.1.2常用协议 10.2 Winsock套接字介绍 10.2.1套接字 (Sockets) 10.2.2服务器 / 客户端模型 10.2.3端口和套接字地址 10.2.4 Winsock编程原理 10.2.5 Winsock程序简单示例 10.3 CAsyncSocket类 10.3.1 CAsyncSocket类成员 10.3.2 CAsyncSocket类使用 10.4 CSocket类 10.4.1 CSocket类成员 10.4.2 CSocket对象与CArchive对象 10.4.3 CSocket编程模型 10.5套接字类派生 10.6套接字编程的阻塞 10.7服务器应用程序Server 10.8客户端应用程序Client 10.9本章小结 第11章 C++托管扩展编程 11.1 C++托管扩展应用简介 11.1.1何时使用C++托管扩展 11.1.2托管C++项目类型 11.2C++托管扩展基本概念 11.2.1托管类型 11.2.2 \_\_gc数组 11.2.3托管应用程序中的指针类型 11.2.4 C++托管扩展中的字符串 11.3托管代码与MFC 11.4使用C++托管扩展创建XML Web services 11.4.1创建XML Web services项目 11.4.2实现XML Web services功能 11.4.3调试XML Web services 11.4.4部署XML Web services 11.5使用C++托管扩展访问XML Web services 11.5.1创建XML Web services客户端项目 11.5.2添加Web引用 11.5.3访问XML Web services项目 11.5.4调试XML Web services客户端 11.6托管代码中的事件处理 11.6.1声明事件 11.6.2定义事件处理程序 11.6.3将事件处理程序与事件挂钩 11.6.4激发事件 11.6.5实例 11.7托管异常处理 11.7.1基本概念 11.7.2 C++托管扩展下的异常处理行为差异 11.8本章小结

## &lt;&lt;应用程序设计编制&gt;&gt;

## 章节摘录

版权页：插图：若要重写基类中定义的处理程序，只需在派生类中定义一个具有相同原型的函数，并创建此处理程序的消息映射项。

此处理程序“重写”类的任何基类中具有相同名称的任何处理程序。

在某些情况下，处理程序应调用基类中重写的处理程序，以便基类和Windows可以操作消息。

在重写中何处调用基类处理程序视不同情况而定。

有时必须首先调用基类处理程序，有时则在最后调用。

有时选择自己处理消息时，则有条件地调用基类处理程序。

有时应调用基类处理程序，然后根据基类处理程序返回的值或状态，有条件地执行自己的处理程序代码。

警告：如果打算将参数传递给基类处理程序，则修改传入处理程序的参数是不安全的。

例如，可能试图修改OnChar处理程序的nChar参数（例如要转换为大写）。

这一行为相当模糊，但如果需要实现此效果，请转而使用CWnd成员函数SendMessage。

如何确定重写给定信息的正确方法？

当“属性”窗口编写某一给定消息的处理函数（如WM\_CREATE的OnCreate处理函数）的主干时，它以所建议的被重写的成员函数格式来构造。

下例建议处理程序首先调用基类处理程序，并且只有当不返回-1时才继续。

按照约定，这些处理程序的名称以前缀“On”开头。

这些处理程序中有些不采用参数，而有些则采用若干参数。

有些还具有void以外的返回类型。

所有WM\_\_消息的默认处理程序都作为CWnd类的成员函数（名称以“On”开头）记录在“MFC参考”中。

CWnd中的成员函数声明以afx\_msg为前缀。

4.5.2命令和控件通知的处理程序 命令或控件通知消息没有默认的处理程序。

因此为这些类别的消息命名处理程序时只受到约定的限制。

当将命令或控件通知映射到处理程序时，“属性”窗口基于命令ID或控件通知代码提供一个建议的名称。

可以接受此建议名称，也可以更改或替换它。

约定建议在这两类中为它们所表示的用户界面对象命名处理程序。

这样，“编辑”菜单中的“剪切”命令的处理程序名可能是：afx\_msg void OnEditCut（）。

<<应用程序设计编制>>

编辑推荐

<<应用程序设计编制>>

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>