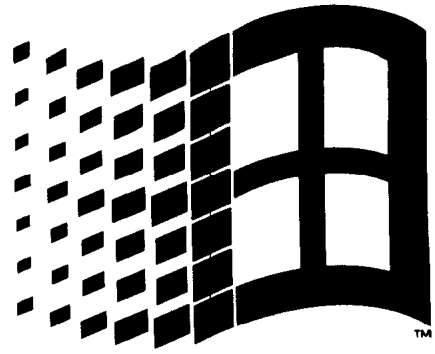

Windows
程序设计基础教程

张李义 尚 涛 王 林 编著

武 汉 大 学 出 版 社

图书在版编目(CIP)数据

Windows 程序设计基础教程/张李义,尚涛,王林编著.—武汉:武汉大学出版社,2001.8
ISBN 7-307-03297-X

Ⅰ.W… Ⅱ.①张… ②尚… ③王… Ⅲ.窗口软件,Windows—程序设计—高等学校—教材 Ⅳ.TP316.7

中国版本图书馆 CIP 数据核字(2001)第 048855 号

责任编辑:郭志安 责任校对:卢 建 版式设计:支 笛

出版:武汉大学出版社 (430072 武昌 珞珈山)
(电子邮件:wdp4@whu.edu cn 网址:www.wdp whu edu cn)
发行:新华书店湖北发行所
印刷:核工业中南三〇九印刷厂
开本:787×1092 1/16 印张:20.125 字数:481 千字
版次:2001 年 8 月第 1 版 2001 年 8 月第 1 次印刷
ISBN 7-307-03297-X/TP·106 定价:28.00 元

前　言

Microsoft 公司的 Windows 98/NT 操作系统是当今世界上最为流行的操作系统之一，这种操作系统的图形功能、友好的用户界面、简单的操作、易学实用的特点使得在 Windows 系列操作系统上使用的应用程序数以千万计，因此编写 Windows 应用程序就成为软件开发人员的不可缺少的技能。

本书适用于已经学习了 Windows 98/NT 操作系统和 C 语言程序设计等基础课程的读者，书中介绍了如何使用 Visual C++ 6.0 所提供的 MFC 开发 Windows 应用程序的技术。其中，第一章介绍 Windows 程序设计的基本知识；第二章介绍菜单的功能及使用方法；第三章介绍 Windows 程序设计的设备环境和画笔、画刷字体、位图及调色板等图形设备接口；第四章介绍鼠标、键盘和定时器等输入设备；第五章、第六章介绍工具栏和状态栏的使用；第七章介绍 Windows 的对话框和控件的开发方法；第八章介绍属性单和属性页的使用；第九章讲述文档和视图的概念及其使用方法；第十章介绍多视图和切分窗口；第十一章讲述 Windows 程序中的异常处理；最后一章介绍打印及其相关的技术。

本书本着易学、实用的原则，在编写过程中，考虑到了多媒体教学的需要，并根据作者多年的教学经验，结合实际科研项目给出了许多具有实际应用价值的例子供参考。可以作为大中专院校的计算机应用相关专业的本科生、研究生的教材，也可以作为电脑培训时 Windows 程序设计和 C++ 课程方面的教材。

本书的主要内容由张李义、尚涛两人完成，书中的应用程序例子部分由王林和硕士研究生赖碧云完成。

目 录

第一章 Windows 程序设计概述

本章主要讲述 Windows 程序设计的特点、编程规则、面向对象程序设计的一些基本概念，并重点介绍 Microsoft Visual C++ 6.0 开发 Windows 应用程序的步骤。

1.1 面向对象程序设计的一些基本概念

面向对象的程序设计(Object-Oriented Programming，简称 OOP)是 20 世纪 90 年代初开始流行的一种程序设计方法，它从根本上改变了以往的程序设计方法，被称为软件设计的一次革命。OOP 以人们描述现实世界的方法来描述软件问题，将一个复杂的问题分解为一个一个能完成独立功能的对象，并把这些对象组合起来来完成一个复杂的功能。要学习面向对象的程序设计，首先必须理解一些基本概念。

(1) 对象 在面向对象的程序设计中，对象是基本的运行实体，它既包括数据(属性)，也包括对数据的操作(行为)，因此对象是把属性和行为封装在一起的一个整体。从程序设计的观点来看，对象是一个程序设计模块。在对象内的操作通常称为方法。

(2) 消息 消息是对象之间进行通信的一种构造。当一个消息发送给某个对象时，要求包括接受的对象去执行某些活动的消息。接受消息的对象对消息进行解释，然后予以响应。这种通信机制称为消息传递。但发送消息的对象不需要知道接受消息的对象如何对请求予以响应。

(3) 类 通常把一组大体上相似的对象称为一个类。一个类所包含的方法和数据描述了一组对象的共同行为和属性。将一组对象的共同特性加以抽象并存储在一个类中的能力，是面向对象技术的一个非常重要的特点。一个优秀的面向对象程序设计语言包括了一个非常丰富的类库。类是在对象之上的抽象，有了类以后，对象则是类的具体化，是类的实例，同时，类可以有子类，也可以有父类，从而形成类的层次结构。

(4) 继承性 继承性是父类和子类之间共享数据和方法的机制。这是类之间的一种关系，在定义和实现一个类的时候，可以在一个已存在的类的基础上来进行，把这个已存在的类所定义的内容作为自己的内容，并根据需要加入新的内容。继承性是面向对象程序设计语言不同于其他语言的最主要的特点，是其他语言所没有的。

(5) 多态性 在收到消息时，对象要予以响应，不同的对象收到同一消息可产生完全不同的结果，这种现象称为多态。在使用多态的时候，用户可以发送一个通用的消息，而实现的细节则由接收对象自行决定，这样，同一消息就可以调用不同的方法。多态的实现受到继承性的支持，通过用类的继承的层次关系，把具有通用功能的消息存放在高层次，而实现这一功能的不同行为放在低层次，在这些低层次上生成的对象能够给通用消息以不同的响应。

多态有几种不同的形式，Cardelli 和 Wegner 将多态分为四种：参数多态、包含多态、过载

多态和强制多态。前两种称为通用多态，后两种称为特定多态。

1.2 Windows 程序设计的特点

Windows 程序设计的编程思想同原来的基于 DOS 的程序设计思想有着本质的不同，下面介绍一些 Windows 程序设计的特点。

1. 事件驱动

事件驱动的程序设计不是以一个设计好的过程为中心，而是以消息为中心。Windows 消息提供应用程序与应用程序之间和应用程序与 Windows 操作系统之间进行通信的手段。几乎每个 Windows 程序所做的事情都是为了响应发送到窗口函数的某条消息。在大多数应用程序中，很大一部分代码是用来处理这些消息函数的。

Windows 的消息由三部分组成：消息标识号、字参数和长参数。消息标识号由事先定义的消息名来标识，字参数和长参数包含与消息标识号相关的值。例如，当光标在窗口用户区时，按下鼠标器的左按钮，Windows 就向该窗口发送一条 WM _ LBUTTONDOWN 消息，其长参数含有光标的 X 坐标和 Y 坐标(分别在其低字节和高字节中)，而其字参数则包含指示各种虚拟键是否被按下的值。窗口收到此消息后，就可以作出相应的处理。Windows 应用程序创建的每一个窗口都有一个窗口函数，用以处理发送到该窗口的消息。窗口函数由 Windows 采用消息驱动的形式直接调用，而不是由应用程序显式调用的。窗口函数处理完消息后，又将控制返回给 Windows。

Windows 中有一个系统消息队列(简称系统队列)。当开始执行一个 Windows 应用程序时，Windows 为该应用程序建立一个“消息队列”，称之为应用程序队列。这个队列存放着程序可能创建的各种窗口的所有消息。程序中含有一段“消息循环”代码，用来从消息队列中检索这些消息，并把它们分发到相应的窗口函数中，有些消息不必放到消息队列中，而是直接送给窗口函数。

2. 用户界面统一

用户界面是衡量一个应用程序质量的一个重要的技术指标。Windows 为设计用户界面提供了一种全新的方法。Windows 是一个图形用户界面(GUI)，各种操作对象都是以图形方式显示在屏幕上，通过鼠标器和键盘用户可以在屏幕上直接操纵这些对象。

Windows 程序的外观相差不大，每个程序占据一个窗口。窗口是屏幕上的一个矩形区域，它是由标题条来标识的。程序中绝大多数功能可以通过菜单来执行，某些菜单会触发对话框，通过对话框，用户可以输入较多的附加信息。图 1-1 是一个较典型的 Windows 应用程序界面。

可以利用 Windows 提供的 Help 工具，为应用程序创建联机求助系统，用户可以很方便地搜索、查询有关信息。

此外，Windows 还提供了一种多文档界面技术。利用这种技术，在一个应用程序中可以创建多个窗口，分别用来显示和处理不同的文档。

用户界面的一致性使得用户可以设计出较为满意的应用程序。

3. 多任务

Windows 98/NT 操作系统是一个多任务系统，可以同时显示和运行多个应用程序，

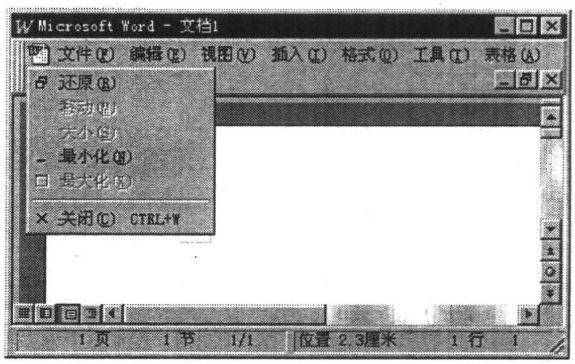

图 1-1　典型 Windows 应用程序界面

每个应用程序有各自的窗口。用户可以在屏幕上移动这些窗口，改变它们的尺寸，在不同的程序间切换，还可以在应用程序之间进行数据交换，并且应用程序必须与所有当前正在运行的其他应用程序共享这些资源。因此在程序设计时要考虑多任务的特点，协调并共享资源，以避免耗尽资源。

Windows 操作系统的多任务是通过在程序之间进行切换来实现的，只有当正在运行的应用程序消息队列中没有消息(或只有 WM _ PAINT 和 WM _ TIMER 消息)时，才能转去执行其他等待消息的程序。因此，Windows 操作系统是一种非抢占式的多任务系统。

4. 设备无关性

Windows 提供了丰富的、与设备无关的图形处理功能。应用程序能很方便地画出各种图形，而不需要直接与具体的输出设备打交道。由于 Windows 提供了设备无关性，因此，应用程序可以使用同一函数在显示屏幕上或在打印机上输出同一图形。

1.3　Windows 程序设计的一些基本约定

1.3.1　Windows 程序的命名规则

Windows 应用程序中变量一般由两部分组成：第一部分为变量前缀，代表变量的类型；第二部分为变量名，代表变量的含义，一般采用大小写混用方法命名变量，每个单词的第一个字母为大写，其余为小写。这种变量的命名既反映了变量的类型又反映了变量的含义。表 1-1 列出了一些常用变量的前缀。

例如：

m _ nWidth：表示一个类的成员变量，变量类型为整型；

lpszMenuName：表示此变量是以 NULL 结尾的字符串长指针；

pDC：表示一个设备环境变量指针；

hWnd：表示一个窗口的句柄。

表 1-1 Windows 应用程序常用变量的前缀

前缀	含义	前缀	含义
a	数组	c	类
ch	字符	p	指针
cb	字节计数	lp	长指针
dw	无符号长型	n	整型
h	句柄	pt	点结构
i	索引	sz	以 NULL 为结束符的字符串
m_	类的成员变量	w	无符号整型

1.3.2 Windows 程序设计的一些编程规则

1. 变量的使用规则

● 为变量赋予表义性强的名称。为了使应用程序的代码更容易被人理解，应对程序中所涉及的变量赋予合乎逻辑的名字。对于一些临时使用的变量，如循环变量，可以使用简单的字母来表示，如 i，j 等，而其他具有一定含义的变量就应该用相应的英文单词组合来表示。例如，用整型数表示的一个矩形的宽度，可以用变量 nWidth 来表示。

● 在变量中混合使用大小写字母。全部使用大写或小写字母的变量名比大小写混合使用的变量名更难阅读。例如：变量 sFIRSTNAME、sfirstname 这两种表示都不合适，合适的应是 sFirstName。

● 尽量避免对变量名进行缩写。即使进行缩写，也要注意整个代码中缩写规则的一致性，例如，在程序中的某些地方使用 Cnt，而另外一些地方使用 Count，则必然增加程序的阅读难度。

● 尽量使用统一的量词。例如在数据库程序中经常使用记录的表示如下所示：

First 第一条记录
Last 最后一条记录
Next 当前记录的下一条记录
Prev 当前记录的上一条记录
Cur 当前记录
Min 最小值
Max 最大值

● 对布尔变量要尽量使用肯定形式。例如：使用 bFound、bDone 而不使用 bNotFound、bNotDone。使用否定形式的布尔变量会增加变量处理出错的机率，特别是在复杂表达式中。

● 尽量缩小变量的作用范围。变量的作用范围越小，出现与该变量相关的错误机会也就越小，对整个应用程序的影响也就越小。

2. 代码的格式化

● 一条太长的语句应分成几行来写。有时程序的一行很长，这时应分成多行，以免阅读代码时需要滚动显示这样的语句，增加程序的可阅读性。

● 缩进后续行。特别是当一条语句太长而需要分成几行时，后续行应缩进，以增加可

阅读性。

● 应用语句缩进来显示代码的组织结构。这在 if、switch、for、while 等结构化程序设计时尤为重要。

● 使用白空间将相关语句组合在一起。所谓白空间是指语句与语句之间留有空白行。一般情况下,在变量定义与其他可执行语句之间、在 if、switch、for、while 等结构与其他语句之间都要有空白行,以增加程序的可阅读性。

3. 代码的注释

注释能使代码更加容易理解,更加容易追踪,一个没有注释的程序是很难读的,对代码的注释主要按照下列规则:

● 用文字对代码进行说明,这种说明要简单易懂。

● 对代码中可能引起误解的地方进行简要的说明。

● 在编写代码前进行说明。一般在编写一个函数之前先写上注释,如果可能,可以编写完整句子的注释或伪代码。一旦用注释对代码进行了概述,就可以在注释之间编写代码,在代码编写时可能需要对注释进行调整。

● 用行尾注释来说明变量。

4. 代码流的控制

● 尽量避免 Goto 语句的使用。

● 正确地使用循环结构。

● 分支结构的判断条件较多时,要采用 switch 结构而不要使用 if 结构。

● 对控制结构中出现的表达式进行格式化,以便进行准确的计算和判断。

5. 用户界面的实现规则

● 为窗口赋予统一的外观和行为特性。窗口是应用程序的基本构件,如果应用程序配有一个用户界面,那么它至少拥有一个窗口,应该为多窗口的边框样式、颜色等进行统一的设计。

● 使窗口中的控件具有标准外观。

● 尽可能使用系统提供的颜色。

● 窗口所挂接的菜单要便于理解和使用。

6. 用户输入和消息

● 精心设计窗口中各个控件的操作顺序。

● 窗口控件要有默认操作。

● 鼠标的外形要与 Windows 系统提供的外形相一致。

● 创建有创意和功能良好的消息框。

1.4 Windows 应用程序的组成

一个 Windows 应用程序通常包括头文件(.H)、CPP 源文件、模块定义文件(.def)和资源文件(.rc)等几个部分。头文件的扩展名为.H,内容为应用程序所需的数据结构和常量的定义、函数原型定义、类的定义等。CPP 源文件的扩展名为.cpp,它是数据结构定义的成员函数和全局函数的具体实现,是应用程序编程的主要部分。

模块定义文件(.def)用于通知连接器如何装配应用程序,一般包括应用程序的名字、

段、内存要求和输出函数等。下面是一个模块定义文件的例子。

```
Example def
NAME            exam
DESCRIPTION       'A Example of Windows
EXETYPE           WINDOWS
STUB              'WINSTUB EXE
CODE            MOVEABLE DISCARDABLE
DATA            MOVEABLE MULTIPLE
HEAPSIZE        4096
STACKSIZE       5120
EXPORTS         MainWindowProc @1
```

资源文件(.rc)用于描述应用程序要用到的资源,这些资源包括:菜单、图标、光标、插入标记(Caret)、消息框、对话框、字体、位图、画笔、画刷等,这些资源都可以通过资源描述语句来预先定义,Windows 应用程序根据资源的定义来调用、显示资源。

定义资源的文件就是资源文件。每个资源也可以有自己的资源文件。各类资源文件的含义如下:

.RC　由一个或多个其他类型的资源组成
.DLG　对话框文本文件,包含一个或多个对话框的描述
.BMP　位图文件,由二进制格式的位图资源组成
.CUR　光标文件,二进制格式,用于描述一个光标
.ICO　图标文件,二进制格式,用于描述一个图标
.FNT　二进制格式字体文件

所有这些文件最后都要加入到扩展名为.rc 的资源文件中。下面是一个资源文件的例子。

```
IDR_MAINFRAME ICON DISCARDABLE "exam.ico"
IDR_MAINFRAME BITMAP MOVEABLE "tool.bmp"
IDR_MAINFRAME MENU PRELOAD DISCARDABLE
IDD_ADD_UPDATE DIALOG DISCARDABLE  0,0,120,80
STYLE DS_MODALFRAME|WS_POPUP|WS_VISIBLE|
    WS_CAPTION|WS_SYSMENU
CAPTION   "This is a example"
FONT  8,"MS Sans Serif"
BEGIN
  EDITTEXT    IDC_EDIT_YEAR,12,22,40,20,ES_AUTOHSCROLL
  EDITTEXT    IDC_EDIT_SALES,70,22,64,20,ES_AUTOHSCROLL
  PUSHBUTTON   "OK",IDOK,11,46,56,20
END
```

在所有代码完成以后,创建一个.MAK 文件用来编译、连接并创建应用程序的可执行文件。一个 MAK 文件的例子如下所示。

```
# Nmake macros for building Windows 32 Bit apps
! include <ntwin32 mak>
all: menu exe
# Update the resource if necessary
menu res: menu rc menu h menu dlg checkon bmp
          checkoff bmp menu ico
  rc -r menu rc
menu.rbj: menu res
cvtres - $ (CPU) menu res o menu rbj
# Update the object file if necessary
menu obj: menu c menu h
  $ (cc) $ (cflags) $ (cvars) $ (cdebug) menu c
# Update the executable file if necessary, and if so,
      add the resource back in
menu exe: menu obj menu.rbj menu def
  $ (link) $ (linkdebug) $ (guiflags) out:menu.exe
  menu obj menu rbj $ (guilibs)
```

1.5 Windows 程序设计工具

1.5.1 Microsoft Visual C++ 6.0 简介

Microsoft Visual C++ 6 0 开发环境 Developer Studio 是在 Windows 98/NT 环境下运行的一套集成开发工具，由文本编辑器、资源编辑器、项目工具、优化编译器、源代码浏览器、集成调试器等组成。

使用 Developer Studio，不仅可以创建由 Visual C++ 6.0 使用的源文件和其他文件，而且可以输出、查看和编写与 ActiveX 部件有关的文档。

1. 使用向导工具开发应用程序

Visual C++ 6.0 的向导用于帮助用户生成各种不同类型的应用程序的基本框架。在生成应用程序的基本框架后，就可以使用 ClassWizard 来创建新类，定义消息处理函数，从对话框、表单视图或记录视图的控件中获取数据并验证数据的合法性，在自动化对象中添加属性、事件和方法。下面简要叙述应用 ClassWizard 创建应用程序的步骤。

为了生成 MFC 应用程序的框架，启动 Visual C++ 6.0，并从 File 菜单中选择 New 菜单项。Visual C++ 6.0 会弹出 New 对话框，切换到 Projects 选项单，如图 1-2 所示。在 Project name 栏中输入应用程序的名字，在 Location 编辑框中选择项目所要放置的位置。然后单击 OK 按钮。

项目的类型有多种，主要的含义如下：

ATL COM AppWizard 创建 ATL(Active Template Library)应用程序。

Cluster Resource Type Wizard Cluster Resource Type Wizard 创建二个项目以便在微软群集服务器上管理和监视群集上的资源。ResourceDLL 项目用于管理和监视指定类型

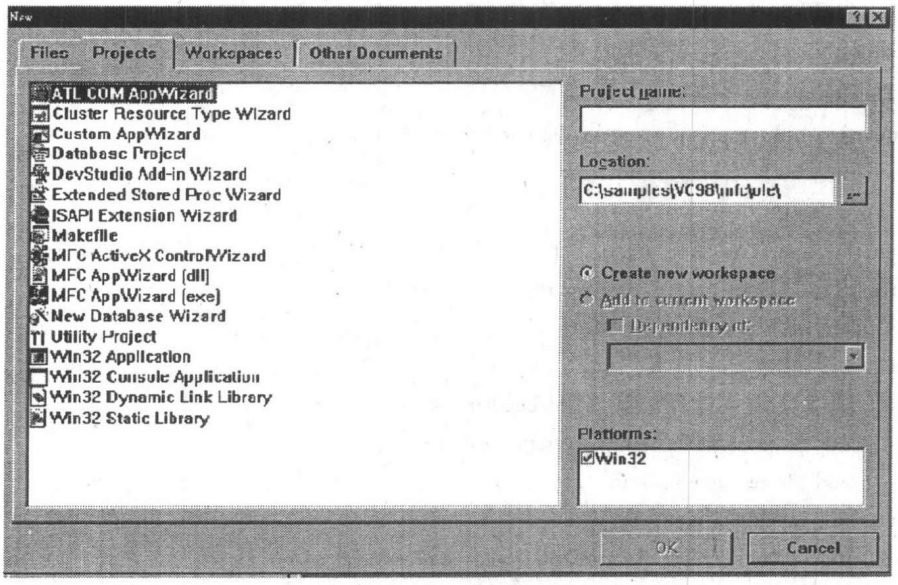

图 1-2 Visual C++ 的 New 对话框

的资源,该项目的名称同给出的 Cluster Resource Type project 名称相同;Cluster Administrator extension DLL 用于配置这种新类型的资源,它由 Cluster Administrator 装载。这种 DLL 项目是一个 COM 和在线处理服务器,项目的名称在给出的 Cluster Resource Type project 名称后面加上"Ex"。

Custom AppWizard 创建自定义的项目类型,并将其添加到创建项目时的可用项目类型列表中。创建新的 AppWizard 类型的启动源文件,允许程序员修改或添加对话框到 AppWizard 中。Custom AppWizard 对于创建可复用的功能组件非常有用。

Database Project 创建数据库应用程序。

DevStudio Add _ in Wizard 创建 Add _ in 项目类型,加载页类似于 Developer Studio 自动化的宏,但是它们是由 C++ 或另一种编程语言编写的。

ISAPI Extension Wizard 创建 Internet 服务器 API 扩展或过滤应用程序。创建完成后,Visual C++ 将产生 Internet 服务器或过滤器 DLL。

Makefile 创建命令行编译所需的文件。一个 MAKEFILE 项目是一个命令行项目,它通知编译程序如何执行命令行。缺省命令是 nmake,但也可以是任何编译工具。

MFC ActiveX ControlWizard 创建 ActiveX (前身为 OLE)控件所需要的模板文件。一旦完成创建过程,Visual C++ 将产生 ActiveX 控件所必须的所有文件,包括:源代码文件、头文件(.H)、资源文件、模块定义文件、工程文件以及对象描述文件等。可以使用 ClassWizard定义控件的事件、属性和方法(有些在 MFC 库中已预先定义)。

MFC AppWizard (dll) 创建 MFC 的 DLL。

MFC AppWizard (exe) 创建 MFC 应用程序。

New Database Wizard 供安装了 Visual InterDev 的用户使用,使创建的应用程序在 Web 页连接 SQL 数据库时得到简化。

Utility Project 当创建 Utility Project 时，并不是在项目中生成任何文件，如 LIB、DLL、EXE，而是把 Utility Project 当作链接时的文件的容器。也可以把它当作所有子项目的主项目。

Win32 Application 创建 32 位应用程序，但是不使用 MFC 和 AppWizard 模板。

Win32 Console Application 创建 32 位控制台应用程序。

Win32 Dynamic Link Library 创建 32 位动态连接库。

Win32 Static Library 创建 32 位静态连接库。

在单击 OK 后，AppWizard 开始与用户交互进行应用程序的有关特性的设定。

2. 应用程序的特性设定

应用程序的特性设定过程包括如下 6 个部分：

(1) 应用程序的类型和语言的选择。

系统提示用户要建立什么类型的应用程序，包括：Single Document(单文档)、Multiple Document(多文档)、Dialog Based(基于对话框的应用)3 种。其中，单文档应用程序中主窗口中只含有一个窗口，而多文档的主窗口中可以含有多个子窗口。基于对话框的应用程序的主窗口只是一个对话框，如 CD Player 应用程序。交互样式如图 1-3 所示。

在这一步完成后单击 Next 按钮进入下一步。

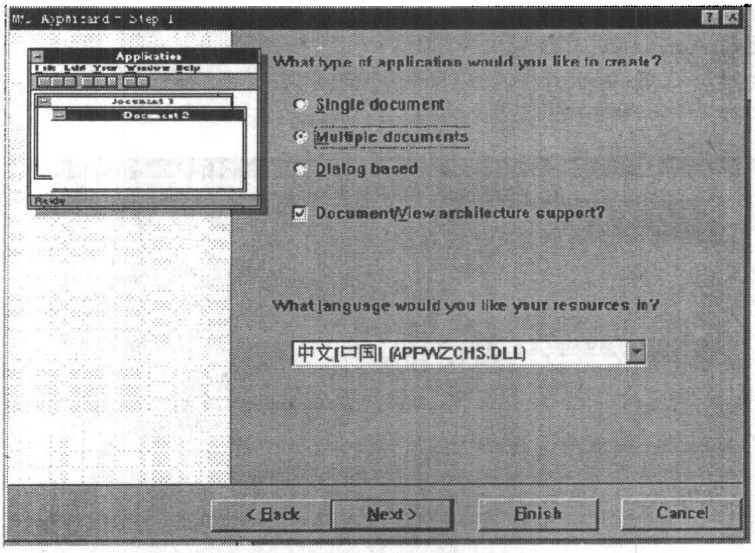

图 1-3 选择应用程序的类型

(2) 是否支持数据库?

这一步决定应用程序是否想利用 ODBC 类去存取一个数据库，图 1-4 显示了这一步的交互。

选择项共有四个。缺省时不包括数据库支持。如果要使用 ODBC 支持，则可以选择 Header File only 或 Database view with file support，如果选择后者，则数据库必须存在，这些操作同控制面板的数据源操作类似。指定一个数据库后，AppWizard 将会生成 CRecordSet 和 CRecordView 的派生类来支持数据库。