

巧学巧用



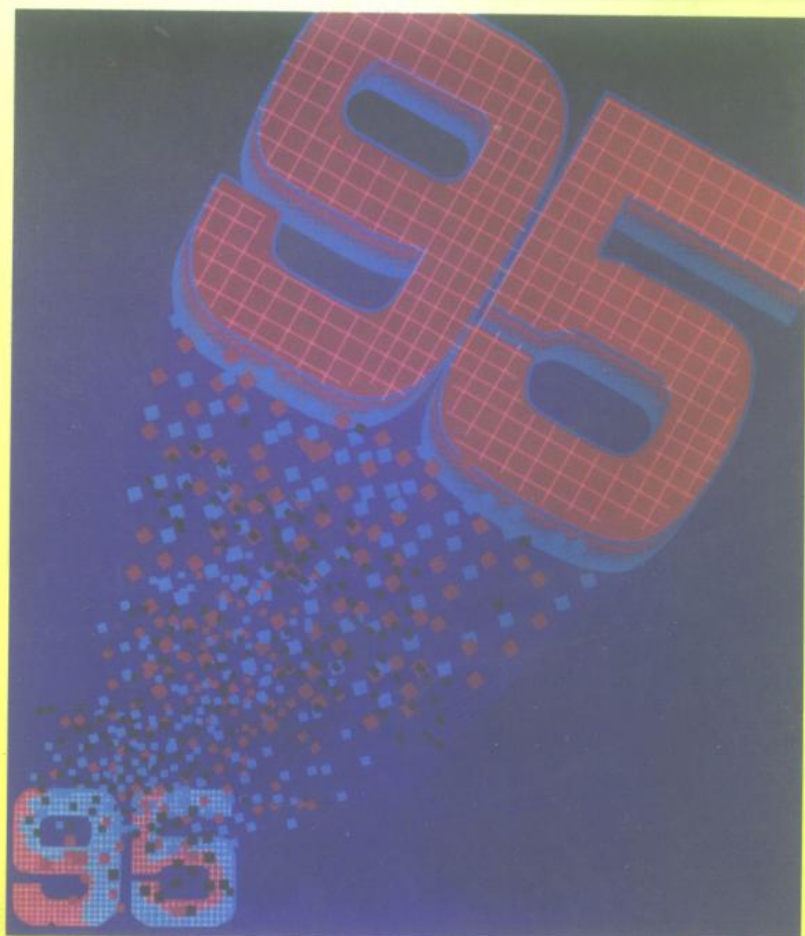
Borland C/C++ OWL 5.0 for Windows 95

黄晓鸣 于伟 武清 编著

Windows

- 学习使用最新C/C++ Windows程序设计开发系统的对象窗口类库
- 详细介绍Borland C/C++ 5.0 for Windows 95对象窗口类库(OWL)之基本内容
- 对280多个类的定义、用法作深入介绍
- 软件设计人员的珍贵参考资料, 面向对象程序设计初学者的入门

引导



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.co.cn>

462605

巧学巧用 Borland C/C++ OWL 5.0 for Windows 95

黄晓鸣 于 伟 武 清 编著



電子工業出版社

Publishing House of Electronics Industry

内 容 简 介

本书是最新 C/C++ Windows 程序设计开发系统 Borland C/C++ 5.0 for Windows 95 对象窗口类库(Object Windows Library, 简称 OWL)的使用参考书。书中详细介绍了对象窗口类库(OWL)的基本内容。包括类定义、类层次、类属关系、成员数据、成员函数的定义及其使用方法等。本书既是不同层次的软件设计人员软件开发时的参考资料,也可以供正在学习面向对象程序设计方法的读者使用。

3324.18

书 名:巧学巧用 Borland C/C++ OWL 5.0 for Windows 95

著 者:黄晓鸣 于伟 武清

责任编辑:张 琼

特约编辑:谭德峰

印 刷 者:北京金特印刷厂

出版发行:电子工业出版社、发行 URL:<http://www.phei.co.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

经 销:各地新华书店经销

开 本:787×1092 1/16 印张:22.75 字数:582.4 千字

版 次:1997 年 9 月第 1 版 1997 年 9 月第 1 次印刷

书 号: ISBN 7-5053-3997-4
TP·1749

定 价:28.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换
版权所有·翻印必究

出版说明

近几年来,随着计算机的发展及其广泛普及,它在人们的工作和生活中发挥着越来越大的作用。作为计算机“灵魂”的软件的发展也是日新月异,一些功能强大、界面友好、易学易用的优秀软件层出不穷,同时一些深受人们欢迎的流行软件的版本也不断更新、提高。

随着这些优秀软件的推出和更新,一些软件开发人员、软件用户及广大的计算机爱好者,都需要了解和学习这其中的新概念、新技术,需要尽快地熟悉并精通这些软件的操作使用和开发的方法与经验。为满足广大读者的迫切需求,我社特邀请了具有丰富经验的计算机专业人员,精心策划、编写了这套“最新流行实用软件巧学巧用”丛书。

这套丛书的主要特点是从最基本的概念入手,以简单易懂的解说,辅以大量的典型应用程序实例及微机运行屏幕显示图例,由浅入深地详细而系统地介绍各种最新流行实用软件的功能、特点、使用和编程方法与技巧。这套书内容翔实,循序渐进,图文并茂,给出了大量的学习经验和使用、编程技巧。因此,读者可以花最少的学习时间,以最佳的捷径,掌握并精通这些流行应用软件;这套书既可作为从零开始的读者的入门向导,又包含了技术内容的深入剖析,把程序设计的专家经验与软件复杂的技术细节紧密地融合在一起,因此,对有经验的程序员来说,也很有帮助。

我们将不断推出多种“最新流行实用软件巧学巧用”选题方面的图书,欢迎广大读者和我们联系,提出宝贵意见和建议。

电子工业出版社

1997年5月

前 言

随着计算机软件技术的不断发展,面向对象程序设计(Object Oriented Programming,简称 OOP)方法正越来越多地被应用于程序设计和软件开发中。如果说八十年代人们对面向对象程序设计技术的概念和方法还只是处在理解阶段的话,那么毫无疑问,从九十年代开始,面向对象程序设计方法就已经开始被成功地应用于软件设计工作中了。可以说,随着 OOP 方法的不断普及和应用,它必将对计算机的应用和发展产生深刻的影响。

C++作为一种有效的 OOP 程序设计语言,已为大多数程序设计人员所使用。而在众多的 C++ 语言产品中,Borland International 公司的系列 C/C++ 产品(TurboC++ 和 Borland C++ 系列)又以其独具的特点和强大的功能成为最受广大程序设计人员喜爱的 OOP 程序开发工具之一。

到目前为止,Borland 公司已经开发了一系列的 C 和 C++ 语言产品。从 1987 年推出的微机上的 C 编译系统 Turbo C 1.0 开始,到 1988 年又推出了 Turbo C 1.5 和 Turbo C 2.0。这是一个非常成功的产品,即使到现在仍然有许多的 C 程序员在使用它来进行通常的 C 程序设计。1989 年,Borland 公司首次推出了支持面向对象程序设计功能的 Turbo C++ 1.0 版,之后又陆续推出了 Borland C++ 2.0 版、3.0 版、3.1 版、4.0 版和 4.5 版。1996 年初又推出了最新版本 Borland C++ 5.0。伴随着每一次版本的升级,该软件的性能都有很大的提高,尤其是在 Microsoft Windows 3.X 和 Windows 95 广为普及的情况下,人们对能在 Windows 环境下运行的软件产品的需求越来越大,因而从事 Windows 程序设计的程序设计人员也越来越多。正是为适应这一潮流,Borland 公司的 C/C++ 产品在保证不断完善和改进其支持面向对象程序设计的性能的同时,更强调了其对 Windows 程序设计的支持。这就为众多从事面向对象程序设计和 Windows 程序设计的人提供了使用方便且功能强大的开发工具。可以说,Borland 公司为当今世界上面向对象程序设计和 Windows 程序设计的普及和推广是功不可没的。

本书主要介绍 Borland C/C++ 5.0 版本中的 OWL 类库中所有类的定义及使用方法。全书共分三章,第一章简单讲述用 OWL 进行 Windows 程序设计的基本过程与方法,并对 OWL 中的少数几个最重要的类做了较详细的说明。第二章是全书的主要部分,按字母顺序介绍了 OWL 类库中所有类的定义以及其成员数据和成员函数的使用方法。第三章中详细地列出了 OWL 中用到的所有常量、枚举、宏、事件处理函数、库函数以及类型定义的格式和使用方法等内容。

在本书的出版过程中,电子工业出版社的张琼小姐为本书的编辑付出了辛勤的劳动并向作者提出了许多很好的建议,作者在此谨向她表示衷心的感谢。

由于编者水平有限,加上时间仓促,书中难免疏漏乃至错误,欢迎读者批评指正。

黄晓鸣

96 年 10 月于青岛

目 录

第一章 OWL 类库与 Windows 程序设计	(1)
§ 1.1 面向对象程序设计(OOP)方法的回顾	(1)
1.1.1 面向对象程序设计与 C++	(1)
1.1.2 封装性(encapsulation)	(2)
1.1.3 继承性(inheritance)	(4)
1.1.4 多态性(polymorphism)	(7)
§ 1.2 OWL 类库与 Windows 程序设计	(8)
1.2.1 什么是 OWL	(8)
1.2.2 用 OWL 编写 Windows 程序的步骤	(12)
§ 1.3 几个重要的 OWL 类	(21)
1.3.1 TApplication 类	(21)
1.3.2 TWindow 类	(26)
1.3.3 TFrameWindow 类	(42)
1.3.4 TControl 类	(48)
1.3.5 TDialog 类	(50)
§ 1.4 本章小结	(56)
第二章 OWL 类定义及其使用方法	(57)
A	(58)
TAnimateCtrl 类	(58)
TAppDictionary 类	(58)
TApplication 类	(60)
TAutoFactory < > 类	(62)
B	(63)
TBandInfo 类	(63)
TBitmap 类	(63)
TBitmapGadget 类	(64)
TBitSet 类	(65)
TBIVbxLibrary 类	(65)
TBrush 类	(66)
TBtnBitmap 类	(66)
TButton 类	(67)
TButtonGadget 类	(68)
TBwccDll 类	(69)

C	(71)
TCelArray 类	(71)
TCharFormat 类	(72)
TCharRange 类	(73)
TCharSet 类	(73)
TCheckBox 类	(73)
TCheckList 类	(74)
TCheckListItem 类	(75)
TChooseColorDialog 类	(75)
TChooseFontDialog 类	(76)
TChooseFontDialog::Tdata 类	(76)
TClientDC 类	(77)
TClipboard 类	(77)
TClipboardViewer 类	(79)
TColor 类	(80)
TColumnHeader 类	(81)
TComboBox 类	(82)
TComboBoxData 类	(83)
TCommandEnabler 类	(84)
TCommCtrlDll 类	(86)
TCommonDialog 类	(87)
TControl 类	(88)
TControlBar 类	(89)
TControlEnabler 类	(90)
TControlGadget 类	(91)
TCreatedDC 类	(91)
TCid3dDll 类	(92)
TCursor 类	(92)
D	(94)
TDatagramSocket 类	(94)
TDataObject 类	(94)
TDC 类	(95)
TDecoratedFrame 类	(101)
TDecoratedMDIFrame 类	(102)
TDesktopDC 类	(103)
TDialog 类	(103)
TDib 类	(105)
TDibDC 类	(107)
TDiBitmap 类	(108)
TDllLoader 类	(109)
TDockable 类	(109)
TDockableControlBar 类	(109)

TDockingSlip 类	(109)
TDocManager 类	(110)
TDocTemplate 类	(112)
TDocTemplateT< D, V > 类	(114)
TDocument 类	(114)
TDocument::TList 嵌套类	(116)
TDragList 类	(117)
TDropTarget 类	(117)
TDvFileOpenDialog 类	(118)
TDvFileSaveDialog 类	(118)
TDvOpenSaveData 类	(118)
E	(120)
TEdgeConstraint 结构	(120)
TEdgeSlip 类	(120)
TEdit 类	(121)
TEditFile 类	(124)
TEditSearch 类	(125)
TEnDropFiles 类	(126)
TEnhMetaFilePict 类	(126)
TEnProtected 类	(127)
TErrorMode 类	(127)
TEventHandler 类	(127)
TEventHandler::TEventInfo 类	(128)
F	(129)
TFileDocument 类	(129)
TFileOpenDialog 类	(130)
TFileSaveDialog 类	(130)
TFilterValidator 类	(130)
TFindDialog 类	(131)
TFindReplaceDialog 类	(131)
TFindReplaceDialog::TData 类	(132)
TFindText 类	(132)
TFloatingFrame 类	(133)
TFloatingSlip 类	(134)
TFont 类	(135)
TFormatRange 类	(136)
TFrameWindow 类	(136)
G	(139)
TGadget 类	(139)
TGadgetControl 类	(141)
TGadgetList 类	(141)

TGadgetWindow 类	(142)
TGadgetWindowFont 类	(144)
TGauge 类	(145)
TGdiObject 类	(146)
TGlyphButton 类	(148)
TGroupBox 类	(149)
H	(150)
THalfTonePalette 类	(150)
THarbor 类	(150)
THatch8x8Brush 类	(151)
THdrItem 类	(152)
THdrNotify 类	(152)
THelpContext 类	(153)
THelpFileManager 类	(153)
THotKey 类	(154)
THSlider 类	(154)
I	(156)
TIC 类	(156)
TIdentityPalette 类	(156)
TImageInfo 类	(156)
TImageList 类	(157)
TSocketAddress 类	(158)
TInputDialog 类	(158)
TInStream 类	(159)
L	(160)
TLangId 类	(160)
TLayoutConstraint 结构	(160)
TLayoutMetrics 类	(161)
TLayoutWindow 类	(162)
TListBox 类	(162)
TListBoxData 结构	(164)
TListView 类	(165)
TListWindColumn 类	(166)
TListWindItem 类	(167)
TListWindow 类	(167)
TLocaleString 类	(169)
TLookupValidator 类	(170)
TLwComparer 类	(170)
TLwDispInfoNotify 类	(170)
TLwFindInfo 类	(170)
TLwHitTestInfo 类	(171)

TLwKeyDownNotify 类	(171)
TLwNotify 类	(171)
M	(173)
TMailer 类	(173)
TMci 类	(173)
TMciHiddenWindow 类	(174)
TMciWaveAudio 类	(174)
TMDIChild 类	(174)
TMDIClient 类	(175)
TMDIFrame 类	(176)
TMemoryDC 类	(177)
TMenu 类	(178)
TMenuDescr 类	(179)
TMenuGadget 类	(182)
TMenuItemEnabler 类	(182)
TMessageBar 类	(183)
TMetaFileDC 类	(184)
TMetaFilePict 类	(184)
TModeGadget 类	(185)
TModule 类	(185)
TModuleProc 类	(187)
TModuleVersionInfo 类	(187)
TMsgFilter 类	(188)
N	(189)
TNoteTab 类	(189)
TNoteTabItem 类	(190)
TNotify 类	(190)
O	(191)
TOleClientDC 类	(191)
TOleDocument 类	(191)
TOleFactoryBase < > 类	(192)
TOleFrame 类	(193)
TOleLinkView 类	(195)
TOleMDIFrame 类	(195)
TOleView 类	(196)
TOleWindow 类	(198)
TOpenSaveDialog 类	(207)
TOutStream 类	(208)
P	(209)
TPaintDC 类	(209)

TPalette 类	(209)
TPaletteEntry 类	(210)
TPaneSplitter 类	(211)
TParaFormat 类	(212)
TPen 类	(212)
TPickListDialog 类	(213)
TPickListPopup 类	(213)
TPictureWindow 类	(214)
TPidl 类	(214)
TPointer < > 类	(215)
TPopupMenu 类	(215)
TPreviewDCBase 类	(216)
TPreviewPage 类	(216)
TPreviewWin 类	(217)
TPrintDC 类	(217)
TPrintDialog 类	(218)
TPrintDialog::TData 类	(219)
TPrinter 类	(220)
TPrinterAbortDlg 类	(221)
TPrintout 类	(222)
TPrintPreviewDC 类	(223)
TProfile 类	(223)
TPropertyPage 类	(224)
TPropertySheet 类	(225)
TPXPictureValidator 类	(226)

R

TRadioButton 类	(227)
TRangeValidator 类	(227)
TRecentFiles 类	(228)
Trect 类	(228)
TRegion 类	(230)
TRegisterOCXWnd 类	(231)
TReplaceDialog 类	(231)
TReqResize 类	(232)
TResponseTableEntry 类	(232)
TRgbQuad 类	(232)
TRgbTriple 类	(233)
TRichEdit 类	(233)
TRichEditPagePreview 类	(235)
TRichEditPreviewFrame 类	(235)
TRichEditPrintout 类	(235)
TRollDialog 类	(236)

S		(237)
	TScreenDC 类	(237)
	TScrollBar 类	(237)
	TScrollBarData 结构	(238)
	TScroller 类	(238)
	TSelChange 类	(240)
	TSeparatorGadget 类	(240)
	TSerializer 类	(240)
	TSerializeReceiver 类	(240)
	TServiceEntry 类	(241)
	TServiceManager 类	(241)
	TServiceWindow 类	(242)
	TShellItem 类	(242)
	TShellItemIterator 类	(245)
	TShellMalloc 类	(245)
	TSizeGripGadget 类	(246)
	TSlider 类	(246)
	TSocket 类	(248)
	TSocketError 类	(250)
	TSocketInfo 类	(251)
	TSocketManager 类	(251)
	TSortedStringArray 类	(252)
	TSplashWindow 类	(253)
	TStatic 类	(253)
	TStatus 类	(254)
	TStatusBar 类	(254)
	TStorageDocument 类	(256)
	TStream 类	(257)
	TStreamSocket 类	(258)
	TStringLookupValidator 类	(259)
	TSystemFont 类	(259)
	TSystemMenu 类	(259)
T		(260)
	TTabControl 类	(260)
	TTabHitTestInfo 类	(261)
	TTabItem 类	(261)
	TTabKeyDown 类	(261)
	TTextGadget 类	(262)
	TTextRange 类	(262)
	TTimeGadget 类	(263)
	TTinyCaption 类	(263)
	TToolBox 类	(265)

TToolInfo 类	(265)
TTooltip 类	(266)
TTooltipEnabler 类	(267)
TTooltipHitTestInfo 类	(267)
TTooltipText 类	(268)
TTreelItem 类	(268)
TTreeNode 类	(268)
TTreeWindow 类	(269)
TTwComparer 类	(270)
TTwDispInfoNotify 类	(271)
TTwHitTestInfo 类	(271)
TTwKeyDownNotify 类	(271)
TTwNotify 类	(271)
U	(272)
TUDAccel 类	(272)
TUIBorder 类	(272)
TUIFace 类	(272)
TUIHandle 类	(273)
TUIPart 类	(275)
TUpDown 类	(275)
TUpDownNotify 类	(276)
V	(277)
TValidator 类	(277)
TVbxControl 类	(278)
TVbxEventHandler 类	(280)
TView 类	(284)
TVSlider 类	(286)
W	(287)
TWindow 类	(287)
TWindow Attr 类	(294)
TWindow DC 类	(295)
TWindowView 类	(295)
TWinG 类	(296)
TWinGDll 类	(296)
TWinSock 类	(297)
TWinSockDll 类	(297)
X	(299)
TXClipboard 类	(299)
TXCommCtrl 类	(299)
TXCompatibility 类	(299)

TGdiBase::TXGdi 类	(300)
TApplication::TXInvalidMainWindow 类	(301)
TModule::TXInvalidModule 类	(301)
TMenu::TXMenu 类	(301)
TXOutOfMemory 类	(302)
TXOwl 类	(302)
TPrinter::TXPrinter 类	(304)
TXShell 类	(304)
TValidator::TXValidator 类	(305)
TWindow::TXWindow 类	(305)
第三章 OWL 常量、类型定义、枚举、宏和事件处理函数	(306)
§ 3.1 OWL 所用到的常量值	(306)
§ 3.2 OWL 枚举量及其类型定义	(317)
§ 3.3 OWL 中的事件处理函数 (Event Handlers)	(326)
§ 3.4 OWL 库函数	(339)
§ 3.5 OWL 所用到的宏定义	(340)
§ 3.6 OWL 中的类型定义	(347)

第一章 OWL 类库与 Windows 程序设计

本章主要介绍利用 Borland C/C++ ObjectWindows 类库(OWL)开发 Windows 程序的基本方法。在此之前,我们将首先回顾一下有关面向对象程序设计(OOP)以及 C++ 语言方面的一些基本知识,然后再分别讲述 OWL 的有关特点,并通过具体的示例程序来说明使用 OWL 进行 Windows 程序设计的具体过程。

§ 1.1 面向对象程序设计(OOP)方法的回顾

面向对象程序设计(OOP)方法是一种全新的程序设计方法。它使人们以一种与传统的面向过程的程序设计方法完全不同的思维方式来认识软件设计的各个方面,并用于指导软件生产的整个过程。那么,这一新的技术究竟会给软件技术的发展和应用带来怎样的好处呢?

我们知道,思维方式决定着人们解决问题的方式。传统的结构化的程序设计是一种面向过程的程序设计方法,它注重于对问题的具体求解过程。在实际应用中通常采用自顶向下、逐步求精的方法来进行软件的设计和开发。其具体过程就是将一个问题划分成若干子问题,再根据需要对子问题作进一步的划分,直到问题能够被编程实现。六十年代诞生的这一方法曾给软件设计领域带来了巨大的发展,并一直主宰着过去 30 多年的软件设计领域。但随着技术的发展以及实际应用问题复杂程度的增大,人们对软件设计的方法也在不断地提出新的要求。面向对象的程序设计方法就是在这种背景下逐步发展起来的。

目前已有很多介绍面向对象程序设计方法的书籍,所以本书不打算详细地讲述面向对象程序设计方法的所有内容,仅仅把这一方法的最主要方面作一个简单的介绍。

1.1.1 面向对象程序设计与 C++

对象是一种特殊的数据,其中不仅包含了数据本身的属性,还包含着对该数据进行操作的过程或函数。因此可以简单地说,对象是被赋予了特定行为动作的数据。对象一旦被建立,它就可以被重用,也可以以它为基础,通过修改或扩充来创建新的对象。面向对象程序设计方法就是根据对象的特点,在程序设计的过程中,把重点放在对象之间的关系上,而不是在于其实现的细节上。对象之间的联系是通过它们之间相互的类属(或称层次)关系确定的。面向对象的程序当然也就包括对象,而对象又是由它的属性和行为组成的。属性不能从对象的外部直接存取,只能由对象的行为来操纵。

面向对象程序设计有几个优点。首先,由于对象包含属性和行为,因此它支持模块化的程序设计,而模块化程序设计又便于程序的开发和维护。其次,面向对象程序设计过程中可以充分地利用已有的类库,在此基础上所做的修改与扩充不会损害原有的类库,因此一个应用程序的新增代码可以明显地减少,这样就进一步增加了应用程序的可靠性。

C++ 是一种支持面向对象程序设计的语言。由于 C++ 是 C 语言超集,所以它不但实

现了面向对象程序设计的要求,而且还秉承了 C 语言本身的所有优点,因而非常有利于众多的 C 程序员进行软件的设计和开发。这就是 C++ 成为最被广泛使用的面向对象程序设计语言的主要原因。

用 C++ 编写的程序可以很好地发挥面向对象程序设计技术所带来的优势。这些优势包括封装性、继承性和多态性等等。下面各节将分别解释这些特性。

1.1.2 封装性(encapsulation)

封装性(encapsulation)是指一个对象既有数据的属性又有对数据进行操作的行为。C++ 通过引入新的数据类型“类”来描述对象并实现“封装”的特性。通过这种方式定义一个对象,对象的内部的实现细节被隐藏起来了,对外部是不可见的,也是不可存取的,这就是封装性,它有利于模块化的程序设计以及代码的可维护性和可重用性。

下面的例子中定义了一个类。

我们考虑数学中的一个简单的几何问题,当要处理角度时,我们希望把角度限制在 0—359 的范围以内。亦即 359 加 1 应该是 0 而不是 360,图 1.1 描述了这个循环计数系统。

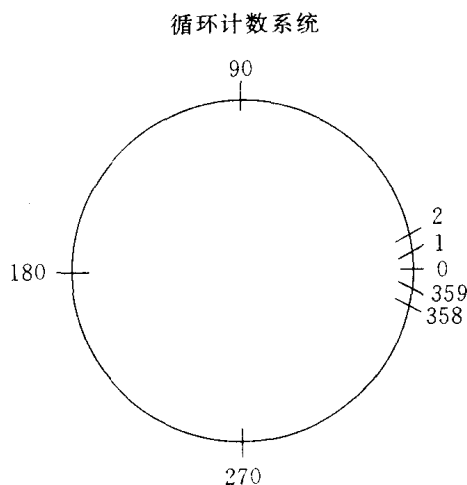


图 1.1 一个循环计数系统的图示

我们可以用 C++ 的类来描述这个系统。给这个类取的类名是 Circnum,为了说明问题的主要方面,在这个类中我们只提供了两种操作,使循环计数系统中的数能够进行增加和减少。有兴趣的读者可以把这个类加以扩充,以实现更多的操作,比如两个数的相加、相减、比较等等。

下面是该例子的源代码:

```
// 定义循环计数类
```

```
class Circnum
```

```
{
```

```
private:
```

```
    unsigned val;
```

```

static const int Circ_MaxVal;

public:
    Circnum (int v)
    {
        while(v > Circ_MaxVal)
            v -= (Circ_MaxVal + 1);
        while(v < 0)
            v += (Circ_MaxVal + 1);
        val = v;
    }

    Circnum (const Circnum& c) : val (c.val)
    {
    }

    Circnum ()
    {
        val = 0;
    }

    ~Circnum ()
    {
    }

    Circnum operator + + ()
    {
        if (val == Circ_MaxVal)
            val = 0;
        else
            val + + ;
        return *this;
    }

    Circnum operator - - ()
    {
        if (val == 0)
            val = Circ_MaxVal;
        else
            val - - ;
        return *this;
    }

    void operator + = (const Circnum& r)
    {

```