**第5课 文本编程**

**1创建插入符**

**CWnd::CreateSolidCaret创建插入符，ShowCaret()显示插入符。GetTextMetrics(),获得当前字体的一些信息。**

CClientDC dc(this);

TEXTMETRIC tm;

dc.GetTextMetrics (&tm); //获取当前字体的相关信息

//用当前字体的宽度和高度设置插入符的大小

CreateSolidCaret(tm.tmAveCharWidth /8,tm.tmHeight);

ShowCaret();

**CWnd::CreateCaret()创建图象插入符**

bitmap.LoadBitmap(IDB\_BITMAP1);

CreateCaret(&bitmap); //注意这里的bitmap是类成员变量，否则函数结束就会收回资源

ShowCaret();

**1 Cstring输出文字**

CTextView::OnDraw(CDC\* pDC/\*pDC\*/)函数中：

CString str(\_T("维?新?科?技?"));

pDC->TextOut(50,50,str);

**通过string资源输出string**

str.LoadStringW (IDS\_weixin);

pDC->TextOut(0,200,str);

**2 路径层的概念：有两种方法创建路径层**

CString str(\_T("维新科技"));

pDC->TextOut(50,50,str);

CSize sz=pDC->GetTextExtent (str); //获取str大小，CSize包括x、y两个方向大小

str.LoadStringW (IDS\_weixin);

pDC->TextOut(0,200,str);

//创建路径层

pDC->BeginPath ();

pDC->Rectangle(50,50,50+sz.cx,50+sz.cy); //路径区域，矩形左上角坐标和右下角坐标

pDC->EndPath ();

pDC->SelectClipPath(RGN\_DIFF); //设置模式，作图避开路径区域，RGN\_只在路径区域绘图

for(int i=0;i<300;i+=10)

{

pDC->MoveTo (0,i);

pDC->LineTo (300,i);

pDC->MoveTo (i,0);

pDC->LineTo (i,300);

}

< xmlnamespace prefix ="v" ns ="urn:schemas-microsoft-com:vml" />

**3 输出文字**

**添加成员变量m\_strLine**

用于存储文字信息

**添加OnLButtonDown响应，用于鼠标点击改变光标位置**

在OnLButtonDown中加入SetCaretPos(point);将光标移动到单击位置。

**添加成员变量m\_ptOrigin**

用于存储文字输出的起始点

在OnLButtonDown中加入：

m\_strLine.Empty ();

m\_ptOrigin=point;

将字符串清空，将起始点移到点击点位置。

**OnChar中输出文字**

CClientDC dc(this);

CFont font; //创建字体对象

font.CreatePointFont(300,\_T("华文行楷"),NULL); //设置字体，字体高度及类型

CFont \*pOldFont=dc.SelectObject(&font); //将字体选择到DC中

TEXTMETRIC tm; //创建字体信息对象

dc.GetTextMetrics(&tm); //获得当前字体信息

if(0x0d==nChar) //如果是回车

{

m\_strLine.Empty (); //字符串清空

m\_ptOrigin.y+=tm.tmHeight; //原始点高度加一个字体高度

}

else if(0x08==nChar) //处理退格键

{

COLORREF clr=dc.SetTextColor(dc.GetBkColor()); //设置字体颜色为背景颜色

dc.TextOut(m\_ptOrigin.x,m\_ptOrigin.y,m\_strLine); //重新输出，覆盖掉原信息

m\_strLine=m\_strLine.Left(m\_strLine.GetLength()-1); //字符串变短1个

dc.SetTextColor(clr); //字体颜色设置回去

}

else

{

m\_strLine=m\_strLine+char(nChar); //按下其他键，添加到字符串

}

CSize sz=dc.GetTextExtent(m\_strLine); //获取字符串的大小

CPoint pt; //处理光标位置，使其随输出文字移动

pt.x=m\_ptOrigin.x+sz.cx;

pt.y=m\_ptOrigin.y;

SetCaretPos(pt);

dc.TextOut(m\_ptOrigin.x,m\_ptOrigin.y,m\_strLine); //输出字体

dc.SelectObject(pOldFont); //将原先字体选择回去

**4 模拟卡啦OK变色的步骤。**

**设置定时器**

OnCreate函数中：

SetTimer(1,100,NULL); //选择定时器1，定时100ms，不用回调函数，window消息来处理

在view类中添加WM\_TIMER消息处理函数OnTimer(UINT\_PTR nIDEvent)，每过100ms OnTimer函数被调用一次，可以通过case语句判断nIDEvent选择感兴趣的定时器添加响应程序。

**Cwnd::DrawText的使用，在OnTimer函数中**

m\_nWidth+=5; //添加成员变量，宽度随时间增大

CClientDC dc(this);

TEXTMETRIC tm;

dc.GetTextMetrics (&tm); //获取字体信息

CRect rect;

rect.left =0;

rect.top=200;

rect.right =m\_nWidth; //矩形宽度随时间增大

rect.bottom=rect.top+tm.tmHeight ;

dc.SetTextColor (RGB(255,0,0));

CString str;

str.LoadStringW (IDS\_weixin);

dc.DrawText (str,rect,DT\_LEFT); //输出的str只有位于矩形范围内才能显示，DT\_LEFT表示str从矩形左边输入

rect.top=150;

rect.bottom=rect.top+tm.tmHeight ;

dc.DrawText (str,rect,DT\_RIGHT); //str右端从矩形右端输入

CSize sz=dc.GetTextExtent(str); //获得字符串的长度

if(m\_nWidth>sz.cx) //当长方形长度大于字符串的长度后，矩形宽度置0

{

m\_nWidth=0;

dc.SetTextColor(RGB(0,255,0));

dc.TextOut(0,200,str);

}