

0 进度

- ☐ 文件
 - ☒ 文本文件的读写方法
 - ☐ XML文件的树形结构的创建、查找
- ☐ 文件系统
 - ☒ 文件的表示方法: ListViewItem
 - ☒ 文件集合的表示方法: ListView控件
 - ☒ 文件夹的树形结构的创建、查找
 - ☒ 文件夹的树形结构的显示: TreeView
- ☒ 进程系统
 - ☒ 进程Process的常用操作、属性
 - ☒ 进程的显示方法 : ListViewItem
 - ☒ 常用的性能计数器 :
 - ☒ 性能计数器的显示方法 : Chart Queue
- ☐ 项目管理
 - ☐ 参数、资源、XML资源
- ☒ 窗口管理
 - ☒ MDI主/子窗体 ,
 - ☒ 菜单
 - ☒ 上下文菜单动态创建、显示
- ☒ 资源管理器
 - ☒ 文件、进程、设备系统的Watcher ?
- ☐ 自定义控件、集成开发
 - ☒ 创建控件、设置事件处理函数
 - ☒ 自定义的事件Event、引发事件RaiseEvent、处理事件
 - ☐ 控件集合
 - ☐ 解决方案与项目的类型、引用关系
- ☐ 并行计算、后台线程与信号量
 - ☒ parrell.for
 - ☒ backgroundWorker
 - ☐ semaphore
 - ☒ 委托类型的定义 , 委托对象的创建 (address of 函数名)
 - ☐ 委托对象的调用(frm.invoke)、参数传递
- ☒ 智能输入

- ☒ 键盘事件与参数KeyPress KeyUp KeyDown
- ☒ 键树的创建、查找
- ☒ 操作历史
 - ☒ 操作类的定义（继承）、操作对象表的管理
 - ☒ Redo与Undo的机制（动态的应用）
- ☒ 图像处理技术
 - ☒ Bitmap对象的像素计算（颜色）、常用运算（旋转、缩放）
 - ☒ 幻灯片技术（Timer、矩阵的局部变换）
 - ☒ 图像的图块计算（图的遍历）
- ☐ 图形技术
 - ☒ 图形类中的继承设计、多态效果
 - ☒ 图形对象的管理：自定义的事件Event、引发事件RaiseEvent、处理事件
 - ☒ 图形对象的选择技术
 - ☐ 图形对象的动态编辑：橡皮筋、拖拽

1 文件

1.1 文本文件的读写方法

1.1.1 C#逐行读

```
using System;
using System.IO;

class Test
{
    public static void Main()
    {
        try
        {
            // Open the text file using a stream reader.
            using (StreamReader sr = new StreamReader("TestFile.txt"))
            {
                // Read the stream to a string, and write the string to the console.
                String line = sr.ReadToEnd();
                Console.WriteLine(line);
            }
        }
        catch (Exception e)
        {
            Console.WriteLine("The file could not be read:");
            Console.WriteLine(e.Message);
        }
    }
}
```

1.1.2 VB读写

```
Dim PathUserData As String = Application.StartupPath & "\实操统计sql.txt"
TextTB.Text = System.IO.File.ReadAllText(PathUserData)
'或者用 System.IO.File.ReadAllText(PathUserData, System.Text.Encoding.UTF8)
```

```
Dim PathUserData As String = Application.StartupPath & "\实操统计sql.txt"
Dim t As System.IO.StreamWriter = New System.IO.StreamWriter(PathUserData,
True, System.Text.Encoding.UTF8)
t.Write(TextTB.Text)
t.Close()
```

1.2 XML文件的树形结构的创建、查找

```
Imports System.Xml

Public Class frmTest

    Dim doc As XmlDocument, amf As XmlElement

    Private Sub btnOpen_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOpen.Click
        Dim fname As String = "x1.amf"
        fname = My.Application.Info.DirectoryPath & "\" & fname
        doc = New XmlDocument : doc.Load(fname)
        amf = doc.SelectSingleNode("amf")
        '遍历整个结构的基本方法
        GetSegment(amf, "object", txtObjects)
        GetSegment(amf, "constellation", txtConstellations)
        GetSegment(amf, "material", txtMaterials)
    End Sub

    Private Sub GetSegment(ByVal amf As XmlElement, ByVal SName As String, ByVal
textbox As TextBox)
        Dim Nodes As XmlNodeList = amf.GetElementsByTagName(SName)
        For i = 0 To Nodes.Count - 1
            Dim XNode As XmlNode = Nodes(i)
            Dim Attributes As String = ""
            For j = 0 To XNode.Attributes.Count - 1
                Attributes = Attributes & XNode.Attributes(j).Name & ": " &
XNode.Attributes(j).Value & vbCrLf
            Next
            textbox.Text = textbox.Text & Attributes & XNode.InnerXml() & vbCrLf &
vbCrLf
        Next
    End Sub

    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
        doc.Save("x.amf")
    End Sub
End Class
```

2 文件系统

2.1 文件的表示方法: ListViewItem

```
Public Class clsFiles
    Public items() As ListViewItem
    Public Sub New(pathname As String)
        Dim files() As String = Directory.GetFiles(pathname, "*.*)")
        ReDim items(files.Count - 1)
        For i = 0 To files.Count - 1
            item = New ListViewItem(files(i), 0)
            item.SubItems.Add("属性")
            items(i) = item
        Next
    End Sub
End Class
```

2.2 文件集合的表示方法: ListView控件

```
'设置列
ListView.items = Files.items
```

2.3 文件夹的树形结构的创建、查找

```
Public Class clsDirectory
    Public Directories As List(Of clsDirectory)
    Public Sub New(PathName As String)
        Create(PathName)
    End Sub
    Private Sub Create(PathName As String)
        directories = New List(Of clsDirectory)
        Dim foldes() As String = Directory.GetDirectories(PathName)
        For i = 0 To foldes.Count - 1
            Directories.Add(New clsDirectory(foldes(i)))
        Next
    End Sub
End Class
```

2.4 文件夹的树形结构的显示: TreeView

```
Public Class clsDirectory
    Public Root As TreeNode
    Public Sub New(pathname As String)
        Root = CreateNode(pathname)
    End Sub
    Public Function CreateNode(pathname As String) As TreeNode
        Dim directories() As String = Directory.GetDirectories(pathname)
        Dim node As TreeNode = New TreeNode(Path.GetFileNameWithoutExtension(pathname))
        For i = 0 To directories.Count - 1
            node.Nodes.Add(CreateNode(directories(i)))
        Next
    End Function
End Class
```

```

        Next
        Return node
    End Function
End ClassSub Display(ByVal lstInfo As ListBox)
    For i = 0 To files.Count - 1
        lstInfo.Items.Add(files(i))
    Next
    For i = 0 To Directories.Count - 1
        Directories(i).Display(lstInfo)
    Next
End Sub

```

3 进程系统

3.1 进程Process的常用操作、属性

- Process.Threads
- proc.ProcessName
- proc.MainModule.FileName
- Process.GetProcesses()
- Proces(i).Kill()

3.2 进程的显示方法：ListViewItem

```

Public Class clsProcess
    Public Process As Process
    Public ProcessName As String
    Public FileName As String
    Public item As ListViewItem

    Public Sub New(proc As Process)
        Me.Process = proc
        ProcessName = proc.ProcessName
        Try
            FileName = proc.MainModule.FileName
        Catch ex As Exception
            FileName = "" ' 为什么拒绝访问？
        End Try
        Me.item = New ListViewItem(ProcessName, 0)
        item.SubItems.Add(FileName)
    End Sub

    Public Function GetThread() As ProcessThreadCollection
        Return Process.Threads
    End Function
End Class

Public Class clsProcesses
    Private Processes As List(Of clsProcess)
    Public Items() As ListViewItem

```

```

Public Sub New()
    Processes = New List(Of clsProcess)
    Dim Procs() As Process = Process.GetProcesses()
    For i = 0 To Procs.Count - 1
        Processes.Add(New clsProcess(Procs(i)))
    Next
    ReDim Items(Procs.Count - 1)
    For i = 0 To Procs.Count - 1
        Items(i) = Processes(i).item
    Next
End Sub

Public Sub ListView_Init(lv As ListView)
    Dim ColumnHeaders As ListView.ColumnHeaderCollection
    ColumnHeaders = New ListView.ColumnHeaderCollection(lv)
    ColumnHeaders.Add("进程名")
    ColumnHeaders.Add("程序名")
    With lv
        .View = View.Details
        .FullRowSelect = True
    End With
    lv.Columns(0).Width = 200
    lv.Columns(1).Width = 600
End Sub

Public Sub ListView_Display(lvItem As ListView)
    lvItem.Items.Clear()
    lvItem.Items.AddRange(Items)
End Sub

Public Sub Kill(FileName As String)
    Dim Procs() As Process = Process.GetProcesses()
    For i = 0 To Procs.Count - 1
        Try
            If Procs(i).MainModule.FileName = FileName Then
                Procs(i).Kill()
            End If
        Catch ex As Exception
        End Try
    Next
End Sub

Public Function GetFileName(index As Integer) As String
    Return Processes(index).FileName
End Function

Public Function GetProcessName(index As Integer) As String
    Return Processes(index).ProcessName
End Function

Public Function GetThread(index As Integer) As ProcessThreadCollection
    Return Processes(index).GetThread()
End Function

End Class

```

```
Public Class frmTest
```

```

    Dim Processes As clsProcesses
    Private Sub btnDisplayProc_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDisplayProc.Click
        Processes = New clsProcesses
        Processes.ListView_Init(lvProcess)
        Processes.ListView_Display(lvProcess)
    End Sub

    Private Sub btnKill_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnKill.Click
        If lvProcess.SelectedIndices.Count = 0 Then Exit Sub
        For i = lvProcess.SelectedIndices.Count - 1 To 0 Step -1
            Dim FileName As String =
Processes.GetFileName(lvProcess.SelectedIndices(i))
            Processes.Kill(FileName)
        Next
    End Sub

    Private Sub lvProcess_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles lvProcess.SelectedIndexChanged
        If lvProcess.SelectedIndices.Count = 0 Then Exit Sub
        Dim k As Integer = lvProcess.SelectedIndices(0)
        Dim Threads As ProcessThreadCollection = Processes.GetThread(k)
        lstThread.Items.Clear()
        For i = 0 To Threads.Count - 1
            Dim Thread As ProcessThread = Threads(i)
            lstThread.Items.Add(Thread.Id & vbTab & Thread.ThreadState.ToString)
        Next
    End Sub
End Class

```

3.3 常用的性能计数器：

```

Server
Processor Information
Cache
USUB
TCPv6
Terminal Services

```

3.4 性能计数器的显示方法：Chart Queue

```

Imports System.Windows.Forms.DataVisualization.Charting

Public Class clsCounter
    Private WithEvents Ticker As Timer
    Protected Counter As PerformanceCounter
    Private chtValue As Chart, lblValue As Label
    Private Q As Queue(Of Single)
    Private Const QueueLength As Integer = 100 ' 队列的最大长度
    Private Const TickerInterval As Integer = 1000 ' 时钟中断的间隔

```

```

Public Sub New(Counter As PerformanceCounter, Chart As Chart, lblValue As Label)
    Me.Counter = Counter
    Me.chtValue = Chart : Me.lblValue = lblValue
    Me.Q = New Queue(Of Single)
    Ticker = New Timer : Ticker.Interval = TickerInterval : Ticker.Enabled = True
End Sub

Private Sub Ticker_Tick(sender As Object, e As System.EventArgs) Handles
Ticker.Tick
    Q.Enqueue(GetValue)
    If Q.Count > QueueLength Then
        Q.Dequeue()
    End If
    lblValue.Text = Q.Last
    DisplayChart()
End Sub

' NextValue和RawValue的区别是什么? 原理是什么?
Private Function GetValue() As Single
    Return Counter.NextValue
End Function

Sub DisplayChart()
    Me.chtValue.Series(0).ChartType = SeriesChartType.Line
    Me.chtValue.Series("Series1").Points.Clear()
    For i = 0 To Q.Count - 1
        Me.chtValue.Series("Series1").Points.AddXY(i, Q(i))
    Next
    chtValue.Legends("Legend1").Docking = Docking.Top
End Sub

End Class

```

```

Imports System.Windows.Forms.DataVisualization.Charting

Public Class frmTest
    Dim Processor_Time As clsCounter
    Dim User_Time As clsCounter

    Private Sub frmTest_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load
        User_Time = New clsCounter(New PerformanceCounter("Process", "% User Time",
"_Total"), Chart1, lblUserTime)
        Processor_Time = New clsCounter(New PerformanceCounter("Process", "% Processor
Time", "_Total"), Chart2, lblProcessorTime)
    End Sub
End Class

```

4 项目管理

4.1 参数、资源、XML资源

xml

```
Imports System.Xml

Public Class clsLight
    Public Name As String
    Public Position As String
    Public Ambient As String
    Public Diffuse As String
    Public Specular As String

    Public Sub New(node As System.Xml.XmlElement)
        Name = node.Name
        Position = node.SelectSingleNode("position").InnerText
        Ambient = node.SelectSingleNode("ambient").InnerText
        Diffuse = node.SelectSingleNode("diffuse").InnerText
        Specular = node.SelectSingleNode("specular").InnerText
    End Sub
End Class

Public Class clsLights
    Public Lights As List(Of clsLight)

    Public Sub New(Content As String)
        Dim XML As New XmlDocument
        XML.LoadXml(Content)
        Lights = New List(Of clsLight)

        Dim node As XmlNode
        node = XML.FirstChild.FirstChild
        While node IsNot Nothing
            Lights.Add(New clsLight(node))
            node = node.NextSibling
        End While
    End Sub
End Class
```

```
Public Class frmTest

    Dim Lights As clsLights

    Private Sub btnXML_Click(sender As System.Object, e As System.EventArgs) Handles
        btnLights.Click
            Lights = New clsLights(My.Resources.ConfigLight)
            lstLight.Items.Clear()
            For i = 0 To Lights.Lights.Count - 1
                lstLight.Items.Add(Lights.Lights(i).Name)
            Next
        End Sub
End Class
```

```

Private Sub lstLight_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles lstLight.SelectedIndexChanged
    If lstLight.SelectedIndex = -1 Then Exit Sub
    Dim Light As clsLight = Lights.Lights(lstLight.SelectedIndex)
    With lstLightPara.Items
        .Clear()
        .Add("Name" & vbTab & Light.Name)
        .Add("Position" & vbTab & Light.Position)
        .Add("Ambient" & vbTab & Light.Ambient)
        .Add("Diffuse" & vbTab & Light.Diffuse)
        .Add("Specular" & vbTab & Light.Specular)
    End With

End Sub
End Class

```

5 窗口管理

5.1 MDI主/子窗体

```

Public Class frmChild

    Private Sub ExitToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles ExitToolStripMenuItem.Click
        Me.Close()
    End Sub
End Class

```

```

Public Class frmParent

    Dim Count As Integer = 0

    Private Sub frmParent_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.IsMdiContainer = True
    End Sub
    Private Sub mnuCreate_Click(sender As Object, e As EventArgs) Handles
mnuCreate.Click
        CreateChild()
    End Sub
    Sub CreateChild()
        Count += 1
        Dim NewChild As Form = New frmChild
        NewChild.MdiParent = Me
        NewChild.Text = "子窗体" & Count
        NewChild.Show()
    End Sub

    Private Sub Arrange_Click(sender As Object, e As EventArgs) Handles
m_ArrangeIcons.Click, m_Cascade.Click, m_TileHorizontal.Click, m_TileVertical.Click

```

```

        If sender Is m_ArrangeIcons Then Me.LayoutMdi(MdiLayout.ArrangeIcons)
        If sender Is m_Cascade Then Me.LayoutMdi(MdiLayout.Cascade)
        If sender Is m_TileHorizontal Then Me.LayoutMdi(MdiLayout.TileHorizontal)
        If sender Is m_TileVertical Then Me.LayoutMdi(MdiLayout.TileVertical)
    End Sub

    Private Sub CloseAll_Click(sender As Object, e As EventArgs) Handles
m_CloseAll.Click
        For i = Me.MdiChildren.Length - 1 To 0 Step -1
            Me.MdiChildren(i).Close()
        Next
        Count = 0
    End Sub

End Class

```

5.2 菜单

```

Public Class frmChild
    ' 提供了访问键盘当前状态（例如，当前按下了什么键）的属性，并提供了向活动窗口发送键击的方法。
    Dim Keyboard As Devices.Keyboard

    Private Sub frmChild_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load
        Timer1.Enabled = True
        Keyboard = New Devices.Keyboard
    End Sub

    Private Sub m_FileNew_Click(sender As System.Object, e As System.EventArgs) Handles
m_FileNew.Click, m_FileOpen.Click, m_FileSave.Click, m_FileExit.Click
        If sender Is m_FileNew Then FileNew()
        If sender Is m_FileOpen Then FileOpen()
        If sender Is m_FileSave Then FileSave()
        If sender Is m_FileExit Then FileExit()
    End Sub
    Private Sub FileNew()
        MsgBox("FileNew")
    End Sub
    Private Sub FileOpen()
        MsgBox("FileOpen")
    End Sub
    Private Sub FileSave()
        MsgBox("FileSave")
    End Sub
    Private Sub FileExit()
        Me.Close()
    End Sub

    Private Sub m_ToolLine_Click(sender As System.Object, e As System.EventArgs)
Handles m_ToolLine.Click, m_ToolRectangle.Click, m_ToolCircle.Click,
m_ToolPolygon.Click, m_ToolSelect.Click
        If sender Is m_ToolLine Then ToolLine()
        If sender Is m_ToolRectangle Then ToolRectangle()
    End Sub

```

```

        If sender Is m_ToolCircle Then ToolCircle()
        If sender Is m_ToolPolygon Then ToolPolygon()
        If sender Is m_ToolSelect Then Toolselect()
    End Sub
    Private Sub ToolLine()
        MsgBox("ToolLine")
    End Sub
    Private Sub ToolRectangle()
        MsgBox("ToolRectangle")
    End Sub
    Private Sub ToolCircle()
        MsgBox("ToolCircle")
    End Sub
    Private Sub ToolPolygon()
        MsgBox("ToolPolygon")
    End Sub
    Private Sub ToolSelect()
        MsgBox("ToolSelect")
    End Sub

    Private Sub m_TransformPan_Click(sender As System.Object, e As System.EventArgs)
Handles m_TransformPan.Click, m_TransformScale.Click, m_TransformRotate.Click
        If sender Is m_TransformPan Then TransformPan()
        If sender Is m_TransformScale Then TransformScale()
        If sender Is m_TransformRotate Then TransformRotate()
    End Sub
    Private Sub TransformPan()
        MsgBox("TransformPan")
    End Sub
    Private Sub TransformScale()
        MsgBox("TransformScale")
    End Sub
    Private Sub TransformRotate()
        MsgBox("TransformRotate")
    End Sub

    Private Sub m_PropertyColor_Click(sender As System.Object, e As System.EventArgs)
Handles m_PropertyColor.Click, m_Propertywidth.Click, m_Propertystyle.Click
        If sender Is m_PropertyColor Then PropertyColor()
        If sender Is m_Propertywidth Then Propertywidth()
        If sender Is m_Propertystyle Then Propertystyle()
    End Sub
    Private Sub PropertyColor()
        MsgBox("PropertyColor")
    End Sub
    Private Sub Propertywidth()
        MsgBox("Propertywidth")
    End Sub
    Private Sub Propertystyle()
        MsgBox("Propertystyle")
    End Sub

```

’ 使用状态栏中的标签控件

```

    Private Sub m_FileNew_MouseEnter(sender As System.Object, e As System.EventArgs)
Handles m_FileNew.MouseEnter
        sslblMnuPrompt.Text = "新建文件"
    End Sub
    Private Sub m_FileNew_MouseLeave(sender As System.Object, e As System.EventArgs)
Handles m_FileNew.MouseLeave
        sslblMnuPrompt.Text = ""
    End Sub

    ' 使用状态栏中的进度条
    Private Sub btnWork_Click(sender As System.Object, e As System.EventArgs) Handles
btnWork.Click
        ssprgWork.Minimum = 0
        ssprgWork.Maximum = 100
        DOWork()
    End Sub
    Private Sub DOWork()
        For i = ssprgWork.Minimum To ssprgWork.Maximum
            System.Threading.Thread.Sleep(10)
            ssprgWork.Value = i
        Next
    End Sub

    Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs) Handles
Timer1.Tick
        sslblTime.Text = DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss")
        If Keyboard.CapsLock = True Then
            sslblCapsLock.Text = "大写"
        Else
            sslblCapsLock.Text = "小写"
        End If
        If Keyboard.NumLock = True Then
            sslblNumLock.Text = "数字"
        Else
            sslblNumLock.Text = "非数字"
        End If
    End Sub
End Class

```

5.3 上下文菜单动态创建、显示

```

Public Class frmTest

    Dim WithEvents MenuOpState As clsMenuOpState

    Private Sub frmTest_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        MenuOpState = New clsMenuOpState
    End Sub

    Private Sub PictureBox1_MouseClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.MouseEventHandler) Handles PictureBox1.MouseClick
        If e.Button = Windows.Forms.MouseButtons.Right Then

```

```

        MenuOpState.Show(Left + PictureBox1.Left + e.X, Top + PictureBox1.Top +
e.Y)
    End If
End Sub

Private Sub MenuOpState_FlagChange(ByVal flag As OpState) Handles
MenuOpState.FlagChange
    Text = flag.ToString
End Sub
End Class

```

```

Public Enum OpState '可扩展的状态类型
    Pan
    Rotate
End Enum

Public Class clsMenuOpState
    Private MenuOpState As ContextMenuStrip ' 菜单条
    Private WithEvents MenuOpStatePan, MenuOpStateRotate As ToolStripMenuItem '菜单项
    Private Flag As OpState ' 状态
    Public Event FlagChange(ByVal flag As OpState)

    Public Sub New()
        MenuOpState = New ContextMenuStrip
        MenuOpStatePan = New ToolStripMenuItem : MenuOpStateRotate = New
ToolStripMenuItem
        MenuOpState.Items.AddRange(New ToolStripItem() {Me.MenuOpStatePan,
Me.MenuOpStateRotate})
        With MenuOpState
            .Name = "MenuOpState"
            .Size = New System.Drawing.Size(113, 48)
        End With
        With MenuOpStatePan
            .Text = "Pan"
            .Size = New System.Drawing.Size(112, 22)
        End With
        With MenuOpStateRotate
            .Text = "Rotate"
            .Size = New System.Drawing.Size(112, 22)
        End With
        Flag = OpState.Pan : SetMenuOpState()
    End Sub

    Private Sub SetMenuOpState()
        Select Case Flag
            Case OpState.Pan
                MenuOpStatePan.Checked = True : MenuOpStateRotate.Checked = False
            Case OpState.Rotate
                MenuOpStatePan.Checked = False : MenuOpStateRotate.Checked = True
        End Select
    End Sub

```

```

    Private Sub MenuOpState_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MenuOpStatePan.Click, MenuOpStateRotate.Click
        If sender Is MenuOpStatePan Then Flag = OpState.Pan
        If sender Is MenuOpStateRotate Then Flag = OpState.Rotate
        SetMenuOpState()
        RaiseEvent FlagChange(Flag)
    End Sub

    Public Sub Show(ByVal x As Integer, ByVal y As Integer)
        MenuOpState.Show(x, y)
    End Sub

End Class

```

6 资源管理器

6.1 文件、进程、设备系统的Watcher

6.1.1 文件watcher

```

Public Class frmTest
Public WithEvents Watcher As FileSystemWatcher
Delegate Sub DeleCreateFile(filename As String)
Delegate Sub DeleRemoveFile(filename As String)
Delegate Sub DeleRenameFile(oldfilename As String, newfilename As String)

Dim opCreate As DeleCreateFile
Dim opRemove As DeleRemoveFile
Dim opRename As DeleRenameFile

Private Sub frmTest_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load
    opCreate = New DeleCreateFile(AddressOf Me.CreateFile)
    opRemove = New DeleRemoveFile(AddressOf Me.RemoveFile)
    opRename = New DeleRenameFile(AddressOf Me.RenameFile)
End Sub

Private Sub btnBrowse_Click(sender As System.Object, e As System.EventArgs) Handles
btnBrowse.Click
    Dim PathName As String = GetFolder()
    If PathName = "" Then Return
    txtPathName.Text = PathName
    Watcher = New FileSystemWatcher(PathName)
    Watcher.NotifyFilter = NotifyFilters.FileName Or NotifyFilters.DirectoryName
    Watcher.EnableRaisingEvents = True
    DisplayPath(PathName)
End Sub

Private Sub DisplayPath(path As String)
    Dim files() As String = Directory.GetFiles(path, "*.*",
SearchOption.TopDirectoryOnly)

```

```

        1stFile.Items.Clear()
        For i = 0 To files.Count - 1
            1stFile.Items.Add(files(i))
        Next
    End Sub
    '增加
    Private Sub watcher_Created(sender As Object, e As System.IO.FileSystemEventArgs)
Handles watcher.Created
        Dim filename As String = e.FullPath
        Invoke(opCreate, filename)
    End Sub
    Private Sub CreateFile(filename As String)
        1stFile.Items.Add(filename)
    End Sub

    '删除
    Private Sub watcher_Deleted(sender As Object, e As System.IO.FileSystemEventArgs)
Handles watcher.Deleted
        Dim filename As String = e.FullPath
        Invoke(opRemove, filename)
    End Sub
    Private Sub RemoveFile(filename As String)
        1stFile.Items.Remove(filename)
    End Sub

    '改名
    Private Sub watcher_Renamed(sender As Object, e As System.IO.RenamedEventArgs)
Handles watcher.Renamed
        Dim oldfilename As String = e.OldFullPath
        Dim newfilename As String = e.FullPath
        Invoke(opRename, New String() {oldfilename, newfilename})
    End Sub
    Private Sub RenameFile(oldfilename As String, newfilename As String)
        Dim k As Integer = 1stFile.Items.IndexOf(oldfilename)
        1stFile.Items(k) = newfilename
    End Sub

    Public Function GetFolder() As String
        Static dlg As New FolderBrowserDialog
        dlg.ShowNewFolderButton = True
        If dlg.ShowDialog() = DialogResult.OK Then Return dlg.SelectedPath
        Return ""
    End Function

End Class

```

6.1.2 进程watcher

```

Public Class frmTest
    Enum OpKind
        Creation
        Modification
        Deletion
    End Enum

```


End Enum

```
Public WithEvents CreationWatcher As ManagementEventWatcher ' 用于观察进程新建的事件
Public WithEvents ModificationWatcher As ManagementEventWatcher ' 用于观察进程变化的事件
Public WithEvents DeletionWatcher As ManagementEventWatcher ' 用于观察进程退出的事件

Delegate Sub deleAddOp(Process As ManagementBaseObject, kind As OpKind)
Dim AddProcessOp As deleAddOp

Private Sub frmTest_Load(sender As Object, e As System.EventArgs) Handles Me.Load
    AddProcessOp = New deleAddOp(AddressOf AddOp)
    ' 建立WMI事件查询对象 WITHIN 1表示采集事件的事件间隔为1秒
    Dim qCreation As EventQuery = New EventQuery("SELECT * FROM __InstanceCreationEvent WITHIN 1 WHERE TargetInstance isa ""Win32_Process""")
    Dim qModification As EventQuery = New EventQuery("SELECT * FROM __InstanceModificationEvent WITHIN 1 WHERE TargetInstance isa ""Win32_Process""")
    Dim qDeletion As EventQuery = New EventQuery("SELECT * FROM __InstanceDeletionEvent WITHIN 1 WHERE TargetInstance isa ""Win32_Process""")
    ' 建立观察对象
    CreationWatcher = New ManagementEventWatcher(qCreation)
    ModificationWatcher = New ManagementEventWatcher(qModification)
    DeletionWatcher = New ManagementEventWatcher(qDeletion)
End Sub

Private Sub btnBrowse_Click(sender As System.Object, e As System.EventArgs) Handles btnStart.Click
    lstCreation.Items.Clear() : lstModification.Items.Clear() : lstDeletion.Items.Clear()
    ' 异步开始侦听
    CreationWatcher.Start() : ModificationWatcher.Start() : DeletionWatcher.Start()
End Sub

Private Sub btnStop_Click(sender As System.Object, e As System.EventArgs) Handles btnStop.Click
    CreationWatcher.Stop() : ModificationWatcher.Stop() : DeletionWatcher.Stop()
End Sub

Private Sub CreationWatcher_EventArrived(sender As Object, e As System.Management.EventArrivedEventArgs) Handles CreationWatcher.EventArrived
    'e.NewEvent的所有成员: PreviousInstance SECURITY_DESCRIPTOR TargetInstance TIME_CREATED
    Dim Process As ManagementBaseObject = e.NewEvent("TargetInstance")
    Me.Invoke(Me.AddProcessOp, Process, OpKind.Creation)
End Sub

Private Sub ModificationWatcher_EventArrived(sender As Object, e As System.Management.EventArrivedEventArgs) Handles ModificationWatcher.EventArrived
    Dim Process As ManagementBaseObject = e.NewEvent("TargetInstance")
    Me.Invoke(Me.AddProcessOp, Process, OpKind.Modification)
End Sub

Private Sub DeletionWatcher_EventArrived(sender As Object, e As System.Management.EventArrivedEventArgs) Handles DeletionWatcher.EventArrived
    Dim Process As ManagementBaseObject = e.NewEvent.Item("TargetInstance")
```

```

        Me.Invoke(Me.AddProcessOp, Process, OpKind.Deletion)
    End Sub

    Private Sub frmTest_FormClosing(sender As Object, e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
        CreationWatcher.Stop() : ModificationWatcher.Stop() : DeletionWatcher.Stop()
    End Sub

    Sub AddOp(Process As ManagementBaseObject, kind As OpKind)
        If kind = OpKind.Creation Then
            lstCreation.Items.Add(Process.Item("Name"))
            lstCreation.SelectedIndex = lstCreation.Items.Count - 1
        End If
        If kind = OpKind.Modification Then
            lstModification.Items.Add(Process.Item("Name"))
            lstModification.SelectedIndex = lstModification.Items.Count - 1
        End If
        If kind = OpKind.Deletion Then
            lstDeletion.Items.Add(Process.Item("Name"))
            lstDeletion.SelectedIndex = lstDeletion.Items.Count - 1
        End If
        '观察进程的所有属性名和值
        With lstProperty.Items
            .Clear()
            For Each p As PropertyData In Process.Properties
                .Add(p.Name & vbTab & p.Value)
            Next
        End With
    End Sub

End Class

```

7 自定义控件、集成开发

7.1 创建控件、设置事件处理函数

```

Public Class Form1

    Dim WithEvents Button1 As System.Windows.Forms.Button

    Private Sub btnCreate_Click(sender As System.Object, e As System.EventArgs) Handles
btnCreate.Click
        Button1 = New System.Windows.Forms.Button
        With Button1
            .Location = New System.Drawing.Point(50, 50)
            .Name = "btnInit"
            .Size = New System.Drawing.Size(25, 25)
            .TabIndex = 0
            .Text = ""
            .UseVisualStyleBackColor = True
        End With
        Me.Controls.Add(Button1)
        Me.ResumeLayout(False) '恢复正常的布局逻辑，可以选择强制对挂起的布局请求立即进行布局。
    End Sub
End Class

```

```

        Me.PerformLayout() '强制控件将布局逻辑应用于其所有子控件。
    End Sub

    Private Sub btnCheck_Click(sender As System.Object, e As System.EventArgs) Handles
btnCheck.Click
        If Button1.FlatStyle = FlatStyle.Standard Then
            Button1.FlatStyle = FlatStyle.Flat
        Else
            Button1.FlatStyle = FlatStyle.Standard
        End If
    End Sub

    Private Sub Button1_Click(sender As Object, e As System.EventArgs) Handles
Button1.Click
        MsgBox("Hello")
    End Sub
End Class

```

```

Public Class clsButtonRow
    Private Buttons As List(Of Button)
    Private width As Integer ' 方形按钮的宽度
    Private XGap As Integer ' x方向按钮的间距
    Private Start As Point ' 按钮集合的排列起点 (左上角)

    Public Sub New(frm As frmTest, Start As Point, ButtonN As Integer)
        Me.Width = 25 : Me.XGap = 1
        Me.Start = Start
        ' 创建按钮集合
        CreateButtons(ButtonN)
        ' 将按钮集合加入frm中
        For i = 0 To ButtonN - 1
            frm.Controls.Add(Buttons(i))
        Next
        frm.ResumeLayout(False) '恢复正常的布局逻辑，可以选择强制对挂起的布局请求立即进行布局。
        frm.PerformLayout() '强制控件将布局逻辑应用于其所有子控件。
    End Sub

    Public Sub CreateButtons(ButtonN As Integer)
        Buttons = New List(Of Button)
        For i = 0 To ButtonN - 1
            Buttons.Add(New Button)
            With Buttons(i)
                .Name = "btn" & i
                .Text = ""
                .TabIndex = i
                .UseVisualStyleBackColor = True
                .Size = New System.Drawing.Size(width, width)

                .Location = New System.Drawing.Point(Start.X + i * (width + XGap),
Start.Y)
            End With
            AddHandler Buttons(i).Click, AddressOf Button_Click
        Next
    End Sub

```

```

End Sub

Private Sub Button_Click(sender As Object, e As EventArgs)
    MsgBox("Hello " & sender.name)
End Sub
End Class

```

7.2 自定义的事件Event、引发事件RaiseEvent、处理事件

'在图形对象集合中，设置事件，以便在集合发生变化时，界面进行绘图操作。

```

Class clsElems           // 图形对象集合类
Private elems as list(of Elem)
Event Append()           // 定义事件
Sub Append(e as Elem)
    Elems.Add(e)
    RaiseEvent Append()  // 产生事件
End sub
End class
frmTest: 界面类
    dim WithEvents Elems as clsElems // 能够接受事件
    Sub Draw( ) handle Elems. Append //响应事件
        clsElems.Draw(....)
    End Sub
end sub

```

// 以智能输入为例，说明自定义事件的使用方法。

```

Public Class frmListBox
    Public Event CharPress(C As Char) // 定义事件
Private Sub lstWords_KeyPress(e)
    ....
    RaiseEvent CharPress(e.KeyChar) '产生事件
End Sub
End Class
Public Class frmTest
Dim WithEvents frm As frmListBox // 能够接受事件
    Private Sub frm_CharPress(C As Char) Handles frm.CharPress //响应事件
        txtContent.SelectedText = C
    End Sub
End Class

```

7.3 控件集合

7.4 解决方案与项目的类型、引用关系

8 并行计算、后台线程与信号量

8.1 parrell.for

```

Imports System.Threading.Tasks

Public Class clsParallelArray
    Const ElemCount As Integer = 1000000

```

```

Private A(ElemCount) As Single
Private TaskCount As Integer '并行的任务个数

Public Sub New(TaskCount As Integer)
    Me.TaskCount = TaskCount
End Sub

' 顺序计算
Public Function RunSequence() As Integer
    Dim sw As Stopwatch = New Stopwatch()
    sw.Start() ' 启动计时
    Do1(0, ElemCount)
    sw.Stop() ' 停止计时
    Dim UsedTime As Integer = sw.ElapsedMilliseconds '获取算法执行时间
    Return UsedTime
End Function

' 并行计算
Public Function RunParallel() As Integer
    Dim sw As Stopwatch = New Stopwatch()
    sw.Start() ' 启动计时
    Parallel.For(0, TaskCount, AddressOf DoParallel) '将任务分成TaskCount个子任务：编号
为[0,TaskCount-1]
    sw.Stop() ' 停止计时
    Dim UsedTime As Integer = sw.ElapsedMilliseconds '获取算法执行时间
    Return UsedTime
End Function
Private Sub DoParallel(i As Integer)
    ' 根据子任务的编号i，计算子任务的参数
    Dim n1 As Integer = ElemCount / TaskCount ' 子任务计算的数据个数
    Dim si As Integer = i * n1 ' 子任务计算的起始下标
    '每个任务的计算范围A(si)...A(si+n1-1)
    Do1(si, n1) '任务之间的数据范围交叉时，不报错
End Sub

'对A(si)...A(si+n1-1)中的每个元素，重复执行1000次基本操作
Private Sub Do1(si As Integer, n1 As Integer)
    For I = si To si + n1 - 1
        For j = 0 To 1000
            A(I) += 1
        Next
    Next
End Sub
End Class

```

8.2 backgroundWorker

```

Imports System.ComponentModel

Public Structure WorkerPara
    Dim id As Integer
    Dim n1 As Integer

```

```

    Dim n2 As Integer
End Structure
Public Structure WorkerResult
    Dim id As Integer
    Dim sum As Single
End Structure

Public Class clsWorkerA
    Private para As WorkerPara
    Private Worker As BackgroundWorker

    Public Event Complete(Cancel As Boolean, result As WorkerResult)

    Public Sub New(para As WorkerPara)
        Me.para = para
        Me.Worker = New BackgroundWorker
        AddHandler Worker.DoWork, AddressOf DoWork
        AddHandler Worker.RunWorkerCompleted, AddressOf Completed
    End Sub
    Public Sub Run()
        Me.Worker.RunWorkerAsync(para)
    End Sub
    Public Sub DoCancel()
        Worker.CancelAsync()
    End Sub

    Private Sub DoWork(sender As Object, e As System.ComponentModel.DoWorkEventArgs)
        Dim sum As Single, k As Integer
        For k = 1 To 2000
            If Worker.CancellationPending Then
                e.Cancel = True : Return
            End If
            sum = 0
            For i = para.n1 To para.n2
                sum += i
            Next
        Next
        Dim result As WorkerResult
        result.id = para.id : result.sum = sum
        e.Result = result
    End Sub

    Private Sub Completed(sender As Object, e As RunWorkerCompletedEventArgs)
        Dim result As WorkerResult = e.Result
        RaiseEvent Complete(e.Cancelled, result)
    End Sub
End Class

```

```
Imports System.Threading.Tasks
```

```

Public Class clsParallelList
    Const ElemCount As Integer = 1000000

```

```

Private A As List(Of Single)
Private TaskCount As Integer '并行的任务个数

Public Sub New(TaskCount As Integer)
    A = New List(Of Single)
    For i = 0 To ElemCount
        A.Add(i)
    Next
    Me.TaskCount = TaskCount
End Sub

' 顺序计算
Public Function RunSequence() As Integer
    Dim sw As Stopwatch = New Stopwatch()
    sw.Start() '启动计时
    Do1(0, ElemCount)
    sw.Stop() '停止计时
    Dim UsedTime As Integer = sw.ElapsedMilliseconds '获取算法执行时间
    Return UsedTime
End Function

' 并行计算
Public Function RunParallel() As Integer
    Dim sw As Stopwatch = New Stopwatch()
    sw.Start() '启动计时
    Parallel.For(0, TaskCount, AddressOf DoParallel) '将任务分成TaskCount个子任务：编号
为[0,TaskCount-1]
    sw.Stop() '停止计时
    Dim UsedTime As Integer = sw.ElapsedMilliseconds '获取算法执行时间
    Return UsedTime
End Function
Private Sub DoParallel(i As Integer)
    ' 根据子任务的编号i，计算子任务的参数
    Dim n1 As Integer = ElemCount / TaskCount ' 子任务计算的数据个数
    Dim si As Integer = i * n1 ' 子任务计算的起始下标
    '每个任务的计算范围A(si)...A(si+n1-1)
    Do1(si, n1) '任务之间的数据范围交叉时，不报错
End Sub

'对A(si)...A(si+n1)中的每个元素，重复执行1000次基本操作
Private Sub Do1(si As Integer, n1 As Integer)
    For I = si To si + n1 - 1
        For j = 0 To 1000
            A(I) += 1
        Next
    Next
End Sub

End Class

```

```
Imports System.Threading.Tasks
```

```

Public Class clsParallelList
    Const ElemCount As Integer = 1000000
    Private A As List(Of Single)
    Private TaskCount As Integer '并行的任务个数

    Public Sub New(TaskCount As Integer)
        A = New List(Of Single)
        For i = 0 To ElemCount
            A.Add(i)
        Next
        Me.TaskCount = TaskCount
    End Sub

    ' 顺序计算
    Public Function RunSequence() As Integer
        Dim sw As Stopwatch = New Stopwatch()
        sw.Start() '启动计时
        Do1(0, ElemCount)
        sw.Stop() '停止计时
        Dim UsedTime As Integer = sw.ElapsedMilliseconds '获取算法执行时间
        Return UsedTime
    End Function

    ' 并行计算
    Public Function RunParallel() As Integer
        Dim sw As Stopwatch = New Stopwatch()
        sw.Start() '启动计时
        Parallel.For(0, TaskCount, AddressOf DoParallel) '将任务分成TaskCount个子任务：编号
        '为[0,TaskCount-1]
        sw.Stop() '停止计时
        Dim UsedTime As Integer = sw.ElapsedMilliseconds '获取算法执行时间
        Return UsedTime
    End Function

    Private Sub DoParallel(i As Integer)
        ' 根据子任务的编号i，计算子任务的参数
        Dim n1 As Integer = ElemCount / TaskCount '子任务计算的数据个数
        Dim si As Integer = i * n1 '子任务计算的起始下标
        '每个任务的计算范围A(si)...A(si+n1-1)
        Do1(si, n1) '任务之间的数据范围交叉时，不报错
    End Sub

    '对A(si)...A(si+n1)中的每个元素，重复执行1000次基本操作
    Private Sub Do1(si As Integer, n1 As Integer)
        For I = si To si + n1 - 1
            For j = 0 To 1000
                A(I) += 1
            Next
        Next
    End Sub
End Class

```


8.3 semaphore

8.4 委托类型的定义，委托对象的创建 (address of 函数名)

```
Public Sub New(para As WorkerPara, frmOut As frmTest)
    Me.para = para : Me.frmOut = frmOut
    Me.worker = New BackgroundWorker
    AddHandler worker.DoWork, AddressOf DoWork
    AddHandler worker.RunWorkerCompleted, AddressOf Completed
    Me.ReportProgress = New AppendMessage(AddressOf frmOut.AppendMessage)
End Sub
Private Sub Completed(sender As Object, e As RunWorkerCompletedEventArgs)
    Dim result As WorkerResult = e.Result
    RaiseEvent Complete(e.Cancelled, result)
End Sub
```

```
Public Class frmTest
    Public Sub AppendMessage(id As Integer, msg As String)
        If id = 1 Then
            lstProgress1.Items.Add(msg)
            lstProgress1.SelectedIndex = lstProgress1.Items.Count - 1
        End If
        If id = 2 Then
            lstProgress2.Items.Add(msg)
            lstProgress2.SelectedIndex = lstProgress2.Items.Count - 1
        End If
    End Sub
End Class
```

8.5 委托对象的调用(frm.invoke)、参数传递

```
Public Structure WorkerPara
    Dim id As Integer
    Dim n1 As Integer
    Dim n2 As Integer
End Structure
Public Structure WorkerResult
    Dim id As Integer
    Dim sum As Single
End Structure

Public Class clsWorkerA
    Delegate Sub AppendMessage(id As Integer, msg As String) '函数类型

    Private para As WorkerPara
```

```

Private frmOut As frmTest      '拥有控件的窗体，有权改写控件
Private Worker As BackgroundWorker
Private ReportProgress As AppendMessage    '函数对象，向界面报告计算的进度

Public Event Complete(sender As clsWorkerA, Cancel As Boolean, result As
workerResult)

Public Sub New(para As WorkerPara, frmOut As frmTest)
    Me.para = para : Me.frmOut = frmOut
    Me.Worker = New BackgroundWorker
    AddHandler Worker.DoWork, AddressOf DoWork
    AddHandler Worker.RunWorkerCompleted, AddressOf Completed
    Me.ReportProgress = New AppendMessage(AddressOf frmOut.AppendMessage)
End Sub
Public Sub Run()
    Me.Worker.RunWorkerAsync(para)
End Sub
Public Sub DoCancel()
    Worker.CancelAsync()
End Sub

Private Sub DoWork(sender As Object, e As System.ComponentModel.DoWorkEventArgs)
    Dim sum As Single, k As Integer
    sum = 0
    For k = 1 To 2000
        If Worker.CancellationPending Then
            e.Cancel = True : Return
        End If

        For i = para.n1 To para.n2
            sum += i
        Next
        '在拥有控件的窗口线程上，用指定的参数列表执行指定委托函数。
        frmOut.Invoke(ReportProgress, para.id, sum.ToString & vbTab & k)
    Next
    Dim result As workerResult
    result.id = para.id : result.sum = sum
    e.Result = result
End Sub

Private Sub Completed(sender As Object, e As RunWorkerCompletedEventArgs)
    Dim result As workerResult = e.Result
    RaiseEvent Complete(Me, e.Cancelled, result)
End Sub
End Class

```

9 智能输入

9.1 键盘事件与参数KeyPress KeyUp KeyDown

```
Public Class frmTest
```

```

Dim words() As String = {"abcde", "bed", "cash"}

Private Sub txtContent_KeyPress(sender As System.Object, e As
System.Windows.Forms.KeyPressEventArgs) Handles txtContent.KeyPress
    Dim lastchar As Char = txtContent.Text(txtContent.SelectionStart - 1) '上一个字
    Me.Text = lastchar
    Dim k As Integer = Find(e.KeyChar)
    If k <> -1 Then
        e.KeyChar = ""
        txtContent.SelectedText = words(k) & " "
    End If
End Sub

Function Find(keychar As Char) As Integer
    For i = 0 To words.Count - 1
        If words(i).Substring(0, 1) = keychar Then
            Return i
        End If
    Next
    Return -1
End Function

Private Sub txtContent_KeyDown(sender As System.Object, e As
System.Windows.Forms.KeyEventArgs) Handles txtContent.KeyDown

End Sub
End Class

```

9.2 键树的创建、查找

```

Imports System.IO

Public Class clsKeyTree
    Class Node
        Public nextc() As Node
        Public Sub New()
            ReDim nextc(25) '默认每个单元均为Nothing
        End Sub
    End Class

    Private Root As Node ' 键树的根节点

    Public Sub New()
        Root = New Node
        Dim content As String
        Using fs As New IO.FileStream("words.txt", FileMode.Open)
            Using sr As New StreamReader(fs, System.Text.Encoding.Default)
                content = sr.ReadToEnd()
            End Using
        End Using
        Dim separator() As Char = {vbLf, vbCr}
    End Sub

```

```

    Dim words() As String = content.Split(separator,
StringSplitOptions.RemoveEmptyEntries)
    For i = 0 To words.Count - 1
        InsertWord(words(i).ToLower)
    Next
End Sub

' 在Root所指的键树中插入单词word(全部由小写字母组成)
Private Sub InsertWord(ByVal word As String)
    Dim p As Node = Root
    For i = 0 To word.Length - 1
        ' 每个字符涉及1个结点,或许需要创建新结点
        Dim f As Integer = Asc(word(i)) - Asc("a")
        If p.nexttc(f) Is Nothing Then p.nexttc(f) = New Node
        p = p.nexttc(f)
    Next
End Sub

Public Function SearchbyPrefix(ByVal prefix As String) As List(Of String)
    Dim p As Node = SearchNodebyPrefix(prefix) ' 找到prefix的终点结点p
    Return TraverLeaf(p) ' 遍历p所指的26叉树,将所有叶子结点对应的
单词存入words
End Function

Private Function SearchNodebyPrefix(ByVal prefix As String) As Node
    Dim p As Node = Root
    For i = 0 To prefix.Length - 1
        Dim f As Integer = Asc(prefix(i)) - Asc("a")
        p = p.nexttc(f)
        If p Is Nothing Then Exit For
    Next
    Return p
End Function

Private Function TraverLeaf(p As Node) As List(Of String)
    Dim words As New List(Of String)
    Dim StackCode As Stack(Of Integer) = New Stack(Of Integer) ' 字符编码栈,用来组
织单词
    TraverLeaf(p, words, StackCode)
    Return words
End Function

Private Sub TraverLeaf(p As Node, words As List(Of String), StackCode As Stack(Of
Integer))
    If p Is Nothing Then Return
    ' 遍历当前结点的所有子树,并判别当前结点是否是叶子结点
    Dim isLeaf As Boolean = True
    For i = 0 To 25
        If p.nexttc(i) IsNot Nothing Then
            StackCode.Push(i)
            TraverLeaf(p.nexttc(i), words, StackCode)
            StackCode.Pop()
            isLeaf = False '不是叶子结点
        End If
    Next

```

```

        Next
        ' 若p是叶子结点
        If isLeaf = True Then
            If StackCode.Count > 0 Then
                words.Add(GetWord(StackCode))
            End If
        End If
    End Sub

    Private Function GetWord(StackCode As Stack(Of Integer)) As String
        Dim word As String = ""
        For i = 0 To StackCode.Count - 1
            word = Chr(Asc("a") + StackCode(i)) & word
        Next
        Return word
    End Function

End Class

```

10 操作历史

10.1 操作类的定义 (继承)、操作对象表的管理

```

Public Enum OpKind
    Insert
    Update
    Delete
End Enum

Public MustInherit Class clsOp
    Public MustOverride Sub DoWork(txtContent As TextBox)
End Class

'在position之前插入newchars
Public Class clsInsertOp
    Inherits clsOp
    Private position As Integer, newchars As String
    Public Sub New(position As Integer, newchars As String)
        Me.position = position : Me.newchars = newchars
    End Sub
    Public Overrides Sub DoWork(txtContent As System.Windows.Forms.TextBox)
        Dim left As String = txtContent.Text.Substring(0, position)
        Dim right As String = txtContent.Text.Substring(position)
        If newchars = vbCr Then
            txtContent.Text = left & vbCrLf & right
        Else
            txtContent.Text = left & newchars & right
        End If
    End Sub
    Public Overrides Function ToString() As String
        Return "Insert: " & position & vbTab & newchars
    End Function
End Class

```

'从position开始删除长度为length的字符串

```
Public Class clsDeleteOp
    Inherits clsOp
    Private position As Integer, length As Integer

    Public Sub New(position As Integer, length As Integer)
        Me.position = position : Me.length = length
    End Sub
    Public Overrides Sub DoWork(txtContent As System.Windows.Forms.TextBox)
        Dim left As String = txtContent.Text.Substring(0, position)
        Dim right As String = txtContent.Text.Substring(position + length)
        txtContent.Text = left & right
    End Sub
    Public Overrides Function ToString() As String
        Return "Delete: " & position & vbCrLf & length
    End Function
End Class
```

'从position开始替换长度为length的字符串，新字符串为

```
Public Class clsUpdateOp
    Inherits clsOp
    Private position As Integer, length As Integer, newchars As String

    Public Sub New(position As Integer, length As Integer, newchars As String)
        Me.position = position : Me.length = length : Me.newchars = newchars
    End Sub
    Public Overrides Sub DoWork(txtContent As System.Windows.Forms.TextBox)
        Dim left As String = txtContent.Text.Substring(0, position)
        Dim right As String = txtContent.Text.Substring(position + length)
        txtContent.Text = left & newchars & right
    End Sub
    Public Overrides Function ToString() As String
        Return "Update: " & position & vbCrLf & length & vbCrLf & newchars
    End Function
End Class
```

```
Public Class frmTest
    Dim Ops As List(Of clsOp)

    Private Sub frmTest_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        Ops = New List(Of clsOp)
    End Sub
    ' 识别Delete键
    Private Sub txtContent_KeyDown(sender As Object, e As System.Windows.Forms.KeyEventArgs) Handles txtContent.KeyDown
        If e.KeyCode = Keys.Delete Then
            Dim Op1 As clsOp = New clsDeleteOp(txtContent.SelectionStart, txtContent.SelectionLength)
            Ops.Add(Op1)
            lstOp.Items.Add(Op1.ToString())
        End If
    End Sub
End Class
```

```

End Sub
' 识别一般按键和BackSpace键
' 不能处理汉字，因为每个汉字引发两次事件
Private Sub txtContent_KeyPress(sender As Object, e As
System.Windows.Forms.KeyPressEventArgs) Handles txtContent.KeyPress
    Dim Op1 As clsOp
    If txtContent.SelectedText = "" Then
        If Asc(e.KeyChar) = Keys.Back Then
            Op1 = New clsDeleteOp(txtContent.SelectionStart - 1, 1)
        Else
            Op1 = New clsInsertOp(txtContent.SelectionStart, e.KeyChar)
        End If
    Else
        If Asc(e.KeyChar) = Keys.Back Then
            Op1 = New clsDeleteOp(txtContent.SelectionStart,
txtContent.SelectionLength)
        Else
            Op1 = New clsUpdateOp(txtContent.SelectionStart,
txtContent.SelectionLength, e.KeyChar)
        End If
    End If
    Ops.Add(Op1)
    lstOp.Items.Add(Op1.ToString)
End Sub

Private Sub btnStep_Click(sender As System.Object, e As System.EventArgs) Handles
btnStep.Click
    Static ith As Integer = 0
    If ith = 0 Then TextBox1.Text = ""
    Ops(ith).DoWork(TextBox1)
    ith += 1
    If ith = Ops.Count Then ith = 0
End Sub

Private Sub btnRedo_Click(sender As System.Object, e As System.EventArgs) Handles
btnRedo.Click
    TextBox1.Text = ""
    For i = 0 To Ops.Count - 1
        Ops(i).DoWork(TextBox1)
    Next
End Sub

End Class

```

10.2 Redo与Undo的机制（动态的应用）

```

Public Class clsOps
    Private txtContent As TextBox
    Private Ops As List(Of clsOp), ith As Integer

    Public Sub New(txtContent As TextBox)
        Me.txtContent = txtContent
    End Sub
End Class

```

```

        Me.Ops = New List(Of clsOp) : Me.ith = 0
    End Sub
    Public Sub Add(op As clsOp)
        Ops.Add(op)
        ith = Ops.Count
    End Sub

    Public Sub Undo()
        If ith = 0 Then Return
        Ops(ith - 1).Undo(txtContent)
        ith -= 1
    End Sub

    Public Sub Redo()
        If ith = Ops.Count Then Return
        Ops(ith).Dowork(txtContent)
        ith += 1
    End Sub
End Class

```

11 图像处理技术

11.1 Bitmap对象的像素计算（颜色）、常用运算（旋转、缩放）

```

Imports System.Drawing.Drawing2D

Public Class clsBitmap
    Private Source, Target As Bitmap
    Private picCanvas As PictureBox

    Public Sub New(picfname As String, picCanvas As PictureBox)
        Me.Source = New Bitmap(picfname) ' 在图像计算中，用作数据源
        Me.Target = New Bitmap(Me.Source)
        Me.picCanvas = picCanvas
        Me.picCanvas.Image = Target : Me.picCanvas.Width = Target.Width :
        Me.picCanvas.Height = Target.Height
    End Sub

    Public Sub Scale(ScaleX As Single, ScaleY As Single)
        Dim g As Graphics = Graphics.FromImage(Target)
        'g.InterpolationMode = InterpolationMode.High ' 设置高质量插值法
        'g.SmoothingMode = SmoothingMode.AntiAlias ' 消除锯齿
        Dim sourceRect As Rectangle = New Rectangle(0, 0, Source.Width, Source.Height)
        Dim w As Integer = Source.Width * ScaleX, h As Integer = Source.Height * ScaleY
        Dim destRect As Rectangle = New Rectangle(0, 0, w, h)
        g.DrawImage(Source, destRect, sourceRect, GraphicsUnit.Pixel)
        Me.picCanvas.Width = w : Me.picCanvas.Height = h
        Me.picCanvas.Refresh()
    End Sub

    Public Sub Rotate()
        Target.RotateFlip(RotateFlipType.Rotate90FlipNone) '顺时针旋转
        Me.picCanvas.Width = Target.Width : Me.picCanvas.Height = Target.Height
    End Sub
End Class

```



```

        Me.picCanvas.Refresh()
    End Sub

End Class

```

11.2 幻灯片技术 (Timer、矩阵的局部变换)

```

Public Enum SlideKind
    Left2Right
    Up2Down
End Enum

Public Class clsImage
    Private fname As String, Image As Bitmap
    Public Sub New(fname As String)
        Me.fname = fname
        Image = New Bitmap(fname)
    End Sub
    Public Sub Save(fname As String, format As Imaging.ImageFormat)
        Image.Save(fname, format)
    End Sub

    Public Sub Display(g As Graphics)
        g.DrawImage(Image, New Rectangle(0, 0, Image.Width, Image.Height))
    End Sub
    Public Sub Display(g As Graphics, destRect As Rectangle, sourceRect As Rectangle)
        g.DrawImage(Image, destRect, sourceRect, GraphicsUnit.Pixel)
    End Sub
    '3点决定一个平行四边形
    Public Sub Display(g As Graphics, p1 As Point, p2 As Point, p3 As Point)
        g.DrawImage(Image, {p1, p2, p3})
    End Sub

    ReadOnly Property Width() As Integer
        Get
            Return Image.Width
        End Get
    End Property
    ReadOnly Property Height() As Integer
        Get
            Return Image.Height
        End Get
    End Property

    Public Sub AddString(s As String, font As Font, p As Point)
        Dim g As Graphics = Graphics.FromImage(Me.Image)
        g.DrawString(s, font, Brushes.Red, p)
        g.DrawLine(Pens.Black, 0, 0, 100, 100)
        g.Dispose() '释放
    End Sub

    Public Sub Slide(g As Graphics, kind As SlideKind, blockCount As Integer,
        TimeInterval As Integer)

```

```

    Dim Timer As clsTimer
    Timer = New clsTimer(g, Image, Kind, blockCount, TimeInterval)
    Timer.Slide()
End Sub
End Class

```

```

Public Class clsTimer
    Private g As Graphics
    Private Image As Bitmap
    Private Kind As SlideKind, BlockCount As Integer, TimeInterval As Integer
    Private Timer As Timer

    Public Sub New(g As Graphics, Image As Bitmap, Kind As SlideKind, blockCount As
Integer, TimeInterval As Integer)
        Me.g = g
        Me.Image = Image
        Me.Kind = Kind
        Me.BlockCount = blockCount
        Me.Timer = New Timer
        Timer.Enabled = False
        Timer.Interval = TimeInterval
        AddHandler Timer.Tick, AddressOf TimerTick
    End Sub

    Sub Slide()
        Timer.Enabled = True
    End Sub

    Private Sub TimerTick(sender As Timer, e As EventArgs)
        Static iBlock As Integer = 0
        If iBlock > BlockCount Then
            Timer.Enabled = False : Timer = Nothing
            Return
        End If
        '显示image的第iBlock块的图像
        Dim rect As Rectangle
        If Kind = SlideKind.Up2Down Then
            Dim dy As Integer = Image.Height / BlockCount
            rect = New Rectangle(0, dy * iBlock, Image.Width, dy)
        End If
        If Kind = SlideKind.Left2Right Then
            Dim dx As Integer = Image.Width / BlockCount
            rect = New Rectangle(dx * iBlock, 0, dx, Image.Height)
        End If
        g.DrawImage(Image, rect, rect, GraphicsUnit.Pixel)
        iBlock += 1
    End Sub
End Class

```

11.3 图像的图块计算 (图的遍历)

```

Class clsRegions

```

```

Private Regions As List(Of clsRegion)

Public Sub New()
    Regions = New List(Of clsRegion)
End Sub

Sub New(ByVal Pixels As Color(), ByVal th As Integer)
    Regions = New List(Of clsRegion)
    Dim width As Integer = Pixels.GetLength(0), height As Integer =
Pixels.GetLength(1)
    Dim Visited(width - 1, height - 1) As Boolean
    For i = 0 To width - 1
        For j = 0 To height - 1
            Visited(i, j) = False
        Next
    Next
    For i = 0 To width - 1
        For j = 0 To height - 1
            If Pixels(i, j).R < th And Visited(i, j) = False Then
                Dim Region As clsRegion = Bfs(Pixels, i, j, th, Visited)
                Regions.Add(Region)
            End If
        Next
    Next
End Sub

Private Function Bfs(ByVal Pixels As Color(), ByVal i As Integer, ByVal j As
Integer, ByVal th As Integer, ByVal visited As Boolean()) As clsRegion
    Dim width As Integer = Pixels.GetLength(0), height As Integer =
Pixels.GetLength(1)
    Dim dx() As Integer = {-1, 1, 0, 0}, dy() As Integer = {0, 0, -1, 1}
    Dim Region As New clsRegion
    Dim Q As New Queue(Of Point)
    Q.Enqueue(New Point(i, j)) : Visited(i, j) = True
    While Q.Count > 0
        Dim p As Point = Q.Dequeue() : Region.Append(p)
        For k = 0 To 3
            i = p.X + dx(k) : j = p.Y + dy(k)
            If i < 0 Or i >= width Then Continue For
            If j < 0 Or j >= height Then Continue For
            If Pixels(i, j).R < th And Visited(i, j) = False Then
                Q.Enqueue(New Point(i, j)) : Visited(i, j) = True
            End If
        Next
    End while
    Return Region
End Function

Public Sub Append(ByVal r As clsRegion)
    Regions.Add(r)
End Sub

ReadOnly Property Count As Integer
    Get
        Return Regions.Count
    End Get
End Property

```

```

        End Get
    End Property
    Function Item(ByVal index As Integer) As clsRegion
        Return Regions(index)
    End Function

    Public Sub Display()
        For i = 0 To Regions.Count - 1
            Regions(i).Display()
        Next
    End Sub

    Function GetBuff(ByVal width As Integer, ByVal Height As Integer, ByVal bkcolor As
Color, ByVal ftcolor As Color) As Bitmap
        Dim newbuff As New Bitmap(width, Height)
        For i = 0 To width - 1
            For j = 0 To Height - 1
                newbuff.SetPixel(i, j, bkcolor)
            Next
        Next
        For i = 0 To Regions.Count - 1
            For j = 0 To Regions(i).Count - 1
                Dim p As Point = Regions(i).Item(j)
                newbuff.SetPixel(p.X, p.Y, ftcolor)
            Next
        Next
        Return newbuff
    End Function
End Class

```

```

Public Class clsRegion
    Private Region As List(Of Point)

    Public Sub New()
        Region = New List(Of Point)
    End Sub
    Public Sub Append(ByVal p As Point)
        Region.Add(p)
    End Sub
    ReadOnly Property Count As Integer
        Get
            Return Region.Count
        End Get
    End Property
    Function Item(ByVal index As Integer) As Point
        Return Region(index)
    End Function

    Public Sub Display()
        For i = 0 To Region.Count - 1
            ' Region(i)
        Next
    End Sub

```

```
End Class
```

12 图形技术

12.1 图形类中的继承设计、多态效果

```
Public Class clsVertex
    Public p As PointF
    Function SelectbyPoint(ByVal p As PointF) As Boolean

    End Function
End Class

Public Class clsVertexs
    Private Vs As List(Of clsVertex)
    Private Selects As List(Of clsVertex)
    Public Event SelectChanged()
    Sub SelectbyPoint(ByVal p As PointF)
        RaiseEvent SelectChanged()
    End Sub

End Class

Public MustInherit Class clsElem
    MustOverride Sub Draw(ByVal g As Graphics)
End Class

Public Class clsLine
    Inherits clsElem
    Private sp As clsVertex, ep As clsVertex
    Public Overrides Sub Draw(ByVal g As System.Drawing.Graphics)
    End Sub
End Class

Public Class clsCircle
    Inherits clsElem
    Private cp As clsVertex
    Public Overrides Sub Draw(ByVal g As System.Drawing.Graphics)
    End Sub
End Class

Public Class clsElems
    Private WithEvents Vs As clsVertexs
    Private Elems As List(Of clsElem)
    Public Event ElemChanged()

    Private Sub Vs_SelectChanged() Handles Vs.SelectChanged
        RaiseEvent ElemChanged()
    End Sub
    Sub Draw(ByVal g As Graphics)
```

```
End Sub
End Class
```

12.2 图形对象的管理：自定义的事件Event、引发事件RaiseEvent、处理事件

```
Public Class clsBitmaps
    Private path As String
    Private Bitmaps As List(Of Bitmap)
    Public N As Integer
    Private TaskCount As Integer = 4

    Event SaveProgress(ByVal ith As Integer)

    Public Sub New(ByVal path As String)
        Me.path = path
        Bitmaps = New List(Of Bitmap)
        Dim files() As String = Directory.GetFiles(path)
        N = files.Count
        For i = 0 To N - 1
            Bitmaps.Add(New Bitmap(files(i)))
        Next
    End Sub

    Public Function GetBitmap(ByVal index As Integer) As Bitmap
        Return Bitmaps(index)
    End Function

    Public Sub Save(ByVal path As String, ByVal si As Integer, ByVal ei As Integer)
        For i = si To ei - 1
            Bitmaps(i).Save(path & "xy" & i & ".bmp", Imaging.ImageFormat.Bmp)
            RaiseEvent SaveProgress(i)
        Next
    End Sub

End Class
```

```
Public Class frmTest
    Dim path As String
    Dim WithEvents Bitmaps As clsBitmaps
    Dim Bkworker As BackgroundWorker

    Private Sub frmTest_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        path = My.Application.Info.DirectoryPath
        path = path & "\Tif\"
    End Sub
```

```

Private Sub btnOpen_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOpen.Click
    Bitmaps = New clsBitmaps(path)
End Sub

Private Sub tbXY_ValueChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles tbXY.ValueChanged
    picCanvasXY.Image = Bitmaps.GetBitmap(tbXY.Value)
End Sub

Structure bkworkerPara
    Public si As Integer
    Public ei As Integer
End Structure

' 在漫长的保存过程中，界面能够响应事件!!!
Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    pbProgress2.TaskCount = Bitmaps.N

    path = My.Application.Info.DirectoryPath
    path = path & "\Tif\Bitmaps\"
    Dim workerCount As Integer = 10
    Dim para As bkworkerPara
    For i = 0 To WorkerCount - 1
        Bkworker = New BackgroundWorker
        AddHandler Bkworker.Dowork, AddressOf Bkworker_Dowork
        para.si = i * Bitmaps.N / WorkerCount
        para.ei = (i + 1) * Bitmaps.N / WorkerCount
        Bkworker.WorkerReportsProgress = True
        Bkworker.WorkerSupportsCancellation = True
        Bkworker.RunWorkerAsync(para)
    Next
End Sub

' 线程之间的通讯与动作
Delegate Sub deleSetProgressValue(ByVal value As Integer)
Private Sub SetProgressValue(ByVal value As Integer)
    pbProgress2.SetValue(value)
End Sub
Private Sub Bitmaps_SaveProgress(ByVal ith As Integer) Handles Bitmaps.SaveProgress
    Dim f As deleSetProgressValue = New deleSetProgressValue(AddressOf
SetProgressValue)
    ' 在拥有控件的基础窗口句柄的线程上，用指定的参数列表执行指定委托。
    Invoke(f, {ith})
End Sub

Private Sub Bkworker_Dowork(ByVal sender As Object, ByVal e As
System.ComponentModel.DoworkEventArgs)
    Dim para As bkworkerPara = e.Argument
    Bitmaps.Save(path, para.si, para.ei)
End Sub

```

End Class

12.3 图形对象的选择技术

```
Public Class clsPolygon
    Inherits clsElem
    Private Points As List(Of PointF)

    Public Sub New(Points As List(Of PointF))
        Me.Points = Points
    End Sub

    Public Overrides Sub Draw(g As Graphics)
        Dim myPen As New Pen(Color.Red, 1)
        g.DrawPolygon(myPen, Points.ToArray)
    End Sub

    Public Overloads Overrides Sub Draw(ByVal g As System.Drawing.Graphics, ByVal color As System.Drawing.Color)
        Dim pen As Pen = New Pen(color, 1)
        g.DrawPolygon(pen, Points.ToArray)
    End Sub

    Public Overrides Function ToString() As String
        Dim s As String = "Polygon: "
        For i = 0 To Points.Count - 1
            s = s & Points(i).ToString & " "
        Next
        Return s
    End Function

    Public Overrides Function SelectByPoint(p As PointF) As Boolean
        Dim sp, ep As PointF
        sp = Points(Points.Count - 1)
        For i = 0 To Points.Count - 1
            ep = Points(i)
            If clsLine.SelectByPoint(p, sp, ep) = True Then Return True
            sp = ep
        Next
        Return False
    End Function
End Class
```

12.4 图形对象的动态编辑：橡皮筋、拖拽