文件

文本文件的读写方法？？

XML文件的树形结构的创建、查找

|  |  |
| --- | --- |
| 创建节点 | Public Class clsLight  Public Name As String  Public Position, Ambient, Diffuse, 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 |
| 创建树形结构 | 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 |
| 查找 | 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 |

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

|  |  |
| --- | --- |
|  | Public Sub New(proc As Process)  Me.Process = proc  ProcessName = proc.ProcessName  Try  FileName = proc.MainModule.FileName  Catch ex As Exception  FileName = ""  End Try  Me.item = New ListViewItem(ProcessName, 0)  item.SubItems.Add(FileName)  End Sub  Public Function GetThread() As ProcessThreadCollection  Return Process.Threads  End Function |
|  | Public Class clsProcesses  Private Processes As List(Of clsProcess)  Public Items() As ListViewItem  Public Sub New()  Processes = New List(Of clsProcess)  Dim Proces() As Process = Process.GetProcesses()  For i = 0 To Proces.Count - 1  Processes.Add(New clsProcess(Proces(i)))  Next  ReDim Items(Proces.Count - 1)  For i = 0 To Proces.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 Proces() As Process = Process.GetProcesses()  For i = 0 To Proces.Count - 1  Try  If Proces(i).MainModule.FileName = FileName Then  Proces(i).Kill()  End If  Catch ex As Exception  End Try  Next  End Sub  End Class |
|  | CPU:%Processor Time, %Priviliaged Time, %User Time, %DPC Time  Memory:Available Bytes, Pages/sec, Pages Input/sec, Pages Output/sec, Page Faults/sec, Private Bytes  Disk:PhysicalDisk\Avg.Disk sec/Read(Write), Physical Disk\Disk Reads(Writes)/sec |
|  | ' Pages/sec 是指为解决硬页错误从磁盘读取或写入磁盘的速度。  ' 这个计数器是可以显示导致系统范围延缓类型错误的主要指示器。  ' 它是 Memory\\Pages Input/sec 和 Memory\\Pages Output/sec 的总和。  ' 是用页数计算的，以便在不用做转换的情况下就可以同其他页计数如: Memory\\Page Faults/sec 做比较，  ' 这个值包括为满足错误而在文件系统缓存(通常由应用程序请求)的非缓存映射内存文件中检索的页。  ' Memory\_Pages.CounterType: RateOfCountsPerSecond32  ' 差异计数器，它显示在采用间隔的每一秒内完成的操作的平均数目。这种类型的计数器用系统时钟的滴答 (Tick) 来测量时间。  Dim Memory\_AvaiableMBytes As PerformanceCounter  Dim QAvailableMBytes As Queue(Of Long)  Dim Memory\_Pages, Memory\_InputPages, Memory\_OutputPages As PerformanceCounter  Dim LastMemoryPages\_RawValue, LastMemoryInputPages\_RawValue, LastMemoryOutputPages\_RawValue As Long  Dim QPages As Queue(Of Long)  Private Sub frmTest\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load  QAvailableMBytes = New Queue(Of Long)  Memory\_AvaiableMBytes = New PerformanceCounter("Memory", "Available MBytes")  QPages = New Queue(Of Long)  Memory\_Pages = New PerformanceCounter("Memory", "Pages/sec")  Memory\_InputPages = New PerformanceCounter("Memory", "Pages Input/sec")  Memory\_OutputPages = New PerformanceCounter("Memory", "Pages Output/sec")  LastMemoryPages\_RawValue = Memory\_Pages.RawValue  LastMemoryInputPages\_RawValue = Memory\_InputPages.RawValue  LastMemoryOutputPages\_RawValue = Memory\_OutputPages.RawValue  Timer1.Interval = 100 : Timer1.Enabled = True  End Sub  Private Sub Timer1\_Tick(sender As System.Object, e As System.EventArgs) Handles Timer1.Tick  lblMemoryAvailableMBytes.Text = Memory\_AvaiableMBytes.RawValue  QAvailableMBytes.Enqueue(lblMemoryAvailableMBytes.Text)  DisplayQAvaiableMBytes()  lblMemory\_Pages.Text = Memory\_Pages.RawValue - LastMemoryPages\_RawValue  lblMemory\_InputPages.Text = Memory\_InputPages.RawValue - LastMemoryInputPages\_RawValue  lblMemory\_OutputPages.Text = Memory\_OutputPages.RawValue - LastMemoryOutputPages\_RawValue  QPages.Enqueue(lblMemory\_Pages.Text)  If QPages.Count > 100 Then  QPages.Dequeue()  End If  LastMemoryPages\_RawValue = Memory\_Pages.RawValue  LastMemoryInputPages\_RawValue = Memory\_InputPages.RawValue  LastMemoryOutputPages\_RawValue = Memory\_OutputPages.RawValue  DisplayQPages()  End Sub  Sub DisplayQAvaiableMBytes()  Me.Chart1.Series(0).ChartType = SeriesChartType.Line  Me.Chart1.Series("Series1").Points.Clear()  For i = 0 To QAvailableMBytes.Count - 1  Me.Chart1.Series("Series1").Points.AddXY(i, QAvailableMBytes(i))  Next  Chart1.Legends("Legend1").Docking = Docking.Top  End Sub  Sub DisplayQPages()  Me.Chart2.Series(0).ChartType = SeriesChartType.Line  Me.Chart2.Series("Series1").Points.Clear()  For i = 0 To QPages.Count - 1  Me.Chart2.Series("Series1").Points.AddXY(i, QPages(i))  Next  Chart2.Legends("Legend1").Docking = Docking.Top  End Sub |

项目管理

参数、资源、XML资源

|  |  |
| --- | --- |
|  | Private Sub DisplaySettings()  txtUserName.Text = My.Settings.UserName  Me.BackColor = My.Settings.BkColor  End Sub  Private Sub btnChange\_Click(sender As System.Object, e As System.EventArgs) Handles btnChange.Click  Dim dlg As New ColorDialog  If dlg.ShowDialog() <> Windows.Forms.DialogResult.OK Then Return  My.Settings.BkColor = dlg.Color  DisplaySettings()  End Sub  Private Sub txtUserName\_TextChanged(sender As System.Object, e As System.EventArgs) Handles txtUserName.TextChanged  My.Settings.UserName = txtUserName.Text  End Sub |
|  | Dim WithEvents ticker As Timer  Private Sub frmTest\_Load(sender As Object, e As System.EventArgs) Handles Me.Load  ticker = New Timer  ticker.Interval = 100  ticker.Enabled = True  End Sub  Private Sub ticker\_Tick(sender As Object, e As System.EventArgs) Handles ticker.Tick  Dim Icons() As Icon = {My.Resources.A1, My.Resources.A2, My.Resources.A3, My.Resources.A4}  Static ith As Integer = 0  Me.Icon = Icons(ith)  ith = (ith + 1) Mod 4  End Sub  Private Sub btnXML\_Click(sender As System.Object, e As System.EventArgs) Handles btnXML.Click  Dim XML As New XmlDocument  XML.LoadXml(My.Resources.ConfigMaterial)  Dim node As System.Xml.XmlElement  node = XML.FirstChild.SelectSingleNode("Model")  Dim ambient As String = node.SelectSingleNode("ambient").InnerXml()  MsgBox(ambient)  End Sub |
|  |  |

窗口管理

MDI主/子窗体

菜单

上下文菜单动态创建、显示

|  |  |
| --- | --- |
|  | 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) 层叠排列  If sender Is m\_Cascade Then Me.LayoutMdi(MdiLayout.Cascade)  If sender Is m\_TileHorizontal Then Me.LayoutMdi(MdiLayout.TileHorizontal) 水平排列  If sender Is m\_TileVertical Then Me.LayoutMdi(MdiLayout.TileVertical) 垂直排列  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 |
|  | ' 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 m\_FileNew\_Click(sender As System.Object, e As System.EventArgs) Handles m\_FileNew.Click, m\_FileExit.Click  If sender Is m\_FileNew Then FileNew()  If sender Is m\_FileExit Then FileExit()  End Sub  Private Sub FileNew()  MsgBox("FileNew")  End Sub  Private Sub FileExit()  Me.Close()  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 |
|  | 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 |

资源管理器

文件、进程、设备系统的Watcher ？

|  |  |
| --- | --- |
|  | 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)  lstFile.Items.Clear()  For i = 0 To files.Count - 1  lstFile.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)  lstFile.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)  lstFile.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 = lstFile.Items.IndexOf(oldfilename)  lstFile.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 |
|  | Enum OpKind  Creation  Modification  Deletion  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("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 |
|  | Enum OpKind  Creation  Modification  Deletion  End Enum  Public WithEvents CreationWatcher As ManagementEventWatcher ' 用于观察设备新建的事件  Public WithEvents DeletionWatcher As ManagementEventWatcher ' 用于观察设备删除的事件  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\_LogicalDisk""")  Dim qDeletion As EventQuery = New EventQuery("SELECT \* FROM \_\_InstanceDeletionEvent WITHIN 1 WHERE TargetInstance isa ""Win32\_LogicalDisk""")  ' 建立观察对象  CreationWatcher = New ManagementEventWatcher(qCreation)  DeletionWatcher = New ManagementEventWatcher(qDeletion)  End Sub  Private Sub btnBrowse\_Click(sender As System.Object, e As System.EventArgs) Handles btnStart.Click  lstCreation.Items.Clear() : lstDeletion.Items.Clear()  ' 异步开始侦听  CreationWatcher.Start() : DeletionWatcher.Start()  End Sub  Private Sub btnStop\_Click(sender As System.Object, e As System.EventArgs) Handles btnStop.Click  CreationWatcher.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 DeletionWatcher\_EventArrived(sender As Object, e As System.Management.EventArrivedEventArgs) Handles DeletionWatcher.EventArrived  Dim Process As ManagementBaseObject = e.NewEvent("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() : 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.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 |

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

parallel.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

|  |  |
| --- | --- |
|  | parallel.for(int fromInculsive, int toExclusive, Action<int> body) |
|  | bacckgroundWorker  方法 RunWorkerAsync(parament) //开始执行后台操作  CancelAsync() //请求取消的挂起的后台操作  属性 CancellationPending //指示应用程序是否已请求取消后台操作  事件 DoWork //操作开始时在另一个线程上运行的事件处理程序  RunWorkerCompleted //当辅助线程完成(无论是成功、失败还是取消)时引发 |
|  |  |
|  |  |
|  |  |

智能输入

键盘事件与参数KeyPress KeyUp KeyDown

键树的创建、查找

|  |  |
| --- | --- |
|  | 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) '上一个字符  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 |
|  | 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}  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.nextc(f) Is Nothing Then p.nextc(f) = New Node  p = p.nextc(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.nextc(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.nextc(i) IsNot Nothing Then  StackCode.Push(i)  TraverLeaf(p.nextc(i), words, StackCode)  StackCode.Pop()  isLeaf = False '不是叶子结点  End If  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 |

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）

|  |  |
| --- | --- |
|  | Public Enum OpKind  Insert  Update  Delete  End Enum  Public MustInherit Class clsOp  Public MustOverride Sub DoWork(txtContent As TextBox)  Public MustOverride Sub UnDo(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 Sub UnDo(txtContent As System.Windows.Forms.TextBox)  Dim left As String = txtContent.Text.Substring(0, position)  Dim right As String  If newchars = vbCr Then  right = txtContent.Text.Substring(position + 2)  Else  right = txtContent.Text.Substring(position + 1)  End If  txtContent.Text = left & right  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  Private deletetxt As String ' 删除的文本  Public Sub New(position As Integer, length As Integer, deletetxt As String)  Me.position = position : Me.length = length  Me.deletetxt = deletetxt  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 Sub UnDo(txtContent As System.Windows.Forms.TextBox)  Dim left As String = txtContent.Text.Substring(0, position)  Dim right As String = txtContent.Text.Substring(position)  txtContent.Text = left & deletetxt & right  End Sub  Public Overrides Function ToString() As String  Return "Delete: " & position & vbTab & 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  Private oldchars As String, newchars As String ' 替换前的旧文本, 替换后的新文本  Public Sub New(position As Integer, length As Integer, oldchars As String, newchars As String)  Me.position = position : Me.length = length  Me.oldchars = oldchars : 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 Sub UnDo(txtContent As System.Windows.Forms.TextBox)  Dim left As String = txtContent.Text.Substring(0, position)  Dim right As String = txtContent.Text.Substring(position + newchars.Length)  txtContent.Text = left & oldchars & right  End Sub  Public Overrides Function ToString() As String  Return "Update: " & position & vbTab & length & vbTab & newchars  End Function  End Class |

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）

|  |  |
| --- | --- |
|  | Dim Fname As String  Dim Source, Target As Bitmap  Private Sub btnOpen\_Click(sender As System.Object, e As System.EventArgs) Handles btnOpen.Click  Fname = "Tulips.jpg"  Source = New Bitmap(Fname) ' 在图像计算中，用作数据源  Target = New Bitmap(Source) ' 用于显示  picCanvas.Image = Target 'Bitmap是Image(虚类)的子类  picCanvas.Width = Target.Width  picCanvas.Height = Target.Height  End Sub  Private Sub picCanvas\_MouseMove(sender As System.Object, e As System.Windows.Forms.MouseEventArgs) Handles picCanvas.MouseMove  If Source Is Nothing Then Exit Sub  Dim p As Point = e.Location  Dim pc As Color = Target.GetPixel(p.X, p.Y)  txtRed.Text = pc.R  txtGreen.Text = pc.G  txtBlue.Text = pc.B  ' “色调-饱和度-亮度”(HSB)  Dim hue As Single = pc.GetHue() ' 色调值，以度为单位[0,360]。  Dim saturation As Single = pc.GetSaturation() ' 饱和度值[0,1]。  Dim brightness As Single = pc.GetBrightness() ' 亮度值[0,1]。  txtHue.Text = hue  txtSaturation.Text = saturation  txtBrightness.Text = brightness  ' CMY相减混色模式  Dim C1 As Integer = 255 - pc.R  Dim M1 As Integer = 255 - pc.G  Dim Y1 As Integer = 255 - pc.B  txtC1.Text = C1  txtM1.Text = M1  txtY1.Text = Y1  ' CMYK印刷四色模式  Dim K As Integer = Math.Min(C1, Math.Min(M1, Y1))  Dim C As Integer = C1 - K  Dim M As Integer = M1 - K  Dim Y As Integer = Y1 - K  txtC.Text = C  txtM.Text = M  txtY.Text = Y  txtK.Text = K  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  Me.picCanvas.Refresh()  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  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 |
|  | ' Buff中Mark的连通分量（区域）  Public Class clsRegion  Private Points As List(Of Point) '区域内部所有的点  Public Sub New(Points As List(Of Point))  Public Sub Draw(Buff As Bitmap, Mark As Color)  Public Function PointCount() As Integer  Public Class clsRegions  Private Regions As List(Of clsRegion)  Public Sub New()  Public Sub Add(Region As clsRegion)  Public Sub Draw(buff As Bitmap, mark As Color)  Public Sub FillList(List As ListBox)  Public Function Item(index) As clsRegion  Public SourceBuff, TargetBuff As Bitmap  Private Mark As Color '区域中的点的表示色  Private Points As List(Of Point)  Public Sub New(fname As String, red As Tuple(Of Integer, Integer), green As Tuple(Of Integer, Integer), blue As Tuple(Of Integer, Integer), Mark As Color)  Me.SourceBuff = New Bitmap(fname)  Me.TargetBuff = New Bitmap(Me.SourceBuff.Width, Me.SourceBuff.Height)  Me.Mark = Mark  ' 将SourceBuff中符合条件的像素值，改为Mark色，写入TargetBuff  Dim c As Color  For x = 0 To Me.SourceBuff.Width - 1  For y = 0 To Me.SourceBuff.Height - 1  c = Me.SourceBuff.GetPixel(x, y)  If c.R >= red.Item1 And c.R <= red.Item2 And c.G >= green.Item1 And c.G <= green.Item2 And c.B >= blue.Item1 And c.B <= blue.Item2 Then  TargetBuff.SetPixel(x, y, Mark)  Else  TargetBuff.SetPixel(x, y, Color.White)  End If  Next  Next  End Sub  '在TargetBuff中，根据Mark识别、创建区域，保存到区域集合对象中  Function GetRegions() As clsRegions  Dim Regions As clsRegions = New clsRegions()  Dim Width As Integer = TargetBuff.Width, Height As Integer = TargetBuff.Height  Dim flagMatrix(Width - 1, Height - 1) As Boolean '访问标记数组  For x = 0 To Width - 1  For y = 0 To Height - 1  flagMatrix(x, y) = False '未访问过  Next  Next  Dim p As Point  For x = 0 To Width - 1  For y = 0 To Height - 1  If TargetBuff.GetPixel(x, y).ToArgb <> Mark.ToArgb Then Continue For  If flagMatrix(x, y) = True Then Continue For '已经访问过  p.X = x : p.Y = y  '以p为起点，对相邻的Mark色的像素点进行层次遍历，同时修改flagMatrix(x, y)为True  Dim PointsInRegion As List(Of Point) = Bfs(TargetBuff, p, flagMatrix) '该区域中所有点的集合  Regions.Add(New clsRegion(PointsInRegion))  Next  Next  Return Regions  End Function  ' 在Buff中，以sp为起点，查找sp所在的连通分量，返回其中的所有点的集合  Private Function Bfs(Buff As Bitmap, sp As Point, flagMatrix(,) As Boolean) As List(Of Point)  Dim region As List(Of Point) = New List(Of Point)()  Dim Width As Integer = Buff.Width, Height As Integer = Buff.Height  Dim Q As New Queue(Of Point)  Q.Enqueue(sp) : flagMatrix(sp.X, sp.Y) = True  Dim p, adjPoints(3) As Point  While Q.Count > 0  p = Q.Dequeue() : region.Add(p)  adjPoints(0) = New Point(p.X - 1, p.Y) ' 左右邻接点  adjPoints(1) = New Point(p.X + 1, p.Y)  adjPoints(2) = New Point(p.X, p.Y - 1) ' 上下邻接点  adjPoints(3) = New Point(p.X, p.Y + 1)  For i = 0 To 3  If adjPoints(i).X < 0 Or adjPoints(i).X >= Width Then Continue For ' 越界检查  If adjPoints(i).Y < 0 Or adjPoints(i).Y >= Height Then Continue For ' 越界检查  If Buff.GetPixel(adjPoints(i).X, adjPoints(i).Y).ToArgb <> Mark.ToArgb Then Continue For ' 点颜色检查  If flagMatrix(adjPoints(i).X, adjPoints(i).Y) = True Then Continue For ' 访问标记检查  Q.Enqueue(adjPoints(i)) : flagMatrix(adjPoints(i).X, adjPoints(i).Y) = True  Next  End While  Return region  End Function |

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

|  |  |
| --- | --- |
|  | Public MustInherit Class clsElem  Protected Pen As Pen  Public MustOverride Sub Draw(ByVal g As Graphics)  Public Sub New()  End Class  Public Class clsElems  Public elems As List(Of clsElem)  Event Redraw()  Public Sub New()  Public Sub Append(e As clsElem)  Public Sub Draw(ByVal g As Graphics)  Public Function Count() As Integer  Public Function GetElem(index As Integer) As clsElem  End Class  Class clsLine  Inherits clsElem  Private sp, ep As PointF  Public Sub New(ByVal sp As PointF, ByVal ep As PointF)  Public Overrides Sub Draw(g As Graphics)  g.DrawLine(Pen, sp, ep)  End Sub  Public Overrides Function ToString() As String  Dim s As String = "Line "  s = s & ": " & sp.ToString & "," & ep.ToString  Return s  End Function  End Class  Public Class clsRectangle  Inherits clsElem  Private sp, ep As PointF  Private points(3) As PointF  Public Sub New(ByVal sp As PointF, ByVal ep As PointF)  Private Sub Init()  Dim xmin, xmax, ymin, ymax As Single  If sp.X < ep.X Then  xmin = sp.X : xmax = ep.X  Else  xmin = ep.X : xmax = sp.X  End If  If sp.Y < ep.Y Then  ymin = sp.Y : ymax = ep.Y  Else  ymin = ep.Y : ymax = sp.Y  End If  points(0) = New PointF(xmin, ymin)  points(1) = New PointF(xmax, ymin)  points(2) = New PointF(xmax, ymax)  points(3) = New PointF(xmin, ymax)  End Sub  Public Overrides Sub Draw(ByVal g As Graphics)  Public Overrides Function ToString() As String  End Class |
|  | ' 对图形对象的所有操作，都是经过选集对象进行的  Public Class clsSelection  Private Elems As List(Of clsElem) ' 被选择的图形对象的集合  Public Sub New()  Public Sub Clear()  ' 对选集的管理：添加、移除、判断  Public Sub Append(ByVal Elem As clsElem) '向选集中添加Elem  Public Sub Remove(ByVal Elem As clsElem) '从选集中移除Elem  '若Elem在选集中，则Elem移除选集；若Elem不在选集中，则将Elem加入选集  Public Sub DoSelect(ByVal Elem As clsElem)  Function Contains(ByVal elem As clsElem) As Boolean  Public Function Count() As Integer  End Class  Enum ToolKind  Line  Rectangle  Polygon  PointSelect  End Enum  Dim Tool As ToolKind  Dim WithEvents Elems As clsElems  Dim sp As PointF, sps As List(Of PointF) ' sp用于存储上一个点, sps用于存储上当前点序列  Private Sub frmTest\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load  Private Sub picCanvas\_MouseDown(sender As Object, e As System.Windows.Forms.MouseEventArgs) Handles picCanvas.MouseDown  Private Sub picCanvas\_MouseUp(ByVal sender As System.Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles picCanvas.MouseUp  Dim ep As PointF = e.Location  If Tool = ToolKind.Line Then 'ToolKind.REctangle  Elems.Append(New clsLine(sp, ep))  lstElem.Items.Add(Elems.Item(Elems.Count - 1).ToString)  Return  End If  If Tool = ToolKind.Polygon Then  sps.Add(ep)  If e.Button = Windows.Forms.MouseButtons.Right Then  Elems.Append(New clsPolygon(sps))  sps.Clear()  lstElem.Items.Add(Elems.Item(Elems.Count - 1).ToString)  End If  Return  End If  If Tool = ToolKind.PointSelect Then  Elems.SelectByPoint(ep)  End If  End Sub  Private Sub Elems\_SelectionChanged() Handles Elems.SelectionChanged  lstElem.SelectedIndices.Clear()  For i = 0 To Elems.Count - 1  If Elems.Selection.Contains(Elems.Item(i)) Then  lstElem.SelectedIndices.Add(i)  End If  Next  Dim g As Graphics = picCanvas.CreateGraphics()  Elems.Draw(g)  End Sub  Private Sub Elems\_Redraw() Handles Elems.Redraw  Private Sub picCanvas\_Paint(sender As System.Object, e As System.Windows.Forms.PaintEventArgs) Handles picCanvas.Paint  Private Sub m\_Tool\_Click(sender As System.Object, e As System.EventArgs) Handles m\_Tool\_Line.Click, m\_Tool\_Rectangle.Click, m\_Tool\_Polygon.Click, m\_Tool\_PointSelect.Click  End Class |
|  |  |
|  |  |