# 用户回车响应事件

第一种：

private void txt\_confirmPwd\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

btn\_submit\_Click(btn\_submit, null);

}

}

第二种：

private void txt\_confirmPwd\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

btn\_submit\_Click(btn\_submit, null);

}

}

# 获取当前系统时间

DateTime dt = DateTime.Now;

dt.ToString();//2005-11-5 13:21:25

dt.ToFileTime().ToString();//127756416859912816

dt.ToFileTimeUtc().ToString();//127756704859912816

dt.ToLocalTime().ToString();//2005-11-5 21:21:25

dt.ToLongDateString().ToString();//2005年¨º11月?5日¨?

dt.ToLongTimeString().ToString();//13:21:25

dt.ToOADate().ToString();//38661.5565508218

dt.ToShortDateString().ToString();//2005-11-5

dt.ToShortTimeString().ToString();//13:21

dt.ToUniversalTime().ToString();//2005-11-5 5:21:25

dt.Year.ToString();//2005

dt.Date.ToString();//2005-11-5 0:00:00

dt.DayOfWeek.ToString();//Saturday

dt.DayOfYear.ToString();//309

dt.Hour.ToString();//13

dt.Millisecond.ToString();//441

dt.Minute.ToString();//30

dt.Month.ToString();//11

dt.Second.ToString();//28

dt.Ticks.ToString();//632667942284412864

dt.TimeOfDay.ToString();//13:30:28.4412864

dt.ToString();//2005-11-5 13:47:04

dt.AddYears(1).ToString();//2006-11-5 13:47:04

dt.AddDays(1.1).ToString();//2005-11-6 16:11:04

dt.AddHours(1.1).ToString();//2005-11-5 14:53:04

dt.AddMilliseconds(1.1).ToString();//2005-11-5 13:47:04

dt.AddMonths(1).ToString();//2005-12-5 13:47:04

dt.AddSeconds(1.1).ToString();//2005-11-5 13:47:05

dt.AddMinutes(1.1).ToString();//2005-11-5 13:48:10

dt.AddTicks(1000).ToString();//2005-11-5 13:47:04

dt.CompareTo(dt).ToString();//0

dt.Add(?).ToString();//问¨º号?为a一°?个?时º¡À间?段?

dt.Equals("2005-11-6 16:11:04").ToString();//False

dt.Equals(dt).ToString();//True

dt.GetHashCode().ToString();//1474088234

dt.GetType().ToString();//System.DateTime

dt.GetTypeCode().ToString();//DateTime

dt.GetDateTimeFormats('s')[0].ToString();//2005-11-05T14:06:25

dt.GetDateTimeFormats('t')[0].ToString();//14:06

dt.GetDateTimeFormats('y')[0].ToString();//2005年¨º11月?

dt.GetDateTimeFormats('D')[0].ToString();//2005年¨º11月?5日¨?

dt.GetDateTimeFormats('D')[1].ToString();//2005 11 05

dt.GetDateTimeFormats('D')[2].ToString();//星?期¨²六¢¨´ 2005 11 05

dt.GetDateTimeFormats('D')[3].ToString();//星?期¨²六¢¨´ 2005年¨º11月?5日¨?

dt.GetDateTimeFormats('M')[0].ToString();//11月?5日¨?

dt.GetDateTimeFormats('f')[0].ToString();//2005年¨º11月?5日¨? 14:06

dt.GetDateTimeFormats('g')[0].ToString();//2005-11-5 14:06

dt.GetDateTimeFormats('r')[0].ToString();//Sat, 05 Nov 2005 14:06:25 GMT

string.Format("{0:d}",dt);//2005-11-5

string.Format("{0:D}",dt);//2005年¨º11月?5日¨?

string.Format("{0:f}",dt);//2005年¨º11月?5日¨? 14:23

string.Format("{0:F}",dt);//2005年¨º11月?5日¨? 14:23:23

string.Format("{0:g}",dt);//2005-11-5 14:23

string.Format("{0:G}",dt);//2005-11-5 14:23:23

string.Format("{0:M}",dt);//11月?5日¨?

string.Format("{0:R}",dt);//Sat, 05 Nov 2005 14:23:23 GMT

string.Format("{0:s}",dt);//2005-11-05T14:23:23

string.Format("{0:t}",dt);//14:23

string.Format("{0:T}",dt);//14:23:23

string.Format("{0:u}",dt);//2005-11-05 14:23:23Z

string.Format("{0:U}",dt);//2005年¨º11月?5日¨? 6:23:23

string.Format("{0:Y}",dt);//2005年¨º11月?

string.Format("{0}",dt);//2005-11-5 14:23:23

string.Format("{0:yyyyMMddHHmmssffff}",dt);

DateTime.Now.ToString(“¡ãyyyy年¨ºMM月?dd日¨?”¡À);//xxxx年¨ºxx月?xx日¨?

# 判断是否有选中的项

## 判断一组RadioButton

private string returninfo()

{

string s = "";

foreach (Control c in groupBox2.Controls)

{

if (c is RadioButton)

{ if ((c as RadioButton).Checked) { s += c.Text; } }

}

return s;

}

## 判断菜单

foreach (ToolStripMenuItem men in contextMenuStrip1.Items)

{

foreach (ToolStripMenuItem a in men.DropDownItems)

{

if (a.Checked == true)

{ a.Checked = false; }

}

}

# 进制转换

/// 将字符串转换为相应的进制串

/// <param name="srcInput">输入需要转换的进制字符串</param>

/// <param name="fromBase">源字符串进制</param>

/// <param name="toBase">转换后的进制</param>

public static string HexConvert(string srcInput, int fromBase, int toBase)

{

try

{

long srcChange = Convert.ToInt64(srcInput, fromBase);

return Convert.ToString(srcChange, toBase);

}

catch (Exception)

{ return "进制转换有误."; }

}

//苑斌写的大门口十进制和十六进制互转：

private void ToOX()//转换成十六进制

{

string str= Convert.ToString(Convert.ToInt64(this.cardID10), 16);

if (str.Length > 6)

{

str = str.Substring(str.Length - 6);

}

this.CardID16 = str.PadLeft(8, '0');

}

private void ToOD()//转换成十进制

{

this.cardID10 =Convert.ToInt32(this.cardID16,16).ToString().PadLeft(8,'0');

}

# 退出系统的正规写法

在主窗体添加FormClosing事件

private void gf\_main\_FormClosing(object sender, FormClosingEventArgs e)

{

e.Cancel = (HDIC\_Message.ShowQuestionDialog(this, "是否确定退出系统?") != DialogResult.OK);

}

# 将一个字符串转化为窗体类

//获取窗体的Text值

Type fType = Type.GetType(txt\_subMenu.Text);

Form f = (Form)fType.InvokeMember("随便写", System.Reflection.BindingFlags.CreateInstance, null, null, null);

string formName = f.Text;

# 利用xml设置app.config文件

例如：修改<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<connectionStrings>

<add name ="connStringName" connectionString ="Data Source=.;Initial Catalog=STBInfo;Persist Security Info=False;User ID=sa; pwd =sa"

providerName="System.Data.SqlClient" />

</connectionStrings>

</configuration>

代码可以这样写：

public static void SetValue(string AppKey, string AppValue)

{

XmlDocument xDoc = new XmlDocument();

xDoc.Load(Application.StartupPath+@"\soc\_nds\_csharp.exe.config");

XmlNode xNode;

XmlElement xElem1;

XmlElement xElem2;

xNode = xDoc.SelectSingleNode("//connectionStrings");

xElem1 = (XmlElement)xNode.SelectSingleNode("//add[@name='" + AppKey + "']");

if (xElem1 != null)

{

xElem1.SetAttribute("connectionString", AppValue);

}

else

{

xElem2 = xDoc.CreateElement("add");

xElem2.SetAttribute("name", AppKey);

xElem2.SetAttribute("connectionString", AppValue);

xNode.AppendChild(xElem2);

}

xDoc.Save(Application.StartupPath+@"\soc\_nds\_csharp.exe.config");

}

# 将xml字符串转换为datatable

/// <param name="xmlData">xml字符串</param>

/// <param name="tagName">节点名称形如：//ROWSET/ROW</param>

/// <returns>datatable</returns>

public DataTable XMLToDataTable(string xmlData, string tagName)

{

XmlDocument Xmldoc = new XmlDocument();

Xmldoc.LoadXml(xmlData);

XmlNodeList xlist = Xmldoc.SelectNodes(tagName);//參數錯誤

DataTable dt = new DataTable();

DataRow dr;

for (int i = 0; i < xlist.Count; i++)

{

dr = dt.NewRow();

XmlElement xe = (XmlElement)xlist.Item(i);

for (int j = 0; j < xe.Attributes.Count; j++)

{

if (!dt.Columns.Contains("@" + xe.Attributes[j].Name))

{

dt.Columns.Add("@" + xe.Attributes[j].Name);

}

dr["@" + xe.Attributes[j].Name] = xe.Attributes[j].Value;

}

for (int j = 0; j < xe.ChildNodes.Count; j++)

{

if (!dt.Columns.Contains(xe.ChildNodes.Item(j).Name))

{

dt.Columns.Add(xe.ChildNodes.Item(j).Name);

}

dr[xe.ChildNodes.Item(j).Name] = xe.ChildNodes.Item(j).InnerText;

}

dt.Rows.Add(dr);

}

return dt;

}

# DataGridView

## dataGridView1单元格选择值的问题

先设置datagridview的属性SelectionMode为FullRowSelect否则dataGridView1.SelectedRows[0].Cells[2].Value.ToString()和dataGridView1.SelectedCells[2].Value.ToString();不能用。

【

其他的如dataGridView1 .Rows[e.RowIndex ].Cells [0].Value可以用例如：

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

MessageBox.Show(dataGridView1 .Rows[e.RowIndex ].Cells [0].Value .ToString () );

}

】

第一：private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString()

和dataGridView1.SelectedRows[0].Cells[2].Value.ToString()

和Convert.ToString(dataGridView1[2, dataGridView1.CurrentCell.RowIndex].Value).Trim();

和dataGridView1.SelectedCells[2].Value.ToString();是相等的

}

第二：

private void 修改ToolStripMenuItem\_Click(object sender, EventArgs e)

{

只能用dataGridView1.SelectedRows[0].Cells[2].Value.ToString()

或者Convert.ToString(dataGridView1[2,dataGridView1.CurrentCell.RowIndex].Value).Trim();

或者dataGridView1.CurrentRow.Cells["roleName"].Value.ToString()

或者dataGridView1 .CurrentCell .Value.ToString()

或者dataGridView1.SelectedCells[2].Value.ToString();

不能用dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString()

}

## dataGridView全选，反选

//全选

private void CheckAll\_CheckedChanged(object sender, EventArgs e)

{

if (dataGridView1.Rows.Count>0)

{

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

dataGridView1.Rows[i].Cells[0].Value = true;

}

}

}

//反选

private void CheckReverse\_CheckedChanged(object sender, EventArgs e)

{

if (dataGridView1.Rows.Count>0)

{

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if ((bool)dataGridView1.Rows[i].Cells[0].EditedFormattedValue)

{

dataGridView1.Rows[i].Cells[0].Value = false;

}

else

dataGridView1.Rows[i].Cells[0].Value = true;

}

}

}

## dataGridView按某列排序

【第一种方法：】

private void Form1\_Load(object sender, EventArgs e)

{

DataTable dt = new DataTable("ddd");

dt.Columns.Add(new DataColumn("a", typeof(int)));

dt.Columns.Add(new DataColumn("b", typeof(int)));

dt.Rows.Add(new object[] { "1", "1" });

dt.Rows.Add(new object[] { "3", "234" });

dt.Rows.Add(new object[] { "2", "32" });

dt.Rows.Add(new object[] { "23", "1" });

dt.Rows.Add(new object[] { "52", "231" });

dt.Rows.Add(new object[] { "34", "31" });

dt.Rows.Add(new object[] { "21", "51" });

dt.Rows.Add(new object[] { "2", "15" });

DataView dv = dt.DefaultView;

dv.Sort = "a desc";

DataGridView1.Source = dv;

}

【第二种：】

DataView dv = new DataView(dt, "", "a asc", DataViewRowState.CurrentRows);

## 获取dgv中控件对象

if (e.Row.RowType == DataControlRowType.DataRow)

{

Label labKong = (Label)e.Row.FindControl("labKong1");

}

# KeyCode和Keychar

e.KeyCode==Keys.Enter 和e.KeyChar==13和e.KeyChar===(Char)Keys.Enter 相等

e.KeyChar == (Char)48 ~ 57 -----> 0~9

e.KeyChar == (Char)8 -----------> Backpace

e.KeyChar == (Char)13-----------> Enter

//实例

private void txtName\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == 13)//判断是否按下Enter键

txtPwd.Focus();

e.Handled = true;//这个地方正规来说是要有的，它表示已经处理过KeyPress事件了

}

键盘 ASCII码 键盘 ASCII码

ESC 27 7 55

SPACE 32 8 56

! 33 9 57

" 34 : 58

# 35 ; 59

$ 36 < 60

% 37 = 61

& 38 > 62

' 39 ? 63

( 40 @ 64

) 41 A 65

\* 42 B 66

+ 43 C 67

' 44 D 68

- 45 E 69

. 46 F 70

/ 47 G 71

0 48 H 72

1 49 I 73

2 50 J 74

3 51 K 75

4 52 L 76

5 53 M 77

6 54 N 78

O 79 g 103

P 80 h 104

Q 81 i 105

R 82 j 106

S 83 k 107

T 84 l 108

U 85 m 109

V 86 n 110

W 87 o 111

X 88 p 112

Y 89 q 113

Z 90 r 114

[ 91 s 115

\ 92 t 116

] 93 u 117

^ 94 v 118

\_ 95 w 119

` 96 x 120

a 97 y 121

b 98 z 122

c 99 { 123

d 100 | 124

e 101 } 125

f 102 ~ 126

另外2个特殊ASCII码：回车=13，换行=10

# split用法

第一：static void Main(string[] args)

{

string aa = "1234,234523,4324,324";//清清月儿 http://blog.csdn.net/21aspnet/

string[] cc = aa.Split(new char[] { ',' }); 或者string[] sArray1=s.Split(new char[3]{‘c’,‘d’,‘e’}) ;

foreach (string bb in cc)

{

Console.Write(bb + "\r\n ");

}

}

第二：string aa = "1234,234523,4324,324";

string[] str = aa.Split(',');

foreach (string bb in str)

{

Console.Write(bb + " \r\n");

}

第三：static void Main(string[] args)

{

string a = "1,2,3,4,5,6,7,8,9";

string b = ","; 注意，第三种方法可以用字符串来切分

string[] c = Split(a, b);

foreach (string bb in c)

{

Console.Write(bb + " \r\n");

}

}

public static string[] Split(string input, string pattern)

{

string[] arr = System.Text.RegularExpressions.Regex.Split(input, pattern);

return arr;

}

# winform 主页模块调用方法

## 【方法一】

/// 每个窗体只允许打开一个

/// <typeparam name="T">窗体名称</typeparam>

/// <param name="mdiParent">父窗体</param>

private T OpenUniqueMDIChildWindow<T>(Form mdiParent) where T : Form, new()

{

foreach (Form subForm in mdiParent.MdiChildren)

{

if (subForm.GetType().Equals(typeof(T)))

{

subForm.Activate();

subForm.WindowState = FormWindowState.Normal;

return subForm as T;

}

}

T newForm = new T();

newForm.MdiParent = mdiParent;

newForm.Show();

newForm.WindowState = FormWindowState.Normal;

return newForm;

}

//例如模块一（门禁开卡锁卡）

private void toolStripButton10\_Click(object sender, EventArgs e)

{

OpenUniqueMDIChildWindow<baseinfo .car\_card .card >(this);

}

## 【方法二】

bool checkChildFrmExist(string childName)

{

foreach (Form childFrm in this.MdiChildren)

{

if (childFrm.Name == childName)

{

if (childFrm.WindowState == FormWindowState.Minimized)

childFrm.WindowState = FormWindowState.Normal;

childFrm.Activate();

return true;

}

}

return false;

}

//例如模块一（班级设置）

private void 班级设置MenuItem\_Click(object sender, EventArgs e)

{

if (!checkChildFrmExist("Class"))

{

Class frm = new Class();

frm.MdiParent = this;

frm.Show();

}

}

## 【方法三】

这个是haier 软件里面的。

private void expando\_ItemClick(object sender, EventArgs e)

{

string MenuName = ((TaskItem)sender).Name;

string MenuNo = ((TaskItem)sender).Tag.ToString();

if (string.IsNullOrEmpty(MenuName)) return;

//如果窗体已经创建，则将其激活，不再创建新事例

foreach (Form form in this.MdiChildren)

{

if (form.GetType().FullName.ToLower() == MenuName.ToLower())

{

form.Activate();

return;

}

}

Form f;

try

{

f = (Form)Activator.CreateInstance(Type.GetType(MenuName));

f.Text = ((TaskItem)sender).Text;

}

catch

{

return;

}

f.TopLevel = false;

f.MdiParent = this;

f.WindowState = FormWindowState.Maximized;

f.Show();

}

# winform工具栏点击事件（两种方法）

//【第一中方法】

private void toolStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)//双击toolStrip后生成的click事件

{

switch (e.ClickedItem.Text)

{

case "统计":

{

if (comboBox1.Text == "用户缴费查询")

sum\_JiaoFei();

else if (comboBox1.Text == "用户取款查询")

sum\_QuKuan();

else if (comboBox1.Text == "缴费押金查询")

sum\_YaJin();

else if (comboBox1.Text == "押金取款查询")

sum\_QuYaJin();

break;

}

case "打印":

{

PrintDGV.Print\_DataGridView(dataGridView1, comboBox1.Text);

break;

}

case "退出":

{

this.Close();

break;

}

}

}

//【第二中方法】

private void toolStripButton8\_Click(object sender, EventArgs e)//双击toolStrip的某个子项后子项的click事件

{

}

# Winform中Treeview绑定

## 非xml加载

private void bindTree() //

{

// 部门信息

SQL.SQL sql = new SQL.SQL();

DataTable dtBase\_Dpt = sql.GetData(sqlcon, "select SupDepBH,DepBH,DepName from DepTB where SupDepBH=0");

// 清空列表

treeDpt.Nodes.Clear();

if (dtBase\_Dpt.Rows.Count > 0)

{

// 添加根节点

treeDpt.Nodes.Add(dtBase\_Dpt.Rows[0]["DepBH"].ToString(), dtBase\_Dpt.Rows[0]["DepName"].ToString());

DataTable dtBase\_Dpt1 = sql.GetData(sqlcon, "select SupDepBH,DepBH,DepName from DepTB");

// 添加子节点

for (int i = 0; i < dtBase\_Dpt1.Rows.Count; i++)

{

if (dtBase\_Dpt1.Rows[i]["SupDepBH"].ToString() == dtBase\_Dpt.Rows[0]["DepBH"].ToString())

{

第一种方法： TreeNode tr = treeDpt.Nodes[0];//我测试发现Nodes[0]应该是0应该是深度的意思，不是第几个的意思

tr.Nodes.Add(dtBase\_Dpt1.Rows[i]["DepBH"].ToString(), dtBase\_Dpt1.Rows[i]["DepName"].ToString());

AddTreeNode(dtBase\_Dpt1, dtBase\_Dpt1.Rows[i]["DepBH"].ToString(), tr);

第二种方法：//｛treeDpt.Nodes [0].Nodes.Add(dtBase\_Dpt1.Rows[i]["DepBH"].ToString(), dtBase\_Dpt1.Rows[i]["DepName"].ToString());

// AddTreeNode(dtBase\_Dpt1, dtBase\_Dpt1.Rows[i]["DepBH"].ToString(), treeDpt.Nodes [0]);｝

}

}

}

treeDpt.ExpandAll();

}

//递归调用

private void AddTreeNode(DataTable nodesData, string RootNode, TreeNode tmpNode)

{

// 定义当前节点

TreeNode tmpRootNode = tmpNode.Nodes[RootNode];

// 添加当前节点的子节点

for (int i = 0; i < nodesData.Rows.Count; i++)

{

// 添加当前节点的子接点

if (nodesData.Rows[i]["SupDepBH"].ToString() == RootNode)

{

tmpRootNode.Nodes.Add(nodesData.Rows[i]["DepBH"].ToString(),

nodesData.Rows[i]["DepName"].ToString());

AddTreeNode(nodesData, nodesData.Rows[i]["DepBH"].ToString(), tmpRootNode);

}

}

}

## 从xml中绑定：这个是长城自主开发

//主页面：

dtTemp = ds.Tables[DIRTREE];//this is Tree xml file

if (dtTemp == null) { printInfo = "Cannot find node,please try again..."; ConsoleU.writeLine(printInfo, ConsoleU.Level.Warning); return; }

if (dtTemp.Rows == null) { printInfo = "Cannot find node rows,please try again..."; ConsoleU.writeLine(printInfo, ConsoleU.Level.Warning); return; }

Int32 index = 0;

foreach (DataRow row in dtTemp.Rows)

{

List<String> tempnode = new List<String>();

if (Int32.Parse(row[PARENT].ToString()) == 0)

{

tempnode.Add(row[CURRENTNMAE].ToString());

tempnode.Add(row[CURRENT].ToString());

treeView\_dir.Nodes.Add(row[CURRENT].ToString(), row[CURRENTNMAE].ToString());

AddTreeNode(dtTemp.Rows, row[CURRENT].ToString(), treeView\_dir.Nodes[index++]);//这个index必须有，否则会错位

}

}

//下面是递归调用的方法：

private void AddTreeNode(DataRowCollection tempDtRows, string RootNodeID, TreeNode tmpNode)

{

Int32 index = 0;

if (tempDtRows == null) return;

// 添加当前节点的子节点

foreach (DataRow mrow in tempDtRows)

{

// 添加当前节点的子接点

if (mrow[PARENT].ToString() == RootNodeID)

{

string a = mrow[CURRENT].ToString();

string b = mrow[CURRENTNMAE].ToString();

tmpNode.Nodes.Add(a, b);

AddTreeNode(tempDtRows, mrow[CURRENT].ToString(), tmpNode.Nodes[index]);

index++;

}

}

}

//【添加图片：长城测试，先添加ImageList控件并往该控件添加图片，然后 treeView1.ImageIndex = 0;则第一级节点显示ImageList控件中第一个图片, treeView1.Nodes[0].ImageIndex = 2;则第二级节点显示第二个图片.

# Winform导出到excel

public void DataTableExportExcel(System.Data.DataTable ExportTable, params string[] HeadText)

{

try

{

Microsoft.Office.Interop.Excel.Application excel = new Microsoft.Office.Interop.Excel.Application();//新的excel文档

excel.Application.Workbooks.Add(true); //新的工作表

excel.Visible = true;

int ColumnsIndex = ExportTable.Columns.Count;

//int RowsIndex = ExportTable.Rows.Count;

int index = 1;

//标题

for (; index <= ColumnsIndex; index++)

{

if (index <= HeadText.Length)

{

excel.Cells[1, index] = HeadText[index - 1];

}

}

//记录内容

for (int RowsIndex = 0; RowsIndex <= ExportTable.Rows.Count - 1; RowsIndex++)

{

for (index = 1; index <= ColumnsIndex; index++)

{

excel.Cells[RowsIndex + 2, index] = ExportTable.Rows[RowsIndex][index - 1].ToString();//行号

}

}

}

catch (Exception ex) { throw new Exception(ex.Message); }

}

private void btn\_daochu\_Click(object sender, EventArgs e)//调用excel，将信息导入到excel中

{

string[] headText = new string[7];

for (int ii = 0; ii < dgvClass.Columns.Count; ii++)

{

headText[ii] = dgvClass.Columns[ii].HeaderText;

}

DataTableExportExcel(dt, headText);

}

# 多个【radioButton】委托实现点击

private void jiajian\_Load(object sender, EventArgs e)

{

//下面是借助委托添加新事件，减轻了为每个radioButton 添加CheckedChanged事件

//this .radioButton1.CheckedChanged +=new EventHandler(radioButton1\_CheckedChanged);

//【上面radioButton1的一定要去掉了，因为 radioButton1\_CheckedChanged是它的，若再有则会执行两次。】

this.radioButton2 .CheckedChanged +=new EventHandler(radioButton1\_CheckedChanged);

this.radioButton3.CheckedChanged += new EventHandler(radioButton1\_CheckedChanged);

groupBox1.Visible = false;

pictureBox1.ImageLocation = a + @"\image\正在考试.gif";

}

//下面是事件方法：

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

if (!((RadioButton)sender).Checked)

{

return;

}

time = "";

switch (((RadioButton)sender).Tag.ToString())

{

case "10":

time = "10";

break;

case "20":

time = "20";

break;

case "30":

time = "30";

break;

default:

break;

}

}

# 多线程自己做的实例

private void Form1\_Load(object sender, EventArgs e)

{

Control.CheckForIllegalCrossThreadCalls = false;

/\*由于不允许在winform中直接跨线程访问控件的属性

加入这句代码以后发现程序可以正常运行了。

\* 这句代码就是说在这个类中我们不检查跨线程的调用是否合法（如果没有加这句话运行也没有异常，那么说明系统以及默认的采用了不检查的方式）。

\* 然而，这种方法不可取。我们查看CheckForIllegalCrossThreadCalls 这个属性的定义，就会发现它是一个static的，也就是说无论我们在项目的什么地方修改了这个值，

\* 他就会在全局起作用。而且像这种跨线程访问是否存在异常，我们通常都会去检查。如果项目中其他人修改了这个属性，那么我们的方案就失败了，我们要采取另外的方案。

\*/

}

Thread td1 = null;

private void btn\_seacher\_Click(object sender, EventArgs e)

{

if (td1 != null && td1.IsAlive)

{

MessageBox.Show("正在进行查询，请稍等……");

return;

}

td1 = new Thread(returndate);

td1.IsBackground = true;

td1.Start();

}

/// <summary>

/// 这个是查询的方法

/// </summary>

private void returndate()

{

try

{

SqlConnection con = new SqlConnection(" server=192.168.10.125;database=tcc;uid=sa;pwd=sa");

string sqlstr = " select \* from Car\_Data where 1=1";

if (dt1.Text.Trim() != "" && dt2.Text.Trim() != "")

{ sqlstr += " and '" + dt1.Text.Trim() + "'<DateTime and '" + dt2.Text.Trim() + "'>DateTime"; }

SqlCommand cmd = new SqlCommand(sqlstr, con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds);

DataTable dt = ds.Tables[0];

dataGridView1.DataSource = dt;

if (dt.Rows.Count > 0)

{

label1.Text = dt.Rows.Count.ToString();

}

//Thread.Sleep(1000);

}

catch { }

}

private void otherbutton\_Click(object sender, EventArgs e)

{

//另外一个按钮

if (td1 != null && td1.IsAlive)

{

MessageBox.Show("查询还没有结束，不能执行这个按钮功能……");

return;

}

else

MessageBox.Show("可以执行这个按钮功能了!!");

}

# 动态添加控件

public Form2()

{

InitializeComponent();

}

//用来添加控件的按钮

private void button1\_Click(object sender, EventArgs e)

{

TextBox txt = new TextBox();

txt.Location = new Point(150, 50);//位置

txt.Size = new Size(100, 20);//控件大小

txt.Name = "bt";//必须有，否则下面无法查看具体添加的的控件

this.Controls.Add(txt);

}

//用来显示新控件输入值的按钮

private void button2\_Click(object sender, EventArgs e)

{

TextBox tx = (TextBox)this.Controls.Find( "bt", true)[0];

button1.Text = tx.Text;

}

# 窗体之间显示顺序

## 控件成为顶级控件

private void 总在最前端ToolStripMenuItem\_Click(object sender, EventArgs e)

{

总在最前端ToolStripMenuItem.Checked = !总在最前端ToolStripMenuItem.Checked;

if (总在最前端ToolStripMenuItem.Checked)

{

this.TopMost = true;

this.Activate();

}

else

this.TopMost = false;

}

## 控件移到 Z 顺序的前面

btn\_ok.BringToFront();

## 控件移到 Z 顺序的后面

btn\_ok.SendToBack();

# ToolTip提示框

//将鼠标放到文本框上出现提示框：

private void txt\_name\_MouseEnter(object sender, EventArgs e)

{

toolTip1.Show("欢迎您点击姓名",txt\_name ,txt\_name .Width ,txt\_name .Height );

}

private void txt\_name\_MouseLeave(object sender, EventArgs e)

{

toolTip1.Hide(txt\_name );

}

# NotifyIcon控件

可以在窗体加载的时间添加如下代码：

notifyIcon1.Visible = true;

notifyIcon1.ShowBalloonTip(20,"通知","右击鼠标，可以进行更多功能操作",ToolTipIcon.Info);

//参数分别是（显示时间，标题，内容，图标类型）

//点击后主窗体最小化

private void notifyIcon1\_Click(object sender, EventArgs e)

{

if (this.WindowState == FormWindowState.Normal)

{

this.WindowState = FormWindowState.Minimized;

}

else if (this.WindowState == FormWindowState.Minimized)

{

this.WindowState = FormWindowState.Normal;

}

}

# DateTimePicker控件

使用 DateTimePicker 控件显示和选择时间 c# 的DateTimePicker 控件默认显示为“年月日”格式，当我们需要显示时间时可以将Format属性设置为"Time"或者自行设定具体的格式。

但是此时当我们点击右侧的小箭头时弹出的依然是选择日期，我们需要的是选择一个时间。怎么做呢？

只需要：将 DateTimePicker 的 ShowUpDown 属性设置为 true即可。

【对于自行设定格式，详解】首先将Format属性设置城Custom，然后将CustomFormat属性根据自己需要设置格式，如yy年-MM月-dd日 HH小时:mm分钟

:ss秒

# Listbox控件

## 判断新添加的信息在原列表中是否存在

（从(list\_left 列表右移到 list\_right表）

【第一种】 for (int i = 0; i < list\_left.SelectedItems.Count; i++)

{

if (list\_right.Items [i].ToString (). Equals(list\_left .SelectedItem.ToString()))

MessageBox.Show("已经存在！！");

else

list\_right.Items.Add(list\_left.SelectedItem.ToString());

}

【第二种】 if(list\_right.Items .Contains (list\_left .SelectedItem.ToString()))

MessageBox.Show("已经存在！！");

else

list\_right.Items.Add(list\_left.SelectedItem.ToString());

# OpenFileDialog控件

openFileDialog1.InitialDirectory = Application.StartupPath;

openFileDialog1.Filter = "音频（\*.mp3,\*.wma,\*wav）|\*.mp3;\*.wma;\*.wav|视频(\*.mp4,\*avi,\*.wmv)|\*.mp4;\*.avi;\*.wmv|所有文件(\*.\*)|\*.\*";

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

String fileExtension = System.IO.Path.GetExtension(openFileDialog1.FileNames[0]);

if (fileExtension != ".xml" || !System.IO.File.Exists(fileName))

{

ConsoleU.writeLine("this file is not xml or path error,please select again!", ConsoleU.Level.Warning);

}

}

# 代码设置控件图片

【1】panel3.BackgroundImage = Image.FromFile(Application .StartupPath+@"/image/fd1fe303e9fa033b3812bb30.jpg");

【2】if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.Image = new Bitmap(openFileDialog1.OpenFile());

}

【3】 if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.ImageLocation = openFileDialog1.FileName;

}

# 门禁指纹（转十六进制）

门禁指纹“userid”（十进制）转换成“卡片号码”（十六进制） sql写法：

select right('00000000'+select right('00000000'+dbo.IntToHex(1196+131072),8)(1196+131072),8)

例如 ：

private void button1\_Click(object sender, EventArgs e)

{ //指纹转换

int a = Convert .ToInt32 ( textBox1.Text.Trim()) + 131072;

int b= Convert.ToInt32(textBox1.Text.Trim()) + 65536;

SqlConnection con = new SqlConnection("server=.;database=tcc;pwd=sa;uid=sa;");

con.Open();

SqlCommand com = new SqlCommand("select right('00000000'+dbo.IntToHex(" + a + "),8) as a , right('00000000'+dbo.IntToHex(" + b + "),8) as b", con);

SqlDataReader dr = com.ExecuteReader();

if (dr.Read())

{

string c = dr.GetString(0);

string d = dr.GetString(1).ToString();

MessageBox.Show("您转换成十六进制后两条分别是：" + c + "和" + d);

}

}

/\*数据库中没有系统自带的dbo.IntToHex方法：

可以在 新建查询 或者 标量值函数 中新建函数：\*/

//方法代码一：

create function [dbo].[hex](@cardno int )

returns varchar (100)

as

begin

declare @temp\_mod int

declare @i int

declare @result varchar(100)

declare @temp\_x int

declare @result\_values int

set @result=''

set @i=1

set @temp\_x=0

while @cardno>0

begin

set @temp\_mod=@cardno%16

set @cardno=@cardno/16

set @result=(case @temp\_mod when 10 then 'A'

when 11 then 'B'

when 12 then 'C'

when 13 then 'D'

when 14 then 'E'

when 15 then 'F'

else ltrim(str(@temp\_mod)) end )+@result end

return @result end

//方法代码二：

Create Function IntToHex(@IntNum int)

returns varchar(16)

as

begin

declare @Mods int,@res varchar(16)

set @res=''

while @IntNum <> 0

begin

set @Mods =@IntNum % 16

if @Mods > 9

set @res = Char(Ascii('A')+@Mods-10)+@res

else

set @res = Cast(@Mods as varchar(4)) + @res

set @IntNum = @IntNum/16 end

return @res end