

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

《现代软件开发——使用.NET与C#》系列第六讲

► WinForms: GUI编程

俞晖
MSDN讲师
dolphin@vip.163.com

《现代软件开发——使用.NET与C#》 系列

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 回顾上节课的内容（*异常的设计*）
- 本节课讲述在**C#**中的**GUI**编程方面的问题
- **LEVEL 200**

Date	Topic
2005年8月1日	<i>WinForms</i>
2005年8月16日	<i>ADO.NET</i>
.	.
.	.

讨论的题目

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

“不论是在.NET或者JAVA中，GUI代码是事件驱动代码。在.NET中我们有两种GUI应用，WinForms 和 WebForms. WinForms 表现传统 Windows 桌面应用程序。在 Visual Studio 中，我们可以像VB6一样对控件进行拖拽和编码。

- 议题：
 - 事件驱动编程
 - WinForms

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

日程

- 事件驱动编程
- WinForms介绍
- WinForms编程

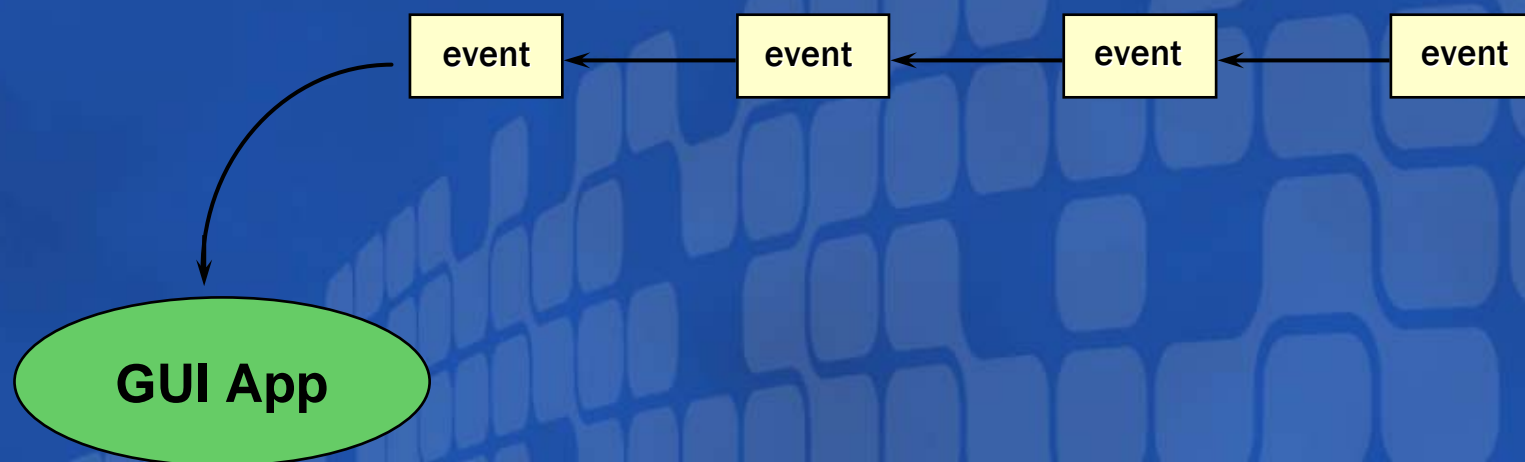
事件驱动

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 简单的思路:

- 用户个人的行为都被封装在很多事件中
- 事件由操作系统来安排顺序执行



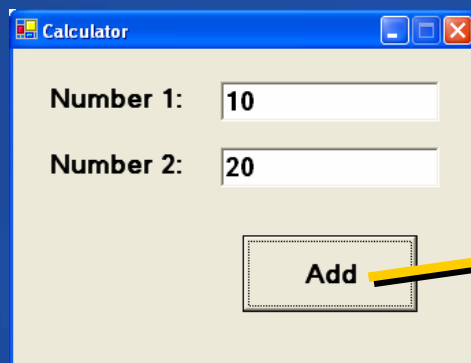
事件处理

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 事件被事件处理器处理

- 一般来说, 这些处理器方法的代码是写在可视化界面的后面的。
- **.NET** 自动处理事件并生成一些基础的调用事件处理器的代码



```
namespace Calculator
{
    public class Form1 : System.Windows.Forms.Form
    {
        DETAILS...

        Windows Form Designer generated code

        [STAThread]
        static void Main()
        {
            Application.Run(new Form1());
        }

        private void cmdAdd_Click(object sender, System.EventArgs e)
        {
            int num1, num2, result;

            num1 = int.Parse(this.txtNumber1.Text);
            num2 = int.Parse(this.txtNumber2.Text);
            result = num1 + num2;
            MessageBox.Show("The sum = " + result, "Calculator");
        }
    }
}
```

事件映射

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 事件是如何映射给事件处理器呢？
Delegates（代理）...

```
namespace Calculator
{
    public class Form1
    {
        DETAILS...

        Windows Form Designer generated code

        [STAThread]
        static void Main()
        {
            Application.Run(new Form1());
        }

        private void cmdAdd_Click(object sender, System.EventArgs e)
        {
            int num1, num2, result;

            num1 = int.Parse(this.txtNumber1.Text);
            num2 = int.Parse(this.txtNumber2.Text);
            result = num1 + num2;
            MessageBox.Show("The sum = " + result, "Calculator");
        }
    }
}
```

private System.Windows.Forms.Button cmdAdd;

this.cmdAdd.Click += new System.EventHandler(this.cmdAdd_Click);

delegate


MSDN Webcasts

事件处理器参数

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 事件处理器有两个参数：
 - *Sender*: 表示哪个对象生成了事件
 - *EventArgs*: 例外的特定事件信息（如果需要）



```
private void cmdAdd_Click(object sender, System.EventArgs e)
{
    int num1, num2, result;

    num1 = int.Parse(this.txtNumber1.Text);
    num2 = int.Parse(this.txtNumber2.Text);
    :
    :
    :
}
```


例子：通用验证方法

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 通用文本框验证方法

- 检验文本框内容是否包括一个有效的整型数据

```
using SWF=System.Windows.Forms;

public static void IsInteger(object sender,
                             System.ComponentModel.CancelEventArgs e)
{
    SWF.TextBox txt = (SWF.TextBox) sender;

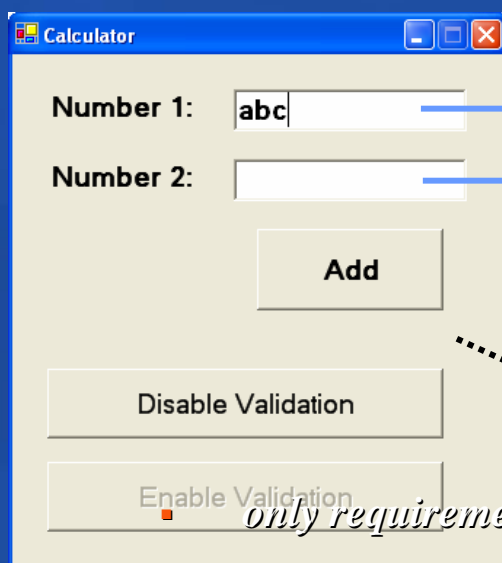
    try {
        int.Parse(txt.Text);
    }
    catch (FormatException) {
        SWF.MessageBox.Show("Please enter a numeric value");
        e.Cancel = true; // tell object validation failed, returns focus back...
    }
    catch (OverflowException) {
        ...
    }
    catch (Exception ex) {
        ...
    }
}
```

一般事件处理器

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 允许把多个事件映射到同一个处理器



```
public static void IsInteger(object sender,  
    System.ComponentModel.CancelEventArgs e)  
{  
    :  
    :  
}
```

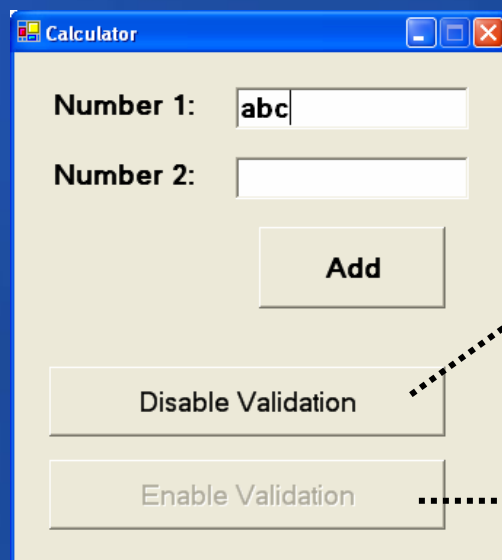
```
using SCM=System.ComponentModel;  
private void Form1_Load(...)  
{  
    this.txtNumber1.Validating +=  
        new SCM.CancelEventHandler(Validation.IsInteger);  
    this.txtNumber2.Validating +=  
        new SCM.CancelEventHandler(Validation.IsInteger);  
}
```

动态映射

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 处理器可以在运行时态增加或移除映射



```
using SCM=System.ComponentModel;  
  
private void cmdDisable_Click(...)  
{  
    this.txtNumber1.Validating -=  
        new SCM.CancelEventHandler(Validation.IsInteger);  
    this.txtNumber2.Validating -=  
        new SCM.CancelEventHandler(Validation.IsInteger);  
}
```

```
private void cmdEnable_Click(...)  
{  
    this.txtNumber1.Validating +=  
        new SCM.CancelEventHandler(Validation.IsInteger);  
    this.txtNumber2.Validating +=  
        new SCM.CancelEventHandler(Validation.IsInteger);  
}
```

■ *NOTE: demo app*

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Demonstration One

demo

Event handling...

Calculator

Number 1: 10

Number 2: 20

Add

Disable Validation

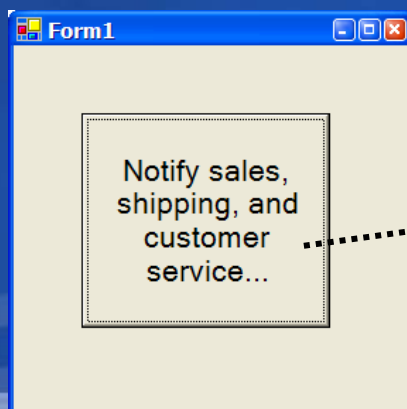
Enable Validation

事件处理器总结

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 你可以“多对一”进行事件处理
 - 不同的事件可以对应到相同的处理程序
- 你也可以“一对多”进行事件处理
 - 一个事件触发可以有多个动作（事件处理器映射）
 - Example: after update, all open forms must be notified
 - Example: after purchase, notify sales, shipping, billing, etc.



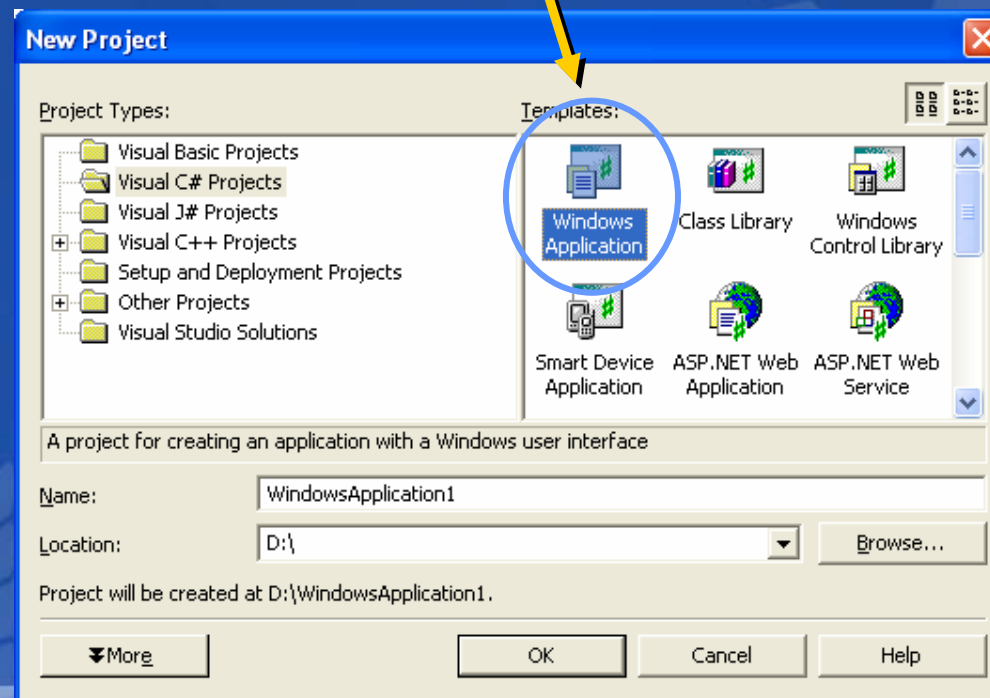
```
private void Form1_Load(...)
{
    this.cmdButton.Click += new System.EventHandler(...);
    this.cmdButton.Click += new System.EventHandler(...);
    this.cmdButton.Click += new System.EventHandler(...);
    .
    .
    .
}
```


WinForms

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- WinForms = .NET 桌面 GUI
- 使用 *Windows Application* 模板建立:

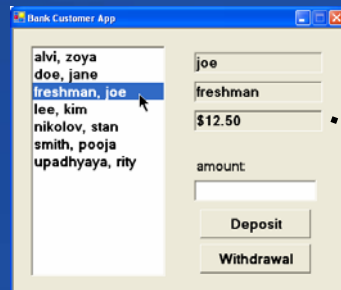


实现

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 定义在 *System.Windows.Forms* 名域
- **Forms** 从 *S.W.F.Form* 类继承行为:



object

instance of

System.Windows.Forms.Form

inherits from

Form1

用法

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 很像 **VB6** !

- 相同的体系：控件，属性，事件
- 相同的控件组：文本框，按钮，下拉框
- 强调简单实施：拖拉，放，编码

- 改进：

- 完全面对对象并且可以跨语言通用
- 有一些更强大的控件 (**DataGrid**)
- 对于数据绑定有更好的支持
- 更好的**resizing**支持 (*Anchor controls; AutoScroll form*)
- **Good-bye twips, hello pixels**

启动Form

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- Visual Studio 在生成 *Main* 代码中自动实例化一个 *Form* 的对象

```
public class Form1 : System.Windows.Forms.Form
{
    :
    :
    [STAThread]
    static void Main()
    {
        Application.Run( new Form1() );
    }
}
```

this form object becomes main form for application (when closed, app halts)

Best Practice #1

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- *把 Main 方法移出 Form Class*
 - Main 更容易找到 (不要成为 GUI 的一部分)
 - Main 很有可能还有其他的职责，这些职责是与 GUI 没有关系的
 - 一般使用 App 当作类名

```
using SWF=System.Windows.Forms;  
  
public class App  
{  
    [STAThread]  
    public static void Main()  
    {  
        .    // setup exception handling, tracing, etc.  
        :  
        :  
        SWF.Application.Run( new Form1() );  
    }  
}
```


Best Practice #2

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

● 一般启动时的通用异常处理

```
using SWF=System.Windows.Forms;
using ST=System.Threading;
using T=System.Diagnostics.Trace;

public class App
{
    [STAThread]
    public static void Main()
    {
        SWF.Application.ThreadException +=
            new ST.ThreadExceptionHandler(App.OnAppException);
        :
        :
        SWF.Application.Run( new Form1() );
    }
}
```

next page...

通用异常处理器

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 最简单的处理是日志然后终止程序

```
private static void OnAppException(object sender, ST.ThreadExceptionEventArgs e)
{
    Exception ex = e.Exception;

    SWF.MessageBox.Show("Halting due to error: " + ex.Message, ...);

    // log entire exception chain...
    T.WriteLine("Halting App in Generic Exception Handler:");
    while (ex != null)
    {
        T.WriteLine(ex.GetType().FullName);
        T.Indent();
        T.WriteLine("Msg: " + ex.Message);
        T.WriteLine("Trace: " + ex.StackTrace);
        T.Unindent();
        T.Flush();
        ex = ex.InnerException;
    }

    // close trace log and exit app
    T.Close();
    SWF.Application.Exit();
}
```

Best Practice #3

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 使用 try-catch 来捕捉异常

```
public static void Main()
{
    try
    {
        SWF.Application.ThreadException +=
            new ST.ThreadExceptionHandler(App.OnAppException);

        .    // setup log file, etc!
        .
        SWF.Application.Run( new Form1() );
    }
    catch(Exception ex)
    {
        OnAppException(null, new ST.ThreadExceptionHandlerEventArgs(ex));
    }
}
```

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Demonstration Two

demo

A few best practices...

The screenshot shows a window titled "Bank Customer App". On the left is a list box containing the following names: "alvi, zoya", "doe, jane", "freshman, joe" (which is highlighted with a blue background and a mouse cursor), "lee, kim", "nikolov, stan", "smith, pooja", and "upadhyaya, rity". On the right side of the window, there are four input fields: the first contains "joe", the second contains "freshman", the third contains "\$12.50", and the fourth is labeled "amount:" and is empty. Below these fields are two buttons: "Deposit" and "Withdrawal".

WinForms 编程

您的潜力，我们的动力

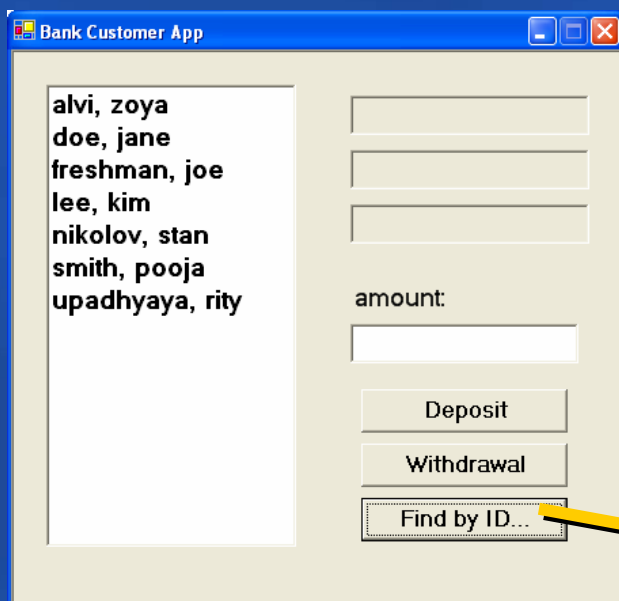
Microsoft
微软(中国)有限公司

- **WinForms 程序设计是一个很大的领域：**
 - 使用控件
 - 建立自定义控件
 - 多线程
 - **MVC (model-view-controller) 设计**
 - ...

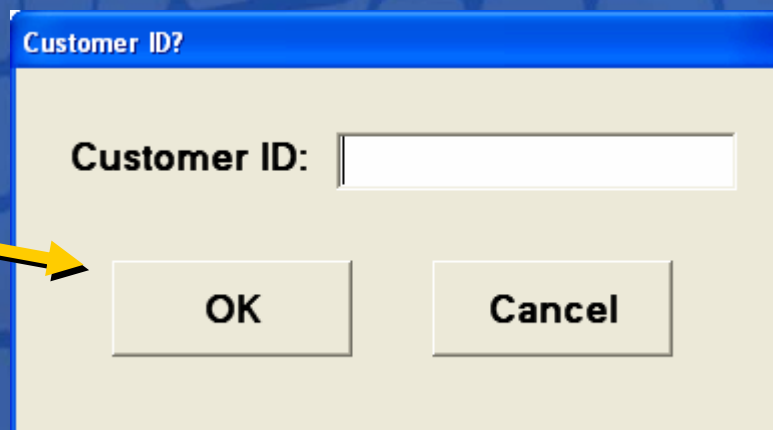
多Form应用程序例子:

您的潜力, 我们的动力
Microsoft
微软(中国)有限公司

- 输入customer id :



A screenshot of a Windows application window titled "Bank Customer App". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is divided into two sections. On the left, there is a list of customer names: "alvi, zoya", "doe, jane", "freshman, joe", "lee, kim", "nikolov, stan", "smith, pooja", and "upadhyaya, rity". On the right, there are three empty text input fields, followed by the label "amount:" and another empty text input field. Below these fields are three buttons: "Deposit", "Withdrawal", and "Find by ID...". A yellow arrow points from the "Find by ID..." button to a separate dialog box.



A screenshot of a small dialog box titled "Customer ID?". The dialog box has a blue title bar. Inside, there is a label "Customer ID:" followed by a single-line text input field. At the bottom of the dialog box are two buttons: "OK" and "Cancel".

The Main Form

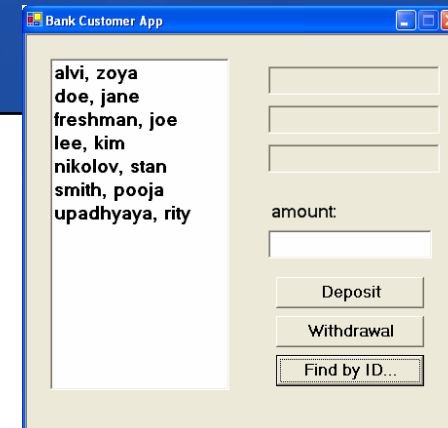
您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 建立, 展现

```
private void cmdFindByID_Click(...)
{
    // show form modally, then have .NET dispose for us...
    using (FormInputID frm = new FormInputID())
    {
        DialogResult dr;
        int id;

        dr = frm.ShowDialog(); // show form and wait for user to decide...
        if (dr == DialogResult.OK)
        {
            id = frm.CustomerID; // retrieve customer id!
            MessageBox.Show("Simulating search for customer " + id);
            :
            :
        }
    }
}
```

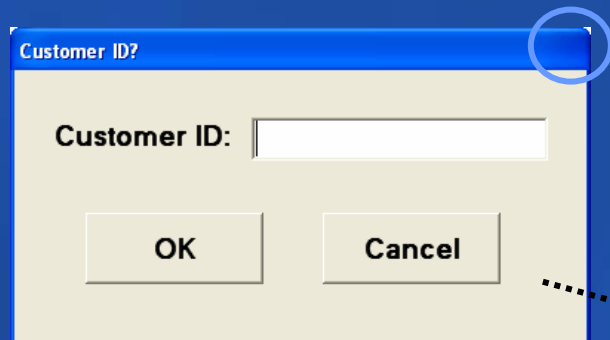


The Modal InputID Form

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 验证输入



```
private void Form1_Load(...)
{
    this.txtCustomerID.Validating += ...;
    this.cmdCancel.CausesValidation = false;
}

public int CustomerID
{
    get { return int.Parse(this.txtCustomerID.Text); }
}

private void cmdOK_Click(...)
{
    this.DialogResult = DialogResult.OK; // close with OK
}

private void cmdCancel_Click(...)
{
    this.txtCustomerID.Text = "-1"; // reset
    this.DialogResult = DialogResult.Cancel; // close w/ Cancel
}
```

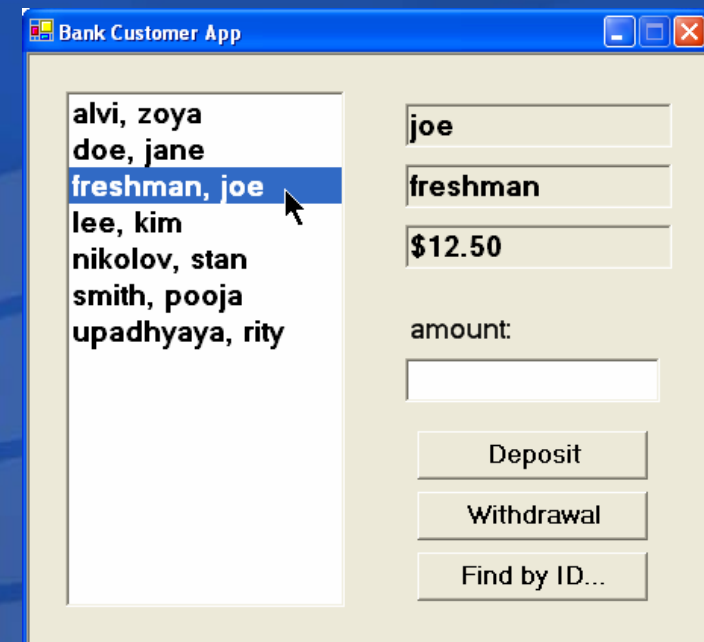
您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Demonstration Three

demo

Modal forms...



The image shows a Windows-style application window titled "Bank Customer App". It features a list of customer names on the left and transaction options on the right. The list includes: alvi, zoya; doe, jane; **freshman, joe** (highlighted with a mouse cursor); lee, kim; nikolov, stan; smith, pooja; and upadhyaya, rity. To the right of the list are three text input fields containing "joe", "freshman", and "\$12.50". Below these is a label "amount:" followed by an empty text input field. At the bottom are three buttons: "Deposit", "Withdrawal", and "Find by ID...".

Customer Name	Account Type	Amount
alvi, zoya		
doe, jane		
freshman, joe	freshman	\$12.50
lee, kim		
nikolov, stan		
smith, pooja		
upadhyaya, rity		

amount:

Deposit
Withdrawal
Find by ID...

(2) 数据绑定

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

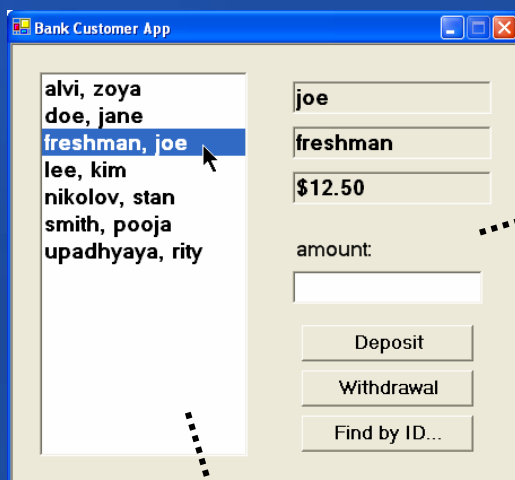
- 控件可以绑定数据源来获得数据
 - 控件可以绑定整个集合**collections**
 - 控件可以绑定单个对象属性
- 数据绑定在.NET中是广泛使用的
 - 自动地 reads & writes 绑定的数据

Examples

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 数据绑定在list上:



```
private void Form1_Load(...)
{
    System.Collections.ArrayList customers;

    customers = BankCustomers.Read("customers.txt");

    this.lstCustomers.DataSource = customers;
}
```

```
private void lstCustomers_SelectedIndexChanged(...)
{
    BankCustomer bc = (BankCustomer) this.lstCustomers.SelectedItem;

    this.txtFirstName.DataBindings.Clear();
    this.txtFirstName.DataBindings.Add("Text", bc, "FirstName");
    :
}
```

control's property

object's property

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Demonstration Four

demo

Data binding with
list & text boxes...

The screenshot shows a Windows application window titled "Bank Customer App". It features a list box on the left containing the following names: "alvi, zoya", "doe, jane", "freshman, joe", "lee, kim", "nikolov, stan", "smith, pooja", and "upadhyaya, rity". The "freshman, joe" entry is selected and highlighted in blue. To the right of the list box are three text boxes. The first text box contains "joe", the second contains "freshman", and the third contains "\$12.50". Below these text boxes is a label "amount:" followed by an empty text box. At the bottom of the window are three buttons: "Deposit", "Withdrawal", and "Find by ID...".

其它参考资源

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- **Webcast:**

- <http://www.microsoft.com/china/msdn/events/webcasts/shared/Webcast/MSDNWebCast.aspx>

- **Resources:**

- MSDN Library
<http://www.microsoft.com/china/MSDN/library/default.mspix>
- **C. Sells**, *“Windows Forms Programming in C#”*

MSDN flash:

- <http://www.microsoft.com/china/newsletter/case/MSDN.asp>

感谢大家的参与!

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 非常感谢大家的参与
- 您的**反馈**对我们来说是非常重要的



Date	Topic
2005年8月1日	C#中WinForms应用
2005年8月16日	ADO.NET




Q&A

您的潜力，我们的动力


Microsoft
微软(中国)有限公司

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问



提问(A)

删除(D)

问题管理器(Q)

您的潜力，我们的动力

Microsoft®
微软(中国)有限公司

Microsoft®

msdn


MSDN Webcasts