

### Проверка за коректно въведени данни

Свойство **CausesValidation** на контролите в **Windows Forms** – (**true**) при загуба на фокуса контролът се проверява; при неуспешна проверка фокусът се връща към контрола.

#### Събития за проверка коректността на данните

**Validating** даден контрол загубва фокуса и фокусът се премества към контрол със свойство **CausesValidation=true**; за да не се промени фокусът, се установява свойството **Cancel** на параметъра от тип **CancelEventArgs**;

**Validated** След събитието **Validating** преди загуба на фокуса от контрола; не може да се анулира.

### Контрол **ErrorProvider**

Представя информация за грешките.

#### Методи

**setError** определя контрола **control**, предизвиква грешката, като го маркира с икона за грешка и извежда подсказка **value**, описваща грешката.

```
public void setError (Control control, string value);
```

### Проверка на коректността на данните:

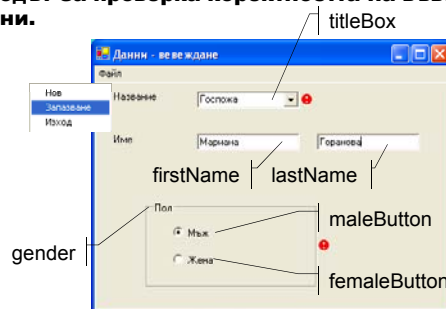
#### 1. От единствен контрол

- събитие **Validating**;
- **CausesValidation=true** за всички контроли.

#### 2. От много контроли – проверка на ниво форма:

- създава се метод за проверка коректността от много контроли;
- извиква се методът при завършване на въвеждането.

**Пример:** Форма със зависимост между данните, въведени в **Название** (контрол с падащ списък от елементи) и **Пол** (групов контрол с радио-бутони). При избор на подменю **Запазване** се извиква методът за проверка коректността на въведените данни.



```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace Validation
{
    public class ValidatingForm : System.Windows.Forms.Form
    {
        private System.Windows.Forms.MainMenu mainMenu;
        private System.Windows.Forms.MenuItem menuItem1;
        private System.Windows.Forms.MenuItem newItem;
        private System.Windows.Forms.MenuItem saveItem;
        private System.Windows.Forms.MenuItem exitItem;
        private System.Windows.Forms.Label titleLabel;
        private System.Windows.Forms.Label nameLabel;
    }
}
```

```
private System.Windows.Forms.ComboBox titleBox;
private System.Windows.Forms.TextBox firstName;
private System.Windows.Forms.TextBox lastName;
private System.Windows.Forms.GroupBox gender;
private System.Windows.Forms.RadioButton femaleButton;
private System.Windows.Forms.RadioButton maleButton;
private System.Windows.Forms.ErrorProvider errorProvider;

private System.ComponentModel.Container components = null;

public ValidatingForm()
{
    InitializeComponent();
    Reset();
}
```

```
protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose( disposing );
}
```

```
#region Windows Form Designer generated code
private void InitializeComponent()
{
    // Всички контроли по подразбиране имат свойството
    // CausesValidation=true
    // ...
    this.newItem.Click +=
        new System.EventHandler(this.newItem_Click);

    this.saveItem.Click +=
        new System.EventHandler(this.saveItem_Click);

    this.exitItem.Click +=
        new System.EventHandler(this.exitItem_Click);
}
#endregion
```

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

private void Reset()
{
    titleBox.Text="Господин";
    firstName.Text="";
    lastName.Text="";
    maleButton.Checked=true;
    errorProvider.SetError(gender,"");
    errorProvider.SetError(titleBox,"");
}
```

```
private void newItem_Click(object sender, System.EventArgs e)
{
    Reset();
}

private void exitItem_Click(object sender, System.EventArgs e)
{
    Close();
}
```

```
private bool checkTitleAndGender()
{
    if(titleBox.Text=="Господин")
    {
        if(!maleButton.Checked)
        {
            errorProvider.SetError(gender,
                "Ако званието е господин, полът трябва да бъде мъж");

            errorProvider.SetError(titleBox,
                "Ако полът е жена, званието трябва да бъде госпожа
                или госпожица");
            return false;
        }
    }
}
```

```
else if(titleBox.Text=="Госпожа" ||
        titleBox.Text=="Госпожица")
{
    if(!femaleButton.Checked)
    {
        errorProvider.SetError(gender,
            "Ако званието е госпожа или госпожица, полът
            трябва да бъде жена");
        errorProvider.SetError(titleBox,
            "Ако полът е мъж, званието трябва да бъде
            господин");
        return false;
    }
    errorProvider.SetError(gender,"");
    errorProvider.SetError(titleBox,"");
    return true;
}
```

```
private void saveItem_Click(object sender,
    System.EventArgs e)
{
    // При неуспешна проверка
    if(!checkTitleAndGender())
        e.Cancel=true;
}
}
```

**Ако проверката е само за един контрол (въвеждане на название), се използва събитието Validating:**

```
private void titleValidating(object sender,
    System.ComponentModel.CancelEventArgs e)
{
    // При неуспешна проверка
    if(!checkTitleAndGender())
        e.Cancel=true;
}
```

**Приложения с интерфейс с един и много документи**

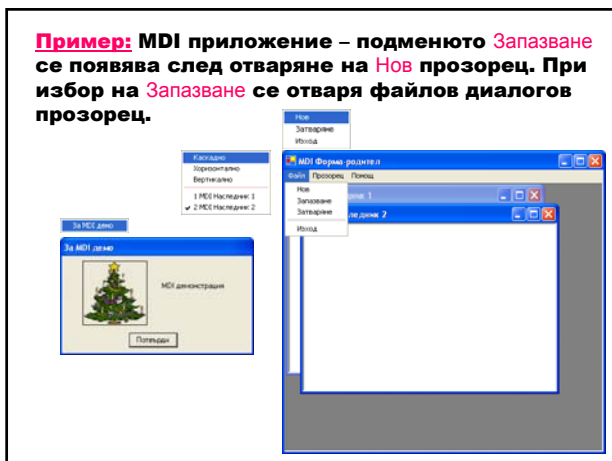
Стилове потребителски интерфейс

1. Интерфейс с един документ (SDI – single-document interface).
2. Интерфейс с много документи (MDI – multiple-document interface).
3. Интерфейс в стил Explorer.

**Създаване на MDI приложение**

1. Създаване на форма-родител
  - IsMdiContainer = true;
  - LayoutMdi () – аранжира интерфейса с много документи.
2. Създаване на форма-наследник.
3. Извикване на наследника от родителя.

Проверка за активна форма-наследник  
Свойство **ActiveMdiChild** връща текущата активна форма-наследник.



```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace MDIDemo
{
    public class MDIParent : System.Windows.Forms.Form
    {
        private System.Windows.Forms.MainMenu mdiMenu;

        private System.Windows.Forms.MenuItem fileItem;
        private System.Windows.Forms.MenuItem newItem;
        private System.Windows.Forms.MenuItem closeItem;
        private System.Windows.Forms.MenuItem exitItem;
    }
}
```

```
private System.Windows.Forms.MenuItem windowItem;
private System.Windows.Forms.MenuItem cascadeItem;
private System.Windows.Forms.MenuItem horizontalItem;
private System.Windows.Forms.MenuItem verticalItem;

private int childCount=0;

private System.Windows.Forms.MenuItem helpItem;
private System.Windows.Forms.MenuItem aboutItem;

private System.ComponentModel.Container components =
    null;

public MDIParent()
{
    InitializeComponent();
}
```

```
protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose( disposing );
}
```

```
#region Windows Form Designer generated code
private void InitializeComponent()
{
    // ...
    // fileItem
    this.fileItem.MergeType =
        System.Windows.Forms.MenuMerge.MergeItems;
    // newItem
    this.newItem.MergeOrder = 0;
    this.newItem.Click +=
        new System.EventHandler(this.newItem_Click);
    // closeItem
    this.closeItem.MergeOrder = 2;
    this.closeItem.Click +=
        new System.EventHandler(this.closeItem_Click);
}
```

```
// exitItem
this.exitItem.MergeOrder = 4;
this.exitItem.Click +=
    new System.EventHandler(this.exitItem_Click);

// windowItem
// Представяне на списък на отворените прозорци
this.windowItem.MdiList = true;

this.cascadeItem.Click +=
    new System.EventHandler(this.cascadeItem_Click);
this.horizontalItem.Click +=
    new System.EventHandler(this.horizontalItem_Click);
this.verticalItem.Click +=
    new System.EventHandler(this.verticalItem_Click);
this.aboutItem.Click +=
    new System.EventHandler(this.aboutItem_Click);
```

```
// MDIParent
this.IsMdiContainer = true;
this.Menu = this.mdiMenu;
}
#endregion

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

```
private void exitItem_Click(object sender, System.EventArgs e)
{
    this.Close();
}

private void newItem_Click(object sender, System.EventArgs e)
{
    // Създаване на форма-наследник
    MDIChild childForm=new MDIChild();
    childForm.MdiParent=this;
    childCount++;
    childForm.Text=childForm.Text+" "+childCount;
    // Изобразяване на формата-наследник
    childForm.Show();
}
```

```
private void cascadelItem_Click(object sender,
                               System.EventArgs e)
{
    this.LayoutMdi(MdiLayout.Cascade);
}

private void horizontalItem_Click(object sender,
                                  System.EventArgs e)
{
    this.LayoutMdi(MdiLayout.TileHorizontal);
}

private void verticalItem_Click(object sender,
                                System.EventArgs e)
{
    this.LayoutMdi(MdiLayout.TileVertical);
}
```

```
private void closeItem_Click(object sender,
                              System.EventArgs e)
{
    // Проверка за активен прозорец
    Form childForm=this.ActiveMdiChild;
    if(childForm != null)
        childForm.Close();
}

private void aboutItem_Click(object sender,
                              System.EventArgs e)
{
    About aboutDialog=new About();
    aboutDialog.ShowDialog();
}
}
```

```
// Project => Add Windows Form
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.IO;

namespace MDIDemo
{
    public class MDIChild : System.Windows.Forms.Form
    {
        private System.Windows.Forms.TextBox editData;
        private System.Windows.Forms.MainMenu childMenu;
        private System.Windows.Forms.MenuItem fileItem;
        private System.Windows.Forms.MenuItem saveItem;
        private System.Windows.Forms.MenuItem menuItem3;
    }
}
```

```
private System.Windows.Forms.SaveFileDialog
    saveFileDialog;
private System.ComponentModel.Container components =
    null;

public MDIChild()
{
    InitializeComponent();
}
```

```
protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if(components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose( disposing );
}
```

```
#region Windows Form Designer generated code
private void InitializeComponent()
{
    // ...
    // fileItem
    this.fileItem.MergeType =
        System.Windows.Forms.MenuMerge.MergeItems;
    this.fileItem.Text = "&Файл";
    // saveItem
    this.saveItem.MergeOrder = 1;
    this.saveItem.Click +=
        new System.EventHandler(this.saveItem_Click);
    // menuItem3
    this.menuItem3.MergeOrder = 3;
    this.menuItem3.Text = "-";
    // MDIChild
    this.Menu = this.childMenu;
}
#endregion
```

```
private void saveItem_Click(object sender,
                        System.EventArgs e)
{
    DialogResult buttonClicked=saveFileDialog.ShowDialog();
    if( buttonClicked.Equals(DialogResult.OK))
    {
        Stream saveStream=saveFileDialog.OpenFile();
        StreamWriter saveWriter=
            new StreamWriter(saveStream);
        foreach(string line in editData.Lines)
            saveWriter.WriteLine(line);
        saveWriter.Close();
    }
}
```

```
// Project => Add Windows Form
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;

namespace MDIDemo
{
    public class About : System.Windows.Forms.Form
    {
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Button ok;
        private System.ComponentModel.IContainer components =
            null;
    }
}
```

```
public About()
{
    InitializeComponent();
}

protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if(components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose( disposing );
}
```

```
#region Windows Form Designer generated code
private void InitializeComponent()
{
    // label1
    this.label1.Image =
        ((System.Drawing.Image)(resources.GetObject(
            "label1.Image")));

    // label2
    this.label2.Text = "MDI демонстрация";

    // ok
    this.ok.Click += new System.EventHandler(this.ok_Click);

    // About
    this.Text = "За MDI демо";
}
#endregion
```

```
private void ok_Click(object sender, System.EventArgs e)
{
    this.Close();
}
}
```

### Лента с инструменти и лента за статуса

#### Контрол ToolBar

Команден контрол – съдържа лента с инструменти (бутони от тип ToolBarButton).

#### Свойства на контрола ToolBar

**Buttons** дава колекция ToolBarButtonCollection от ToolBarButton контроли с методи: **Add** – добавяне, **Remove** – изтриване, **Clear** – изчистване, **IndexOf** – връща индекса на бутона в колекцията;

**ImageList** дава/установява колекция от изображения за бутоните;

**ShowToolTips** (**true**) изобразява подсказка за бутоните.

**Свойства на класа ToolBarButton**

**Tag** дава/установява обект с данни за бутона;

**ImageIndex** дава/установява стойността на индекса на изображението от списък с изображения;

**Style** дава/установява стила на бутона: **DropDownButton**, **PushButton**, **Separator**, **ToggleButton**;

**TooltipText** текст-подсказка за бутона.

**Събития за контрола ToolBar**

**ButtonClick** вдига се при натискане на бутон от тип **ToolBarButton** от лентата с инструменти **ToolBar**.

**Контрол StatusBar**

**Текстов контрол** – съдържа статус-лента в долната част на прозореца с панели от тип **StatusBarPanel**.

**Свойства на контрола StatusBar**

**Panels** дава колекция **StatusBarPanelCollection** от **StatusBarPanel** контроли с методи: **Add** – добавяне, **Clear** – изчистване;

**ShowPanels** (**true**) изобразява панелите;

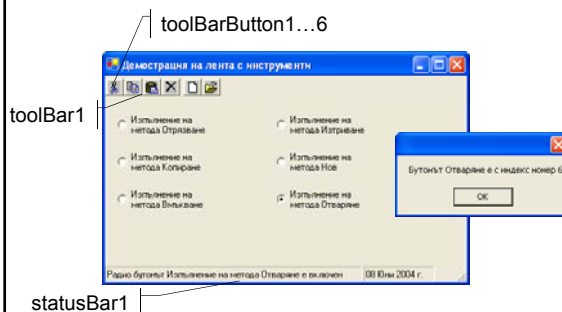
**Свойства на класа StatusBarPanel**

**AutoSize** **Contents** – ширината на панела се определя от неговото съдържание; **None** – не променя размера си; **Spring** – споделя наличното пространство с другите панели с тази стойност;

**Text** текст на панела в статус лентата;

**TooltipText** текст-подсказка за панела.

**Пример:** Приложение с лента с инструменти и статус-лента. При натискане на бутон от лентата с инструменти съответният радио-бутон преминава в състояние включено.



```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace ToolBarDemo
{
    public class ToolBarForm : System.Windows.Forms.Form
    {
        private string panelText;

        private System.Windows.Forms.ToolBar toolbar1;
        private System.Windows.Forms.ToolBarButton toolbarButton1;
        private System.Windows.Forms.ToolBarButton toolbarButton2;
    }
}
```

```
private System.Windows.Forms.ToolBarButton toolbarButton3;
private System.Windows.Forms.ToolBarButton toolbarButton4;
private System.Windows.Forms.ToolBarButton toolbarButton5;
private System.Windows.Forms.ToolBarButton toolbarButton6;
private System.Windows.Forms.ToolBarButton toolbarButton7;
private System.Windows.Forms.ImageList imageList1;

private System.Windows.Forms.RadioButton cutRadioButton;
private System.Windows.Forms.RadioButton copyRadioButton;
private System.Windows.Forms.RadioButton pasteRadioButton;
```

```
private System.Windows.Forms.RadioButton
deleteRadioButton;
private System.Windows.Forms.RadioButton
newRadioButton;
private System.Windows.Forms.RadioButton
openRadioButton;

private System.Windows.Forms.StatusBar statusBar1;
private System.ComponentModel.IContainer components;

public ToolBarForm()
{
    InitializeComponent();
}
```

```
protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose( disposing );
}
```

Windows Forms Designer generated code

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

```
private void toolBar1_ButtonClick(object sender,
System.Windows.Forms.ToolBarButtonClickEventArgs e)
{
    switch(e.Button.Tag.ToString())
    {
        case "Отрязване":
            this.cutRadioButton.Checked = true;
            panelText = "Радио бутонът "+this.cutRadioButton.Text +
                " е включен";
            break;
        case "Копиране":
            this.copyRadioButton.Checked = true;
            panelText = "Радио бутонът "+this.copyRadioButton.Text
                + " е включен";
            break;
    }
```

```
case "Вмъкване":
    this.pasteRadioButton.Checked = true;
    panelText = "Радио бутонът "+this.pasteRadioButton.Text
        + " е включен";
    break;
case "Изтриване":
    this.deleteRadioButton.Checked = true;
    panelText = "Радио бутонът "+this.deleteRadioButton.Text
        + " е включен";
    break;
case "Нов":
    this.newRadioButton.Checked = true;
    panelText = "Радио бутонът "+this.newRadioButton.Text +
        " е включен";
    break;
```

```
case "Отваряне":
    this.openRadioButton.Checked = true;
    panelText = "Радио бутонът "+this.openRadioButton.Text
        + " е включен";
    break;
}
this.statusBar1.Panels[0].Text = panelText;
MessageBox.Show("Бутонът " + e.Button.Tag
    + " е с индекс номер " + toolBar1.Buttons.IndexOf(e.Button));
}
```

```
private void formLoad(object sender, System.EventArgs e)
{
    StatusBarPanel panel1 = new StatusBarPanel();
    StatusBarPanel panel2 = new StatusBarPanel();

    panel1.AutoSize = StatusBarPanelAutoSize.Contents;
    panel2.AutoSize = StatusBarPanelAutoSize.Contents;

    panel1.ToolTipText = "Текущо избраният радио бутон";
    panel2.ToolTipText = "Текущата дата";

    panel1.Text = "Радио бутонът "+this.cutRadioButton.Text +
        " е включен";
    panel2.Text = System.DateTime.Now.ToLongDateString();
}
```



```
// Изчиства съществуващите панели и добавя panel1 и
// panel2
statusBar1.Panels.Clear();
statusBar1.Panels.Add(panel1);
statusBar1.Panels.Add(panel2);
statusBar1.ShowPanels = true;
this.cutRadioButton.Checked = true;
}
}
```

### Компоненти за графичен потребителски интерфейс

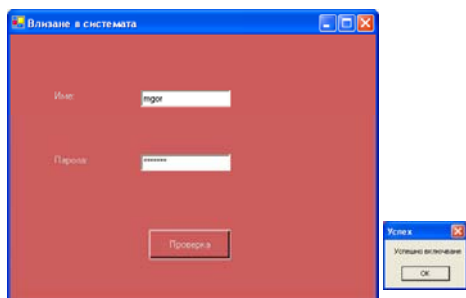
#### Потребителски контрол:

- контейнер с контроли;
- наследник на класа **Control** – съдържа функционалността на контролите с допълнителни свойства, методи и събития.

#### Предимства:

- използва се в много приложения;
- приложението взима автоматично модифицираната версия на потребителския контрол, без да се рекомпилира.

**Пример:** Потребителски контрол **Login** за влизане в система.



// File => New => Project => Templates: Windows Control Library



```
using System;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Windows.Forms;

namespace LoginControl
{
    public class Login : System.Windows.Forms.UserControl
    {
        // Дефиниране на събития за успешно и неуспешно
        // включване – чрез делегати
        public event System.EventHandler LoginSuccess;
        public event System.EventHandler LoginFail;

        private System.Windows.Forms.Label userNameLabel;
        private System.Windows.Forms.Label passwordLabel;
```

```
private System.Windows.Forms.TextBox userNameText;
private System.Windows.Forms.TextBox passwordText;

private System.Windows.Forms.ErrorProvider loginError;
private System.Windows.Forms.Button loginButton;

private System.ComponentModel.IContainer components
    = null;

public Login()
{
    InitializeComponent();
}
```

```
protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if( components != null )
            components.Dispose();
    }
    base.Dispose( disposing );
}

```

Windows Designer generated code

```
// Проверка за съответствието парола-въведено
// потребителско име
private bool userNameAndPasswordAreValid
    (string userName, string password)
{
    return password.Equals("TrustMe");
}

private void loginClick(object sender, System.EventArgs e)
{
    if(userNameText.Text.Length==0)
    { loginError.SetError( userNameText,
        "Моля въведете име на потребител");
        return;
    }
    if(passwordText.Text.Length==0)
    { loginError.SetError(passwordText,
        "Моля въведете парола");
        return;
    }
}

```

```
if(userNameAndPasswordAreValid(userNameText.Text,
    passwordText.Text))
{
    if(LoginSuccess != null)
    {
        LoginSuccess(this, new System.EventArgs());
    }
}
else
{
    if(LoginFail != null)
    {
        LoginFail(this, new System.EventArgs());
    }
}
}

```

```
// Дефиниране на свойства на потребителския контрол
// Class View => <R> клас Login => Add => Add Property
public string UserNameLabel
{
    get { return userNameLabel.Text; }
    set { userNameLabel.Text=value; }
}

public string PasswordLabel
{
    get { return passwordLabel.Text; }
    set { passwordLabel.Text=value; }
}

public string LoginButtonText
{
    get { return loginButton.Text; }
    set { loginButton.Text=value; }
}

```

```
// Не позволява преглеждане
[Browsable(false)]
public string UserName
{
    set { userNameText.Text=value; }
}

[Browsable(false)]
public string Password
{
    set { passwordText.Text=value; }
}
}

```

#### Компилиране на LoginControl

#### Добавяне на потребителския контрол Login към Toolbox

Tools => Add/Remove Toolbox Items...

Brows... => ../LoginControl/bin/Debug/LoginControl.dll

Контролът Login се появява към табулатора General на Toolbox.

```
// Тестване на потребителския контрол
// File => New => Project
// Project: Set As Start Up Project
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace LoginTest
{
    public class Form1 : System.Windows.Forms.Form
    {
        private LoginControl.Login login1;

        private System.ComponentModel.IContainer components =
            null;
```

```
public Form1()
{
    InitializeComponent();
}

protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
        base.Dispose( disposing );
    }
}
```

```
#region Windows Form Designer generated code
private void InitializeComponent()
{
    this.login1 = new LoginControl.Login();
    this.SuspendLayout();

    // login1
    this.login1.Location = new System.Drawing.Point(20, 20);
    this.login1.LoginButtonText = "Проверка";
    this.login1.Name = "login1";
    this.login1.PasswordLabel = "Парола:";
    this.login1.Size = new System.Drawing.Size(400, 300);
    this.login1.TabIndex = 0;
    this.login1.UserNameLabel = "Име:";
    this.login1.LoginFail +=
        new System.EventHandler(this.loginFail);
    this.login1.LoginSuccess +=
        new System.EventHandler(this.loginSuccess);
```

```
// Form1
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.BackColor = System.Drawing.Color.IndianRed;
this.ClientSize = new System.Drawing.Size(448, 326);
this.Controls.Add(this.login1);
this.ForeColor = System.Drawing.Color.LightGray;
this.Name = "Form1";
this.Text = "Влизане в системата";
this.ResumeLayout(false);
}
#endregion
```

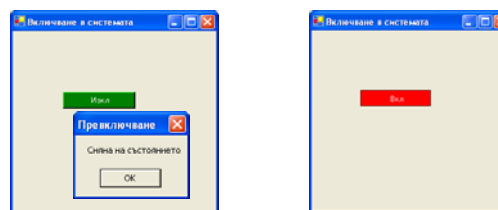
```
static void Main()
{
    Application.Run(new Form1());
}

private void loginFail(object sender, System.EventArgs e)
{
    MessageBox.Show("Неуспешно включване", "Неуспех");
}

private void loginSuccess(object sender, System.EventArgs e)
{
    MessageBox.Show("Успешно включване", "Успех");
}
}
```

### Специализирани контроли, използващи наследяване

**Пример:** Контрол **ToggleButton** за включване в системата – има две състояния включено/изключено; при натискане се превключва и остава в новото състояние до повторно натискане. Наследява класа **CheckBox**.



```
// Създаване на библиотечен клас
// File => New => Project => Templates: Class Library
using System;
using System.Windows.Forms;
using System.Drawing;

namespace SubclassesControls
{
    public class ToggleButton : CheckBox
    {
        // Текст в състояние включено
        private string checkedText;
        public string CheckedText
        {
            get { return checkedText; }
            set { this.checkedText=value; }
        }
    }
}
```

```
// Текст в състояние изключено
private string uncheckedText;
public string UncheckedText
{
    get { return uncheckedText; }
    set { this.uncheckedText=value; }
}

// Цвят на бутона при състояние включено
private Color checkedColor;
public Color CheckedColor
{
    get { return checkedColor; }
    set { this.checkedColor=value; }
}

// Цвят на бутона при състояние изключено
private Color uncheckedColor;
public Color UncheckedColor
{
    get { return uncheckedColor; }
    set { this.uncheckedColor=value; }
}
}
```

```
public ToggleButton()
{
    // Появява се като бутон
    this.Appearance=Appearance.Button;
    this.CheckedText="Включено";
    this.UncheckedText="Изключено";
    this.CheckedColor=Color.Gray;
    this.UncheckedColor=this.BackColor;
}
}
```

```
// Предефиниран метод OnClick, който вдига събитието
// Click
protected override void OnClick(EventArgs e)
{
    base.OnClick(e);
    if(this.Checked)
    {
        this.Text=this.CheckedText;
        this.BackColor=this.CheckedColor;
    }
    else
    {
        this.Text=this.UncheckedText;
        this.BackColor=this.UncheckedColor;
    }
}
}
```

#### Компилиране на SubclassesControl

Добавяне на контрола ToggleButton към Toolbox

Tools => Add/Remove Toolbox Items...

Brows... =>

.../SubclassesControl/bin/Debug/SubclassesControl.dll

Контролът ToggleButton се появява към табулатора General на Toolbox.

Преместване на ToggleButton от табулатора General в табулатора Windows Forms на Toolbox

General => <R> ToggleButton => Cut => OK

Windows Forms => <R> произволен контрол => Paste => OK

#### // Тестване на контрола ToggleButton

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace WindowsApplication1
{
    public class Form1 : System.Windows.Forms.Form
    {
        private SubclassesControls.ToggleButton toggleButton1;

        private System.ComponentModel.IContainer components = null;
    }
}
```

```

public Form1()
{
    InitializeComponent();
}

protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose( disposing );
}

```

```

#region Windows Form Designer generated code
private void InitializeComponent()
{
    this.toggleButton1 = new SubclassesControls.ToggleButton();
    this.SuspendLayout();

    // toggleButton1
    this.toggleButton1.Appearance =
        System.Windows.Forms.Appearance.Button;
    this.toggleButton1.BackColor = System.Drawing.Color.Green;
    this.toggleButton1.CheckedColor =
        System.Drawing.Color.Red;
    this.toggleButton1.CheckedText = "Вкл";
    this.toggleButton1.ForeColor = System.Drawing.Color.White;
    this.toggleButton1.Location =
        new System.Drawing.Point(72,88);
    this.toggleButton1.Name = "toggleButton1";
    this.toggleButton1.TabIndex = 0;
}

```

```

this.toggleButton1.Text = "Изкл";
this.toggleButton1.TextAlign =
    System.Drawing.ContentAlignment.MiddleCenter;
this.toggleButton1.UncheckedColor =
    System.Drawing.Color.Green;
this.toggleButton1.UncheckedText = "Изкл";
this.toggleButton1.Click +=
    new System.EventHandler(this.toggleClick);

// Form1
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(292, 266);
this.Controls.Add(this.toggleButton1);
this.Name = "Form1";
this.Text = "Включване в системата";
this.ResumeLayout(false);
}
#endregion

```

```

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

private void toggleClick(object sender, System.EventArgs e)
{
    MessageBox.Show
        ("Смяна на състоянието","Превключване");
}
}

```