

## Приложения за Windows

### Въведение в Windows формите

#### Приложения за Windows (графичен потребителски интерфейс – GUI)

- създават върху екрана прозорци;
- съдържат визуални елементи за:
  - а) изобразяване на информация;
  - б) взаимодействие на потребителя с програмата чрез мишка или клавиатура;
- класове от наименованото пространство `System.Windows.Forms`.

#### Компонента (`System.ComponentModel.Component`)

- базов клас;
- реализира интерфейса `IComponent` за дефиниране поведението на компонентите.

#### Контрол (`System.Windows.Forms.Control`)

- компонента с графична част;
- видим (компонентите без графична част не са видими).

#### Контейнер (`System.ComponentModel.Container`)

- капсулира компоненти;
- метод `Dispose` освобождава заетите ресурси от контейнера.

#### System.Object

#### System.MarshalByRefObject

#### System.ComponentModel.Component

#### System.Windows.Forms.Control

#### System.Windows.Forms.DataGrid

#### System.Windows.Forms.DateTimePicker

#### System.Windows.Forms.GroupBox

#### System.Windows.Forms.Label

#### System.Windows.Forms.ListControl

#### System.Windows.Forms.ListView

#### System.Windows.Forms.MonthCalendar

#### System.Windows.Forms.PictureBox

#### System.Windows.Forms.PrintPreviewControl

#### System.Windows.Forms.ProgressBar

#### System.Windows.Forms.ScrollBar

#### System.Windows.Forms.Splitter

#### System.Windows.Forms.StatusBar

#### System.Windows.Forms.TabControl

#### System.Windows.Forms.TextBoxBase

#### System.Windows.Forms.ToolBar

#### System.Windows.Forms.TrackBar

#### System.Windows.Forms.TreeView

#### Форма (`System.Windows.Forms.Form`)

- контрол-контейнер за компоненти и контроли;
- видове:

прозорец;

диалогов прозорец;

прозорец за интерфейс с много документи (MDI).

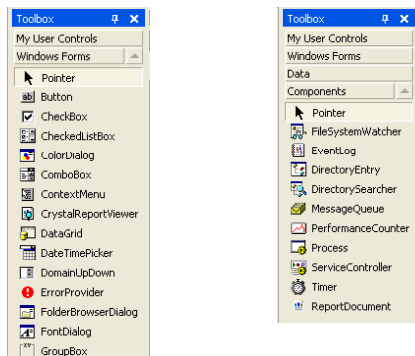
- свойства – дефинират външния ѝ вид;

- методи – дефинират нейното поведение;

- събития – дефинират взаимодействието на формата с потребителя.

## Контроли и компоненти в прозореца

### Toolbox



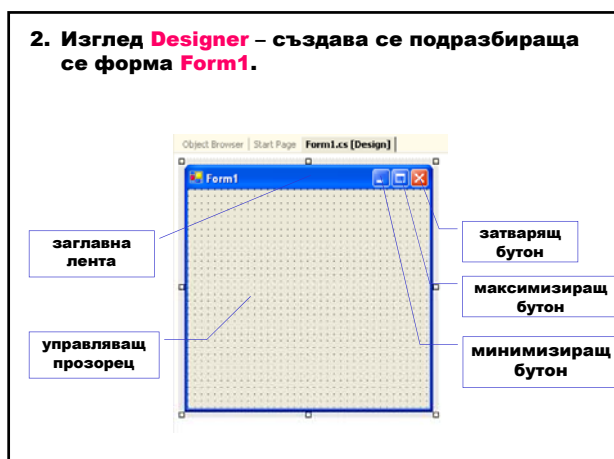
## Създаване на форма

### 1. Създаване на проект

- тип **Visual C# projects**
- шаблон **Windows Application**



**2. Изглед Designer – създава се подразбираща се форма Form1.**



**3. View ⇒ Code**

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
namespace MyWindowsForm
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.ComponentModel.Container 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
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.components =
        new System.ComponentModel.Container();
    this.Size = new System.Drawing.Size(300,300);
    this.Text = "Form1";
}
#endregion
[STAThread]
static void Main()
{
    Application.Run(new Form1());
}
}
```

**4. Изпълнение**

**Build ⇒ Build Solution**  
**Debug ⇒ Start Without Debugging**

**Методът Main създава и показва формата.**

**Методът Run на класа System.Windows.Forms.Application стартира стандартен цикъл на съобщения на приложението за текущата нишка и показва формата.**

**За изпълнение на формата трябва да се добави атрибутът [STAThread] към Main.**

**Приложението се затваря при затваряне на формата. Трябва да предефинира метода Dispose, който се извиква автоматично за главната форма на приложението; за всяка друга форма трябва да се извика явно.**

**Designer генерира код, затворен между директивите #region и #endregion, който не трябва да се модифицира или изтрива.**

### Свойства, методи и разполагане на контролите

Клас `Control` (`System.Windows.Forms`)

#### Свойства

<code>BackColor</code>	Фонов цвят на контрола.
<code>BackgroundImage</code>	Фоново изображение на контрола.
<code>Controls</code>	Дава колекцията от контроли, съдържащи се в контрола.
<code>Enable</code>	Разрешава/забранява ( <code>true/false</code> ) на потребителя да взаимодейства с контрола. Забраненият контрол „посивява“.
<code>Focused</code>	Определя ( <code>true/false</code> ) дали контролът има фокус.

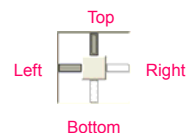
<code>Font</code>	Шрифт на изобразявания текст.
<code>ForeColor</code>	Цвят за изобразяване на текста.
<code>TabIndex</code>	Последователност от табулатори на контрола. При натискане на <code>&lt;Tab&gt;</code> фокусът се премества към контрола в нарастващ ред на табулаторите.
<code>TabStop</code>	Разрешава/забранява ( <code>true/false</code> ) използването на <code>&lt;Tab&gt;</code> за избор на контрола.
<code>Text</code>	Изобразяван текст в заглавната лента.
<code>TextAlign</code>	Подравняване на текста в контрола.
<code>Visible</code>	Показва ( <code>true/false</code> ) дали контролът е видим.

#### Методи

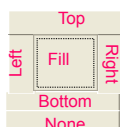
<code>Focus</code>	Премества фокуса към контрола.
<code>Hide</code>	Скрива контрола (установява <code>Visible</code> във <code>false</code> ).
<code>Show</code>	Изобразява контрола (установява <code>Visible</code> в <code>true</code> ).
<code>SuspendLayout</code>	Временно преустановява логиката на разполагането за контрола.
<code>ResumeLayout</code>	Връща логиката на нормалното разполагане на контрола.

#### Свойства на разположението на контролите във формата

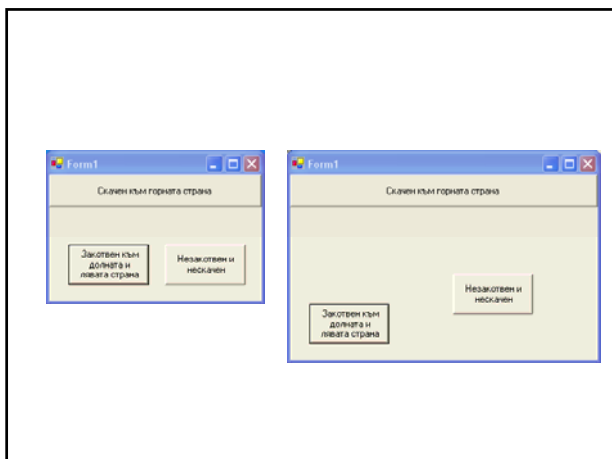
`Anchor` Страната на контейнера-родител, към която контролът се закотвя – контролите се разполагат на фиксирано разстояние от страните на контейнера, което не се променя при преоразмеряване. Стойностите могат да се комбинират.



`Dock` Страната на контейнера-родител за скачване („пристан“) на контрола. Контролите се разширяват според страните на контейнера. Стойностите не се комбинират. `Fill` запълва целия родител.



`DockPadding` Установява разстоянието от контролите до стените на контейнера. Подразбира се `0`.  
 (за контейнери)  
`Location` Разположението на горния ляв ъгъл на контрола спрямо своя контейнер.  
`Size` Размер на контрола. Използва се структурата `Size`, която има свойствата `Height` и `Width`.  
`MinimumSize` Минималният и максималният размер на формата.  
`MaximumSize` (за `Windows Forms`)



### Жизнен цикъл на форма – събития

1. Load
2. Activated
3. GotFocus
4. Closing
5. Closed
6. LostFocus
7. Deactivate
8. Dispose

#### 1. Създаване на форма

- инициализиране на променливите, преместване и преоразмеряване на контролите – кодът се добавя към конструктора след извикването на метода `InitializeComponent`;
  - събитието `Load` зарежда формата в паметта и я изобразява.
2. Формата става активна – вдига се събитието `Activated` (формата е видима или получава фокус от друга форма в същия проект).
3. Формата получава фокуса – вдига се събитието `GotFocus`.

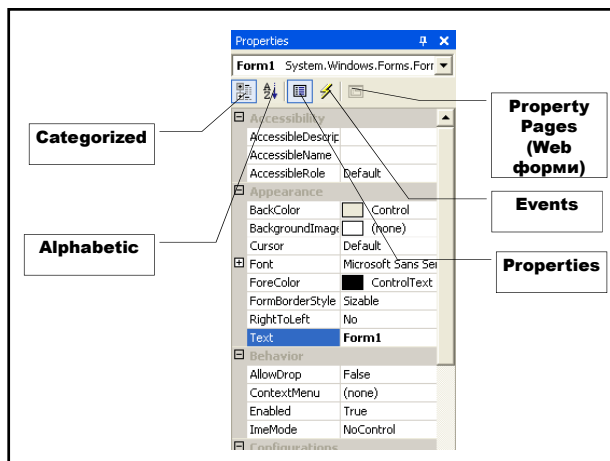
#### 4. Заявка за затваряне на форма – вдига се събитието `Closing` и формата се затваря. Ако формата проверява валидността на данните и данните не са коректни, тя остава отворена.

5. Затваряне на формата – събитието `Closed`.
6. Загубване на фокуса – събитието `LostFocus`.
7. Формата губи активността си – вдига се събитието `Deactivate` (формата е загубила фокуса).
8. Освобождава на ресурсите – събитието `Dispose`.

### Свойства на формите

#### Свойства

- в изглед `Designer` чрез прозореца `Properties`:  
`View` ⇒ `Properties Windows`
- чрез писане на код.  
`View` ⇒ `Code`



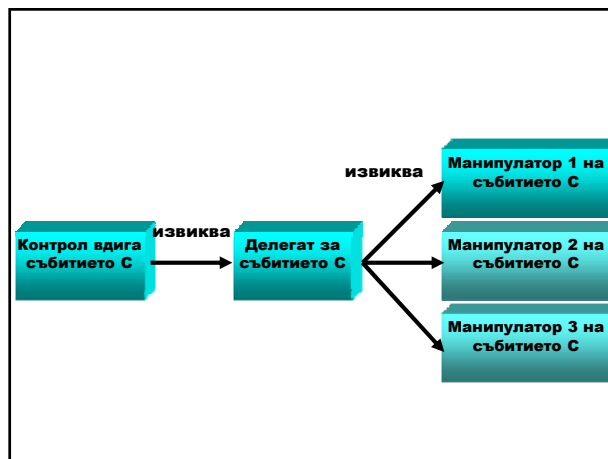
(Name)	Име на формата в проекта.
AcceptButton	Бутон при натискане на <Enter>.
AutoScaleBaseSize	Дава/установява основния размер за автоматично оразмеряване на формата
CancelButton	Бутон при натискане на <ESC>.
ClientSize	Дава/установява размера на клиентската част на формата (без рамките и заглавната лента).
ControlBox	Определя (true/false) изобразяването на бутоните Minimize, Maximize, Help и Close в заглавната лента.

FormBorderStyle	Определя (true/false) изобразяването на рамка около формата, бутоните и вида на заглавната лента (None, Sizable, Fix3D).
IsMdiContainer	Определя (true/false) дали формата е контейнер за интерфейс с много документи.
MaximizeBox	Определя (true/false) изобразяването на бутона Maximize в заглавната лента.
MinimizeBox	Определя (true/false) изобразяването на бутона Minimize в заглавната лента.
StartPosition	Определя мястото на формата върху екрана при първоначалното ѝ появяване.

Size	Размер на формата (по подразбиране 300; 300).
------	---

Методи на формите	
Close	Затваря формата.
Dispose	Освобождава ресурсите, заети от компонентите.
LayoutMdi	Аранжира интерфейс с много документи – MdiLayout параметърът показва начина на аранжиране на документите в клиентската област на формата-родител: ArrangIcons – всички документи са вътре в клиентската област, Cascade – каскадно разполагане, TileHorizontal – хоризонтално разполагане, TileVertical – вертикално разполагане.

Обработка на събития	
<b>Събитие</b> – съобщение, което даден контрол изпраща, за да сигнализира за действие, предизвикано от потребителя или програмата:	
– източник	– генерира събитието;
– приемник	– обработва събитието;
– делегат	– свързва източника с приемника и се обръща към метод за обработка на събитието – <b>манипулатор на събитието</b> .



### Манипулатор на събитие за контрол в .NET

#### 1. Създаване на манипулатор на събитие

Изглед **Designer** ⇒ <L> контрол-источник ⇒ <L<sup>2</sup>> ⇒  
 създава се манипулатор с име <име на контрол>\_<име на събитие по подразбиране>  
 или  
 Изглед **Designer** ⇒ <L> контрол-источник ⇒ Прозорец **Properties** ⇒ ⚡ ⇒ <L> събитие ⇒  
 въвеждаме текст <Enter> ⇒ създава се манипулатор с име <име на събитие>

#### 2. Автоматично регистриране на манипулатора на събитието в метода `InitializeComponent()`

```
this.<име на контрол>.<име на събитие> +=  
new System.EventHandler(this.<име на манипулатор>);
```

#### 3. Автоматично добавяне на метода за обработка на събитието, който реализира програмната логика на манипулатора

```
private void <име на манипулатор>(object sender,  
System.EventArgs e)  
{  
    // потребителски код за обработка на събитието  
}
```

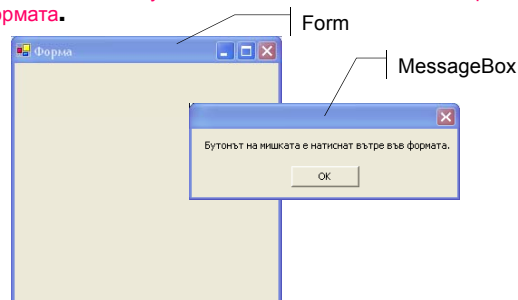
#### Добавяне на манипулатор на събитие на контрол по време на изпълнение

```
this.<име на контрол>.<име на събитие> +=  
new System.EventHandler(this.<име на манипулатор>);
```

#### Премахване на манипулатор на събитие на контрол по време на изпълнение

```
this.<име на контрол>.<име на събитие> -=  
new System.EventHandler(this.<име на манипулатор>);
```

**Пример:** – приложение за Windows, което обработва събитието **натискане бутон на мишката** – изобразява се прозорец със съобщението Бутонът на мишката е натиснат вътре във формата.



Клас `MessageBox` (`System.Windows.Forms`) изобразява прозорец със съобщение, който може да съдържа текст, бутони и символи за информиране и инструктиране на потребителя.

Методът `MessageBox.Show` изобразява прозорец за съобщение с определен текст `text`.

```
public static DialogResult Show (string text );
```

Изобразява диалогов прозорец за съобщение с определен текст `text`, заглавие `caption`, бутони `buttons` (`Abort`, `RetryIgnore`, `OK`, `OKCancel`, `RetryCancel`, `YesNo`, `YesNoCancel`) и икона `icon` (`Asterisk`, `Error`, `Exclamation`, `Hand`, `Information`, `None`, `Question`, `Stop`, `Warning`).

```
public static DialogResult Show( string text, string caption,  
    MessageBoxButtons buttons, MessageBoxIcon icon );
```

```
using System;  
using System.Drawing;  
using System.Collections;  
using System.ComponentModel;  
using System.Windows.Forms;  
using System.Data;  
namespace MyForm  
{  
    public class MyForm : System.Windows.Forms.Form  
    {  
        private System.ComponentModel.IContainer components =  
            null;  
        public MyForm()  
        {  
            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.AutoScaleBaseSize =
        new System.Drawing.Size(5, 13);
    this.ClientSize = new System.Drawing.Size(292, 266);
    this.Name = "MyForm";
    this.Text = "Форма";
    this.Click +=
        new System.EventHandler(this.MyForm_Click);
}
#endregion

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

```
private void MyForm_Click(object sender,
                        System.EventArgs e)
{
    MessageBox.Show
    ("Бутонът на мишката е натиснат вътре във формата.");
}
}
```

### Работа с контроли

**Контроли** – компоненти, които се използват многократно в Windows приложенията и капсулират функционалността на потребителския интерфейс.

#### Добавяне на контроли във формата

1. **Метод Add** на класа `Form.ContainerCollection`, който представя колекцията от контроли във формата.  
`public override void Add (Control value);`
2. **Метод AddRange** на класа `Form.ContainerCollection`, като добавя масив от контроли към колекцията.  
`public virtual void AddRange (Control[] controls);`

#### Използване на контроли във формите

1. **Дефиниране на контрол**  
`private <Контрол> <контрол>;`
2. **Създаване на контрол**  
`this.<контрол> = new <Контрол>();`
3. **Установяване свойствата на контрола**  
`this.<контрол>.Location=new System.Drawing.Point(10,10);`  
`this.<контрол>.Name = "myControl";`  
`this.<контрол>.Size = new System.Drawing.Size(50, 20);`  
`this.<контрол>.TabIndex = 0;`  
`this.<контрол>.Text = "Моят контрол";`  
 ...

4. **Добавяне на контрол към формата**  
`this.Controls.Add(this.<контрол>);`  
 или  
`this.Controls.AddRange`  
`(new System.Windows.Forms.Control[] {this.<контрол>});`
5. **Добавяне на манипулатор на събитие за контрола**  
`this.<контрол>.<събитие> +=`  
`new System.EventHandler(this.<манипулатор>);`
6. **Реализиране на манипулатора**  
`private void <манипулатор> (object sender,`  
`System.EventArgs e)`  
`{`  
`// код за обработка на събитието`  
`}`

**Видове контроли според тяхната функционалност:**

**1. Командни контроли**

- **Button** – бутон за пускане, спиране или превключване на процес;
- **ToolBar** – лента с набор от контроли-бутони.

**2. Текстови контроли**

- **TextBox** – текст, въведен по време на проектиране; редактира се при изпълнение; променя се програмно;
- **RichTextBox** – текст с форматиране;
- **Label** – текст, който не се редактира директно;
- **StatusBar** – информация за текущото състояние на приложението – обикновено в долната част на прозореца-родител.

**3. Контроли за избор на стойност от списък**

- **CheckedListBox** – списък с плъзгач от елементи, като към всеки елемент има бутон за избор;
- **ComboBox** – падащ списък от елементи;
- **ListBox** – списък с текстови и графични елементи;
- **TreeView** – йерархично дърво от възли, които съдържат текст с бутон за избор или икона.

**4. Контроли за установяване на стойност**

- **CheckBox** – прозорец и етикет с текст;
- **RadioButton** – бутон в състояние включено/изключено;
- **DateTimePicker** – графичен календар за избор на дата или време;
- **MonthCalendar** – графичен календар за избор на област от дати.

**5. Контроли-менюта**

- **MainMenu** – осигурява интерфейс при проектиране за създаване на менюта;
- **ContextMenu** – реализира меню, което се появява при натискане десния бутон на мишката.

**6. Диалогови прозорци**

- **ColorDialog** – диалогов прозорец за избор на цвят от палитра с цветовете;
- **FontDialog** – диалогов прозорец за избор на шрифт и неговите атрибути;
- **OpenFileDialog** – диалогов прозорец за разглеждане и избор на файл;
- **PrintDialog** – диалогов прозорец за избор на принтер и неговите атрибути;
- **PrintPreviewDialog** – диалогов прозорец за вида на документа при отпечатване;

- **PageSetupDialog** – диалогов прозорец за установяване размерите на страницата;
- **SaveFileDialog** – диалогов прозорец за запазване на файл.

**7. Контейнери**

- **Panel** – групира контроли в рамка;
- **GroupBox** – групира контроли (напр. радиобутони) в рамка с етикет без плъзгач;
- **TabControl** – осигурява табулирана страница за организиране и ефективен достъп до групирани обекти.

**8. Контроли-графики**

- **ImageList** – съхранява изображения;
- **PictureBox** – изобразява графични файлове (битмапи и икони).

**Контрол Label**

Изобразява текст или изображение, които не могат да се редактират от потребителя.

Компютърни системи и технологии

**Свойства**

- Text** – заглавие на етикета;
- Alignment** – подравняване на текста;
- AutoSize** – true за динамично оразмеряване на етикета;
- TabIndex** – ред на контрола при използване на табулатор в неговия контейнер;
- UseMnemonic** true за установяване клавиш за достъп – клавиш, пред който е написан символа & в свойството **Text**.

- Не получава фокус.
- Създаване на клавиш за достъп до други контроли:

1. В свойството **Text** на другия контрол се използва **&**.
2. Преди контрола се добавя етикет със свойство **TabIndex** с 1 по-малко от това на контрола.
3. Свойството **UseMnemonic** на етикета се установява в **true**.



### Контрол Button

Изобразява бутон, който извършва действие при натискането му.



#### Свойства

**Text** текст върху бутон; чрез & се използва за клавиш за достъп (ALT+клавиш);

### Събития

**Click** при натискане на бутона; подразбиращо се събитие при <L2> в изглед Designer;

**MouseEnter** при въвеждане на контрола от указателя на мишката;

**MouseDown** указателят на мишката е над контрола и бутонът се натисне;

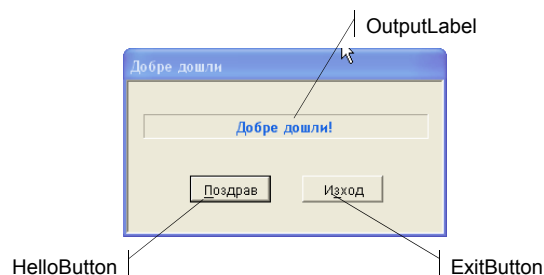
**MouseUp** указателят на мишката е над контрола и бутонът се отпусне.

Не поддържа събитие за двукратно натискане на мишката.

### Начини за избор на бутон:

1. Натискане на бутон чрез мишка.
2. Извикване на събитието Click в код.
3. Преместване на фокуса върху бутон чрез натискане на <TAB>, след което се натиска <SPACEBAR> или <ENTER>.
4. Натискане на клавиш за достъп (<ALT> + подчертан символ).
5. Натискане на <ENTER> за потвърждаващ бутон.
6. Натискане на <ESC> за отменящ бутон.
7. Програмно натискане на бутон чрез извикване на метода Button.PerformClick.

**Пример:** Създава форма без бутони в заглавната лента с един етикет и два бутона. При натискане на бутона Поздрав (ALT+П) в етикета се изобразява текстът Добре дошли!, а при натискане на бутона Изход (ALT+з) формата се затваря.



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

namespace Welcome
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.Label OutputLabel;
        private System.Windows.Forms.Button HelloButton;
        private System.Windows.Forms.Button ExitButton;
        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.OutputLabel = new System.Windows.Forms.Label();
    this.HelloButton = new System.Windows.Forms.Button();
    this.ExitButton = new System.Windows.Forms.Button();
    this.SuspendLayout();
}
```

```
// Етикет OutputLabel
this.OutputLabel.BorderStyle =
    System.Windows.Forms.BorderStyle.Fixed3D;
// Шрифт Trebuchet MS, Bold, 10pt, Cyrillic
this.OutputLabel.Font = new System.Drawing.Font
("Trebuchet MS", 9.75F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point,
(System.Byte)(204));
// Цвет System, ActiveCaption (син)
this.OutputLabel.ForeColor =
    System.Drawing.SystemColors.ActiveCaption;
this.OutputLabel.Location =
    new System.Drawing.Point(14, 30);
this.OutputLabel.Name = "OutputLabel";
this.OutputLabel.Size =
    new System.Drawing.Size(264, 23);
```

```
this.OutputLabel.TabIndex = 0;
this.OutputLabel.TextAlign =
    System.Drawing.ContentAlignment.MiddleCenter;
// Бутон HelloButton
this.HelloButton.Location =
    new System.Drawing.Point(57, 87);
this.HelloButton.Name = "HelloButton";
this.HelloButton.Size = new System.Drawing.Size(75, 25);
this.HelloButton.TabIndex = 1;
this.HelloButton.Text = "&Поздрав";
this.HelloButton.Click +=
    new System.EventHandler(this.HelloButton_Click);
```

```
// Бутон ExitButton
this.ExitButton.DialogResult =
    System.Windows.Forms.DialogResult.Cancel;
this.ExitButton.Location =
    new System.Drawing.Point(161, 87);
this.ExitButton.Name = "ExitButton";
this.ExitButton.Size = new System.Drawing.Size(75, 25);
this.ExitButton.TabIndex = 2;
this.ExitButton.Text = "И&зход";
this.ExitButton.Click +=
    new System.EventHandler(this.ExitButton_Click);
// Форма Form1
this.AcceptButton = this.HelloButton;
this.AutoScaleBaseSize =
    new System.Drawing.Size(6, 16);
```

```
this.CancelButton = this.ExitButton;
this.ClientSize = new System.Drawing.Size(290, 139);
this.ControlBox = false;
this.Controls.Add(this.ExitButton);
this.Controls.Add(this.HelloButton);
this.Controls.Add(this.OutputLabel);
this.Font = new System.Drawing.Font("Trebuchet MS",
9.75F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point,
((System.Byte)(204)));
this.FormBorderStyle =
    System.Windows.Forms.FormBorderStyle.Fixed3D;
this.Name = "Form1";
this.Text = "Добре дошли";
this.ResumeLayout(false);
}
#endregion
```

```
[STAThread]
static void Main()
{
    Application.Run(new Form1());
}
private void HelloButton_Click(object sender,
System.EventArgs e)
{
    OutputLabel.Text = "Добре дошли!";
}
private void ExitButton_Click(object sender,
System.EventArgs e)
{
    this.Close();
}
}
```

### Контрол **CheckBox**

Контрол с две състояния (*true/false*). Дава възможност за избор – истина/лъжа или да/не.

Компютърни системи и технологии

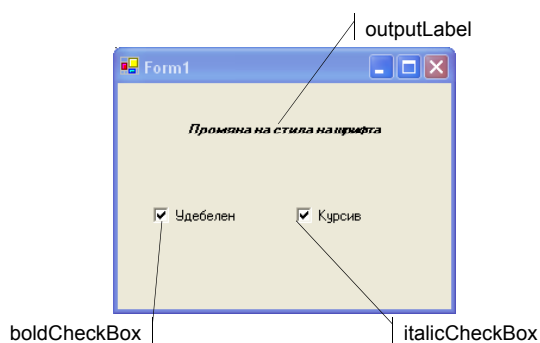
#### Свойства

- Checked**      дава/установява стойността на контрола (*true/false*);
- CheckState**    *Checked, Unchecked, Indeterminate* (с проверка и със сянка);
- CheckStatus**    дали е включена проверка (*Checked, Unchecked, Indeterminate*);
- Text**            изобразяван текст от дясно на контрола.

### Събития

- CheckedChanged**    вдига се при промяна на състоянието на контрола; подразбиращо се събитие при **<L<sup>2</sup>>** в изглед **Designer**.
- CheckStateChanged**    вдига се при промяна на свойството **CheckState**.

**Пример:** Форма за промяна стила на шрифта – чрез използване на поле за избор **CheckBox**.



Клас **Font (System.Drawing)** – дефинира шрифт с определено име **familyName**, размер **emSize** и стил **style**.

```
public Font( string familyName, float emSize, FontStyle style );
```

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

namespace CheckBoxes
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.Label outputLabel;
        private System.Windows.Forms.CheckBox boldCheckBox;
        private System.Windows.Forms.CheckBox italicCheckBox;
        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.outputLabel = new System.Windows.Forms.Label();
    this.boldCheckBox=new System.Windows.Forms.CheckBox();
    this.italicCheckBox=new System.Windows.Forms.CheckBox();
    this.SuspendLayout();
    // outputLabel
    this.outputLabel.Location = new System.Drawing.Point(56,32);
    this.outputLabel.Name = "outputLabel";
    this.outputLabel.Size = new System.Drawing.Size(176, 23);
    this.outputLabel.TabIndex = 0;
    this.outputLabel.Text = "Промяна на стила на шрифта";
}
```

```
// boldCheckBox
this.boldCheckBox.Location =
    new System.Drawing.Point(30,100);
this.boldCheckBox.Name = "boldCheckBox";
this.boldCheckBox.TabIndex = 1;
this.boldCheckBox.Text = "Удебелен";
this.boldCheckBox.CheckedChanged += new
System.EventHandler(this.boldCheckBox_CheckedChanged);
// italicCheckBox
this.italicCheckBox.Location =
    new System.Drawing.Point(150, 100);
this.italicCheckBox.Name = "italicCheckBox";
this.italicCheckBox.TabIndex = 2;
this.italicCheckBox.Text = "Курсив";
this.italicCheckBox.CheckedChanged += new
System.EventHandler(this.italicCheckBox_CheckedChanged);
```

```
// Form1
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(280, 190);
this.Controls.Add(this.italicCheckBox);
this.Controls.Add(this.boldCheckBox);
this.Controls.Add(this.outputLabel);
this.Name = "Form1";
this.Text = "Form1";
this.ResumeLayout(false);
}
#endregion
[STAThread]
static void Main()
{
    Application.Run(new Form1());
}
```

```
private void boldCheckBox_CheckedChanged(object sender,
    System.EventArgs e)
{
    outputLabel.Font=new Font(outputLabel.Font.Name,
        outputLabel.Font.Size,
        outputLabel.Font.Style^FontStyle.Bold);
}
private void italicCheckBox_CheckedChanged(object sender,
    System.EventArgs e)
{
    outputLabel.Font=new Font(outputLabel.Font.Name,
        outputLabel.Font.Size,
        outputLabel.Font.Style^FontStyle.Italic);
}
}
```

### Контрол **TextBox**

Въвежда или изобразява текст.



#### Свойства

**Text** дава/установява текущия текст в текстовия прозорец;

**PasswordChar** символ, който се изобразява при въвеждане на текст – за въвеждане на парола;

**Lines** дава/установява редовете с текст в контрола.

#### Методи

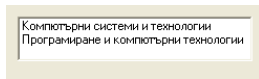
**Clear** изчиства съдържанието на текстовия прозорец.

#### Събития

**TextChanged** вдига се при промяна на текста; подразбиращо се събитие при <L<sup>2</sup>> в изглед **Designer**.

### Контрол **ListBox**

Изобразява списък от елементи, които се избират чрез двукратно натискане на левия бутон на мишката. **ListBox** не позволява добавяне на нови елементи списъка.



#### Свойства

- Items** съдържа колекция от данни в контрола;
- MultiColumn** (**true/false**) позволява представяне на списъка в много колони;

- SelectedIndex** връща индекса на избрания елемент (-1, ако не е избран елемент);
- SelectedIndices** връща колекция от индексите на избраните елементи;
- SelectedItem** връща избрания елемент;
- SelectedItems** връща колекция от избраните елементи;
- SelectionMode** определя броя на избраните елементи (**None**, **One**, **MultiSimple** и **MultiExtended** – с комбинация от клавиши-стрелки, <Shift> и управляващи клавиши);
- Sorted** Определя дали (**true/false**) елементите са сортирани в списъка.

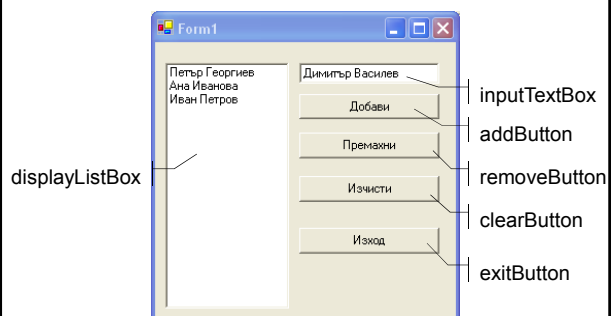
#### Методи

- Clear** изчиства колекцията;
- Add** добавя обект към колекцията;
- Remove** премахва даден обект;
- RemoveAt** премахва обект с даден индекс;
- GetSelected** (**true/false**) при зададен индекс връща **true**, ако елементът е избран.

#### Събития

- SelectedIndexChanged** вдига се при промяна на избрания индекс; подразбиращото се събитие при <L2> в режим **Designer**.

**Пример:** Форма за добавяне, премахване и изчистване на елементи от списък – чрез **ListBox**.



```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
namespace ListBoxes
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.ListBox displayListBox;
        private System.Windows.Forms.TextBox inputTextBox;
        private System.Windows.Forms.Button addButton;
        private System.Windows.Forms.Button removeButton;
        private System.Windows.Forms.Button clearButton;
        private System.Windows.Forms.Button exitButton;
        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.displayListBox = new System.Windows.Forms.ListBox();
    this.inputTextBox = new System.Windows.Forms.TextBox();
    this.addButton = new System.Windows.Forms.Button();
    this.removeButton = new System.Windows.Forms.Button();
    this.clearButton = new System.Windows.Forms.Button();
    this.exitButton = new System.Windows.Forms.Button();
    this.SuspendLayout();

    // displayListBox
    this.displayListBox.Location =
        new System.Drawing.Point(10, 20);
    this.displayListBox.Name = "displayListBox";
    this.displayListBox.Size =
        new System.Drawing.Size(120,238);
    this.displayListBox.TabIndex = 0;
```

```
// inputTextBox
this.inputTextBox.Location =
    new System.Drawing.Point(140,20);
this.inputTextBox.Name = "inputTextBox";
this.inputTextBox.Size = new System.Drawing.Size(136, 20);
this.inputTextBox.TabIndex = 1;
this.inputTextBox.Text = "";

// addButton
this.addButton.Location=new System.Drawing.Point(140,50);
this.addButton.Name = "addButton";
this.addButton.Size = new System.Drawing.Size(136, 25);
this.addButton.TabIndex = 2;
this.addButton.Text = "Добави";
this.addButton.Click +=
    new System.EventHandler(this.addButton_Click);
```

```
// removeButton
this.removeButton.Location =
    new System.Drawing.Point(140, 88);
this.removeButton.Name = "removeButton";
this.removeButton.Size = new System.Drawing.Size(136,25);
this.removeButton.TabIndex = 3;
this.removeButton.Text = "Премахни";
this.removeButton.Click +=
    new System.EventHandler(this.removeButton_Click);

// clearButton
this.clearButton.Location =
    new System.Drawing.Point(140, 130);
this.clearButton.Name = "clearButton";
this.clearButton.Size = new System.Drawing.Size(136, 25);
this.clearButton.TabIndex = 4;
this.clearButton.Text = "Изчисти";
this.clearButton.Click +=
    new System.EventHandler(this.clearButton_Click);
```

```
// exitButton
this.exitButton.Location =
    new System.Drawing.Point(140, 180);
this.exitButton.Name = "exitButton";
this.exitButton.Size = new System.Drawing.Size(136, 25);
this.exitButton.TabIndex = 5;
this.exitButton.Text = "Изход";
this.exitButton.Click +=
    new System.EventHandler(this.exitButton_Click);

// Form1
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(292, 266);
this.Controls.Add(this.exitButton);
this.Controls.Add(this.clearButton);
this.Controls.Add(this.removeButton);
this.Controls.Add(this.addButton);
this.Controls.Add(this.inputTextBox);
this.Controls.Add(this.displayListBox);
```

```
this.Name = "Form1";
this.Text = "Form1";
this.ResumeLayout(false);
}
#endregion

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

```
private void addButton_Click(object sender,
    System.EventArgs e)
{
    displayListBox.Items.Add(inputTextBox.Text);
    inputTextBox.Clear();
}

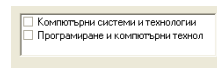
private void removeButton_Click(object sender,
    System.EventArgs e)
{
    if(displayListBox.SelectedIndex != -1)
        displayListBox.Items.RemoveAt
            (displayListBox.SelectedIndex);
}
```

```
private void clearButton_Click(object sender,
                               System.EventArgs e)
{
    displayListBox.Items.Clear();
}

private void exitButton_Click(object sender,
                               System.EventArgs e)
{
    Application.Exit();
}
}
```

### Контрол **CheckedListBox**

Изобразява списък от елементи с възможност за проверка пред всеки елемент; избират се чрез двукратно натискане на левия бутон на мишката.



#### Свойства

- CheckedItems** връща колекция от проверените елементи;
- CheckedIndices** връща индексите на проверените елементи;
- SelectionMode** определя броя на проверяваните елементи (**None**, **One** – много елементи за проверка).

**Методи**

**GetItemChecked** при зададен индекс връща **true**, ако елементът е проверен.

**Събития**

**ItemCheck** вдига се при промяна на проверката на елемент; подразбиращото се събитие при <L2> в режим **Designer**.

**Свойства на ItemCheckEventArgs**

- CurrentValue** показва дали елементът е проверен/непроверен (**Checked**, **Unchecked**, **Indeterminate**);
- Index** връща индекса на променения елемент;
- NewValue** определя ново състояние на елемента.

**Пример:** Форма за изобразяване на избрани елементи от списък – чрез **CheckedListBox**. Добавя/премахва в контрола **ListBox** избрания елемент от контрола **CheckedListBox**.

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

namespace CheckListBoxes
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.CheckedListBox
            inputCheckedListBox;
        private System.Windows.Forms.ListBox displayListBox;
        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.inputCheckedListBox =
        new System.Windows.Forms.CheckedListBox();
    this.displayListBox =
        new System.Windows.Forms.ListBox();
    this.SuspendLayout();
    // inputCheckedListBox
    this.inputCheckedListBox.Items.AddRange(new object[] {
        "C++", "Java", "C#", "VB", "Perl", "Python"});
    this.inputCheckedListBox.Location =
        new System.Drawing.Point(20, 20);
    this.inputCheckedListBox.Name = "inputCheckedListBox";
    this.inputCheckedListBox.Size =
        new System.Drawing.Size(120, 199);
    this.inputCheckedListBox.TabIndex = 0;
    this.inputCheckedListBox.ItemCheck += new
        System.Windows.Forms.ItemCheckEventHandler
        (this.inputCheckedListBox_ItemCheck);
}
```

```
// displayListBox
this.displayListBox.Location =
    new System.Drawing.Point(160, 20);
this.displayListBox.Name = "displayListBox";
this.displayListBox.Size =
    new System.Drawing.Size(120,199);
this.displayListBox.TabIndex = 1;
// Form1
this.AutoScaleBaseSize = new System.Drawing.Size(5,13);
this.ClientSize = new System.Drawing.Size(292, 266);
this.Controls.Add(this.displayListBox);
this.Controls.Add(this.inputCheckedListBox);
this.Name = "Form1";
this.Text = "Form1";
this.ResumeLayout(false);
}
#endregion
```

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

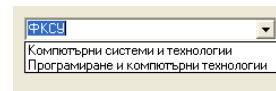
private void inputCheckedListBox_ItemCheck(object sender,
    System.Windows.Forms.ItemCheckEventArgs e)
{
    string item=inputCheckedListBox.SelectedItem.ToString();
    if(e.NewValue==CheckState.Checked)
        displayListBox.Items.Add(item);
    else
        displayListBox.Items.Remove(item);
}
}
```

## Контрол **ComboBox**

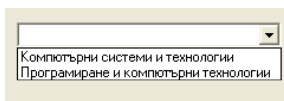
Изобразява падащ списък от елементи.

### Свойства

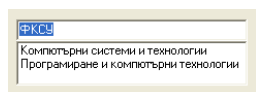
**DropDownStyle** определя стила:  
**DropDown** – текстът в прозореца  
 може да се редактира;  
 за изобразяване на списъка  
 трябва да се натисне бутонът  
 стрелка;



**DropDownList** – текстът не може да се  
 редактира директно;  
 за изобразяване на списъка  
 трябва да се натисне бутонът  
 стрелка;



**Simple** – текстът може да се  
 редактира; списъкът е видим  
 винаги.



**Items** съдържа колекция от данни  
 в контрола.

**MaxDropDownItems** определя максималния брой  
 елементи (1 до 100);  
 при надвишаване на  
 максималния брой се  
 появява плъзгач;

**SelectedIndex** връща индекса на избрания  
 елемент (-1, ако не избран  
 елемент);

**SelectedItem** връща избрания елемент;

**Sorted** Определя дали (**true/false**)  
 елементите са сортирани в  
 списъка.



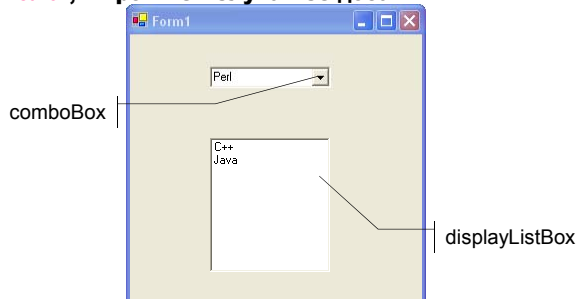
**Методи**

**Clear**            изчиства колекцията;  
**Add**             добавя обект към колекцията;  
**Remove**         премахва даден обект.

**Събития**

**SelectedIndexChanged**    вдига се при промяна на избрания индекс; подразбиращото се събитие при <L2> в режим **Designer**.

**Пример:** Форма за изобразяване на избрани елементи от списък – чрез **ComboBox**. Ако избраният елемент от **ComboBox** съвпада с елемент от **ListBox**, елементът се изтрива от **ListBox**, в противен случай се добавя.



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

namespace CheckListBoxes
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.ListBox displayListBox;
        private System.Windows.Forms.ComboBox comboBox;

        private System.ComponentModel.Container 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.comboBox =
        new System.Windows.Forms.ComboBox();
    this.displayListBox =
        new System.Windows.Forms.ListBox();
    this.SuspendLayout();

    // displayListBox
    this.displayListBox.Location =
        new System.Drawing.Point(80, 104);
    this.displayListBox.Name = "displayListBox";
    this.displayListBox.Size =
        new System.Drawing.Size(120, 134);
    this.displayListBox.TabIndex = 1;
}
```

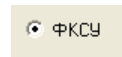
```
// comboBox
this.comboBox.Items.AddRange(new object[] {
    "C++", "Java", "C#", "VB", "Perl", "Python"});
this.comboBox.Location=new System.Drawing.Point(80,32);
this.comboBox.Name = "comboBox";
this.comboBox.Size = new System.Drawing.Size(121, 21);
this.comboBox.TabIndex = 2;
this.comboBox.Text = "Избери";
this.comboBox.SelectedIndexChanged +=
    new System.EventHandler
    (this.comboBox_SelectedIndexChanged);

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

```
[STAThread]
static void Main()
{
    Application.Run(new Form1());
}
private void comboBox_SelectedIndexChanged(object sender,
    System.EventArgs e)
{
    string item=comboBox.SelectedItem.ToString();
    for(int i=0; i<displayListBox.Items.Count; i++)
        if(displayListBox.Items[j]==item)
        {
            displayListBox.Items.Remove(item);
            return;
        }
    displayListBox.Items.Add(item);
}
}
```

### Контрол **RadioButton**

Контрол с две състояния (**true/false**) – възможност за избор/изчистване. При разполагане на радиобутоните в група (чрез **GroupBox** или **Panel**) изборът на един радиобутон води до изчистване на останалите в групата.



#### Свойства

**Checked** дава/установява стойността на контрола (**true/false**);

**Text** изобразяван текст от дясната страна на радио бутона.

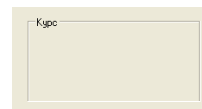
**Събития**

**Click** при натискане на бутона;

**CheckedChanged** вдига се всеки път, когато радио бутонът си промени състоянието; подразбиращото се събитие при <L<sup>2</sup>> в режим **Designer**.

### Контрол **GroupBox**

Контрол, който съдържа и аранжира други компоненти.



#### Свойства

**Controls** връща колекция от съдържащите се контроли;

**Text** изобразявания текст в горната част на контрола.

**Пример: Форма за избор на типа на прозореца със съобщения **MessageBox** – чрез радио бутони се избира типа на бутоните и иконата.**

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

namespace RadioButtons
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.Label promptLabel;
        private System.Windows.Forms.GroupBox groupButtons;
        private System.Windows.Forms.GroupBox groupIcons;
        private System.Windows.Forms.RadioButton okButton;
        private System.Windows.Forms.RadioButton okCancelButton;
        private System.Windows.Forms.RadioButton abortRetryIgnoreButton;
    }
}
```

```
private System.Windows.Forms.RadioButton
yesNoCancelButton;
private System.Windows.Forms.RadioButton yesNoButton;
private System.Windows.Forms.RadioButton
retryCancelButton;
private System.Windows.Forms.RadioButton asteriskButton;
private System.Windows.Forms.RadioButton errorButton;
private System.Windows.Forms.RadioButton
exclamationButton;
private System.Windows.Forms.RadioButton handButton;
private System.Windows.Forms.RadioButton
informationButton;
private System.Windows.Forms.RadioButton noneButton;
private System.Windows.Forms.RadioButton questionButton;
private System.Windows.Forms.RadioButton stopButton;
private System.Windows.Forms.RadioButton warningButton;
```

```
private System.Windows.Forms.Button displayButton;
private System.Windows.Forms.Label displayLabel;
// Избор на потребителя
private MessageBoxButtons buttonType =
MessageBoxButtons.OK;
private MessageBoxIcon iconType=MessageBoxIcon.Error;
private System.ComponentModel.Container 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.promptLabel = new System.Windows.Forms.Label();
this.groupButtons = new System.Windows.Forms.GroupBox();
this.groupIcons = new System.Windows.Forms.GroupBox();
this.okButton = new System.Windows.Forms.RadioButton();
this.okCancelButton =
new System.Windows.Forms.RadioButton();
this.abortRetryIgnoreButton =
new System.Windows.Forms.RadioButton();
this.yesNoCancelButton =
new System.Windows.Forms.RadioButton();
this.yesNoButton =
new System.Windows.Forms.RadioButton();
this.retryCancelButton =
new System.Windows.Forms.RadioButton();
this.errorButton = new System.Windows.Forms.RadioButton();
```

```
this.exclamationButton =
new System.Windows.Forms.RadioButton();
this.informationButton =
new System.Windows.Forms.RadioButton();
this.questionButton =
new System.Windows.Forms.RadioButton();
this.handButton = new System.Windows.Forms.Button();
this.displayButton = new System.Windows.Forms.Button();
this.displayLabel = new System.Windows.Forms.Label();
this.asteriskButton =
new System.Windows.Forms.RadioButton();
this.handButton = new System.Windows.Forms.RadioButton();
this.noneButton = new System.Windows.Forms.RadioButton();
this.stopButton = new System.Windows.Forms.RadioButton();
this.warningButton =
new System.Windows.Forms.RadioButton();
this.groupButtons.SuspendLayout();
this.groupIcons.SuspendLayout();
this.SuspendLayout();
```

```
// promptLabel
this.promptLabel.Location=new System.Drawing.Point(24,24);
this.promptLabel.Name = "promptLabel";
this.promptLabel.Size = new System.Drawing.Size(248, 23);
this.promptLabel.TabIndex = 0;
this.promptLabel.Text =
"Изберете типа на Прозореца със съобщения";

// groupButtons
this.groupButtons.Controls.Add(this.retryCancelButton);
this.groupButtons.Controls.Add(this.yesNoButton);
this.groupButtons.Controls.Add(this.yesNoCancelButton);
this.groupButtons.Controls.Add(this.abortRetryIgnoreButton);
this.groupButtons.Controls.Add(this.okCancelButton);
this.groupButtons.Controls.Add(this.okButton);
this.groupButtons.Location =
new System.Drawing.Point(20,50);
this.groupButtons.Name = "groupButtons";
```

```

this.groupButtons.Size = new System.Drawing.Size(136, 150);
this.groupButtons.TabIndex = 1;
this.groupButtons.TabStop = false;
this.groupButtons.Text = "Тип на бутона";

// groupIcons
this.groupIcons.Controls.Add(this.warningButton);
this.groupIcons.Controls.Add(this.stopButton);
this.groupIcons.Controls.Add(this.noneButton);
this.groupIcons.Controls.Add(this.handButton);
this.groupIcons.Controls.Add(this.asteriskButton);
this.groupIcons.Controls.Add(this.questionButton);
this.groupIcons.Controls.Add(this.informationButton);
this.groupIcons.Controls.Add(this.exclamationButton);
this.groupIcons.Controls.Add(this.errorButton);
this.groupIcons.Location=new System.Drawing.Point(170,50);
this.groupIcons.Name = "groupIcons";
this.groupIcons.Size = new System.Drawing.Size(136, 206);
    
```

```

this.groupIcons.TabIndex = 2;
this.groupIcons.TabStop = false;
this.groupIcons.Text = "Икона";

// okButton
this.okButton.Location = new System.Drawing.Point(20, 20);
this.okButton.Name = "okButton";
this.okButton.TabIndex = 0;
this.okButton.Text = "OK";
this.okButton.CheckedChanged += new
    System.EventHandler(this.buttonType_CheckedChange);

// okCancelButton
this.okCancelButton.Location =
    new System.Drawing.Point(20,40);
this.okCancelButton.Name = "okCancelButton";
this.okCancelButton.TabIndex = 1;
this.okCancelButton.Text = "OKCancel";
this.okCancelButton.CheckedChanged += new
    System.EventHandler(this.buttonType_CheckedChange);
    
```

```

// abortRetryIgnoreButton
this.abortRetryIgnoreButton.Location =
    new System.Drawing.Point(20, 60);
this.abortRetryIgnoreButton.Name = "abortRetryIgnoreButton";
this.abortRetryIgnoreButton.Size =
    new System.Drawing.Size(112, 24);
this.abortRetryIgnoreButton.TabIndex = 2;
this.abortRetryIgnoreButton.Text = "AbortRetryIgnore";
this.abortRetryIgnoreButton.CheckedChanged += new
    System.EventHandler(this.buttonType_CheckedChange);

// yesNoCancelButton
this.yesNoCancelButton.Location =
    new System.Drawing.Point(20, 80);
this.yesNoCancelButton.Name = "yesNoCancelButton";
this.yesNoCancelButton.TabIndex = 3;
this.yesNoCancelButton.Text = "YesNoCancel";
this.yesNoCancelButton.CheckedChanged += new
    System.EventHandler(this.buttonType_CheckedChange);
    
```

```

// yesNoButton
this.yesNoButton.Location =
    new System.Drawing.Point(20,100);
this.yesNoButton.Name = "yesNoButton";
this.yesNoButton.TabIndex = 4;
this.yesNoButton.Text = "YesNo";
this.yesNoButton.CheckedChanged += new
    System.EventHandler(this.buttonType_CheckedChange);

// retryCancelButton
this.retryCancelButton.Location =
    new System.Drawing.Point(20, 120);
this.retryCancelButton.Name = "retryCancelButton";
this.retryCancelButton.TabIndex = 5;
this.retryCancelButton.Text = "RetryCancel";
this.retryCancelButton.CheckedChanged += new
    System.EventHandler(this.buttonType_CheckedChange);
    
```

```

// asteriskButton
this.asteriskButton.Location =
    new System.Drawing.Point(20,20);
this.asteriskButton.Name = "asteriskButton";
this.asteriskButton.TabIndex = 0;
this.asteriskButton.Text = "Asterisk";
this.asteriskButton.CheckedChanged += new
    System.EventHandler(this.iconType_CheckedChange);

// errorButton
this.errorButton.Location = new System.Drawing.Point(20, 40);
this.errorButton.Name = "errorButton";
this.errorButton.TabIndex = 1;
this.errorButton.Text = "Error";
this.errorButton.CheckedChanged += new
    System.EventHandler(this.iconType_CheckedChange);
    
```

```

// exclamationButton
this.exclamationButton.Location =
    new System.Drawing.Point(20, 60);
this.exclamationButton.Name = "exclamationButton";
this.exclamationButton.TabIndex = 2;
this.exclamationButton.Text = "Exclamation";
this.exclamationButton.CheckedChanged += new
    System.EventHandler(this.iconType_CheckedChange);

// handButton
this.handButton.Location = new System.Drawing.Point(20,80);
this.handButton.Name = "handButton";
this.handButton.TabIndex = 3;
this.handButton.Text = "Hand";
this.handButton.CheckedChanged += new
    System.EventHandler(this.iconType_CheckedChange);
    
```

```
// informationButton
this.informationButton.Location =
    new System.Drawing.Point(20, 100);
this.informationButton.Name = "informationButton";
this.informationButton.TabIndex = 4;
this.informationButton.Text = "Information";
this.informationButton.CheckedChanged += new
    System.EventHandler(this.iconType_CheckedChange);

// noneButton
this.noneButton.Location=new System.Drawing.Point(20,120);
this.noneButton.Name = "noneButton";
this.noneButton.TabIndex = 5;
this.noneButton.Text = "None";
this.noneButton.CheckedChanged += new
    System.EventHandler(this.iconType_CheckedChange);
```

```
// questionButton
this.questionButton.Location =
    new System.Drawing.Point(20, 140);
this.questionButton.Name = "questionButton";
this.questionButton.TabIndex = 6;
this.questionButton.Text = "Question";
this.questionButton.CheckedChanged += new
    System.EventHandler(this.iconType_CheckedChange);

// stopButton
this.stopButton.Location=new System.Drawing.Point(20,160);
this.stopButton.Name = "stopButton";
this.stopButton.TabIndex = 7;
this.stopButton.Text = "Stop";
this.stopButton.CheckedChanged += new
    System.EventHandler(this.iconType_CheckedChange);
```

```
// warningButton
this.warningButton.Location =
    new System.Drawing.Point(20,180);
this.warningButton.Name = "warningButton";
this.warningButton.TabIndex = 8;
this.warningButton.Text = "Warning";
this.warningButton.CheckedChanged += new
    System.EventHandler(this.iconType_CheckedChange);

// displayButton
this.displayButton.Location =
    new System.Drawing.Point(16,272);
this.displayButton.Name = "displayButton";
this.displayButton.Size = new System.Drawing.Size(136, 23);
this.displayButton.TabIndex = 3;
this.displayButton.Text = "Изобрази";
this.displayButton.Click += new
    System.EventHandler(this.displayButton_Click);
```

```
// displayLabel
this.displayLabel.Location =
    new System.Drawing.Point(160,272);
this.displayLabel.Name = "displayLabel";
this.displayLabel.Size = new System.Drawing.Size(136, 23);
this.displayLabel.TabIndex = 4;
// Form1
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(328, 326);
this.Controls.Add(this.displayLabel);
this.Controls.Add(this.displayButton);
this.Controls.Add(this.groupIcons);
this.Controls.Add(this.groupButtons);
this.Controls.Add(this.promptLabel);
this.Name = "Form1";
this.Text = "Form1";
this.groupButtons.ResumeLayout(false);
this.groupIcons.ResumeLayout(false);
this.ResumeLayout(false);
}
#endregion
```

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

```
private void buttonType_CheckedChange(object sender,
    System.EventArgs e)
{
    if(sender==okButton)
        buttonType=MessageBoxButtons.OK;
    else if(sender==okCancelButton)
        buttonType=MessageBoxButtons.OKCancel;
    else if(sender==abortRetryIgnoreButton)
        buttonType=MessageBoxButtons.AbortRetryIgnore;
    else if(sender==yesNoCancelButton)
        buttonType=MessageBoxButtons.YesNoCancel;
    else if(sender==yesNoButton)
        buttonType=MessageBoxButtons.YesNo;
    else
        buttonType=MessageBoxButtons.RetryCancel;
}
```

```
private void iconType_CheckedChange(object sender,
                                     System.EventArgs e)
{
    if(sender==asteriskButton)
        iconType=MessageBoxIcon.Asterisk;
    else if(sender==errorButton)
        iconType=MessageBoxIcon.Error;
    else if(sender==exclamationButton)
        iconType=MessageBoxIcon.Exclamation;
    else if(sender==handButton)
        iconType=MessageBoxIcon.Hand;
    else if(sender==informationButton)
        iconType=MessageBoxIcon.Information;
    else if(sender==noneButton)
        iconType=MessageBoxIcon.None;
    else if(sender==questionButton)
        iconType=MessageBoxIcon.Question;
    else if(sender==stopButton)
        iconType=MessageBoxIcon.Stop;
    else
        iconType=MessageBoxIcon.Warning;
}
```

```
private void displayButton_Click(object sender,
                                 System.EventArgs e)
{
    DialogResult result = MessageBox.Show
        ("Потребителски прозорец със съобщения",
        "Съобщения", buttonType, iconType);
    switch(result)
    {
        case DialogResult.OK:
            displayLabel.Text="Натиснат е бутонът OK";
            break;
        case DialogResult.Cancel:
            displayLabel.Text="Натиснат е бутонът Cancel";
            break;
        case DialogResult.Abort:
            displayLabel.Text="Натиснат е бутонът Abort";
            break;
    }
}
```

```
case DialogResult.Retry:
    displayLabel.Text="Натиснат е бутонът Retry";
    break;
case DialogResult.Ignore:
    displayLabel.Text="Натиснат е бутонът Ignore";
    break;
case DialogResult.Yes:
    displayLabel.Text="Натиснат е бутонът Yes";
    break;
case DialogResult.No:
    displayLabel.Text="Натиснат е бутонът No";
    break;
}
}
```

### Контрол DateTimePicker

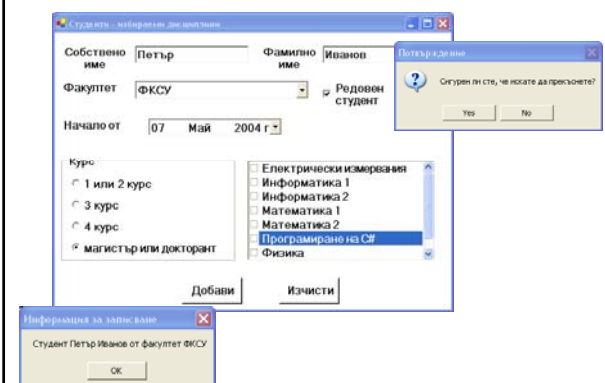
Използва се за избор на дати или времена от списък.



#### Свойства

**Value** съдържа установените дата и време.

### Пример: Форма за записване на студенти в избираеми дисциплини.



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

```
namespace Students
{
    public class StudentForm : System.Windows.Forms.Form
    {
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox firstName;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.TextBox lastName;
```

```
private System.Windows.Forms.Label label3;
private System.Windows.Forms.ComboBox faculty;
private System.Windows.Forms.CheckBox student;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.DateTimePicker startFrom;
private System.Windows.Forms.GroupBox experience;
private System.Windows.Forms.RadioButton novice;
private System.Windows.Forms.RadioButton intermediate;
private System.Windows.Forms.RadioButton experienced;
private System.Windows.Forms.RadioButton accomplished;
private System.Windows.Forms.CheckedListBox courses;
private System.Windows.Forms.Button add;
private System.Windows.Forms.Button clear;
private System.ComponentModel.Container components =
    null;
```

```
public StudentForm()
{
    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()
{
    this.label1 = new System.Windows.Forms.Label();
    this.firstName = new System.Windows.Forms.TextBox();
    this.label2 = new System.Windows.Forms.Label();
    this.lastName = new System.Windows.Forms.TextBox();
    this.label3 = new System.Windows.Forms.Label();
    this.faculty = new System.Windows.Forms.ComboBox();
    this.student = new System.Windows.Forms.CheckBox();
    this.label4 = new System.Windows.Forms.Label();
    this.startFrom =
        new System.Windows.Forms.DateTimePicker();
```

```
this.experience =
    new System.Windows.Forms.GroupBox();
this.accomplished =
    new System.Windows.Forms.RadioButton();
this.experienced =
    new System.Windows.Forms.RadioButton();
this.intermediate =
    new System.Windows.Forms.RadioButton();
this.novice =
    new System.Windows.Forms.RadioButton();
this.courses =
    new System.Windows.Forms.CheckedListBox();
this.add = new System.Windows.Forms.Button();
this.clear = new System.Windows.Forms.Button();
this.experience.SuspendLayout();
this.SuspendLayout();
```

```
// Етикет label1 – Собствено име
this.label1.Location = new System.Drawing.Point(10, 20);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(100, 36);
this.label1.TabIndex = 0;
this.label1.Text = "Собствено име";
this.label1.TextAlign =
    System.Drawing.ContentAlignment.MiddleCenter;

// Текстово поле firstName
this.firstName.Location =
    new System.Drawing.Point(120,20);
this.firstName.Name = "firstName";
this.firstName.Size = new System.Drawing.Size(170, 26);
this.firstName.TabIndex = 1;
this.firstName.Text = "";
```

```
// Етикет label2 – Фамилно име
this.label2.Location=new System.Drawing.Point(300,20);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(100, 36);
this.label2.TabIndex = 2;
this.label2.Text = "Фамилно име";
this.label2.TextAlign =
    System.Drawing.ContentAlignment.MiddleCenter;

// Текстово поле lastName
this.lastName.Location =
    new System.Drawing.Point(400, 20);
this.lastName.Name = "lastName";
this.lastName.Size = new System.Drawing.Size(170, 26);
this.lastName.TabIndex = 3;
this.lastName.Text = "";
```

```
// Етикет label3 – Факултет
this.label3.Location = new System.Drawing.Point(10,72);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(96, 23);
this.label3.TabIndex = 4;
this.label3.Text = "Факултет";

// Комбо поле faculty
this.faculty.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
this.faculty.Location=new System.Drawing.Point(120,72);
this.faculty.Name = "faculty";
this.faculty.Size = new System.Drawing.Size(260, 28);
this.faculty.TabIndex = 5;
```

```
// Поле за проверка student
this.student.Location =
    new System.Drawing.Point(400,72);
this.student.Name = "student";
this.student.Size = new System.Drawing.Size(100, 40);
this.student.TabIndex = 6;
this.student.Text = "Редовен студент";

// Етикет label4 – Начало от
this.label4.Location=new System.Drawing.Point(10,128);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(120, 23);
this.label4.TabIndex = 7;
this.label4.Text = "Начало от";

// Контрол за избор на дата startFrom
this.startFrom.Location =
    new System.Drawing.Point(140, 128);
this.startFrom.Name = "startFrom";
this.startFrom.TabIndex = 8;
```

```
// Контрол за аранжиране experience – Курс
this.experience.Controls.Add(this.accomplished);
this.experience.Controls.Add(this.experienced);
this.experience.Controls.Add(this.intermediate);
this.experience.Controls.Add(this.novice);
this.experience.Location =
    new System.Drawing.Point(10, 184);
this.experience.Name = "experience";
this.experience.Size =
    new System.Drawing.Size(260,160);
this.experience.TabIndex = 9;
this.experience.TabStop = false;
this.experience.Text = "Курс";
```

```
// Радио бутон novice – 1 или 2 курс
this.novice.Location = new System.Drawing.Point(16,32);
this.novice.Name = "novice";
this.novice.Size = new System.Drawing.Size(220, 24);
this.novice.TabIndex = 10;
this.novice.Text = "1 или 2 курс";

// Радио бутон intermediate – 3 курс
this.intermediate.Location =
    new System.Drawing.Point(16, 64);
this.intermediate.Name = "intermediate";
this.intermediate.Size =
    new System.Drawing.Size(220, 24);
this.intermediate.TabIndex = 11;
this.intermediate.Text = "3 курс";
```

```
// Радио бутон experienced – 4 курс
this.experienced.Location =
    new System.Drawing.Point(16, 96);
this.experienced.Name = "experienced";
this.experienced.Size =
    new System.Drawing.Size(220, 24);
this.experienced.TabIndex = 12;
this.experienced.Text = "4 курс";

// Радио бутон accomplished
this.accomplished.Location =
    new System.Drawing.Point(16, 128);
this.accomplished.Name = "accomplished";
this.accomplished.Size =
    new System.Drawing.Size(220, 24);
this.accomplished.TabIndex = 13;
this.accomplished.Text = "магистър или докторант";
```

```
// Списък с проверка courses
this.courses.Location =
    new System.Drawing.Point(288, 192);
this.courses.Name = "courses";
this.courses.Size = new System.Drawing.Size(280, 151);
this.courses.Sorted = true;
this.courses.TabIndex = 10;

// Бутон add – Добави
this.add.Location = new System.Drawing.Point(190,368);
this.add.Name = "add";
this.add.Size = new System.Drawing.Size(90, 40);
this.add.TabIndex = 11;
this.add.Text = "Добави";
this.add.Click +=
    new System.EventHandler(this.addClick);
```



```
// Бутон clear – Изчисти
this.clear.Location=new System.Drawing.Point(335,368);
this.clear.Name = "clear";
this.clear.Size = new System.Drawing.Size(90, 40);
this.clear.TabIndex = 12;
this.clear.Text = "Изчисти";
this.clear.Click +=
    new System.EventHandler(this.clearClicked);
```

```
// Форма StudentForm
this.AutoScaleBaseSize =
    new System.Drawing.Size(8, 19);
this.ClientSize = new System.Drawing.Size(592, 416);
this.Controls.Add(this.clear);
this.Controls.Add(this.add);
this.Controls.Add(this.courses);
this.Controls.Add(this.experience);
this.Controls.Add(this.startFrom);
this.Controls.Add(this.label4);
this.Controls.Add(this.student);
this.Controls.Add(this.faculty);
this.Controls.Add(this.label3);
this.Controls.Add(this.lastName);
this.Controls.Add(this.firstName);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
```

```
this.Font = new System.Drawing.Font
    ("Microsoft Sans Serif", 12F,
    System.Drawing.FontStyle.Bold,
    System.Drawing.GraphicsUnit.Point,
    (System.Byte)(204));
this.Name = "StudentForm";
this.Text = "Студенти - избираеми дисциплини";
this.Closing += new
    System.ComponentModel.CancelEventHandler
    (this.memberFormClosing);
this.experience.ResumeLayout(false);
this.ResumeLayout(false);
}
#endregion
```

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

public void Reset()
{
    // Изчистване на текстовите полета
    firstName.Text="";
    lastName.Text="";

    // Изчистване и добавяне на данни за факултетите в
    // контрола ComboBox
    faculty.Items.Clear();
    faculty.Items.Add("ФА");
    faculty.Items.Add("ФЕТТ");
    faculty.Items.Add("ФКСУ");
```

```
faculty.Items.Add("ФКТТ");
faculty.Items.Add("ЕФ");
faculty.Items.Add("СФ");
faculty.Items.Add("МФ");
faculty.Items.Add("МТФ");

// Установяване на текущата дата на контрола
// DateTimePicker
startFrom.Value=DateTime.Today;

// Установяване "редовен студент" на контрола
// CheckBox
student.Checked=true;

// Установяване студент "1 или 2" курс за групата от
// радио-бутони
novice.Checked=true;
```

```
// Изчистване и добавяне на дисциплини към
// контрола CheckListBox
courses.Items.Clear();
courses.Items.Add("Информатика 1");
courses.Items.Add("Математика 1");
courses.Items.Add("Математика 2");
courses.Items.Add("Физика");
courses.Items.Add("Химия");
courses.Items.Add("Информатика 2");
courses.Items.Add("Програмиране на С#");
courses.Items.Add("Електрически измервания");
}

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

```
private void addClick(object sender, System.EventArgs e)
{
    string details;
    details="Студент "+firstName.Text+" "+lastName.Text+
        " от факултет "+faculty.Text;
    MessageBox.Show(details,
        "Информация за записване");
}

private void memberFormClosing(object sender,
    System.ComponentModel.CancelEventArgs e)
{
    DialogResult key=MessageBox.Show
        ("Сигурен ли сте, че искате да прекъснете?",
        "Потвърждение", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question);
    e.Cancel=(key==DialogResult.No);
}
}
```