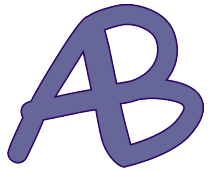
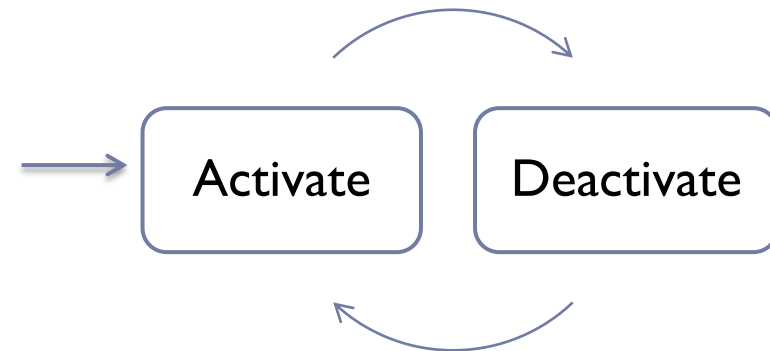
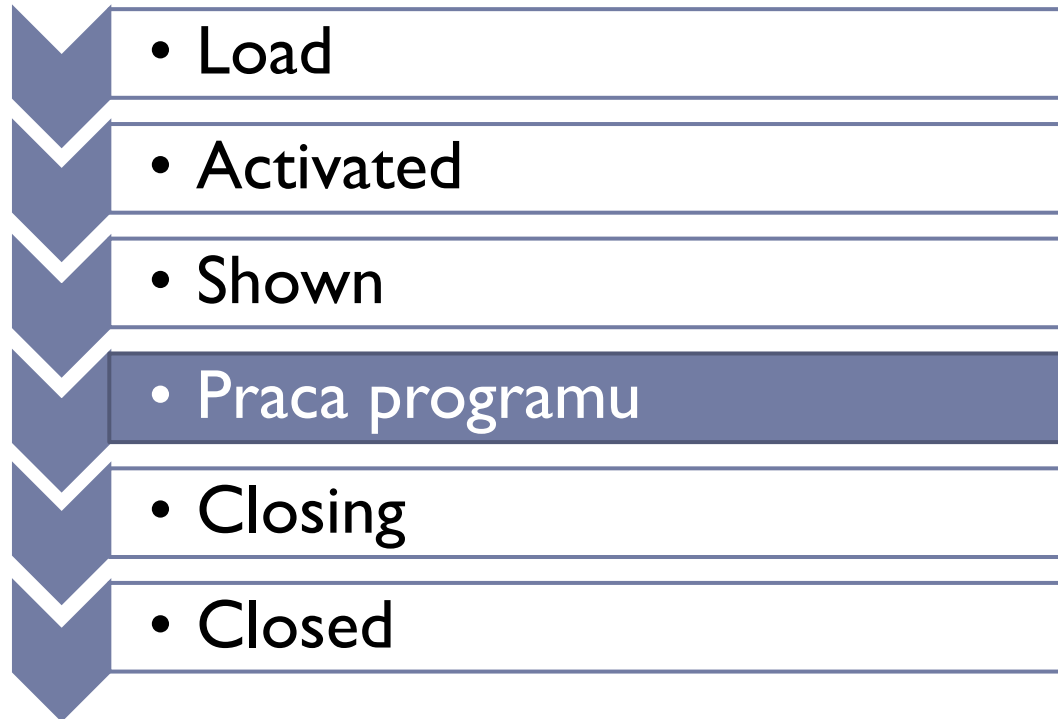


Wykład Zdarzenia



Cykl życia aplikacji

Cykl życia programu



Cykl życia programu

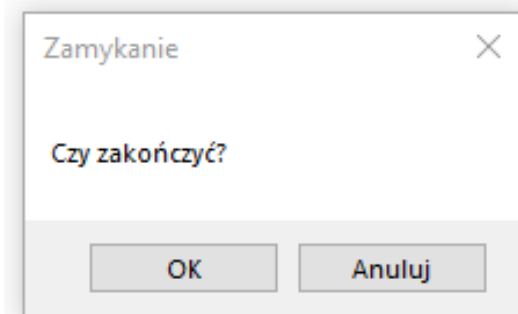
Zdarzenia dotyczą całego okna – nie poszczególnych komponentów.

Behavior	
ChangeUICues	
ControlAdded	
ControlRemoved	
FormClosed	Form1_FormClosed
FormClosing	Form1_FormClosing
HelpButtonClicked	
HelpRequested	
ImeModeChanged	
InputLanguageChanged	
InputLanguageChanging	
Load	Form1_Load
QueryAccessibilityHelp	
Shown	Form1_Shown
StyleChanged	
SystemColorsChanged	
Data	
(DataBindings)	
Drag Drop	
DragDrop	
DragEnter	
DragLeave	
DragOver	
GiveFeedback	
QueryContinueDrag	
Focus	
Activated	Form1_Activated
Deactivate	Form1_Deactivate

Zdarzenia

Przykład:

Potwierdzenie zamknięcia programu

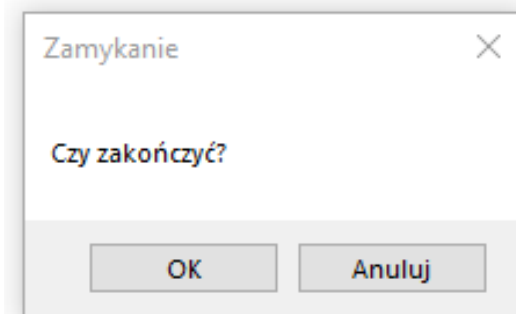


```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (MessageBox.Show("Czy zakończyć?", "Zamykanie",
        MessageBoxButtons.OKCancel) == DialogResult.Cancel)
    {
        e.Cancel = true;
    }
}
```

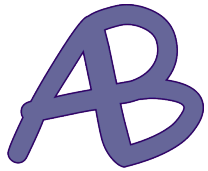
Zdarzenia

Przykład:

Potwierdzenie zamknięcia programu



```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (MessageBox.Show("Czy zakończyć?", "Zamykanie",
        MessageBoxButtons.OKCancel) == DialogResult.Cancel)
    {
        e.Cancel = true;
    }
}
```



Wygląd / rozmiar / położenie aplikacji

Zdarzenia

▣ Action

Click	
DoubleClick	
MouseCaptureChanged	
MouseDown	
MouseDoubleClick	
ResizeBegin	
ResizeEnd	
Scroll	

Początek zmiany rozmiaru okna/
zakończenie zmiany rozmiaru

Zdarzenie wywoływane
podczas przesuwania okna

▣ Layout

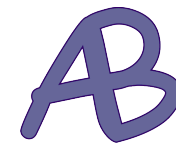
Layout	
MdiChildActivate	
Move	
PaddingChanged	
Resize	

Zdarzenie wywoływane
podczas zmiany rozmiaru (w
sposób ciągły)

Zdarzenie wywoływane
podczas przerysowania okna
lub jego fragmentu

▣ Appearance

Paint	
-------	--



Zdarzenia

Przykład:

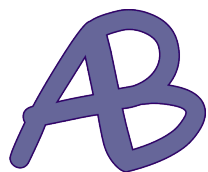
Wyświetlenie aktualnej pozycji lub rozmiaru okna na pasku statusu

```
private void Form1_Move(object sender, EventArgs e)
{
    statusStrip1.Items[0].Text =
        Form1.ActiveForm.Location.X.ToString() + " " +
        Form1.ActiveForm.Location.Y.ToString();
}
```

Pozycja

Rozmiar

```
private void Form1_Resize(object sender, EventArgs e)
{
    statusStrip1.Items[0].Text =
        Form1.ActiveForm.Width.ToString() + " " +
        Form1.ActiveForm.Height.ToString();
}
```



Zdarzenia myszy

Zdarzenia myszy

Action	
Click	Form1_Click
DoubleClick	Form1_DoubleClick
MouseCaptureChanged	
MouseClicked	Form1_MouseClick
MouseDoubleClick	Form1_MouseDoubleClick
ResizeBegin	

Zdarzenia **Click** i **DoubleClick** można (na niektórych obiektach) wywołać za pomocą entera.

Zdarzenia **MouseClicked** i **MouseDoubleClick** są związane tylko z myszą





Zdarzenia myszy

☐ Mouse

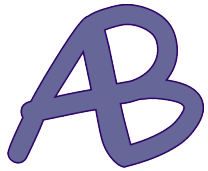
<code>MouseDown</code>	- Naciśnięcie przycisku myszy
<code>MouseEnter</code>	- Wjechanie kursora na obiekt
<code>MouseHover</code>	- Wjechanie i zatrzymanie kursora na obiekcie
<code>MouseLeave</code>	- Opuszczanie obiektu przez kursor
<code>MouseMove</code>	- Poruszanie się kursora nad obiektem (zdarzenie ciągłe)
<code>MouseUp</code>	- Puszczanie przycisku myszy

Informacje o zdarzeniu – w szczególności:

- który przycisk został naciśnięty,
- jaka jest pozycja myszy

można wyodrębnić z obiektu „e”

```
if (e.Button == MouseButton.Left)
    richTextBox1.AppendText(e.X.ToString());
```



Zdarzenia Klawiatury



Zdarzenia Klawiatury

[-] Key

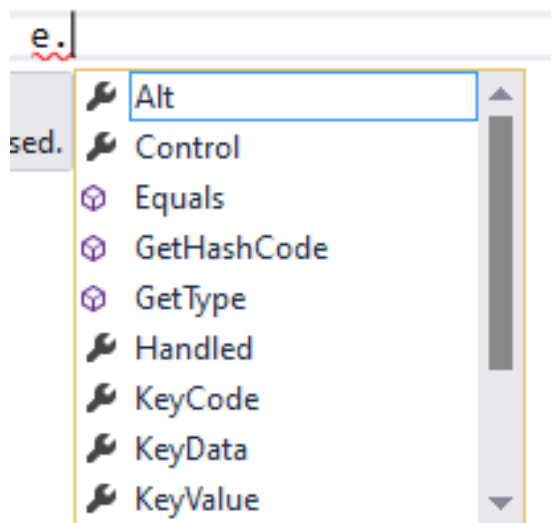
KeyDown	- Naciśnięcie klawisza
KeyPress	- Pełny cykl (naciśnięcie + zwolnienie)
KeyUp	- Zwolnienie klawisza
PreviewKeyDown	- Obsługa klawiszy modyfikatorów (Alt, Ctrl, Shift)

Zdarzenia Klawiatury

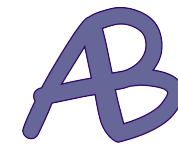
Odczytanie wartości klawisza z metody **KeyPress**

```
private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    char klawisz = e.KeyChar;
}
```

W metodzie **KeyDown** mamy nieco więcej możliwości



Możemy między innymi odczytać klawisze modyfikatorów.

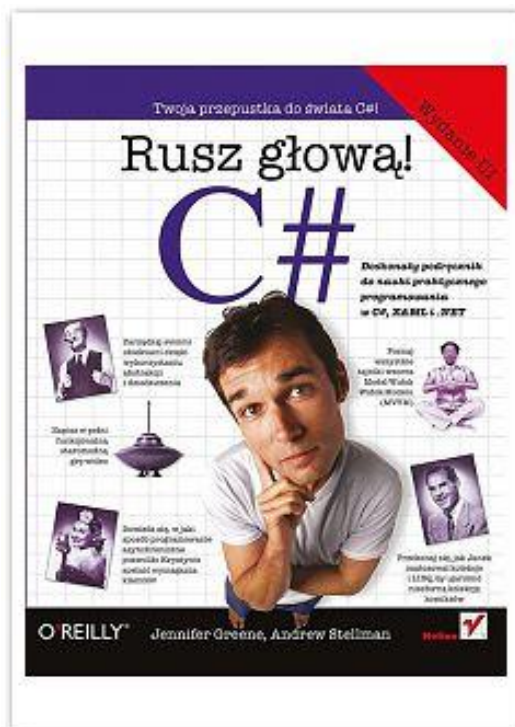
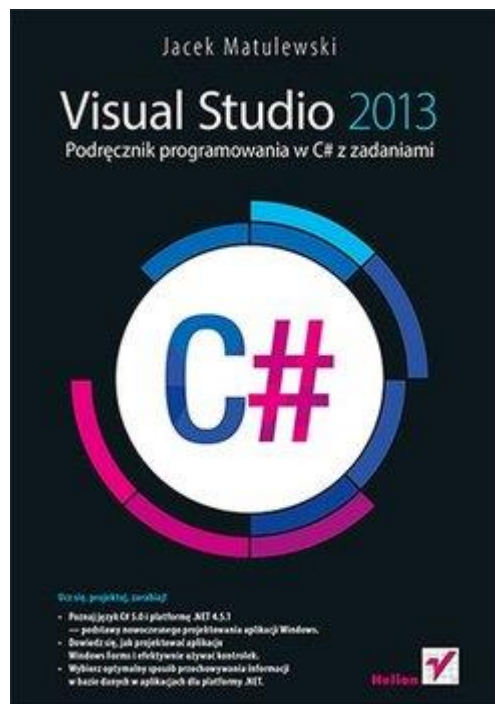


Zdarzenia Klawiatury

- ✓ Każdy element (kontrolka) ma swoją własną kolejkę zdarzeń klawiatury.
- ✓ Zdarzenie przechwytuje ta kontrolka, która posiada „fokus”
- ✓ Aby zdarzenie zostało dodatkowo przechwycone przez okno główne (Form1) – należy zmienić jego właściwość **KeyPreview** na **true**

Misc	
AcceptButton	(none)
CancelButton	(none)
KeyPreview	True

Literatura:



Użyte w tej prezentacji tabelki pochodzą z książki: Visual Studio 2013. Podręcznik programowania w C# z zadaniami
Autor: Matulewski Jęcek, Helion