

Tematický celek 04

4.1 Události



Aplikace po spuštění přejde do stavu nečinnosti a další její chod je řízen událostmi (*Events*).

Činnost vyvolávající událost	Název události ve VB
Pohnutí myši	MouseMove
Stisknutí tlačítka myši	MouseDown
Uvolnění tlačítka myši	MouseUp
Kliknutí na tlačítko myši	Click
Dvojité kliknutí na tlačítko myši	DoubleClick
Při stisknutí klávesy na klávesnici	KeyDown
Při uvolnění klávesy na klávesnici	KeyUp
Stisknutí klávesy na klávesnici	KeyPress
Získání fokusu	GotFokus
Ztráta fokusu	LostFokus

- Události *KeyDown* a *KeyPress* vypadají na první pohled stejné. Liší se ale v předávaném kódu.
- Zatímco událost *KeyPress* poskytuje běžnou ASCII hodnotu stisknutého znaku, událost *KeyDown* (a také *KeyUp*) poskytuje interní kód stisknuté klávesy. Pomocí události *KeyDown* můžeme zjišťovat stisknutí i kláves, které nerepresentují znaky (klávesy F1–F12, navigační klávesy apod.).
- Událost *GotFocus* u ovládacího prvku vzniká v okamžiku, kdy s ním začneme pracovat. Například umístíme-li do textového pole kurzor, abychom do pole mohli něco napsat, vyvolá to v textovém poli událost *GotFocus*. Naopak v okamžiku, kdy ovládání (fokus) přeneseme jinam (například klikneme na některé tlačítko v aplikaci – klikneme úplně mimo námi vytvořená okna), vyvolá to v textovém poli událost *LostFocus*.

Samotný formulář má navíc ještě další specifické události. Uvedme nejdůležitější z nich:

- *Load* a *Open* – nastávají při otevření okna s formulářem. Typicky se používají pro inicializaci proměnných a počáteční nastavení ovládacích prvků v okně (například seznamy naplníme údaji).
- *Unload* a *Close* – nastávají při zavření okna. Dáváme do nich příkazy, které chceme, aby se při zavření provedly (například uložení údajů).

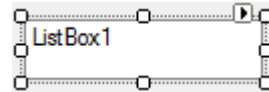
Při sestavování programu u jednotlivých ovládacích prvků rozhodujeme, na které události a jak má aplikace reagovat. K událostem připojujeme příkazy, které se při nich mají vykonat. Připojení příkazů k události je ve VB řešeno přes proceduru. Ta je volána pokaždé, když událost nastane.

Hlavička procedury je složena z údajů:

- jména ovládacího prvku (vlastnost *Name*),
- názvu události,
- parametrů předávaných při události.

4.2 Ovládací prvek ComboBox a ListBox

- nabízejí uživateli seznam možností, ze kterých si může vybrat,
- seznam je zobrazen jako pole se Scroll barem napravo,
- pole se seznamem je zobrazeno jako pole, které má v pravé části tlačítko, které nám umožňuje rozvinout seznam nabídek.
- položky naplňujeme pomocí příkazu `Items.Add`



- ComboBox má tyto typické vlastnosti:
 - o Text (text, který je možné vložit do prvního řádku – ostatní položky je nutné vytvořit pomocí příkazu `additem!!!`)
 - o Font (formát vepsaného textu)
 - o ForeColor (barva vepsaného textu)
 - o BackColor (barva pozadí ovládacího prvku)

cvičný 2

Metody jsou důležité zejména při práci s ovládacím prvkem typu seznam (*Listbox a ComboBox*), neboť nám v něm umožňují přidávat položky či odebírat položky. Jsou to metody:

Metoda	Funkce metody
<i>Items.Clear</i>	Vymaže obsah seznamu.
<i>Items.Add řetězec</i>	Přidá položku do seznamu. Přidaná položka se v seznamu objeví jako poslední. Položka je určena parametrem řetězce.
<i>Items.RemoveAt index</i>	Odstraní položku ze seznamu. Pozice položky je určena parametrem <code>index</code> .
<i>Items.Insert index řetězec</i>	Přidá položku do seznamu na danou pozici. Pozice položky je určena parametrem <code>index</code> , položka je určena parametrem řetězce.

cvičný 3

Vedle metod se při práci se seznamy rovněž hojně využívají jejich vlastnosti, které poskytují důležité údaje o seznamu:

Vlastnost	Význam
<i>Items.Count</i>	Počet položek v seznamu.
<i>Items.Item</i>	Obsah položky v seznamu určené parametrem čísla řádku.
<i>Items.Index</i>	Index položky, která je v seznamu vybrána.
<i>SelectedItem</i>	Hodnota je <i>True</i> , jestliže položka s tímto indexem je vybrána.

Indexování položek je od 0 (první položka v seznamu) po hodnotu *Items.Count-1* (poslední položka v seznamu).

4.3 Logické operátory

- *And* (konjunkce),
- *Eqv* (ekvivalence),
- *Imp* (implikace),
- *Not* (negace),
- *Or* (disjunkce),
- *Xor* (nonekvivalence).

AND

- výsledek operace je True pokud jsou splněny obě části výrazu
- výsledek operace je False, pokud není splněna jedna nebo obě části výrazu
- např: **1 = 1 And 2 + 6 = 8** **výsledkem je True (pravda)**
- např: **1 > 2 And 2 < 5** **výsledkem je False (nepravda)**

Syntaxe:

`výraz a AND výraz b`

OR

- výsledek operace je True pokud je splněna alespoň jedna nebo obě strany výrazu
- výsledek operace je False, pokud není splněna ani jedna část výrazu
- např: **1 = 1 Or 2 + 6 = 7** **výsledkem je True (pravda)**
- např: **1 > 2 Or 2 > 5** **výsledkem je False (nepravda)**

Syntaxe:

`výraz a OR výraz b`

4.4 Základní příkazy

Items.Add

- Slouží pro naplnění ovládacích prvků ComboBox a ListBox položkami.
- Každá položka musí mít svůj příkaz, není možné pomocí jednoho příkazu Items.Add vložit více položek naráz.

Syntaxe:

`Objekt.Items.Add_položka`

Objekt je jméno ovládacího prvku (může být pouze typu ComboBox a ListBox).

Položka může být textový řetězec (je nutné jej napsat do uvozovek, např: „Táta“), číslo (není nutné psát do uvozovek, např: 1589), datum atd...

“““

- Takzvaný prázdný řetězec.
- Můžeme pomocí něho mazat obsah ovládacích prvků či proměnných (u některých je ale nutné použít výraz Empty – prázdná).
- Např: **TextBox1.Text = “““** vymaže obsah textového pole TextBox1.

- Nebo jej můžeme použít pro testování zda je daný ovládací prvek prázdný či nikoliv.
- Např: **If TextBox1.Text = "" Then TextBox1.Text = 5** pokud je pole prázdné, vloží se do něj číslo 5.

Porovnávání řetězců



- Existuje celá řada metod jak porovnat dva textové řetězce.
- Pro naše potřeby je nutné si pamatovat, že pokud porovnáváme dva řetězce (nebo obsahy dvou ovládacích prvků) a chceme, aby porovnání dopadlo jako **TRUE** (tzn. že jsou shodné), musí být oba řetězce **NAPROSTO!!!** shodné (velikost písmen, počet a velikost mezer atd...)
- Např: **“Mám maso“** je shodné jako **“Mám maso“**
- Ale **“Mám maso“** není shodné jako **“Mám Maso“**
- Visual Basic 2010 k porovnávání používá ANSI tabulku znaků a v ní mají jiné číslo např. „m“ a „M“.