

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-103
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech I
Téma didaktického materiálu	Některé ovládací prvky
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 16–17 let
Úroveň žáků	začátečníci
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Ovládací prvky – tlačítko, textové pole, přiřazení
Anotace	Studenti chápou pojmy vlastnosti a události ovládacích prvků, učí se je nastavovat v návrhu i za běhu programu
Použité zdroje	VYSTAVĚL, Radek. <i>Moderní programování: učebnice pro začátečníky</i> . Ondřejov: moderníProgramování s.r.o, 2007, 2 sv. ISBN 978-80-903951-0-7. VYSTAVĚL, Radek. <i>Moderní programování: sbírka úloh k učebnici pro začátečníky</i> . 2. vyd. Ondřejov: moderníProgramování, 2008, 2 sv. ISBN 978-80-903951-5-2.
Typy k metodickému postupu učitele, doporučené výukové metody, způsob hodnocení, typy k individualizované výuce apod.	Text je možno využít ke společné práci, samostatné přípravě studentů, domácímu studiu apod. V pracovním listu je zadání cvičení – většinou se jedná o úlohy, které by měli studenti vyřešit samostatně. Není nutné, aby všichni zpracovali všechno, vhodné je diferencovat podle jejich zájmu a schopností. Součástí materiálu je řešení těchto příkladů. Návrh způsobu hodnocení: ohodnocení samostatné práce během hodiny např. podle volby a počtu úloh a elaborace řešení (efektivnost, komentáře...).

Metodický list k didaktickému materiálu

Prohlášení autora

Tento materiál je originálním autorským dílem. K vytvoření tohoto didaktického materiálu nebyly použity žádné externí zdroje s výjimkou zdrojů citovaných v metodickém listu.

Obrázky (schémata a snímky obrazovek) pocházejí od autora.

103. Některé ovládací prvky (komponenty)

Ovládací prvky jako tlačítka, popisky a textová pole mají své **vlastnosti** a **události**. K obojím přistupujeme přes **editor vlastností a událostí** (Okno **Properties**, kde se můžeme přepnout ikonou s bleskem na **Events**) Už víme, že mezi vlastnosti patří **Name** a **Text**, v okně **Properties** je vždy vlevo uveden vlastnost a vpravo nastavujeme její hodnotu. Hodnoty vlastností mohou být různé – například text nebo čísla přímo zapisujeme čísla, barvy volíme ze seznamu.

Události vznikají například při klepnutí myši, stisknutí klávesy, uzavření okna apod., to, co chceme, aby se při dané události stalo, musíme naprogramovat.

Tlačítko (Button)

Tlačítko známe z Windows jako obdélníček, který při zmáčknutí vykoná nějakou akci.

Nejdůležitější vlastnosti:

Name	buttonVhodnéPojmenování – název může obsahovat i diakritiku, ale ne např. mezery, tečky, pomlčky, apostrofy.
Text	nápis na tlačítku – vyjadřuje jeho funkci, podobně jako VhodnéPojmenování, můžete používat libovolné znaky
Enabled	hodnoty v editoru True nebo False (při programování nutno psát s malým písmenem) vyjadřují, zda se dá tlačítko použít – True nebo ne. (False , je odlišeno světlejší barvou)
Font	zobrazí známé okno s volbou fontu

Hlavní událost

(buď vybereme v okně **Events** nebo poklepeme na tlačítko)

Click	nastane při klepnutí na tlačítko, vytvoří se přitom kostra, do které budeme zapisovat kód.
--------------	--

```
private void buttonVhodnéPojmenování_Click(object sender, EventArgs e)
{
}

```

Textové políčko (textBox)

Slouží ke vstupu i výstupu (zobrazení) textových údajů. Dá se do něj psát a text editovat, je možné také pracovat se schránkou. (Ctrl+C, Ctrl+V)

Vlastnosti uvádíme jen některé, můžete si projít a vyzkoušet i ostatní.

Nejdůležitější vlastnosti:

Name	textBoxAkce
Text	
Enabled	
BackColor	barva pozadí, vybírá se ze seznamu

ForeColor	barva textu, analogicky
TextAlign	zarovnání textu v políčku (Vzpomeňte na Excel – text doleva, čísla doprava...)
ReadOnly	je-li True , je okno pouze ke čtení, nedá se editovat
Multiline	pokud nastavíte na True , můžete políčko myši rozvinout do více řádků a do nich pak zadávat text.
Scrollbars	posuvníky, vybereme ze seznamu

Font

Nastavování vlastností v programu

provádíme pomocí přiřazovacího příkazu, který má tvar: **KAM = CO;**

To co je na pravé straně, uloží se do strany levé. Samozřejmě KAM i CO musí být stejného druhu – do textu text, do barvy barva apod. Příkazy se oddělují středníky.

Příklady:

```
textBoxAkce.Text = "Ahoj";
```

```
textBoxReadOnly = true;
```

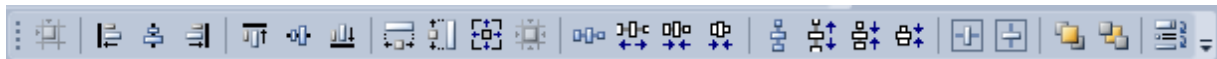
Pokud je více prvků v Designeru – vlastnosti se zobrazují pro aktivní prvek (kolem tečkovaný rámeček). Chceme-li, aby byl některý prvek aktivní, stačí na něj klepnout myší. (Pozor – klepnout jednou, poklepání vytvoří hlavní událost prvku)

Příklad 1

Připravíme si projekt se 4 tlačítky. 1. zobrazí zprávu, 2. změní text na 1. tlačítku, 3. tlačítko 1. tlačítko zakáže a 4. tlačítko je opět povolí.

Doporučené pojmenování tlačítek podle jejich funkce: buttonInfo, buttonPrejmenuj, buttonZakaz a buttonPovol.

Vyberte si barvu i font, pro zarovnání tlačítek na formuláři lze použít jednak nabízející se vodící linky, jednak nabídku Formát, ale také paletu **Layout Toolbar** pod panelem nástrojů. Například vyberete-li všechna tlačítka, objeví se následující možnosti:



Proberte si je s bublinkovou nápovědou. (Např. 9. ikona zprava rozmístí objekty stejně daleko od sebe ve svislém směru)

Příkazy důsledně oddělujte středníky a věnujte pozornost radám našeptávače.

Aby uživatel nemohl měnit rozměry formuláře (a náš layout), nastavíme vlastnosti formuláře **FormBorderStyle** na **FixedSingle** (nebo 3D) a **MaximizeBox** na **False**. Před spuštěním projekt uložte pod vhodným jménem (**Save All**).

```

private void buttonInfo_Click(object sender, EventArgs e)
{
    MessageBox.Show("Různé funkce tlačítek");
}

private void buttonPrejmenuj_Click(object sender, EventArgs e)
{
    buttonInfo.Text = "Emil";
}

private void buttonZakaz_Click(object sender, EventArgs e)
{
    buttonInfo.Enabled = false;
}

private void buttonPovol_Click(object sender, EventArgs e)
{
    buttonInfo.Enabled = true;
}

```

Příklad 2

Na formulář umístíte dvě textová pole a tři tlačítka. Klepnutím na první tlačítko, vložíme do prvního textBoxu text "Ahoj", klepnutím na druhé tlačítko zkopírujeme text z 1. textBoxu do druhého, klepnutím na třetí přebarvíme pozadí obou políček na žluté.

Názvy komponent:

Textová pole: textBoxZdroj, textBoxKopie

Tlačítka: buttonAhoj, buttonKopie, buttonBarva.

Pokud nastavíte vlastnost Font pro formulář, bude automaticky nastavená pro všechny komponenty na něm. (Dodatečně jde samozřejmě změnit)

```

private void buttonAhoj_Click(object sender, EventArgs e)
{
    textBoxZdroj.Text = "Ahoj";
}

private void buttonKopie_Click(object sender, EventArgs e)
{
    textBoxKopie.Text = textBoxZdroj.Text;
}

```

```

    }

    private void buttonBarva_Click(object sender, EventArgs e)
    {
        textBoxZdroj.BackColor = Color.Yellow;
        textBoxKopie.BackColor = Color.Yellow;
    }

```

Všimněte si: skutečnost, že textové políčko `textBoxZdroj` má vlastnost **BackColor**, vyjadřujeme tím, že použijeme zápis **textBoxZdroj.BackColor**. Podobně **Yellow** je zase vlastnost objektu **Color** – našeptávač vám při zápisu **Color**. nabídne seznam barev.

Poznámky:

Pokud chceme nastavovat vlastnosti formuláře za běhu programu, píšeme v přiřazovacím příkazu jen jméno vlastnosti: (Pro ostatní prvky musíme uvádět i jméno prvku: `button1.BackColor`)

```

BackColor = Color.Beige;
Text = "Formulář";

```

Vodorovné zarovnání se provádí například následujícím způsobem:

```

textBox1.TextAlign = HorizontalAlignment.Center;

```

Důležité:

Uživatelské rozhraní programu vytváříme tak, že na formulář přetahujeme komponenty z **ToolBoxu**.

Všechny komponenty mají vlastnost **Name**, kterou nastavujeme tak, že ponecháváme začátek, slova, aby bylo jasné o jaký prvek se jedná a připojujeme charakteristiku funkce prvku. Pomocí této vlastnosti pak s komponentou pracujeme při programování.

Editor vlastností pracuje s prvkem, který je na formuláři vybrán.

Tlačítko – Button, nejdůležitější vlastnosti: Name, Text (nápis na tlačítku)

Hlavní událost tlačítka je klepnutí, chceme-li ji vytvořit buď ji vybereme v editoru, nebo můžeme na tlačítko poklepat.

Textové políčko – textBox, nejdůležitější vlastnosti: Name, Text (text v políčku)

Chceme-li nastavit vlastnost komponenty za běhu programu, použijeme přiřazovací příkaz:

Kam = Co, přičemž na obou stranách musí být stejný typ údajů. (Do barvy barva, do textu text)

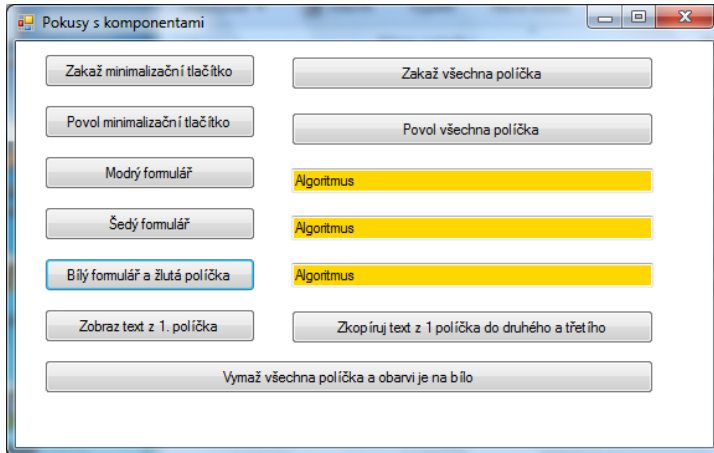
Textové řetězce se uzavírají mezi uvozovky.

Všechno s čím pracujeme, jsou objekty, vybavené vlastnostmi a událostmi, ke kterým přistupujeme pomocí tečkované notace. Např. **textBox1.Text**, **textBox1.Enabled**, **textBox1.ForeColor** apod.

Pracovní list

Cvičení

1. Vyzkoušejte si program Komponenty.exe a naprogramujte svůj projekt, který bude dělat totéž.



2. Vyzkoušejte si v dalším projektu nějaké vlastní nápady s textem, barvami, zarovnáním, textovými políčky a tlačítky.

Řešení

Metody většinou nejsou okomentovány, co mají dělat by mělo být zřejmé z jejich názvu.

1.

```
private void buttonZakMinTlac_Click(object sender, EventArgs e)
{
    MinimizeBox = false;
}

private void buttonPovMinTl_Click(object sender, EventArgs e)
{
    MinimizeBox = true;
}

private void buttonModrForm_Click(object sender, EventArgs e)
{
    BackColor = Color.Aqua;
}

private void buttonSilverForm_Click(object sender, EventArgs e)
```

```
{
    BackColor = Color.Silver;
}

private void buttonBilACerv_Click(object sender, EventArgs e)
{
    MessageBox.Show("Zákaz políček se neprojeví barevně, ale psát do nich
nejde.");

    BackColor = Color.White;

    textBox1.BackColor = Color.Gold;
    textBox2.BackColor = Color.Gold;
    textBox3.BackColor = Color.Gold;
}

private void buttonZakPolicka_Click(object sender, EventArgs e)
{
    textBox1.Enabled = false;
    textBox2.Enabled = false;
    textBox3.Enabled = false;
}

private void buttonPovolPolicka_Click(object sender, EventArgs e)
{
    textBox1.Enabled = true;
    textBox2.Enabled = true;
    textBox3.Enabled = true;
}

private void buttonZobraz1_Click(object sender, EventArgs e)
{
    MessageBox.Show(textBox1.Text);
}

private void buttonKopie_Click(object sender, EventArgs e)
{
    textBox2.Text = textBox1.Text;
}
```

```
        textBox3.Text = textBox1.Text;
    }
    private void buttonReset_Click(object sender, EventArgs e)
    //Vymaže políčka a nastaví jejich barvu na bílou
    {
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox1.BackColor = Color.White;
        textBox2.BackColor = Color.White;
        textBox3.BackColor = Color.White;
    }
}
```

2. Individuální řešení