

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-105
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech I
Téma didaktického materiálu	Datové typy
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	Datový typ int, string, bool, Deklarace, typová konverze
Anotace	Studenti se učí pracovat s datovými typy, deklarovat je a používat při přiřazení
Použité zdroje	<p>DRÓZD, Januš a Rudolf KRYL. <i>Začínáme s programováním</i>. 1.vyd. Praha: Grada, 1992, 306</p> <p>TÖPFER, Pavel. <i>Algoritmy a programovací techniky</i>. 1. vyd. Praha: Prometheus, 1995, 299 s. ISBN 80-858-4983-6.</p> <p>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.</p> <p>VYSTAVĚL, Radek. <i>Moderní programování: učebnice pro pokročilé</i>. Ondřejov: moderníProgramování, 2011, 149 s. ISBN 978-80-903951-7-6.</p> <p>VYSTAVĚL, Radek. <i>Moderní programování: učebnice pro středně pokročilé</i>. Ondřejov: moderníProgramování s.r.o, 2008. ISBN 978-80-903951-2-1.</p> <p>VYSTAVĚL, Radek. <i>Moderní programování: učebnice pro začátečníky</i>. Ondřejov: moderníProgramování s.r.o, 2007, 2 sv. ISBN 978-80-903951-0-7.</p>
Typy k metodickému postupu učitele, doporučené výukové metody, způsob hodnocení, typy k individualizované výuce apod.	<p>Text je možno využít ke společné práci, samostatné přípravě studentů, domácímu studiu apod.</p> <p>Při společné práci je vhodné nejprve obtížnější úlohy rozebrat, potom společně se studenty</p>

	<p>implementovat na počítači. (Rozbor nejlépe na tabuli, synchronní řešení s promítáním)</p> <p>Prezentace obsahuje stručné shrnutí poznatků potřebných pro řešení příkladů. V pracovním listu je zadání cvičení – většinou se jedná o úlohy, které by měli studenti naprogramovat samostatně. Není nutné, aby všichni zpracovali všechno, vhodné je diferencovat podle jejich zájmu a schopností. Obtížnější úlohy jsou označeny hvězdičkou. Součástí materiálu je zdrojový kód těchto příkladů.</p> <p>Návrh způsobu hodnocení: ohodnocení samostatné práce během hodiny např. podle volby a počtu úloh a elaborace řešení (efektivnost, komentáře...).</p>
--	---

## 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.

## 105. Datové typy, práce s textem a čísly

Hodnoty vlastností, které jsme používali, jsou různých druhů – text (Text), číslo (TabIndex) barva výběrem ze seznamu (BackColor) , True nebo False (Enabled). Tento druh hodnot nazýváme v programování **datovým typem (typem)**. **Typ určuje, jakých hodnot může vlastnost nabývat a jaké operace se s ní dají dělat.**

### Názvy typů a jejich možné hodnoty

Textový řetězec	<b>string</b>	“Ahoj, babi”
Barva	<b>Color</b>	Color.Blue
Logická hodnota	<b>bool</b>	true, false
Celé číslo	<b>int</b>	5, -2010

### Operace s typy, typová konverze

Pokud sčítáme dva řetězce, dostaneme spojení řetězců: “15” + “10” = “1510”, součet čísel 15 + 10 =25.

Komponenta TextBox, kterou můžeme použít pro vstup i výstup údajů ovšem obsahuje pouze text. Chceme-li pracovat s čísly, je nezbytné provést typovou konverzi, k tomu slouží třída **Convert** a

například její metody **Convert.ToString** (výsledkem je řetězec) a **Convert.ToInt32** (výsledkem je celé číslo)

Pokud bychom tedy chtěli realizovat v programu výpočty, musíme nejprve převést údaje z textových polí na čísla, provést výpočty a výsledky opět převést na text, aby je bylo možné zobrazit.

## Komentáře

slouží k lepší orientaci v programu, můžeme pomocí nich zapsat tzv. **pseudokód** – návrh algoritmu v textové podobě. (Pokud používáme složitější postupy, vyplatí se nakreslit si dopředu vývojový diagram, jinak vystačíme s komentáři.)

Vývojové prostředí komentář ignoruje, je to každý text, který začíná dvěma lomítky do konce řádku. Také bývá barevně odlišen

```
// Toto je komentář
```

## Přiřazovací příkaz, proměnná

V **přiřazovacím příkaze** kam = co, může být „co“ výrazem. Ten se nejprve vyhodnotí, pak přiřadí. Levá i pravá strana musejí být téhož typu.

Výsledky a mezivýsledky si program ukládá do proměnných.

**Proměnná je symbolické označení místa v operační paměti, kde je uložena její hodnota.**

Proměnná se vytváří pomocí **deklarace**: (příkaz deklarace)

```
typ_proměnné jméno_proměnné;
```

Např. int cislo; string jmeno;

Deklarace s inicializací: (dosazení počáteční hodnoty)

```
typ_proměnné jméno_proměnné = počáteční_hodnota;
```

Např. int číslo = 10; string jmeno = "Anna";

Poznámka: Jméno proměnné může obsahovat diakritiku, ale ne mezery, nesmí také začínat číslicí. Jedná se o identifikátor, stejně jako u vlastnosti **name** ovládacích prvků.

### Příklad 1

Připravíme si formulář se třemi textovými poli (textBoxVstup1, textBoxVstup2, textBoxVystup) a dvěma tlačítky. Jedno tlačítko (buttonPlus) s textem Součet čísel, druhé (buttonSpoj) s textem Spojení slov. U textBoxVystup nastavíme vlastnost ReadOnly na True, aby do něho uživatel nemohl zapisovat. Je asi zřejmé, co by měl program dělat:

```
private void buttonSpoj_Click(object sender, EventArgs e)
{
    string vysledek;
    //deklarace výsledného řetězce
```

```

        vysledek = textBoxVstup1.Text + textBoxVstup2.Text;

        //spojeni řetězců

        textBoxVystup.Text = vysledek;
    }

```

Pokud bychom například chtěli oddělit vstupní řetězce mezerou a na začátek textu připsat slovo Spojeno:

```

vysledek = "Spojeno: "+textBoxVstup1.Text +" "+ textBoxVstup2.Text;

private void buttonPlus_Click(object sender, EventArgs e)
{
    int vysledek; //deklarace výsledného čísla

    vysledek = Convert.ToInt32(textBoxVstup1.Text) +
        Convert.ToInt32(textBoxVstup2.Text);

    //sečtení čísel převedených z textových poliček

    textBoxVystup.Text = Convert.ToString(vysledek);

    //Zobrazení výsledku převedeného na text
}

```

Zatímco spojování řetězců funguje vždy, pokud zadáte něco, co není celé číslo a budete chtít sčítat, program spadne. Jak se tomu dá zabránit se dozvíte příště.

## Příklad 2

Připravte si program se dvěma popisky (třeba LabelPes a LabelVes s adekvátním textem) a jedním tlačítkem buttonVymen a textem Výměna. Při kliknutí na tlačítko se text popisků vymění.

S výměnou obsahu dvou proměnných se v programování setkáte často a funguje to stejně, jako byste měli vyměnit obsah dvou stejných sklenic – potřebujete třetí (stejnou) sklenici.

```

private void buttonVymen_Click(object sender, EventArgs e)
{
    string pomocna; //prázdná sklenice

    pomocna = textBoxLes.Text;

    //schováme text do prázdné sklenice

    textBoxLes.Text = textBoxPes.Text;

    textBoxPes.Text = pomocna;
}

```

## Důležité

**Datový typ:** Každá vlastnost nebo hodnota je určitého typu. Typ určuje jakých hodnot může vlastnost nabývat a jaké operace se s ní dají dělat.

**string** – typ textových řetězců

**int** – typ celých čísel

**bool** – typ logických hodnot

**Color** – typ barev

V **přiřazovacím příkazu** kam = co, může být co výrazem. Ten se nejprve vyhodnotí, pak přiřadí. Levá i pravá strana musejí být téhož typu.

Výsledky a mezivýsledky si program ukládá do proměnných.

**Proměnná je symbolické označení místa v operační paměti, kde je uložena její hodnota**

Proměnná se vytváří pomocí **deklarace**:

**typ\_proměnné jméno\_proměnné;**

**Deklarace s inicializací:**

**typ\_proměnné jméno\_proměnné = počáteční\_hodnota;**

Typová konverze převádí typ hodnoty:

**Convert.ToInt32** – na celé číslo

**Convert.ToString** – na řetězec

**Výměna dvou obsahu objektů** – přes pomocnou proměnnou, jako při výměně obsahu dvou sklenic

## Pracovní list

### Cvičení

1. Naprogramujte sčítání, odčítání, násobení a celočíselné dělení vstupujících čísel.  
(Pro operaci násobení se užívá \*, pro dělení / jako v Excelu. V C# platí, že při dělení celých čísel je výsledek celočíselný podíl. ( $12 / 5 = 2$ ))
2. Naprogramujte výměnu barev dvou textových políček. (Použijte pomocnou proměnnou typu Color)
3. Uživatel zadá podmět a přísudek, program vypíše větu.
4. Co bude v proměnných a, b po provedení následujících příkazů?  
int a = 10; int b = 20;  
a = b;  
b = a;

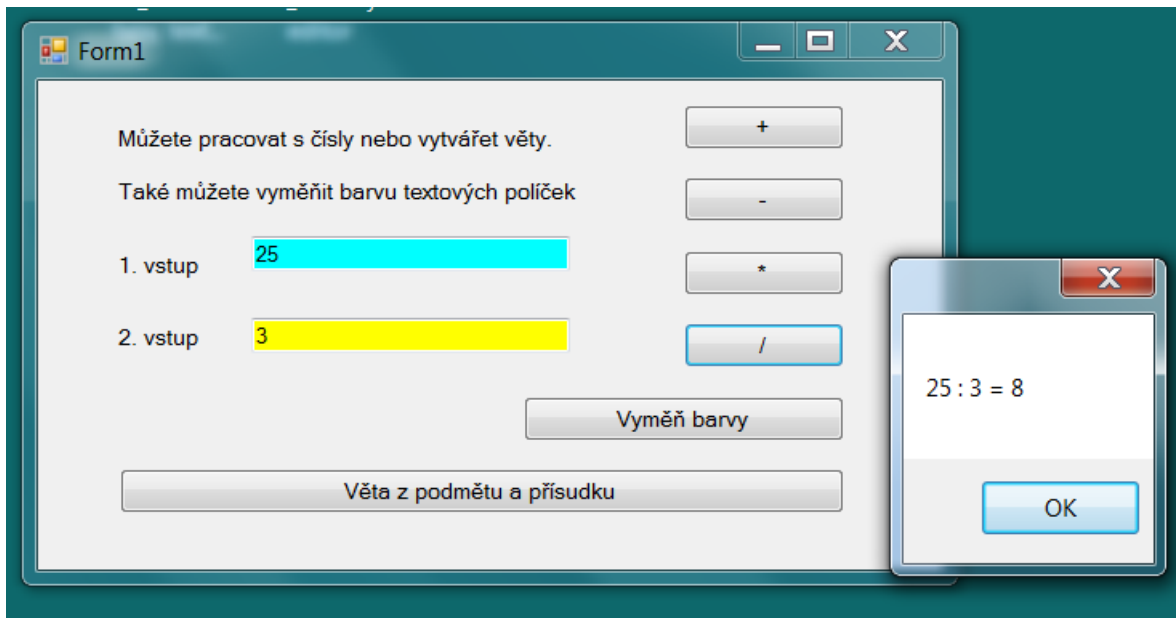
5. Co bude v proměnných a, b po provedení následujících příkazů?

```
int a = 10; int b = 20;
```

```
a = a + b;
```

```
b = a - b;
```

```
a = a - b;
```



### Řešení

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

```
{  
  
    int x, y, z;  
  
    x = Convert.ToInt32(textBox1.Text);  
    y = Convert.ToInt32(textBox2.Text);  
    z = x + y;  
  
    MessageBox.Show(Convert.ToString(x) + " + " + Convert.ToString(y) +  
        " = " + Convert.ToString(z)) ;  
}
```

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

```
{  
  
    int x, y, z;  
  
    x = Convert.ToInt32(textBox1.Text);  
    y = Convert.ToInt32(textBox2.Text);
```

```

        z = x - y;

        MessageBox.Show(Convert.ToString(x) + " - " + Convert.ToString(y) +
            " = " + Convert.ToString(z));
    }

private void buttonKrat_Click(object sender, EventArgs e)
{
    int x, y, z;

    x = Convert.ToInt32(textBox1.Text);
    y = Convert.ToInt32(textBox2.Text);
    z = x * y;

    MessageBox.Show(Convert.ToString(x) + " * " + Convert.ToString(y) +
        " = " + Convert.ToString(z));
}

private void buttonDeleno_Click(object sender, EventArgs e)
{
    int x, y, z;

    x = Convert.ToInt32(textBox1.Text);
    y = Convert.ToInt32(textBox2.Text);
    z = x / y;

    MessageBox.Show(Convert.ToString(x) + " : " + Convert.ToString(y) +
        " = " + Convert.ToString(z));
}

private void buttonVymen_Click(object sender, EventArgs e)
{
    Color pom; //deklarace pomocné barvy

    pom = textBox1.BackColor;

    textBox1.BackColor = textBox2.BackColor;

    textBox2.BackColor = pom;
}

```

```
}
```

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

```
{
```

```
    string veta;
```

```
    veta = textBox1.Text + " " + textBox2.Text + ".";
```

```
    MessageBox.Show(veta);
```

```
}
```

4.  $a = 20$ ,  $b = 20$ , k výměně nelze použít.

5.  $a = 20$ ,  $b = 10$ , lze použít k výměně číselných proměnných bez pomocné proměnné.