

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-108
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech I
Téma didaktického materiálu	Podmíněné příkazy
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	If, else, typ bool
Anotace	Studenti se učí programovat jednodušší úlohy, jejichž řešení vyžaduje větvení programu.
Použité zdroje	<p>DRÓZD, Januš a Rudolf KRYL. <i>Začínáme s programováním</i>. 1.vyd. Praha: Grada, 1992, 306 s. ISBN 80-854-2441-X.</p> <p>KUKAL, Jaromír. <i>Myšlením k algoritmům</i>. Praha: Grada, 1992, 131 s. ISBN 80-854-2447-9.</p> <p>LIBICHER, Ivan a Pavel TÖPFER. <i>Od problému k algoritmu a programu: sbírka řešených úloh z programování</i>. 1. vyd. Praha: Grada, 1992, 119 s. Educa '99. ISBN 80-854-2482-7.</p> <p>TÖPFEROVÁ, Dana a Pavel TÖPFER. <i>Sbírka úloh z programování</i>. Vyd. 1. Praha: Grada, 1992, 98 s. Educa '99. ISBN 80-854-2499-1.</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 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 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.

108. Podmíněné vykonávání a větvení programu

Naše programy dosud vykonávaly příkazy v tom pořadí, jak byly zapsány. (s výjimkou výjimky) Často ale potřebujeme provést nějaký příkaz jen tehdy, je-li splněna jistá podmínka (odmocnit pouze kladné číslo) nebo vykonat různé příkazy podle určitých okolností.

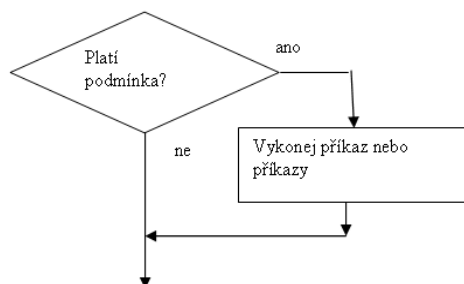
Konstrukce if (neúplný podmíněný příkaz)

má tvar **if (podmínka)**

```

příkaz;
nebo if (podmínka)
{
příkaz 1;
příkaz 2;
...
}

```



Podmínka je výraz, (vlastně výrok) jehož hodnota je pravda nebo nepravda – tedy **true** nebo **false**, Datový typ se jmenuje **bool**.

(Stejně jako v matematice – výrok má hodnotu pravda nebo nepravda)

Jestliže je podmínka splněna, provede se požadovaný příkaz, jinak se nevykoná nic a program pokračuje dál.

Zápisy podmínek

$a = b$ (**$a == b$**) je třeba odlišit rovnost od přiřazovacího příkazu, proto se píší dvě rovnítka za sebou

$a \leq b$ (**$a <= b$**)

$a < b$ (**$a < b$**)

$a \neq b$ (**$a != b$**)

Příklad 1

Do dvou textových políček budeme vkládat čísla, přitom chceme, aby v prvním políčku bylo číslo větší nebo rovné číslu druhému. Kdyby uživatel zadal čísla obráceně, vyměníme je.

```
{  
  
    double vetsi = Convert.ToDouble(textBoxVetsi.Text);  
  
    double mensi = Convert.ToDouble(textBoxmensi.Text);  
  
    double pom;  
  
    if (vetsi < mensi)  
    {  
  
        pom = vetsi;  
  
        vetsi = mensi;  
  
        mensi = pom;  
  
        textBoxVetsi.Text = Convert.ToString(vetsi);  
  
        textBoxmensi.Text = Convert.ToString(mensi);  
  
    }  
  
}
```

Rozvětvení programu

K větvení programu se používá konstrukce **if-else**.

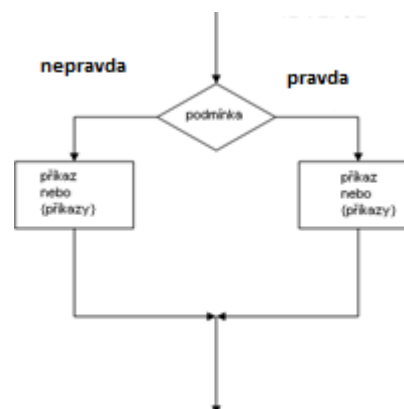
if (podmínka)

Příkaz 1;

else

Příkaz 2;

Jestliže je podmínka splněna, provede se Příkaz 1,



pokud splněna není, provede se Příkaz 2.

Příkaz 1 a příkaz 2 mohou být jednoduché, ale i složené – posloupnosti příkazů, další větvení apod., pak je třeba je uzavřít mezi složené závorky.

Příklad 2

Vstupem bude reálné číslo, výstupem by měla být odmocnina z něj, pokud je číslo nezáporné, jinak zpráva pro uživatele, že odmocnina neexistuje.

```
private void buttonOdmoc_Click(object sender, EventArgs e)
{
    double odmoc;

    try
    {
        double x = Convert.ToDouble(textBoxCislo.Text);

        if (x >= 0)
        {
            odmoc = Math.Sqrt(x);

            MessageBox.Show(Convert.ToString(odmoc));
        }

        else
            MessageBox.Show("Odmocnina ze záporného čísla v reálných
číislech neexistuje");
    }

    catch
    {
        MessageBox.Show("Musíte zadat číslo");
    }
}
```

Příklad 3

Vstupem je celé číslo, program by měl určit, zda je dělitelné třemi.

Pravidlo o ciferném součtu zatím naprogramovat neumíme, spokojíme se tedy s (jednodušším) řešením – číslo je dělitelné třemi, když je jeho zbytek po vydělení třemi roven nule.

```
private void buttonD3_Click(object sender, EventArgs e)
```

```
{  
  
    int cislo = Convert.ToInt32(textBoxCislo.Text);  
  
    int zbytek=cislo % 3;  
  
    if (zbytek==0)  
  
        MessageBox.Show("ano");  
  
    else  
  
        MessageBox.Show("ne");  
  
}
```

Důležité

Podmíněný příkaz:

if (podmínka)

příkaz;

nebo **if (podmínka)**

{

příkaz 1;

příkaz 2;

...

}

Podmínka je typu **bool**, může nabývat hodnot true nebo false.

Zápisy podmínek

rovná se: **(a == b)**, menší nebo rovno: **(a <= b)**, různó: **(a != b)**

Úplné větvení

if (podmínka)

Příkaz 1;

else

Příkaz 2;

|x| **Math.Abs(x)**

Pracovní list

Cvičení

1. Vstupem bude číslo, výstupem jeho absolutní hodnota podle definice. (Třída Math má samozřejmě metodu Math.Abs, ale zkuste to vymyslet po svém)
2. (*) Jsou známy údaje o dvou autech. První ujelo trasu s_1 v čase t_1 , druhé trasu s_2 v čase t_2 . Napište program, který rozhodne, zda jsou rychlosti automobilů stejné a v případě že ne, které je rychlejší a o kolik je jeho rychlost větší.

Výsledky zobrazte pomocí MessageBoxů, opatřete je vysvětlujícím textem, alespoň v jednom příkladu ošetřete výjimku špatně zadaného vstupu.

Řešení

1.

```
private void buttonAbs_Click(object sender, EventArgs e)
{
    //abs. hodnota
    double Absx;

    try
    {
        double x = Convert.ToDouble(textBoxCislo.Text);

        if (x >= 0)
            Absx = x;

        else
            Absx = -x;

        MessageBox.Show("|" + Convert.ToString(x) + "| = " +
Convert.ToString(Absx));
    }

    catch
    {
        MessageBox.Show("Zadejte pouze reálné číslo");
        textBoxCislo.Text = null;
        textBoxCislo.Focus();
    }
}
```

2.

```
private void buttonPorovnani_Click(object sender, EventArgs e)
{
    //porovnání rychlostí aut
    double s1, s2, t1, t2, v1, v2,okolik;
    //deklarace proměnných - názvy obvyklé ve fyzice
    // o kolik bude absolutní hodnota rozdílu obou rychlostí
    s1 = Convert.ToDouble(textBoxS1.Text);
    s2 = Convert.ToDouble(textBoxS2.Text);
    t1 = Convert.ToDouble(textBoxT1.Text);
    t2 = Convert.ToDouble(textBoxT2.Text);
    //načtení vstupních hodnot
    v1 = s1 / t1;
    v2 = s2 / t2;
    //výpočet rychlostí
    if (v1 == v2)
        MessageBox.Show("v1 = v2 = " + Convert.ToString(v1)+"km/h");
    else
    {
        okolik = Math.Abs(v1 - v2);
        //o kolik se liší
        if (v1 > v2)
            MessageBox.Show("První je rychlejší o " +
Convert.ToString(okolik) + "km/h");
        else
            MessageBox.Show("Druhé je rychlejší o " +
Convert.ToString(okolik) + "km/h");
    }
}
```