

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-302
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech III
Téma didaktického materiálu	Vlastní pomocné metody – parametry metod
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 17–18 let
Úroveň žáků	Středně pokročilí
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Formální a skutečné parametry, parametry předávané hodnotou a odkazem, strukturované programování
Anotace	Studenti se seznamují s programováním vlastních metod, rozlišují typy parametrů a jejich použití, sestavují projekty z podprogramů
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>ELLER, Frank. <i>C# - začínáme programovat: podrobný průvodce začínajícího uživatele</i>. 1. vyd. Praha: Grada, 2002, 240 s. ISBN 80-247-0324-6.</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>TÖPFER, Pavel. <i>Algoritmy a programovací techniky</i>. 1. vyd. Praha: Prometheus, 1995, 299 s. ISBN 80-858-4983-6.</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: 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>
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</p>

	<p>tabuli, synchronní řešení s promítáním) 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. 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.

302. Vlastní pomocné metody – parametry metod

Parametry metod

Na parametry metod se můžeme dívat z více hledisek.

Formální a skutečné parametry

V hlavičce metod (ne nutně všech) jsou parametry, kterým říkáme formální. Říkáme, že jejich prostřednictvím metoda komunikuje se svým okolím – při jejím volání se za ně dosadí parametry skutečné. Samozřejmě si musejí odpovídat co do typu, počtu a pořadí, podle kterého tato substituce probíhá. Situaci můžeme přirovnat ke scénáři, v němž se vyskytují postavy (formální parametry), které při natáčení budou hrát skuteční herci. (Skutečné parametry)

Parametry předávané hodnotou a odkazem

Parametry, se kterými jsme dosud pracovali, jsou předávány hodnotou – předává se pouze hodnota, která je metodou zpracována, původní hodnota proměnných se nemění. Takto předávané parametry se v těle metody chovají jako lokální proměnné. Pokud ale potřebujeme původní proměnnou změnit, je třeba předat odkaz na ni. Tímto způsobem může být také parametr strukturovaný (např. pole) nebo jich může být víc. V hlavičce procedury při deklaraci i volání musí tento parametr předcházet slovo **ref**.

Příklad 1.

Napište metodu pro výměnu obsahu dvou řetězcových proměnných a použijte ji pro výměnu dvou textových polí.

```
private void Vymen(ref string a, ref string b)
{
    //výměna obsahu dvou řetězcových proměnných
    string pom = a; //lokální pomocná proměnná
    a = b;
    b = pom;
}
```

Všimněte si, že také uvnitř vlastních metod můžete používat potřebné lokální proměnné.

Použití:

```
private void buttonVymen_Click(object sender, EventArgs e)
{
    string x = textBoxVstup1.Text;
    string y = textBoxVstup2.Text;
    Vymen(ref x, ref y);
    textBoxVstup1.Text = x;
    textBoxVstup2.Text = y;
}
```

Poznámky:

Pokud vynecháte označení parametrů jako ref, neobjeví se chyba, ale metoda nic vyměňovat nebude.

Pokud byste chtěli při volání napsat přímo `Vymen(ref textBoxVstup1.Text, ref textBoxVstup2.Text)`, systém vám to nedovolí, protože očekává odkaz na proměnnou. Skutečný parametr zde tedy může být pouze proměnná.

Příklad 2.

Napište metodu, do které bude vstupovat trojice čísel a jejím výstupem budou tatáž čísla, přičemž budou seřazena vzestupně. (První menší, druhé větší, třetí největší)

Setřídít trojici čísel můžeme šikovně probubláním, ale pozor – naše metoda na výměnu se týká řetězců, musíme ji tedy nejdřív upravit pro čísla.

```
private void VymenCisla(ref int a1, ref int b1)
{
```

```

        //výměna obsahu dvou celočíselných proměnných
        int pom = a1;//lokální pomocná proměnná
        a1 = b1;
        b1 = pom;
    }
    private void Serad(ref int aa, ref int bb, ref int cc)
    {
        if (aa>bb) VymenCisla(ref aa, ref bb);
        if (bb > cc) VymenCisla(ref bb, ref cc);//maximum na konec
        if (aa>bb) VymenCisla(ref aa, ref bb);
        //Kdyby na začátku ještě nebylo minimum
    }
}

```

Příklad 3.

Dokonalé číslo je součtem svých vlastních dělitelů ($6 = 1 + 2 + 3$).

Naprogramujte metodu, která bude vracet true v případě, že číslo je dokonalé, jinak false.

Tento příklad není na parametry předávané odkazem, ale může sloužit jako ukázka toho, že návratová hodnota metody může být v zásadě jakýkoliv typ – tedy i logický.

Budeme procházet čísla do poloviny z testovaného čísla a pokud je toto číslo jimi dělitelné, posčítáme je. Návratová metody pak bude výsledek rovnosti
 součet dělitelů = testované číslo

```

private bool DokCislo(int c)
{
    int soucet = 0;
    int pul = c / 2;
    for (int i = 0; i <= pul; i++)
        if (c % i == 0)
            soucet += i;
    return (soucet == c);
}

```

Důležité

Parametr může být předávaný pouze hodnotou (vhodné pro vstup) nebo odkazem, kde mu při deklaraci i volání předchází slovo **ref**.

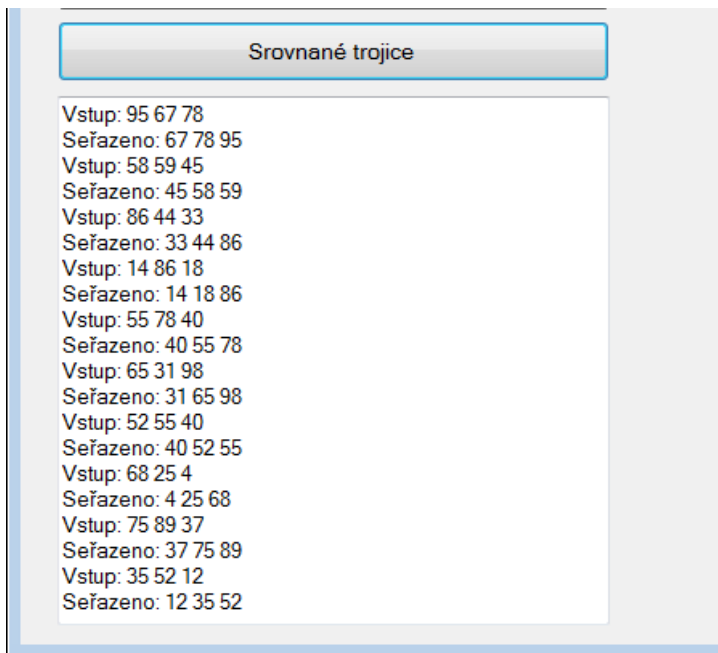
Pracovní list

Cvičení

1. Napište metodu, která vypočítá obsah a obvod kruhu, vstupem bude poloměr. Vyzkoušejte ji v programu. Poloměr je zde vstupní parametr, takže se předává hodnotou, zatímco obsah a obvod budou parametry předávané odkazem.

```
private void ObsahAObvod(double r, ref double S, ref double O)
```

2. Napište program, který vygeneruje deset trojic náhodných celých čísel a vytiskne každou trojici nejprve nestrhříděnou, pak setříděnou metodou Serad.



3. Použijte metodu DokonalCislo k vygenerování všech dokonalých čísel menších než 100.

Řešení

1.

```
private void ObsahAObvod(double r, ref double S, ref double O)
{
    //Protože výstupem budou dvě hodnoty, použijeme místo return
    // předání výsledku odkazem
    S = Math.PI * r * r;
    O = Math.PI * r * 2;
}

private void button00_Click(object sender, EventArgs e)
{
```

```

double r = Convert.ToDouble(textBoxVstup1.Text);

double S1 = 0; double O1 = 0;

ObsahAObvod(r, ref S1, ref O1);

//Při volání metody se vypočítané hodnoty přiřadí do

//proměnných S1 a O1

MessageBox.Show("S = " + S1.ToString("F2") + Environment.NewLine +
"O = " + O1.ToString("F2"));
}

```

2.

```

private void buttonSeTr_Click(object sender, EventArgs e)
{
// Generování a řazení deseti trojic celých čísel

Random nahoda = new Random();

for (int i = 0; i < 10; i++)
{
//Generujeme trojici

int x = nahoda.Next(100);

int y = nahoda.Next(100);

int z = nahoda.Next(100);

//Zobrazíme

textBoxVystup.Text += "Vstup: " + x.ToString() + " " +
y.ToString() + " " + z.ToString()+Environment.NewLine;

//Setřídíme

Serad(ref x, ref y, ref z);

//Zobrazíme

textBoxVystup.Text += "Seřazeno: " + x.ToString() + " " +
y.ToString() + " " + z.ToString() +Environment.NewLine;
}
}

```

3.

```

private void buttonDC_Click(object sender, EventArgs e)
{
// Vypis dokonalých čísel menších než 100

```

```
for (int i = 1; i <= 100; i++)  
    if (DokCislo(i))  
        textBoxVystup.Text += i.ToString() + Environment.NewLine;  
}
```