Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-301
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
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	Metoda, třída, návratová hodnota, strukturované programování
Anotace	Studenti se seznamují s programováním vlastních metod, chápou pojem návratové hodnoty, sestavují projekty z podprogramů
Použité zdroje	DRÓZD, Januš a Rudolf KRYL. *Začínáme s programováním*. 1.vyd. Praha: Grada, 1992, 306 s. ISBN 80-854-2441-X. ELLER, Frank. *C# - začínáme programovat: podrobný průvodce začínajícího uživatele*. 1. vyd. Praha: Grada, 2002, 240 s. ISBN 80-247-0324-6. LIBICHER, Ivan a Pavel TÖPFER. *Od problému k algoritmu a programu: sbírka řešených úloh z programování*. 1. vyd. Praha: Grada, 1992, 119 s. Educa '99. ISBN 80-854-2482-7. TÖPFER, Pavel. *Algoritmy a programovací techniky*. 1. vyd. Praha: Prometheus, 1995, 299 s. ISBN 80-858-4983-6. TÖPFEROVÁ, Dana a Pavel TÖPFER. *Sbírka úloh z programování*. Vyd. 1. Praha: Grada, 1992, 98 s. Educa '99. ISBN 80-854-2499-1. VYSTAVĚL, Radek. *Moderní programování: učebnice pro pokročilé*. Ondřejov: moderníProgramování, 2011, 149 s. ISBN 978-80-903951-7-6. VYSTAVĚL, Radek. *Moderní programování: učebnice pro středně pokročilé*. Ondřejov: moderníProgramování s.r.o, 2008. ISBN 978-80-903951-2-1.
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. Při společné práci je vhodné nejprve obtížnější úlohy rozebrat, potom společně se studenty implementovat na počítači. (Rozbor nejlépe na 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…).

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.

301. Vlastní pomocné metody

S metodami, které jsou připraveny v knihovnách C# pracujeme od začátku, užíváme je, aniž bychom potřebovali vědět, jak jsou naprogramovány.

Použijeme-li například Convert.ToInt32(textBox1.Text), metoda si vezme řetězec ve tvaru čísla v textovém políčku, rozloží ho na cifry a z nich poskládá číslo. Řetězec v závorkách je jejím vstupním parametrem, získané číslo pak můžeme dál používat.

Na rozdíl od toho metoda třídy Environment Environment.NewLine zajistí přechod na nový řádek a žádné další údaje nepotřebuje.

V C# jsou metody nedílnou součástí tříd, se kterými se seznámíme později a rozhodně byste tak měli programovat větší programové celky, kde se uživatelské rozhraní od vlastní logiky aplikace odděluje. U menších programů nám ale metody, které definujeme přímo ve třídě okna programu, mohou usnadnit a zjednodušit práci i přehled v kódu.

Tvorbu vlastních pomocných metod si ukážeme na několika jednoduchých příkladech. Deklarace metody, kde vlastně popisujeme, co má metoda provádět, umístíme na začátek programu – tam, kde deklarujeme členské proměnné.

Připomeňme si, jak vypadají obslužné metody, kterými jsme se už zabývali.

```
private void Form1_Load(object sender, EventArgs e)
        {
```

```
            textBoxVstup1.ReadOnly = true;
        }
```

První řádek obsahuje hlavičku, která se skládá z názvu a pokud metoda pracuje s parametry, pomocí kterých komunikuje se svým okolím, uvádějí se v závorce včetně typu, odděleny čárkami.

Před jménem metody je slovo **void** u metod, které provádějí nějakou akci nebo typ tzv. návratové hodnoty. Slovo **private** souvisí s možnostmi přístupu k metodě. Více až se budeme zabývat vlastními třídami)

Příklad 1.

Naprogramujte metodu, která přebarví formulář na červeno.

```
public partial class Form1 : Form
    {
        private void Cervena()
        {
            //přebarví formulář na červeno
            BackColor=Color.Red;
        }
```

Použití:

```
private void buttonCervena_Click(object sender, EventArgs e)
        {
            //volání metody bez parametrů
            Cervena();
         }
```

Příklad 2.

Naprogramujte metodu, která přebarví formulář na barvu, která bude jejím parametrem.

```
private void Barva(Color b)
        {
            //přebarví formulář na barvu b
            BackColor = b;
        }
```

Použití:

```
private void buttonModra_Click(object sender, EventArgs e)
        {
            //volání metody s parametrem
            Barva(Color.Blue);
        }
```

Připomeňme si ještě jednou metodu ToString třídy Convert. Chceme-li převést řetězec v textovém poli na číslo, můžeme ji použít například ve tvaru: `int x = Convert.ToInt32(textBoxVstup1.Text);`

V závorce uvedeme text – vstupní parametr metody a podobně, jako zacházíme s funkční hodnotou v matematice, můžeme číslo, které jsme tímto způsobem získali, přiřadit do proměnné.
V programování hovoříme o návratové hodnotě metody.

Příklad 3.

Naprogramujte metodu, do které budou vstupovat dvě čísla a která bude vracet větší z nich.

```
private int vetsi(int a, int b)
        {
            //vrací větší číslo ze dvou
            if (a > b)
                return a;
            else return b;
        }
```

Před uvedením návratové hodnoty v těle metody musí být uvedeno slovo **return**.

Použití:

```
private void buttonVetsi_Click(object sender, EventArgs e)
        {
            int x = Convert.ToInt32(textBoxVstup1.Text);
            int y = Convert.ToInt32(textBoxVstup2.Text);
            int max = vetsi(x, y);
            MessageBox.Show("Větší je " + max);
        }
```

Příklad 4.

Naprogramujeme metodu, do které budou vstupovat celočíselný (přirozený) základ a exponent a vystupovat přirozená mocnina – tedy základ na exponent. Použijeme opakované násobení základu.

```
private int Mocnina(int z, int exp)
        {
            int moc = 1;//mocnění = opakované násobení
            for (int i = 0; i < exp; i++)
                moc *= z;
            return moc;
        }
```

Použijte tuto metodu k výpisu mocnin 2 od 2^1 do 2^{10}.

```
        private void buttonMoc2_Click(object sender, EventArgs e)
        {
            textBoxVypis.Text += "Mocniny 2" + Environment.NewLine;
            for (int i = 1; i < 11; i++)
            {
                int m2 = Mocnina(2, i);
textBoxVypis.Text += i.ToString() + " " +   m2.ToString()+Environment.NewLine;
            }
          }
```

Důležité

Vlastní metody nejprve deklarujeme, pak voláme.

Deklarace se skládá z hlavičky, která obsahuje název a parametry s uvedením jejich typu a těly, kde jsou příkazy, které má metoda vykonat.

Pokud mají metody návratovou hodnotu, musí být v deklaraci označena slovem **return.**

Pracovní list

Cvičení

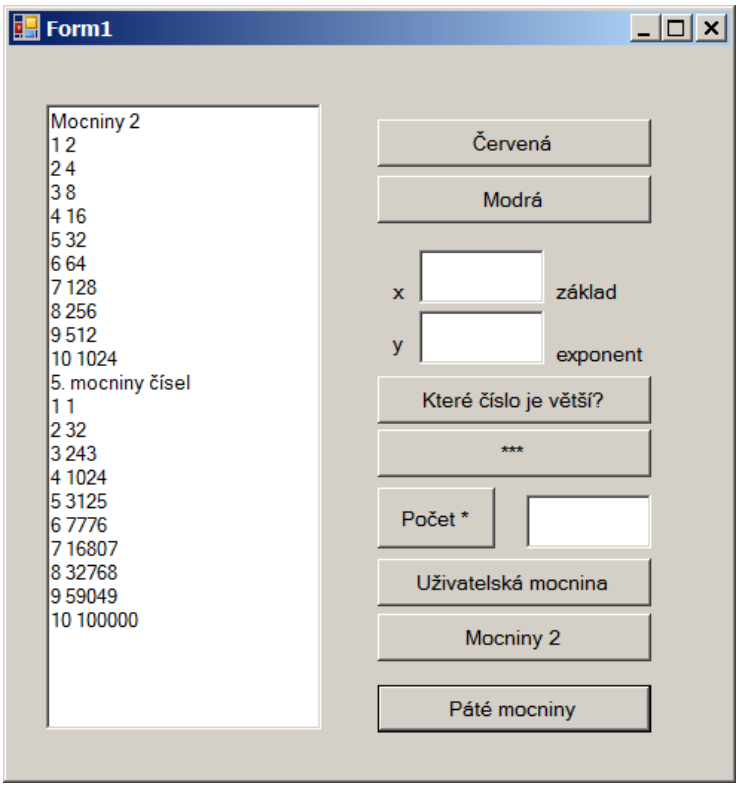
Všechny metody nejprve nadeklarujte a pak vhodně použijte ve svém programu.

1. Využijte metodu, která počítá přirozenou mocninu (Příklad 4 z teoretické části) a zobrazte pátou mocninu čísel od 1 do 10.

2. Naprogramujte metodu bez parametrů, které zapíše do textového pole tři hvězdičky.

3(*). Naprogramujte metodu, jejímž vstupním parametrem bude počet hvězdiček, zadaný uživatelem, který zapíše do textového pole

4. Vymyslete si vlastní metodu, která bude počítat podle nějakého matematického nebo fyzikálního vzorce.



Řešení

1.

```
private void button5moc_Click(object sender, EventArgs e)
        {
           //výpis pátých mocnin čísel 1 až 10
            textBoxVypis.Text += "5. mocniny čísel" + Environment.NewLine;
            for (int i = 1; i < 11; i++)
            {
                int m5 = Mocnina(i, 5);
                textBoxVypis.Text += i.ToString() + " " + m5.ToString() +
Environment.NewLine;
            }
        }
```

2.

```
private void Hvezdy3()
```

```
        {
            //Metoda vypíše tři hvězdičky do textového pole
            textBoxVystup.Text += "***" + Environment.NewLine;
        }
```

3.

```
        private void Hvezdy(int n)
        {
          //Metoda vypíše n hvězdiček do textového pole
            string pom = null;
            //Nejprve vytvoříme do pomocného řetězce potřebný počet hvězdiček
            for (int i = 0; i < n; i++)
            {
                pom += "*";
            }
            textBoxVystup.Text += pom + Environment.NewLine;
        }
```

4. Příklad – obsah kruhu

```
        private double Obsah(double r)
        {
            return Math.PI * r * r;
        }
```

Použití metod v programu:

```
private void button3H_Click(object sender, EventArgs e)
        {
            Hvezdy3();
        }
        private void buttonHv_Click(object sender, EventArgs e)
        {
            Hvezdy(Convert.ToInt32(textBoxVstup3.Text));
        }
        private void buttonObsah_Click(object sender, EventArgs e)
```

```
{
    double r = Convert.ToDouble(textBoxVstuo4.Text);
    MessageBox.Show(Obsah(r).ToString("F2"));
}
```