

<b>Číslo a název šablony</b>	<b>III/2 Inovace a zkvalitnění výuky prostřednictvím ICT</b>
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-120
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech I
Téma didaktického materiálu	Další příklady na cyklus for
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	Faktoriál, aritmetický průměr, přetečení
Anotace	Studenti programují známé vztahy z matematiky a seznamují se s dalším ošetřením výjimek
Použité zdroje	<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, způsoby 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>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. 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</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.

## 120. Další příklady na cyklus for

### Příklad 1

Naprogramujte výpočet aritmetického průměru 10 náhodných jednociferných čísel.

Čísla je třeba nejdřív sečíst a výsledek vydělit 10 – viz komentáře

```
private void buttonPrum_Click(object sender, EventArgs e)
{
    //aritmetický průměr 10 čísel
    Random nahod = new Random();
    double suma = 0;
    //proměnná pro součet čísel - aby šlo přesně reálně dělit 10
    for (int i = 0; i < 10; i++)
    {
        //10* za sebou
        int x = nahod.Next(0, 10);
        textBoxNahoda.Text += x.ToString() + Environment.NewLine;
        //generujeme a zapisujeme nové číslo
        suma += x;
        //přičítáme ho do součtu
    }
    double prumer = suma / 10;
```

```

//vypočítáme průměr
MessageBox.Show("Průměr je " + prumer.ToString());
}

```

## Příklad 2

Naprogramujte výpočet faktoriálu:  $n! = 1*2*3*...*n$ . (n zadá uživatel to textBoxu)

$0! = 1$ ;  $1! = 1$ ;  $2! = 1*2 = 2$ ;  $3! = 1*2*3 = 6$ , ...

Budeme-li mít proměnnou pro faktoriál – faktor, na začátku do něj vložíme jedničku, potom ho vynásobíme dvěma, pak třemi atd., k tomuto opakovanému násobení můžeme s výhodou použít řídicí proměnnou cyklu.

```

private void buttonFaktor_Click(object sender, EventArgs e)
{
    int n;
    try
    {
        n = Convert.ToInt32(textBoxFaktor.Text);
        if (n < 0)
            throw new Exception();
    }
    catch
    {
        MessageBox.Show("Chybný vstup");
        return;
    }
    //odchycení nekorektního vstupu
    int faktor = 1;
    for (int i = 1; i <= n; i++)
    {

```

```

        faktor = faktor * i;
    }

    MessageBox.Show(n.ToString() + "! = " + faktor.ToString("N0"));
}
}

```

V programu ošetříme výjimku nekorektního vstupu – ale to by mohlo být i zadání záporného čísla, pro které faktoriál nedefinujeme.

Proto testujeme, zda je vstupující číslo kladné a jestliže ne, vyvolá se příkazem **throw** výjimka, která se zachytí v následujícím bloku **catch** a předá uživateli zprávu.

Poznámka: `faktor.ToString("N0")` – N0 je formátovací řetězec (prohlédněte si nabídku našeptávače) N zajistí seskupení větších čísel po trojicích a 0 znamená 0 desetinných míst.

## Přetečení

Výsledky fungují dobře až někam k faktoriálu 12. Pak začnou být podivné a od určité hodnoty bude systém oznamovat nulu. Problém je v tom, že výsledky jsou příliš velká čísla a nevejdou se do paměti přidělené proměnné typu `int`.

Pro nás zatím nejprostší řešení je rozšířit podmínku vyvolání výjimky:

```

if ((n < 0) || (n>12))
    throw new Exception();

```

## Příklad 3

Napište program, který vytiskne tabulku hodnot funkce  $y = x^2$ . Proměnná `x` bude nabývat hodnot od -2 do 2 a mezi nimi se bude zvětšovat vždy o 0,1.

```

private void buttonTab_Click(object sender, EventArgs e)
{
    string radek; //vystup ve tvaru x dvě mezery y
    for (double x = -2; x < 2.1; x += 0.1)
    {
        double y = x * x;
        radek = x.ToString("F2") + "    " + y.ToString("F2");
        //F2 zobrazení na dvě desetinná místa
    }
}

```

```

        textBoxNahoda.Text += radek + Environment.NewLine;
    }
}

```

**Poznámka:** Všimněte si, že nestačí přidávat hodnoty do  $x \leq 2$ , poslední řádek by chyběl. (Vyzkoušejte). Je to dáno tím, že aritmetika reálných čísel je díky omezené paměti pro číslo a také vnitřní reprezentaci čísla ve dvojkové soustavě nepřesná.

## Pracovní list

### Cvičení

1. Vygenerujte 10 náhodných celých čísel od 0 do 5 a vypočítejte součet lichých a součet sudých.
2. Vypočítejte přirozenou mocninu opakovaným násobením. Vstupem bude základ a exponent, zadané uživatelem. Zformátujte výsledek pro velká čísla. Ohlídejte výjimkou zadání pouze kladných čísel.

mocnina = základ\*základ...\*základ – počet činitelů je roven exponentu

3. Vypočítejte součet čísel od 1 do n. (Je na to jednoduchý vzoreček:  $n*(n-1)/2$  – vyzkoušejte si ho, ale tady to proveďte cyklem). Přirozené číslo n zadá uživatel.

### Řešení

1.

```

private void buttonSoucLich_Click(object sender, EventArgs e)
{
    //součet lichých čísel z 10 náhodně generovaných
    Random nahod = new Random();

    int sumaLich = 0;
    int sumaSud = 0;

    for (int i = 0; i < 10; i++)
    {
        int x = nahod.Next(0, 6);

        textBoxNahoda.Text += x.ToString() + Environment.NewLine;

        if (x % 2==0)
            sumaSud += x;
        else
            sumaLich += x;
    }
}

```

```
    }  
  
    MessageBox.Show("Součet lichých je " + sumaLich.ToString()+  
        Environment.NewLine + "Součet sudých je "+sumaSud.ToString());  
}
```

2.

```
private void buttonMoc_Click(object sender, EventArgs e)  
{  
    //Výpočet přirozené mocniny opakovaným násobením základu  
    int zak, exp, mocnina;  
  
    try  
    {  
        zak = Convert.ToInt32(textBoxZak.Text);  
        exp = Convert.ToInt32(textBoxExp.Text);  
  
        if ((zak <= 0) || (exp <= 0))  
            throw new Exception();  
    }  
  
    catch  
    {  
        MessageBox.Show("Zadejte pouze kladné hodnoty");  
        return;  
    }  
  
    mocnina = 1;  
  
    for (int i = 0; i < exp; i++)  
    {  
        mocnina *= zak;  
    }  
  
    MessageBox.Show(zak.ToString() + " na "+exp.ToString()+  
        " je "+ mocnina.ToString("N0"));  
}
```

3.

```
private void buttonsoucet_Click(object sender, EventArgs e)
{
    //výpočet součtu 1+2+...n
    int n = Convert.ToInt32(textBoxN.Text);
    int soucet = 0;
    for (int i = 1; i < n+1; i++)
    {
        soucet += i;
    }
    MessageBox.Show("Součet je " + soucet.ToString());
}
```