Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-118
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech I
Téma didaktického materiálu	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	Cyklus, cyklus for
Anotace	Studenti se učí zpracovávat větší množství dat pomocí cyklu s předem známým počtem opakování
Použité zdroje	DRÓZD, Januš a Rudolf KRYL. *Začínáme s programováním*. 1.vyd. Praha: Grada, 1992, 306 s. ISBN 80-854-2441-X. 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: sbírka úloh k učebnici pro začátečníky*. 2. vyd. Ondřejov: moderníProgramování, 2008, 2 sv. ISBN 978-80-903951-5-2. VYSTAVĚL, Radek. *Moderní programování: učebnice pro začátečníky*. Ondřejov: moderníProgramování s.r.o, 2007, 2 sv. ISBN 978-80-903951-0-7.
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.

118. Cyklus For

Často se setkáváme se situací, kdy je třeba nějakou akci mnohokrát opakovat. Někdy víme předem kolikrát (obejdi náměstí desetkrát), jindy ukončení závisí na splnění určité podmínky. (Obcházej náměstí, dokud nepotkáš tu slečnu…) V obou případech programování používá konstrukci cyklu (eventuelně rekurze). S oběma typy cyklů se seznámíme.

Příklad 1.

Vygenerujte 100 náhodných čísel a zobrazte je do textBoxu.

Jedno číslo vygenerovat můžeme, asi bychom mohli příslušný příkaz stokrát nakopírovat, ale jednak by to dalo hodně práce, jednak někdy nemusíme dopředu vědět, kolik kopií (počet čísel zadá uživatel – cvičení)

Napoprvé si necháme poradit od systému:

Napíšeme program pro generování a zobrazení jednoho náhodného čísla:

```
public partial class Cykly : Form
    {
        Random Nahoda = new Random();
     …
private void buttonNahoda100_Click(object sender, EventArgs e)
```

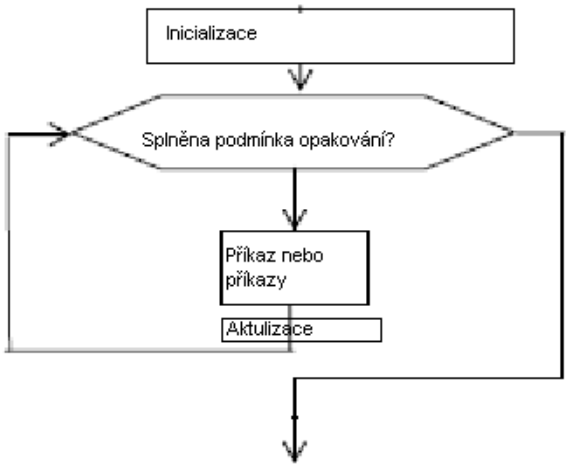
```
        {
                int x = Nahoda.Next(1, 100);
                textBoxCisla.Text += x.ToString() + Environment.NewLine;
            }
```

Oba příkazy vybereme a z místní nabídky (jako když obsluhujeme výjimku) zvolíme for. V prvním řádku se objeví:

```
for (int i = 0; i < length; i++)
```

a naše příkazy se ocitnou ve složených závorkách. Počet opakování udává length…nahradíme ho tedy číslem 100.

```
for (int i = 0; i < 100; i++)
              { …
              }
```



Jak tedy funguje tato konstrukce obecně:

for (inicializace; podmínka opakování; aktualizace)

Před vstupem do cyklu se provede příkaz inicializace – zde se deklarovala proměnná **i** (také řídicí proměnná cyklu) a dosadila se do ní nula.

(int i = 0)

Následně se vyhodonotí podmínka opakování a je-li splněna, provedou se všechny příkazy těla cyklu.

(i < 100)

Po jejich splnění se provede aktualizace – nové nastavení řídicí proměnné.

(i ++ hodnota i se zvětší o jedničku)

Cyklus se ukončí v okamžiku, kdy podmínka opakování přestane být splněna.

Konstrukci příkazu můžeme také zapisovat ručně.

Ještě poznámka: protože i celé číslo x je objektem, není nutné k jeho převedení na řetězec použít třídu Convert – má převodní metodu samo: x.Tostring();

Příklad 2

Uživatel zadá dvě celá čísla – min a max, program vypíše všechna čísla mezi nimi.

```
private void buttonZInt_Click(object sender, EventArgs e)
        {
            int min = Convert.ToInt32(textBoxMin.Text);
            int max = Convert.ToInt32(textBoxMax.Text);
            for (int i = min + 1; i < max; i++)
            {
                textBoxCisla.Text += i.ToString() + Environment.NewLine;
            }
        }
```

Cyklus For se používá hlavně, když je třeba prozkoumat předem známý počet objektů z hlediska nějaké jejich vlastnosti

Příklad 3

Kolik z čísel v příkladu 2 je sudých?

```
private void buttonSud_Click(object sender, EventArgs e)
        {
            int min = Convert.ToInt32(textBoxMin.Text);
            int max = Convert.ToInt32(textBoxMax.Text);
            int pocSud = 0;//počítadlo sudých čísel
            for (int i = min + 1; i < max; i++)
            {
                textBoxCisla.Text += i.ToString() + Environment.NewLine;
                if (i % 2 == 0)
                {
                    pocSud++; //když je zbytek čísla po dělení dvěma nula,
                              //je sudé - zvýšíme počet sudých čísel o jedničku.
                }
```

```
            }
            MessageBox.Show("počet sudých je " + pocSud.ToString());
        }
```

Příklad 4

Které z čísel generovaných v příkladu 1 je největší?

Budeme postupně generovat a vypisovat náhodná čísla a v celočíselné proměnné max si budeme pamatovat to, které je momentálně největší. (Buď do něj na začátku umístíme první číslo, nebo ho inicializujeme tak malou hodnotou, aby se přepsala hned porovnáním s prvním číslem.)

```
private void buttonMax_Click(object sender, EventArgs e)
        {
            int max = -1;
            for (int i = 0; i < 10; i++)
            {
                int x = Nahoda.Next(1, 100);
                if (max < x)
                    max = x;
                textBoxCisla.Text += x.ToString() + Environment.NewLine;

            }
            MessageBox.Show("Největší je " + max.ToString());
        }
```

Důležité

Cyklus s předem známým počtem opakování

for (inicializace; podmínka opakování; aktualizace)

Před vstupem do cyklu se provede příkaz inicializace

Následně se vyhodonotí podmínka opkaování a je-li splněna, provedou se všechny příkazy těla cyklu.

Po jejich splnění se provede aktualizace – nové nastavení řídicí proměnné.

Cyklus se ukončí v okamžiku, kdy podmínka opakování přestane být splněna.

Pracovní list

Cvičení

1.

Upravte příklad 2 tak, že kdyby uživatel zadal min a max obráceně, (větší číslo jako miniimum) hodnoty se vymění.

2.

Upravte příklad 3 (počet sudých čísel z intervalu), že se všechna sudá čísla vypíší.

3.

K určení počtu sudých čísel z intervalu není třeba cyklus. Zkuste vymyslet vhodný vzorec a naprogramovat ho.

4.

Upravte příklad 1 (generování náhodných čísel) tak, aby program vypsal, kolik náhodných čísel padne do intervalu min–max (čísla zadaná z textBoxů, čísla mají být větší než min a menší než max)

5.

Upravte příklad 1 (generování náhodných čísel) tak, aby program vypsal, kolik náhodných čísel je dělitelných 3.

Řešení

1.

```
int pom;
 if (min > max)
 {
     pom = min;
     min = max;
     max = pom;
     textBoxMin.Text = min.ToString();
     textBoxMax.Text = max.ToString();
 }
```

2.

```
            for (int i = min + 1; i < max; i++)
                {
                    if (i % 2 == 0)
                    {
                        textBoxCisla.Text += i.ToString() +Environment.NewLine;
                        pocSud++;//když je zbytek čísla po dělení dvěma nula,
                        //je sudé - zvýšíme počet sudých čísel o jedničku.
                    }
                }
```

3.

```
    if ((min % 2 == 0) && (max % 2 == 0))
                pocSud = (max - min) / 2 - 1;
            else
                pocSud = (max - min) / 2;
```

4.

```
      private void buttonPOcetMM_Click(object sender, EventArgs e)
       {    int min = Convert.ToInt32(textBoxMin.Text);
            int max = Convert.ToInt32(textBoxMax.Text);
            int pocet = 0;
            for (int i = 0; i < 10; i++)
            {
                int x = Nahoda.Next(1, 100);
                textBoxCisla.Text += x.ToString() + Environment.NewLine;
                if ((x <= max) && (x >= min))
                    pocet++;
            }
            MessageBox.Show("Počet čísel z intervalu je " + pocet.ToString());
        }
```

5.

```
int p3 = 0;
        for (int i = 0; i < 10; i++)
        {
            int x = Nahoda.Next(1, 100);
            textBoxCisla.Text += x.ToString() + Environment.NewLine;
            if (x % 3 == 0)
            {
                p3++;
            }
        }
        MessageBox.Show("Počet čísel dělitelných třemi je " + p3.ToString());
```