Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-206
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech II
Téma didaktického materiálu	Jednorozměrné pole
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 16–17 let
Úroveň žáků	Mírně pokročilí
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Strukturovaná data, pole, jednorozměrné pole
Anotace	Studenti se seznamují s datovou strukturou pole, učí se ji procházet, vytvářet a zpracovávat
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. 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. VYSTAVĚL, Radek. *Moderní programování: sbírka úloh k učebnici pro středně pokročilé*. 1. vyd. Ondřejov: moderníProgramování, 2008-2009, 2 sv. ISBN 978-80-903951-3-8. 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.

206. Jednorozměrné pole

Hodnoty údajů se uchovávají v proměnných, ty mohou být členské nebo lokální. Deklarace proměnné určuje její typ – a tím je dáno jakých hodnot může nabývat, kolik paměti potřebuje pro své uložení a jaké operace se s ní dají dělat. Proměnné, se kterými jsme pracovali až doposud, byly jednoduché – uchovávaly vždy jedinou hodnotu. Často si ale potřebujeme pamatovat řadu údajů – databáze osob, výsledky měření apod. K jejich uchovávání se používají strukturované datové typy, mezi které patří také **pole**.

Pole je datová struktura skládající se ze složek stejného typu, které rozlišujeme (a ke kterým přistupujeme) pomocí indexu. (pořadové číslo, vždy se čísluje od 0)

Jako příklad použijme skupinu osob, které si očíslujeme:

Číslo osoby	Jméno osoby
0	Nováková
1	Král
2	Morkes
3	Králová
4	Dvořák
5	Jouda
6	Kůrka
7	Mrázková

Pokud toto pole pojmenujeme osoby, bud platit:

osoby[0] = “Nováková”

osoby[4] = “Dvořák”…index se uvádí vždy v hranatých závorkách, index poslední osoby je počet osob -1.

Při zpracování dat bez pole, bychom museli pracovat s mnoha proměnnými (prakticky také několika tisíci), což by bylo velmi komplikované.

Pokud užijeme pole, můžeme k jednotlivým objektům snadno přistupovat pomocí indexů, zpracovávat celou strukturu v cyklech a celou paměť vyhradit jediným příkazem.

Deklarace pole:

Typ složek [];

Deklarace a vytvoření proměnné tohoto typu (instance)

Typ složek [] Jméno pole = new Typ složek [počet složek]

Pro náš příklad:

```
string[] osoby = new string[8];
```

Počet prvků pole – (zde 8): osoby.Length

Při vytvoření je pole naplněno nulami, chceme-li ho naplnit hodnotami, je možno použít postup:

```
string[] osoby = new string[8] {“Nováková”, “Král”,…”Mrázková”};
```

– tedy hodnoty prvků pole se uvedou do složených závorek za jeho deklaraci.

Zpracování pole v cyklech

```
for (int i = 0; i < osoby.Length; i++)
        {osoby[i] –zpracuj  i.tou položku}
```

Příklad

Nadeklarujeme pole jmen pro osoby v předchozí úloze. Při zobrazení formuláře pole naplníme hodnotami a budeme řešit následující úkoly:

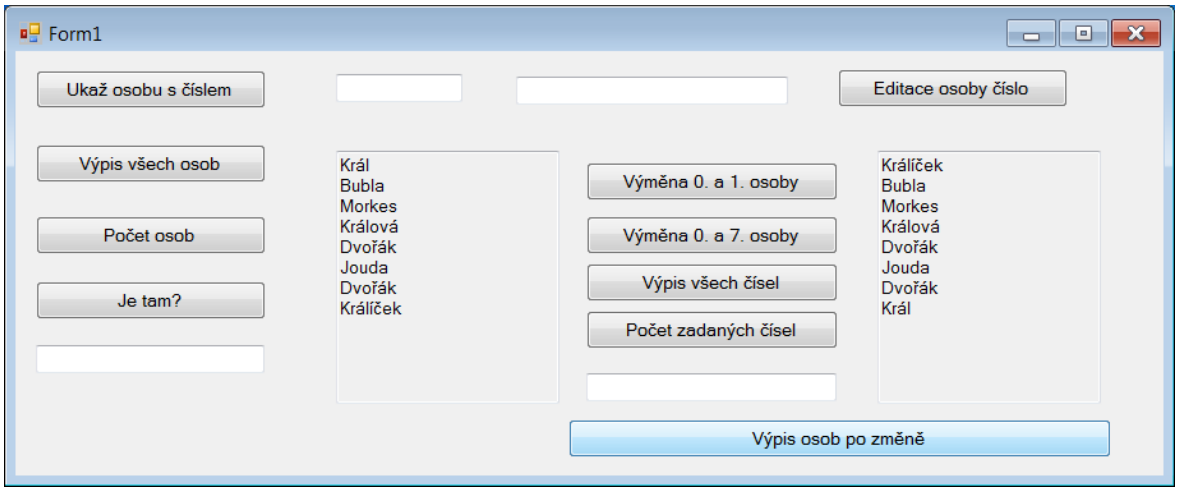
Zobrazení osoby, jejíž číslo zadá uživatel

Výpis všech osob do textového pole

Editaci osoby, jejíž číslo si uživatel vybere

Výměna 1. a 2. osoby v poli

Zjištění, zda se v poli nachází osoba, jejíž jméno zadá uživatel.




```
namespace Pole_jmen
{
    public partial class Form1 : Form
    {
        string[] osoby = new string[8]
            {“Nováková”, “Král”,…”Mrázková”};
```

```
        //deklarace pole 8 řetězců s inicializací- musí být členská proměnná, aby
s ní šlo pracovat v celém programu
        public Form1()
        {
            InitializeComponent();
        }
        private void buttonUkaz1_Click(object sender, EventArgs e)
        {
         //zobrazení osoby, jejíž číslo zadá uživatel
         //pokud je index mimo pole (0-7), program spadne
         int cislo = Convert.ToInt32(textBoxCislo.Text);
            textBoxEdit.Text = osoby[cislo];
        }

        private void buttonVypis_Click(object sender, EventArgs e)
        {
            //ve for cyklu zobrazíme postupně osoby[0] až osoby[7]
            for (int i = 0; i < 8; i++)
            {
                textBoxOsoby.Text += osoby[i]+Environment.NewLine;
            }
        }
        private void buttonEdit_Click(object sender, EventArgs e)
        {
            //osoba s daným číslem se načte do textBoxu Edit,
            //kde můžeme jméno upravit
            int cislo = Convert.ToInt32(textBoxCislo.Text);
            osoby[cislo] = textBoxEdit.Text;
        }
```

```
private void buttonPozmene_Click(object sender, EventArgs e)
{
    //kopíruje výpis, abychom mohli změněné a původní pole porovnat.
    textBoxVypis2.Text = null;
    for (int i = 0; i < 8; i++)
    {
        textBoxVypis2.Text += osoby[i] + Environment.NewLine;
    }
}
private void buttonVymen12_Click(object sender, EventArgs e)
{
    //vymění v poli 1. a 2. osobu
    string pom;
    pom = osoby[0];
    osoby[0] = osoby[1];
    osoby[1]=pom;
}
private void buttonVyhl_Click(object sender, EventArgs e)
{
    //Zjistí, zda se v poli nachází osoba udaného jména
    string hledany = textBoxHledany.Text;
    int i=0;
    while((i<8)&&(osoby[i]!=hledany))
        i++;
    if (i<8)
        MessageBox.Show("Našli");
    else
        MessageBox.Show("Nenašli");
}
```

```
        private void buttonKolik_Click(object sender, EventArgs e)
        {
            int pocetOsob=osoby.Length;
            MessageBox.Show(pocetOsob.ToString());
        }
    }
}
```

Důležité

Pole je datová struktura skládající se ze složek stejného typu, které rozlišujeme (a ke kterým přistupujeme) pomocí indexu. (pořadové číslo)

Deklarace a vytvoření proměnné tohoto typu (instance)

Typ složek [] Jméno pole = new Typ složek [počet složek]

Pole 10 řetězců:.

```
string[] slova = new string[10];
```

Počet prvků pole – (zde 10): slova.Length

Při vytvoření je pole naplněno nulami, chceme-li ho naplnit hodnotami, je možno hodnoty prvků pole uvést do složených závorek za jeho deklaraci.

Zpracování pole v cyklech

```
for (int i = 0; i < slova.Length; i++)
        {slova[i] –zpracuj  i.tou položku}
```

Pracovní list

Cvičení

1. Ošetřete zobrazení osoby, jejíž číslo zadá uživatel, výjimkou, aby jednak muselo být zadáno číslo, jednak aby bylo číslo z intervalu 0–7.

2. Vyměňte první a poslední osobu

3. Nadeklarujte pole dvaceti celých čísel a naplňte ho při vytvoření formuláře tak, aby se tam některá čísla opakovala. Pole zobrazte do textBoxu a zjistěte počet výskytů čísla zadaného uživatelem.

Řešení

1.

```
private void buttonUkaz1_Click(object sender, EventArgs e)
        {
            int cislo=-1;
            try
            {
                cislo = Convert.ToInt32(textBoxCislo.Text);

                if ((cislo < 0) || (cislo > 7))
                    throw new Exception();
               }
            catch
            {
                MessageBox.Show("Zadejte přirozené číslo do nuly do sedmi");
                textBoxCislo.Text = null;
                textBoxCislo.Focus();
                return;
            }
            textBoxEdit.Text = osoby[cislo];
          }
```

2.

```
…
pom = osoby[0];
osoby[0] = osoby[7];
osoby[7] = pom;
```

3.

```
int[] cisla = new int[20];
```

```
…
cisla[0] = 1;
cisla[1] = 10;
…
cisla[9] = 1;
…
//výpis pole čísel
            for (int i = 0; i < 10; i++)
            {
                textBoxVypis2.Text += cisla[i].ToString() + Environment.NewLine;
            }
…
private void buttonPocet_Click(object sender, EventArgs e)
        {
            //počet výskytů čísla zadaného uživatelem
            int x = Convert.ToInt32(textBoxHledane.Text);
            int pocet=0;
            for (int i = 0; i < 10; i++)
            {
                if (cisla[i] == x)
                    pocet++;
            }
            MessageBox.Show(x.ToString()+" je tam "+pocet.ToString());
        }
```