

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-305
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech III
Téma didaktického materiálu	Dvourozměrné pole
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	Dvourozměrné pole, jeho zpracování a zobrazení do komponenty dataGridView
Anotace	Studenti deklarují, plní, zobrazují a zpracovávají dvourozměrné pole.
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 tabuli, synchronní řešení s promítáním)</p> <p>Prezentace obsahuje stručné shrnutí poznatků</p>

	<p>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.</p> <p>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.

305. Dvourozměrné pole

Jako prvky pole mohou vystupovat také pole. Potom hovoříme o poli dimenzionálním, dvourozměrném, případně i vícerozměrném. (Prakticky se moc nepoužívá, více než tři rozměry obvykle komplikují další rozšiřování programu.)

Dvourozměrné pole je zvláště užitečné při zpracování tabulkových dat, matic v matematice a podobně.

Deklarace obecně:

```
TypPrvků[, ] Jméno = new TypPrvků[početřádků, početsloupců];
```

Deklarace pole o dvou řádcích a dvou sloupcích a naplnění hodnotami ukazuje následující příklad.

První index je řádkový, druhý sloupcový, tak jako v matematice, čísluje se od nuly.

```
1 2 3  
7 8 9
```

```
int[,] tabul = new int[2, 3];  
tabul[0, 0] = 1;  
tabul[0, 1] = 2;  
tabul[0,2] = 3;  
tabul[1, 0] = 7;
```

```
tabul[1,1] = 8;
tabul[1, 2] = 9;
```

Chceme-li např. zobrazit všechny čísla v 1. řádku, můžeme použít cyklus for:

```
textBoxVystup.Text += "1. řádek"+Environment.NewLine;
    for (int j = 0; j < 3; j++)
        textBoxVystup.Text += tabul[0, j] + " ";
```

Celou tabulku můžeme zpracovat ve dvou cyklech for nebo v jednom cyklu foreach.

```
foreach (int x in tabul)
    zpracuj x;
```

V tomto případě ovšem jednak nevíme, kde se prvek nachází a také nemůžeme jeho hodnotu v tabulce měnit.

Proto pro tisk raději použijeme vnořený cyklus **for** – ve vnitřním cyklu vždy vytiskneme jeden řádek a odřádkujeme, ve vnějším vystřídáme řádky.

```
textBoxVystup.Text += "Celá tabulka"+Environment.NewLine;
for (int i = 0; i < 2; i++)
{ //tiskne se i-tý řádek
    for (int j = 0; j < 3; j++)
        textBoxVystup.Text += tabul[i, j] + " ";
    textBoxVystup.Text += Environment.NewLine;
}
```

Zpracování sloupců – např. tisk druhého sloupce:

```
textBoxVystup.Text += "Druhý sloupec" + Environment.NewLine;

for (int i = 0; i < 2; i++)
    textBoxVystup.Text += tabul[i, 1] + " ";

}
```

Ke zobrazení tabulky můžeme využít komponentu **dataGridView**. Zatím budeme pracovat s vlastnostmi

dataGridView.RowCount – počet řádků, číslováno od nuly

dataGridView.ColumnCount – počet sloupců, číslováno od nuly

dataGridView[sloupec, řádek].value – hodnota uložená v daném sloupci a řádku.

Abychom se s tímto ovládacím prvkem naučili pracovat, naplníme tuto tabulku hodnotami tak, že hodnota v i-tém řádku a j-tém sloupci bude vždycky $10 \cdot i + j$.

▶	0	1	2	3	4	5	6	7
	10	11	12	13	14	15	16	17
	20	21	22	23	24	25	26	27
	30	31	32	33	34	35	36	37
*	40	41	42	43	44	45	46	47

```

private void button10ij_Click(object sender, EventArgs e)
{
    //Plnění tabulky vypočítanými čísly
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 10; j++)
        {
            int x = 10 * i + j;
            dataGridView1[j, i].Value = x.ToString();
        }
}

```

Všimněte si, že zatímco tabulku typicky procházíme po řádcích, do této komponenty zapisujeme po sloupcích (první index je sloupcový a druhý řádkový, jak jsme zvyklí např. z Excelu). Proto je nutné před vložením čísla do buňky tabulky prohodit indexy.

V dalším příkladu vygenerujeme dvourozměrné pole náhodných čísel, zobrazíme ho do dataGridView, případně do víceřádkového textBoxu a budeme s ním pracovat – vyhledáme maximum 1. řádku a součet čísel ve druhém sloupci. Tentokrát bude **tab** členská (globální) proměnná, protože ji potřebujeme použít ve více metodách.

```

public partial class Form1 : Form
{
    int[,] tab = new int[5, 10];
    Random Nahod = new Random();
...
private void Form1_Load(object sender, EventArgs e)
{
    //plnění pole náhodnými čísly
    for (int i=0;i<5;i++)
        for (int j=0;j<10;j++)
            tab[i,j]=Nahod.Next(10);
    //nastavení vlastností tabulky
    dataGridView1.RowCount = 5;
    dataGridView1.ColumnCount = 10;
}

private void buttonNahoda_Click(object sender, EventArgs e)
{
    // zobrazení pole do textBoxu
    for (int i = 0; i < 5; i++)
        //tisk řádku
        {
            for (int j = 0; j < 10; j++)
                textBoxVystup.Text += tab[i, j] + " ";
            textBoxVystup.Text += Environment.NewLine;
        }
    // zobrazení pole do DataGridView

```

```

for (int i = 0; i < 5; i++)
    //v DataGridView je 1. index sloupcový, proto prohodíme indexy.
    for (int j = 0; j < 10; j++)
        dataGridView1[j, i].Value = tab[i, j];
}

private void buttonMax1_Click(object sender, EventArgs e)
{
    //Maximum 1. řádku
    int Max=-10;
    for (int j = 0; j < 10; j++)
        if (tab[0, j] > Max)
            Max = tab[0, j];
    MessageBox.Show(Max.ToString());
}

private void buttonSum2_Click(object sender, EventArgs e)
{
    //Součet 2. sloupce
    int Sum = 0;
    for (int i = 0; i < 5; i++)
        Sum += tab[i, 1];
    MessageBox.Show(Sum.ToString());
}

```

The screenshot shows a Windows application window titled "Form1". On the left, there is a text box containing a 5x10 grid of numbers:

5	0	6	8	4	0	1	6	2	2	
6	2	0	5	4	9	0	5	8	2	
3	0	8	4	5	9	9	9	6	5	
3	0	4	5	0	3	6	1	5	6	
*	0	6	6	9	0	6	5	1	2	6

In the center, there are several buttons: "DimPole1", "Pole 10*1j", "Součet 2. sloupce", "Maximum tabulky", "Náhodné pole", "Maximum 1.", "Počet sudých v 1. řádku", "Zvětšení posledního sloupce 10*", "Výskyty maxima", and "Výskyty maxima II".

On the right side, there is a list box containing the following values: 1.5, 2.5, 2.6, 2.7, 4.3.

Důležité

Dvourozměrné pole může být pole, jehož prvky jsou řádky tabulky.

Deklarace: `TypPrvků[,] Jméno = new TypPrvků[početřádků, početsloupců];`

dataGridView – ovládací prvek pro zobrazování tabulek

dataGridView.RowCount – počet řádků, číslováno od nuly

dataGridView.ColumnCount – počet sloupců

dataGridView[sloupec, řádek].value – hodnota uložená v daném sloupci a řádku

Pracovní list

Cvičení

1. Určete počet sudých čísel v 1. řádku.
2. Vyhledejte největší číslo tabulky. (Lze řešit i cyklem foreach)
Nalezněte řádkový a sloupcový index posledního výskytu maxima.
(*) Vypište všechny řádkové a sloupcové výskyty maxima do listBoxu.
3. Zvětšete všechna čísla posledního sloupce 10krát.

Řešení

```
private void buttonSud_Click(object sender, EventArgs e)
{
    //Počet sudých v 2. řádku
    int PocSud=0;
    for (int j = 0; j < 10; j++)
        if (tab[0, j] % 2 == 0)
            PocSud++;
    MessageBox.Show(PocSud.ToString());
}

private void buttonMaxTab_Click(object sender, EventArgs e)
{
    //K určení největšího prvku je třeba tabulku projít celou
    int Max = -10;
    int imax=-1;int jmax=1;
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 10; j++)
        {
            if (tab[i, j] > Max) Max = tab[i, j];
            imax = i; jmax = j;
        }
    //Pamatujeme si maximum a jeho indexy
    MessageBox.Show("Maximum: "+Max.ToString()+Environment.NewLine+
        "Řádek: "+imax.ToString()+Environment.NewLine+
        "Sloupec: "+jmax.ToString());
}
```

```

}
private void buttonVyskyty_Click(object sender, EventArgs e)
{
    //Maximum a všechny jeho výskyty
    //1. varianta: Maximum nejprve najdeme a pak znovu projdeme tabulku a
    // jeho výskyty uložíme do listBoxu.
    int Max = -10;
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 10; j++)
        {
            if (tab[i, j] > Max) Max = tab[i, j];
        }
    MessageBox.Show("Maximum: " + Max.ToString());
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 10; j++)
        {
            if (tab[i, j] == Max)
            {
                listBox1.Items.Add(i.ToString() + ", " + j.ToString());
            }
        }
}

private void buttonVyskyt2_Click(object sender, EventArgs e)
{
    //Maximum a všechny jeho výskyty
    //2. varianta: Úlohu vyřešíme na jeden průchod tabulkou:
    //nalezneme-li nové maximum, seznam vyprázdníme a výskyt do něj ho přidáme,
    //nalezneme-li staré maximum, přidáme jeho výskyt do existujícího seznamu.
    int Max = -10;
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 10; j++)
        {
            if (tab[i, j] > Max)
            {
                //nové maximum
                Max = tab[i, j];
                listBox1.Items.Clear();
                listBox1.Items.Add(i.ToString() + ", " + j.ToString());
            }
            else
            {
                if (tab[i, j] == Max)
                {

```

```
        listBox1.Items.Add(i.ToString() + ", " + j.ToString());
    }
}
MessageBox.Show("Maximum: " + Max.ToString());
}
private void buttonPosl10_Click(object sender, EventArgs e)
{
    //Vynásobení čísel v posledním sloupci 10
    for (int i = 0; i < 5; i++)
        tab[i,9]*=10;
}
}
```