

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-306
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 II
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>V pracovním listu je zadání cvičení – většinou se</p>

	<p>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.

## Dvourozměrné pole – procvičování

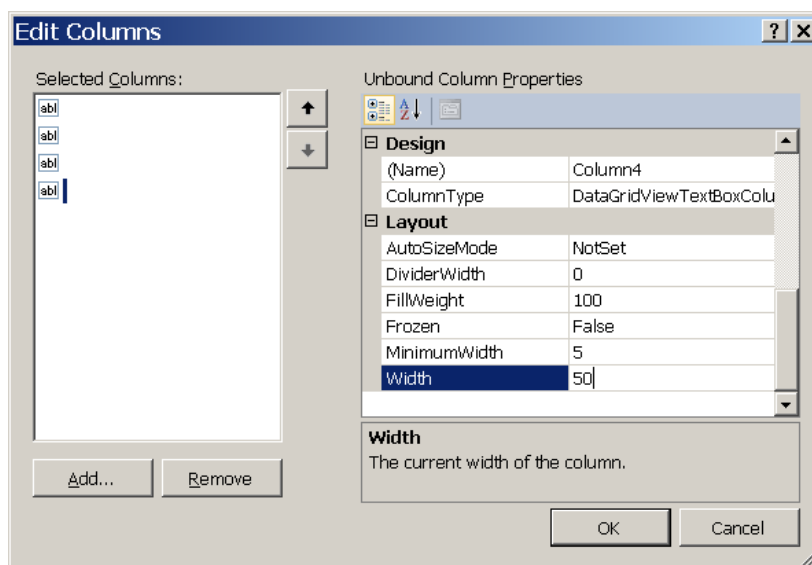
Připravíme si projekt pro zajímavější příklady na zpracování číselných tabulek. Budeme pracovat s tabulkou o 5 řádcích a 5 sloupcích naplněnou z generátoru při vytvoření formuláře. Na formulář zkopírujeme dvakrát **dataGridView**, abychom mohli současně zobrazovat výchozí i upravenou tabulku. Připravíme si zobrazovací metodu, která bude mít jako parametr jednak tabulku, jednak **dataGridView**.

Aby nebyly sloupce zbytečně široké, nastavíme jejich šířku editorem sloupců – vlastnost Columns. Nejprve 5 sloupců přidáme, pak nastavíme jejich šířku na 50.

```
namespace DimPOle_II
{
    public partial class
    Form1 : Form
    {
        Random nahoda = new
        Random();

        int[,] a = new int[5,
        5];
```

```
private void UkazPole(int [,]b,DataGridView d)
```



```

{
    //zobrazení matice b do DataGridView d
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
            d[i, j].Value = b[j, i].ToString();
}

private void vymenCisla(ref int a, ref int b)
{
    //výměna dvou celých čísel
    int pom = a;
    a = b;
    b = pom;
}

public Form1()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    //plnění výchozího pole a jeho zobrazení
    dataGridView1.RowCount = 5;
    dataGridView2.RowCount = 5;
    dataGridView1.ColumnCount = 5;
    dataGridView2.ColumnCount = 5;
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
            a[i, j] = nahoda.Next(20);
    UkazPole(a, dataGridView1);
}

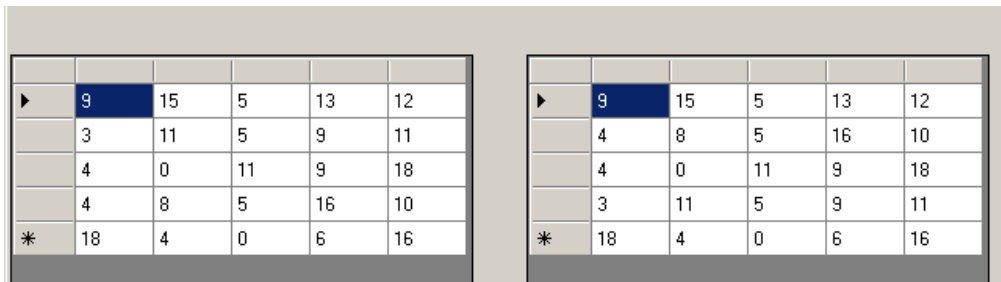
```

### Příklad 1.

Vyměňte 2. a 4. řádek tabulky.

Oba řádky budeme procházet současně a vyměňovat prvky se stejným sloupcovým indexem. Využijeme i dále metodu na výměnu dvou celých čísel `vymenCisla`.

```
private void buttonVymen13_Click(object sender, EventArgs e)
{
    //vyměň 1. a 3. řádek
    for (int j = 0; j < 5; j++)
        vymenCisla(ref a[1, j], ref a[3, j]);
    UkazPole(a, dataGridView2);
}
```



▶	9	15	5	13	12
	3	11	5	9	11
	4	0	11	9	18
	4	8	5	16	10
*	18	4	0	6	16

▶	9	15	5	13	12
	4	8	5	16	10
	4	0	11	9	18
	3	11	5	9	11
*	18	4	0	6	16

## Příklad 2.

Setřídíte hlavní diagonálu tabulky.

Hlavní diagonálu můžeme procházet jediným cyklem, protože indexy prvků na ní jsou si rovny.

```
private void buttonSetrDiag_Click(object sender, EventArgs e)
{
    //setřídění hlavní diagonály
    bool menil;
    do
    {
        menil = false;
        for (int i = 0; i < 4; i++)
        {
            if (a[i, i] > a[i + 1, i + 1])
            {
                vymenCisla(ref a[i, i], ref a[i + 1, i + 1]);
            }
        }
    }
}
```

```

        menil = true;
    }
}
}
while (menil);
UkazPole(a, dataGridView2);
}

```

## Cvičení

1. Vyměňte třetí a pátý sloupec.
2. (\*)Vyhledejte minimum vedlejší diagonály.(Stačí opět jeden cyklus for, je třeba si uvědomit, jak se mění indexy prvků na vedlejší diagonále.
3. Vypište všechny prvky na hlavní diagonále.
4. Vypočítejte součet čísel v jednotlivých řádcích.

## Řešení

1.

```

private void buttonVymen24_Click(object sender, EventArgs e)
{
    //vyměň 3. a 5. sloupec
    for (int j = 0; j < 5; j++)
    {
        vymenCisla(ref a[j, 2], ref a[j, 4]);
    }

    UkazPole(a, dataGridView2);
}

```

2.

```

private void button1_Click(object sender, EventArgs e)
{
    //Minimum vedlejší diagonály
    int min = 1000;
    for (int i = 0; i < 5; i++)

```

```

    {
        if (a[i, 4 - i] < min)
            min = a[i, 4 - i];
    }

    MessageBox.Show(min.ToString());
}

```

3. `private void buttonVypisHD_Click(object sender, EventArgs e)`

```

{
    //výpis čísel na hlavní diagonále
    string hd = null;
    for (int i = 0; i < 5; i++)
    {
        hd += a[i, i].ToString() + " ";
    }
    MessageBox.Show(hd);
}

```

4. `private void buttonSoucRad_Click(object sender, EventArgs e)`

```

{
    //Součet čísel v řádcích
    string vys=null;
    int[] soucet = new int[5];
    for (int i = 0; i < 5; i++)
    {
        soucet[i] = 0;
        for (int j = 0; j < 5; j++)
            soucet[i] += a[i, j];
    }
    for (int i = 0; i < 5; i++)
        vys += soucet[i]+" ";
    MessageBox.Show(vys);
}

```

