

Metodický list k didaktickému materiálu

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-308
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech III
Téma didaktického materiálu	Zpracování tabulkových dat
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	Logický typ, výrok, operace s výroky
Anotace	Studenti programují tabulky operací s výroky – tabulky logických hodnot a jejich aplikace, přitom ukládají data ve formě csv souborů
Použité zdroje	BUŠEK, Ivan. <i>Řešené maturitní úlohy z matematiky</i> . 2. vyd. Praha: SPN, 1985. ISBN 14-290-88. 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.
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) 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...).

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.

308. Programování operací s výroky

Příklad 1.

Chtěli bychom naprogramovat tabulku, která definuje logické spojky a ověřuje, že implikace je ekvivalentní své obměně. (Vzpomínáte na matematiku z 1. ročníku?)

Zopakujme si nejprve potřebné logické funkce a jejich možnou implementaci v C#.

Výrok je proměnná typu bool, má hodnotu false nebo true.

Operátor negace(not): !

Konjunkce (and): &&

Disjunkce (or): ||

Ekvivalence: Tady můžeme použít skutečnost, že dva výroky jsou ekvivalentní, právě když mají stejnou pravdivostní hodnotu a jako operátor ekvivalence použít rovnost: ==

Implikace: Zatímco v Excelu nebo např. Delphi jsou logické hodnoty uspořádané a tedy implikaci můžeme nahradit operátorem \leq (Promyslete si proč), v C# si musíme poradit jinak. Jedna možnost je nahradit implikaci pomocí disjunkce:

$\text{not } (A \Rightarrow B) = (A \text{ and } \text{not } B)$

$\text{not not } (A \Rightarrow B) = (A \Rightarrow B) = (\text{not } A \text{ or } B)$

Jinak také víme, že implikace je pravdivá, když jsou oba výroky pravdivé nebo když předpoklad neplatí a tedy: $(A = \text{false}) \text{ or } ((A = \text{true}) \text{ and } (B = \text{true}))$

	A	B	not A	not B	A and B	A or B	A=>B	not B => not A	A<=>B
	False	False	True	True	False	False	True	True	True
	False	True	True	False	False	True	True	True	False
	True	False	False	True	False	True	False	False	False
	True	True	False	False	True	True	True	True	True

Jakým způsobem vyplňujeme tabulku? Výrok A nabyde hodnoty false, výrok B rovněž, doplníme řádek a zapíšeme do tabulky.

Řádek tabulky zde bude tvořen polem řetězců:

```
string[] radek = new string[9];
```

```
A = false; B = false;
```

```
radek[0] = A.ToString();
```

```
radek[1]=B.ToString();
```

...

```
radek[8] = (A == B).ToString();
```

```
dataGridView.Rows.Add(radek);
```

Čtyři řádky tabulky za sebou bychom tímto způsobem sice zvládli vyplnit, ale kdyby bylo vstupních dat víc (viz aplikační příklad), bylo by řešení poněkud zdlouhavé.

Protože víme, že chceme vyplňovat řádky tak, že výrok A nejprve nabyde hodnota false, výrok B k němu false a true, pak výrok A true atd., nabízí se mechanismus vnořeného cyklu, kdy řídící proměnné budou postupně nabývat těchto hodnot. Můžeme využít např. cyklus foreach.

Nejprve si naplníme pomocné pole logickými hodnotami:

```
bool [] hodnoty=new bool[2] {false,true};
```

A potom aplikujeme vnořený cyklus:

```
foreach (bool A in hodnoty)
```

```
    foreach (bool B in hodnoty)
```

```
        {naplň řádek}
```

Tedy dohromady:

```
bool [] hodnoty=new bool[2] {false,true};
```

```
string[] radek = new string[9];
```

```
foreach (bool A in hodnoty)
```

```
    foreach (bool B in hodnoty)
```

```
    {
```

```
        radek[0] = A.ToString();
```

```
        radek[1]=B.ToString();
```

```
        ...
```

```
        radek[8] = (A == B).ToString();
```

```
        dataGridView.Rows.Add(radek);
```

```
    }
```

```
}
```

Pracovní list

Cvičení

1. Naprogramujte doplnění dalších sloupečků tabulek.
2. Naprogramujte uložení tabulky logických hodnot do csv souboru.
3. (*)Naprogramujte řešení následující aplikační úlohy:
Pro provozní dobu tří benzínových stanic A, B, C v určitém městě platí tyto podmínky:
Vždy je v provozu stanice A nebo C.
Stanice C je mimo provoz, právě když je otevřeno ve stanici A.
Jakmile je otevřená stanice C (právě když), pak stanice A není v provozu a je v činnosti stanice B.
Určete všechny možnosti provozu těchto tří benzínových stanic.

Řešení

```
private void buttonNapln_Click(object sender, EventArgs e)
{ //Vyplnění tabulky

    bool [] hodnoty=new bool[2] {false,true};
    string[] radek = new string[9];
    foreach (bool A in hodnoty)
        foreach (bool B in hodnoty)
        {
            radek[0] = A.ToString();
            radek[1]=B.ToString();
            radek[2] = (!A).ToString();
            radek[3] = (!B).ToString();
            radek[4] = (A && B).ToString();
            radek[5] = (A || B).ToString();
            radek[6] = (!A ||B).ToString();
            radek[7] = ((!A==true)&&!B==true) || (!B==false)).ToString();
            radek[8] = (A == B).ToString();
            dataGridView.Rows.Add(radek);
        }
    }
}
```

```
2. private void buttonUloz_Click(object sender, EventArgs e)
{
    //Uložení tabulky do souboru
    if (saveFileDialog.ShowDialog() != DialogResult.OK)
        //Zjištění jména souboru
```

```

        return;
    string Vystup = saveFileDialog.FileName;
    StreamWriter data = new StreamWriter(Vystup, false, Encoding.Default);
    //hlavička
    string hlavicka = "A;B;not A;not B;A and B;A or B;A=>B;not B=>not A;A<=>B";
    data.WriteLine(hlavicka);
    //zapisujeme tabulku po řádcích
    int pocetR = dataGridView.Rows.Count;//počet řádků
    int pocetS = dataGridView.Columns.Count;//počet sloupců
    for (int i = 0; i < pocetR - 1; i++)
    {
        string radek = null;
        for (int j = 0; j < pocetS; j++)
            //poskládání sloupečků a doplnění středníků
            {
                radek += dataGridView[j, i].Value.ToString();
                if (j < pocetS - 1)
                    radek += ";";
            }
        data.WriteLine(radek);
    }
    data.Close();
}

```

3. Můžete si nechat vytisknout celou tabulku (podobně jako v první úloze – zde vytisknuté řešení je z Excelu, ani tady trocha opakování neuškodí) nebo jen možnosti, které vyhovují podmínkám úlohy (viz program zde)

	A	B	C	D	E	F	G	H	I
1	A	B	C	not A	not C	not A and B	A or C	not C <=> A	C <=> (not A and B)
2	NEPRAVDA	NEPRAVDA	NEPRAVDA	PRAVDA	PRAVDA	NEPRAVDA	NEPRAVDA	NEPRAVDA	PRAVDA
3	NEPRAVDA	NEPRAVDA	PRAVDA	PRAVDA	NEPRAVDA	NEPRAVDA	PRAVDA	PRAVDA	NEPRAVDA
4	NEPRAVDA	PRAVDA	NEPRAVDA	PRAVDA	PRAVDA	PRAVDA	NEPRAVDA	NEPRAVDA	NEPRAVDA
5	NEPRAVDA	PRAVDA	PRAVDA	PRAVDA	NEPRAVDA	PRAVDA	PRAVDA	PRAVDA	PRAVDA
6	PRAVDA	NEPRAVDA	NEPRAVDA	NEPRAVDA	PRAVDA	NEPRAVDA	PRAVDA	PRAVDA	PRAVDA
7	PRAVDA	NEPRAVDA	PRAVDA	NEPRAVDA	NEPRAVDA	NEPRAVDA	PRAVDA	NEPRAVDA	NEPRAVDA
8	PRAVDA	PRAVDA	NEPRAVDA	NEPRAVDA	PRAVDA	NEPRAVDA	PRAVDA	PRAVDA	PRAVDA
9	PRAVDA	PRAVDA	PRAVDA	NEPRAVDA	NEPRAVDA	NEPRAVDA	PRAVDA	NEPRAVDA	NEPRAVDA
10									

```

private void buttonAplikace_Click(object sender, EventArgs e)
{
    //úloha o benzínových pumpách
    bool [] hodnoty=new bool[2] {false,true};
    string vysledek;
    foreach (bool A in hodnoty)

```

```
foreach (bool B in hodnoty)
    foreach (bool C in hodnoty)
        if ((A || C) && (!C == A) && (C == (!A && B)))
        {
            vysledek = "A: " + A.ToString() + " B: " + B.ToString() +
                " C: " + C.ToString();
            textBoxVysledek.Text += vysledek + Environment.NewLine;
        }
}
```

Řešení aplikační úlohy

A: False B: True C: True
A: True B: False C: False
A: True B: True C: False