Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-117
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech I
Téma didaktického materiálu	Obrázky v grafice, události myši
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	Třída Image, událost MouseDown
Anotace	Studenti se učí pracovat s obrázky, používat vestavěná data a programovat obsluhu událostí, které souvisejí s pozicí myši
Použité zdroje	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

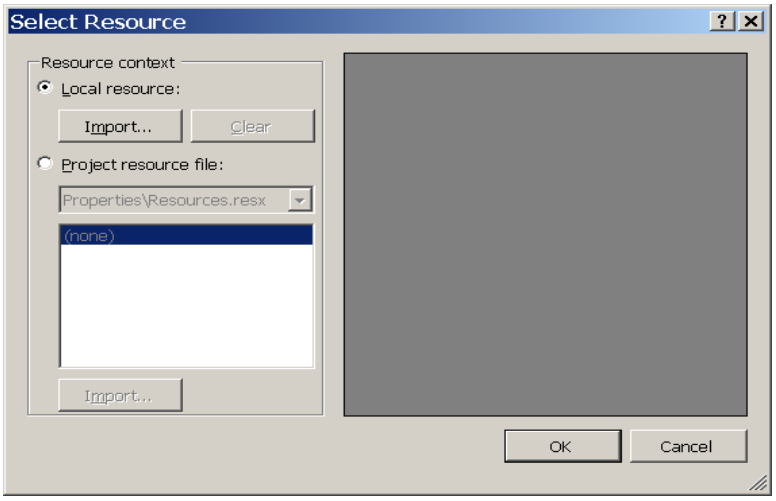
Prohlášení autora

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.

117. Grafika III, události myši

Ve svých projektech můžete pracovat s obrázky buď tak, že si je (předem připravené) zkopírujete do složky Bin/Debug, kde se dají dále upravovat, ale musí být pořád součástí projektu nebo použít tzv. **vestavěné obrázky,** které jsou přímo součástí spustitelného souboru. Ukážeme si oba způsoby.



Příklad 1

Umístěte na pozadí formuláře obrázek.

Postup:

1. Uložíme projekt do vhodně nazvané složky.

2. Obrázek si připravíme v grafickém editoru (podporovány jsou všechny grafické formáty) a okopírujeme ho do podsložky projektu Bin/Debug

3. Ve vlastnostech formuláře zvolíme BackgroundImage a po stisknutí tlačítka se třemi tečkami vybereme v následujícím okně Local resource“ Import a dále zvolíme obrázek.

4. Vlastnost formuláře BackgroundImageLayout nám umožní, jak má být obrázek v okně umístěn.

Příklad 2

Umístěte na formulář komponentu Image s vestavěným obrázkem kuličky a čtyři tlačítka pro volbu směru. Na klepnutí na tlačítko se kulička posune o 10 pixelů příslušným směrem.

Soubor s obrázkem připravíme jako v předchozí úloze. Soubor v programu načteme do proměnné typu **Image** a v obsluze **Paint** formuláře budeme obrázek vykreslovat metodou **DrawImage**.

Aktuální pozici kuličky si bude program pamatovat v členských proměnných x, y, které se budou měnit stisknutím pohybových tlačítek.

Obrázek ze souboru načte metoda třídy **Image.FromFile(jméno souboru)**

```
namespace Obrázky
{
    public partial class Obrazky : Form
    {
        int x =250;
        int y = 100;//vychozí pozice kuličky
        int s = 32;
        int v = 32;//rozměry kuličky, nemění se
        Image obr = Image.FromFile("Kulicka.png");//načtení obrázku při startu
programu
        public Obrazky()
        {
            InitializeComponent();
        }
        private void Obrazky_Paint(object sender, PaintEventArgs e)
        {
            Graphics kp=e.Graphics;
            kp.DrawImage(obr,x,y,s,v);
        }
        private void buttonNahoru_Click(object sender, EventArgs e)
        {
            y -= 10;//y roste směrem dolů
```

```
            Refresh();
        }

        private void buttonDolu_Click(object sender, EventArgs e)
        {
            y += 10;
            Refresh();
        }
        private void buttonDoleva_Click(object sender, EventArgs e)
        {
            x -= 10;
            Refresh();
        }
        private void buttonDoprava_Click(object sender, EventArgs e)
        {
            x += 10;//y roste směrem dolů
            Refresh();
      }
    }
}
```

Použití vestavěného obrázku

Z nabídky **Project** vybereme **Properties** (Jméno projektu Properties). Na další obrazovce nalevo zvolíme **Resources** a dále z **Add Resource** možnost **Add Existing File,** vybreme obrázek.

Přitom se ve složce projektu vytvořila podsložka **Resources,** do které se obrázek zkopíroval. Navíc se začlenil do projektu jako **vestavěná (binární) data**.

Použití zdroje v programu pak vypadá následovně:

```
Image obr2 = Properties.Resources.Kulicka;
```

Události myši

Kromě události klik má většina komponent také události související s polohou a tlačítkem myši – MouseHover, MouseDown, MouseMove, MouseUp apod.

```
private void Form1_MouseDown(object sender, MouseEventArgs e)
```

Objekt MouseEventArgs e má mj. vlastnost e.X a e.Y, souřadnice na formuláři, kde bylo klepnuto myší.

Příklad 3

V místě, kde klepneme myší, se nakreslí kružnice s náhodným poloměrem.

```
int x,y,r;//souřadnice místa kliknutí a poloměr kružnice
Random nahod = new Random();
public Form1()
{
    InitializeComponent();
}
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    x = e.X;
    y = e.Y;
    r = nahod.Next(10, 30);
    Refresh();
}
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics p = e.Graphics;
    p.DrawEllipse(Pens.DarkGoldenrod, x - r, y - r, 2 * r, 2 * r);
}
```

Důležité

Obrázky jsou objekty třídy **Image**. Ze souboru se dají načíst statickou metodou **ImageFromFile,** která má jako parametr v závorce jméno souboru s obrázkem.

Obrázek se vykreslí metodou **DrawImage** objektu kreslicí plochy.

Pokud chceme, aby byl obrázek součástí .exe souboru jako vestavěná data, vložíme ho do projektu přes **Project – Properties – Resources – Add Resorce – Add Existing File.** Z programu pak obrázek získáme jako vlastnost třídy **Properties.Resources.**

Pracovní list

Cvičení

1.

S využitím události MouseDown formuláře přidáme do víceřádkového textového pole souřadnice místa, kde bylo klepnuto myší

2.

Při každém stisknutí tlačítka se zvětší obrázek, použijte vestavěný obrázek. Když by přerostl ven z formuláře, splaskne na původní velikost.

Řešení

```
namespace Myši

{

    public partial class Form1 : Form

    {

        Image obr = Properties.Resources.Ksicht;

        int a = 50;//rozměr obrázku

        public Form1()

        {

            InitializeComponent();

        }


        private void Form1_MouseDown(object sender, MouseEventArgs e)

        {

          //zápis souřadnic kliknutí

            int x = e.X;

            int y = e.Y;

            textBox1.Text += "x: " + Convert.ToInt32(x) + " y: " +
            Convert.ToInt32(y) + Environment.NewLine;

        }
```

```
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
       //vykreslení obrázku se změněnými rozměry
        Graphics p = e.Graphics;
        p.DrawImage(obr, 300 - a, 150 - a, 2 * a, 2 * a);
    }

    private void buttonRostu_Click(object sender, EventArgs e)
    {

      //zvětšení rozměru obrázku
        a += 10;
        if (a > 150)
            a = 50;
        Refresh();
    }
}
```