

## 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-112
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech I
Téma didaktického materiálu	Grafika I
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	Kreslicí plocha (canvas), metody kreslicí plochy, pera, událost Paint
Anotace	Studenti se seznamují s objektem kreslicí plochy, kreslením a vyplňováním jednoduchých obrazců, procvičují si práci se souřadným systémem okna
Použité zdroje	VYSTAVĚL, Radek. <i>Moderní programování: sbírka úloh k učebnici pro začátečníky</i> . 2. vyd. Ondřejov: moderníProgramování, 2008, 2 sv. ISBN 978-80-903951-5-2.  VYSTAVĚL, Radek. <i>Moderní programování: učebnice pro začátečníky</i> . 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 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...).

## 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.

## 112. Grafika I

### Kreslicí plocha, událost Paint

Připravíme si nový projekt, text okna nastavíme na Kreslení a v editoru vlastností a událostí vybereme událost **Paint** formuláře. Tuto událost programujeme, chceme-li změnit vzhled okna jinak než vkládáním komponent. Nastává např. při prvním vykreslení okna, obnovení okna z minimalizace apod. Do obsluhy události Paint vkládáme příkazy pro kreslení grafických obrazců.

Okno se vykresluje často – než se poprvé zobrazí, po minimalizaci a obnovení... Každý požadavek na vykreslení vyvolává událost Paint.

Všechny kreslicí metody náleží objektu **kreslicí plochy (plátno, canvas)**. Budeme používat plochu okna, panelu, ale také tiskárny nebo rastrového obrázku v paměti. Datový typ kreslicích ploch je **Graphics**.

V události Paint – např. formuláře, je třeba deklarovat a inicializovat objekt kreslicí plochy:

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics Plocha= e.Graphics;
    ...
}
```

Potom budeme mít k dispozici její metody a vlastnosti.

### Pera, souřadný systém okna

**Pero** charakterizuje komplexně čáru – barvu, tloušťku a styl. V nejjednodušším případě můžeme používat připravenou sadu **Pens** – našeptávač nabídne barvu. Získáme tak tenkou plnou čáru požadované barvy.

**Souřadný systém okna** – je stejný jako třeba v Malování nebo v Corelu (trochu nám to zkomplikuje život, až budeme kreslit graf funkce) – levý horní roh má souřadnice [0,0], x roste doprava a y směrem dolů.

Pokud si programátor nezvolí jinak, jednotky jsou vždy **pixely**.

### Některé metody kreslicí ploch a jejich parametry:

<b>DrawLine</b>	kreslení úsečky	Pero, souřadnice koncových bodů
<b>DrawRectangle</b>	kreslení obdélníka	Pero, souřadnice levého horního rohu, šířka, výška

## DrawEllipse

kreslení elipsy

Pero, parametry opsaného obdélníka

### Příklad 1:

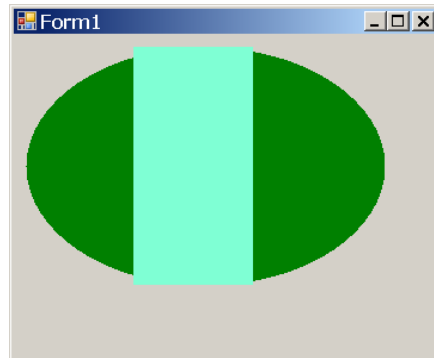
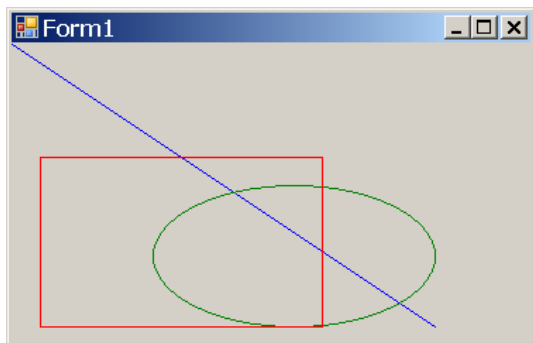
```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics Plocha;

    Plocha = e.Graphics;

    Plocha.DrawLine(Pens.Blue, 0, 0, 300, 200);

    Plocha.DrawEllipse(Pens.Green, 100, 100, 200, 100);

    Plocha.DrawRectangle(Pens.Red, 20, 80, 200, 120);
}
```



## Vyplňování obrazců

Metody jsou velmi podobné, jen místo **Draw** se píše **Fill** a místo per se užívají **štětce**.

### Příklad 2

```
Plocha.FillEllipse(Brushes.Green, 10, 10, 300, 200);
Plocha.FillRectangle(Brushes.Aquamarine, 100, 10, 100, 200);
```

## Důležité

Pro kreslení používáme událost **Paint** komponenty.

Získání kreslicí plochy: **Graphics Plocha = e.Graphics ;**

Metody:

xLH, yLH – x-ová souřadnice levého horního rohu, yLH –y-ová

**DrawLine(pero,x1,y1, x2,y2)**

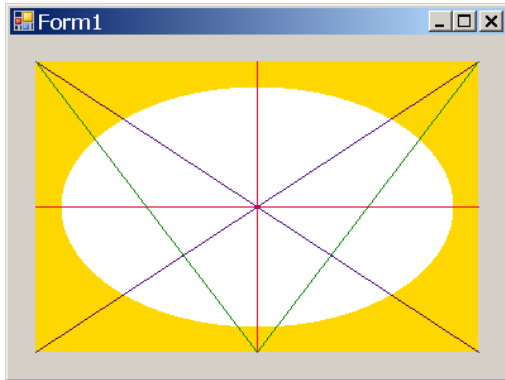
**DrawEllipse (pero, xLH,yLH,sirka, delka)**

**DrawRectangle (pero, xLH,yLH,sirka, delka)**

**FillEllipse, FillRectangle**

## Pracovní list

### Cvičení



1. Naprogramujte vykreslení obrázku vlevo. (Barvy si zvolte svoje)

2. Naprogramujte kreslení vlastního obrázku z čar, obdélníků a elips.

### Řešení

Je vhodné si nejprve nastavit rozměry formuláře: (Vlastnost **size** – např. šířku 400, výšku 300)

1.

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics p = e.Graphics;

    p.FillRectangle(Brushes.Gold, 20, 20, 350, 230);
    p.FillEllipse(Brushes.White, 40, 40, 310, 190);
    p.DrawLine(Pens.Indigo, 20, 20, 370, 250);
    p.DrawLine(Pens.Indigo, 370, 20, 20, 250);
    p.DrawLine(Pens.Crimson, 195, 20, 195, 250);
    p.DrawLine(Pens.Crimson, 20, 135, 370, 135);
    p.DrawLine(Pens.Crimson, 20, 20, 195, 250);
    p.DrawLine(Pens.Green, 20, 20, 195, 250);
    p.DrawLine(Pens.Green, 195, 250, 370, 20);
}
```

2. Individuální