

# Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázkou knihy se vztahují autorská práva, tzv. copyright.

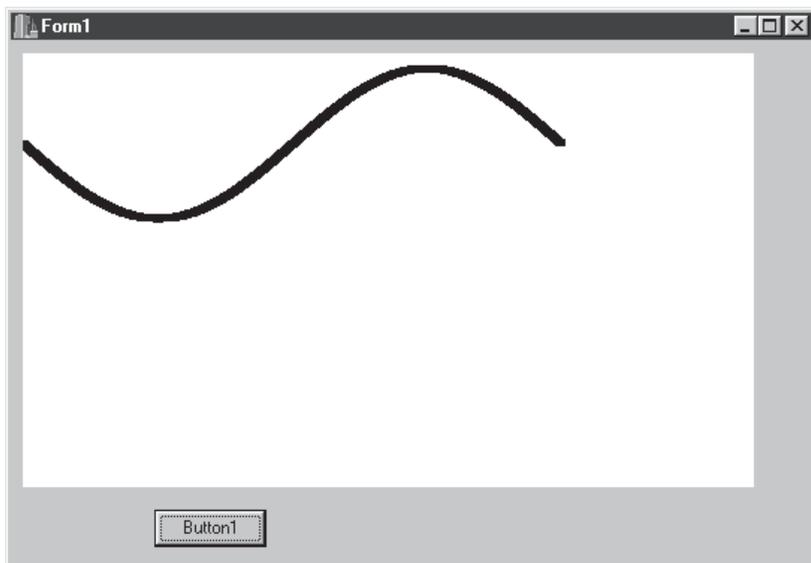
To znamená, že ukáзка má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázkou jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umístováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

*redakce nakladatelství BEN – technická literatura*  
[redakce@ben.cz](mailto:redakce@ben.cz)



## Pero



Obr. 40 G03.BPR

### Co to dělá:

Nakreslí graf funkce sinus tlustou čarou.

### Jak:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int fi,x,y;

    Image1->Canvas->Pen->Width = 5;
    Image1->Canvas->MoveTo( 0,60 );

    for (fi=0;fi<=360;fi++)
    {
        x = fi;
        y = 60+50*sin( fi*2*M_PI/360 );
        Image1->Canvas->LineTo( x,y );
    }
}
```

### Vysvětlení:

Canvas má vlastnost **Pen** – pero. Pero má vlastnost **Width** – tloušťka čáry.

# Ještě Pero



Obr. 41 G04.BPR

## Co to dělá:

Kreslí a mění tloušťku čáry.

## Jak:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int fi,x,y;

    Image1->Canvas->MoveTo( 0,60 );

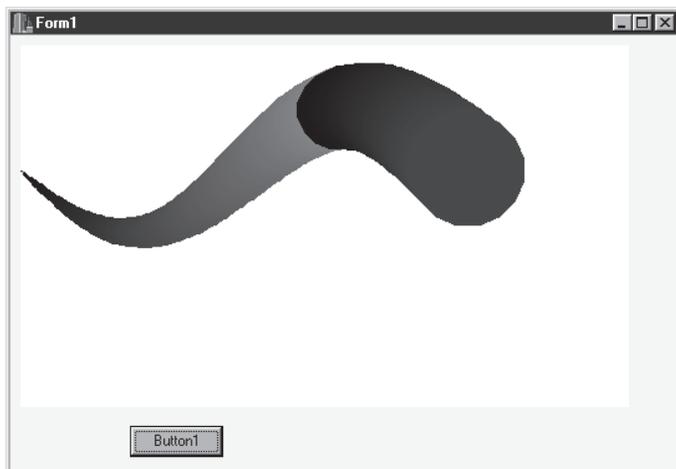
    for (fi=0;fi<=360;fi++)
    {
        x = fi;
        y = 100+50*sin( fi*2*M_PI/360 );

        Image1->Canvas->Pen->Width = 1 + fi/4;
        Image1->Canvas->LineTo( x,y );
    }
}
```

## Vysvětlení:

- Snad ani není třeba. Měníme tloušťku pera v cyklu.
- A přece – ta svislá čárka na levém kraji obrázku – tento graf je (kvůli tloušťce čáry) posunut více dolů („100+...“ místo „60+“). Počáteční **MoveTo()** však nastavuje pero na pozici [0, 60].

## Poprvé barva



Obr. 42 G05.BPR

### Co to dělá:

Kreslí a mění barvu čáry.

### Jak:

Vlastnost **Form1->Color** jsme inspektorem objektů nastavili na **clYellow** (po dvojitém kliknutí na hodnotu lze vybrat).

```
...  
    Image1->Canvas->Pen->Color = fi;  
    Image1->Canvas->Pen->Width = 1 + fi/4;  
    Image1->Canvas->LineTo( x, y );  
...
```

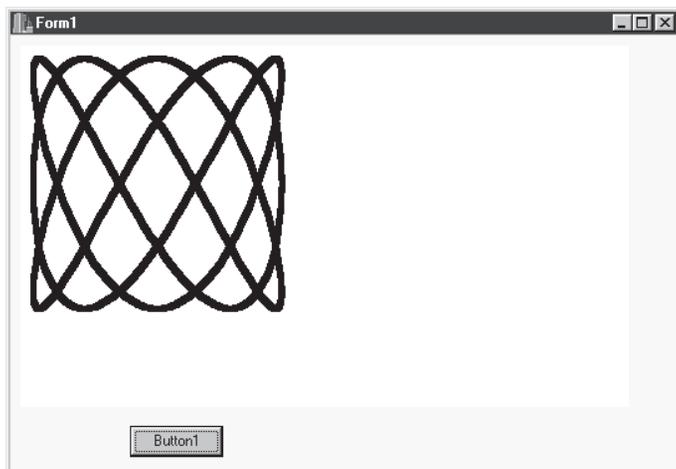
### Vysvětlení:

- Pero má vlastnost **Color** – barva.
- Barva (nejen barva Pera) je typu **TColor** – 4bytové dlouhé číslo. Nejnižší tři bajty udávají podíl červené, zelené a modré barvy jasnost každé je v rozmezí 0 (žádná) až 255 (maximální). Vyjadřujeme-li barvu v šestnáctkové soustavě, odpovídají každé složce (každému bajtu) dvě číslice. Tedy např. hodnota 0xFF00FF reprezentuje součet syté modré (FF = 255), syté červené a žádné zelené – tedy fialovou barvu.

### Poznámka

*U nastavení používajících více než 24bitové barvy (True Color) je barva reprezentována jinak.*

## Lissajousovy obrazce



Obr. 43 G06.BPR

### Co to dělá:

Kreslí Lissajousův obrazec.

### Jak:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int fi,x,y;

    Image1->Canvas->Pen->Width = 5;

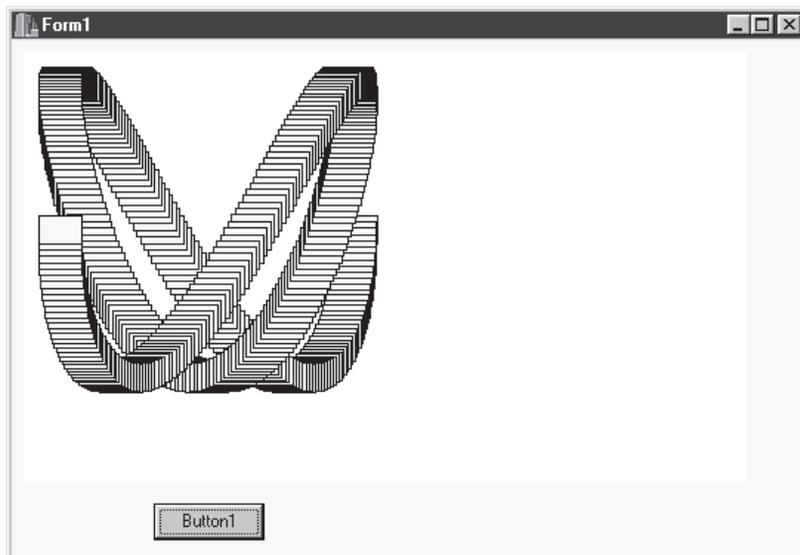
    for (fi=0;fi<=360;fi++)
    {
        x = 110+100*cos( 3* fi*2*M_PI/360 );
        y = 110+100*sin( 5* fi*2*M_PI/360 );

        if (fi==0) Image1->Canvas->MoveTo( x,y );
        else Image1->Canvas->LineTo( x,y );
    }
}
end;
```

### Vysvětlení:

- Skládáním sinusoid různých fází a kmitočtů vznikají tzv. Lissajousovy obrazce. Kružnice je jedním z nich.
- Podmínka „if (fi==0)...“ zajišťuje přesunutí výchozího bodu kreslení. Nahrazuje počáteční **MoveTo()** z minulých příkladů.

## Lissajous 2



Obr. 44 G07.BPR

### Co to dělá:

Kreslí Lissajousův obrazec složený z obdélníků.

### Jak:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int fi,x,y;

    Image1->Canvas->Pen->Width = 1;
    Image1->Canvas->Brush->Color = clYellow;

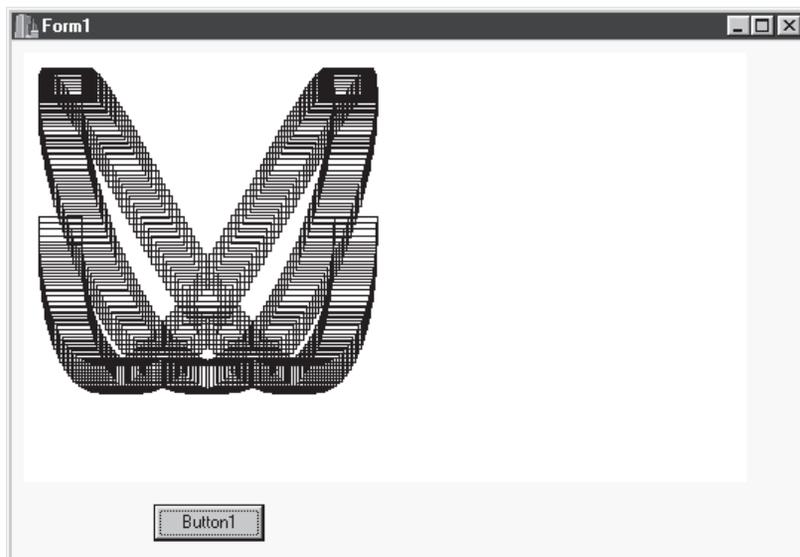
    for (fi=0;fi<=360;fi++)
    {
        x = 110+100*cos( 3* fi*M_PI/360 );
        y = 110+100*sin( 5* fi*M_PI/360 );

        Image1->Canvas->Rectangle( x,y, x+30,y+20 );
    }
}
```

### Vysvětlení:

- Canvas má metodu **Rectangle()**, která nakreslí obdélník.
- Canvas má vedle vlastnosti **Pen** určující způsob kreslení čar i vlastnost **Brush** – štětec – určující způsob kreslení ploch.

## Lissajous 3



Obr. 45 G08.BPR

### Co to dělá:

Kreslí Lissajousův obrazec z průhledných obdélníků.

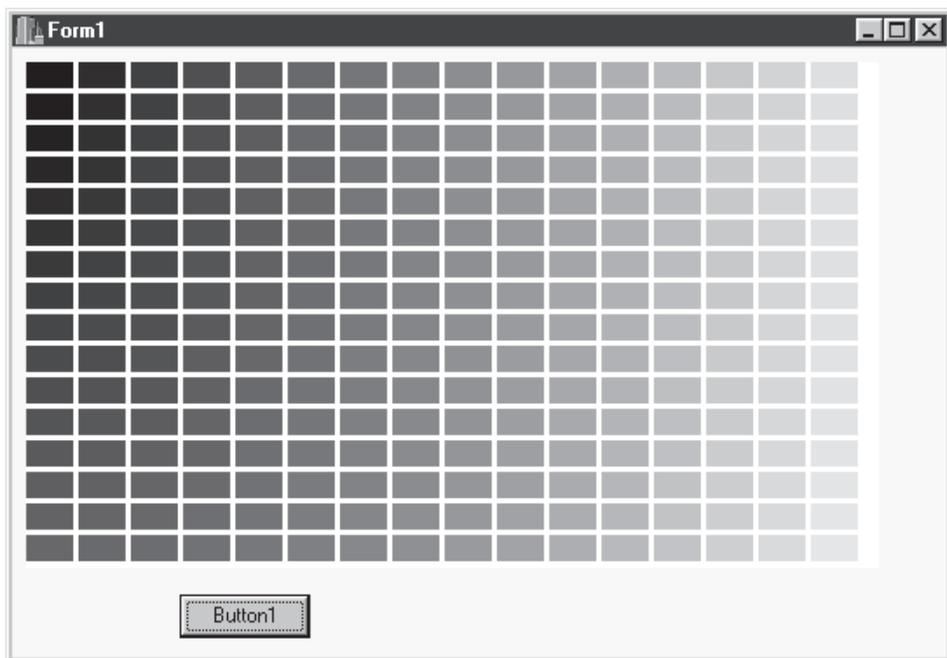
### Jak:

```
...  
    Image1->Canvas->Brush->Color = clYellow;  
    Image1->Canvas->Brush->Style = bsClear;  
...
```

### Vysvětlení:

- Brush má vedle vlastnosti **Color** i vlastnost **Style** udávající, zda má být plocha kreslena plnou barvou (jak odpovídá implicitní hodnotě **bsSolid**), některým ze způsobů šrafování nebo nevyplňována vůbec (**bsClear**).

## Tabulka barev



Obr. 46 G09.BPR

### Co to dělá:

Nakreslí tabulku barev.

### Jak:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int x,y,dx,dy;
    const int sx = 2;
    const int sy = 2;
    const int c1 = 0x001100;
    const int c2 = 0x110000;

    dx = Image1->Width / 16;
    dy = Image1->Height / 16;
    Image1->Canvas->Pen->Style = psClear;
    for (x=0; x<=15; x++){
        for (y=0; y<=15; y++){
            Image1->Canvas->Brush->Color = x *c1 + y*c2;
        }
    }
}
```

```

        Image1->Canvas->Rectangle( dx*x, dy*y,
                                   dx*(x+1)-sx,
                                   dy*(y+1)-sy);
    }
}
}
end;

```

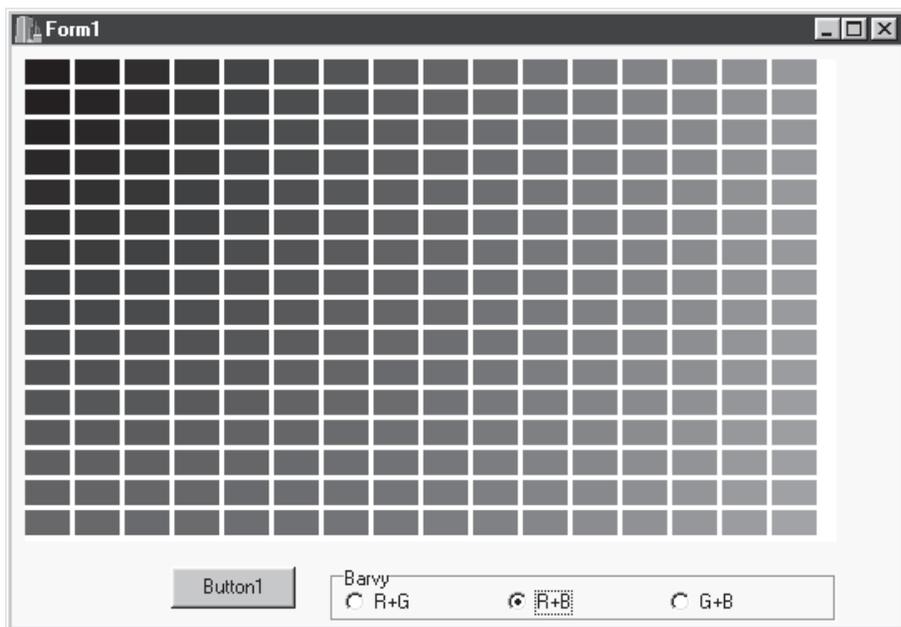
### Vysvětlení:

- Barva, kterou bude vyplněn obdélník, je tentokrát počítána jako určité množství zelené (roste s osou x) a určité množství modré (roste s osou y).
- Konstanty c1 a c2 určují „jednotky“, ze kterých budou sčítány barvy. Hodnoty jsou určeny tak, aby nejvyšší násobek použitý v programu (15×) dával největší sytost té které barevné složky.
- I **Pen** má vlastnost **Style**. Implicitní hodnota je **psSolid** (plná čára), lze zvolit různé druhy přerušovaných čar (např. **psDot** – tečkovaná); hodnota **psClear** znamená, že se čára kreslit nebude.

### Poznámka

*Přerušované čáry se kreslí přerušovaně jen při tloušťce (**Width**) 1.*

## Tabulka barev 2



Obr. 47 G10.BPR

### Co to dělá:

Jako minulý příklad, ale lze zvolit, které barvy se budou míchat.

### Jak:

Nastavte tyto vlastnosti komponenty **RadioGroup1** typu **TRadioGroup**:

- **Caption** na 'Barvy',
- **Columns** na 3,
- **Items.Strings** na ('R+G', 'R+B', 'G+B').

```
class TForm1 : public TForm
{
...
public:          // User declarations
    TColor c1,c2;
...
};

void __fastcall TForm1::RadioGroup1Click
    (TObject *Sender)
{
    switch (RadioGroup1->ItemIndex)
```

```

{
  case 0:
    c1 = 0x000011;
    c2 = 0x001100;
    break;
  case 1:
    c1 = 0x000011;
    c2 = 0x110000;
    break;
  case 2:
    c1 = 0x001100;
    c2 = 0x110000;
  }
  Button1Click( 0 );
}
end;

```

### Vysvětlení:

- c1 a c2 jsou proměnné nastavované přepnutím komponenty **RadioGroup1**.
- **Button1Click(0)** zavolá proceduru pro vykreslení tabulky.
- Metoda **RadioGroup1Click** (událost **RadioGroup1->OnClick**) je zavolána, kdykoliv se změní stav přepínače **RadioGroup1**.