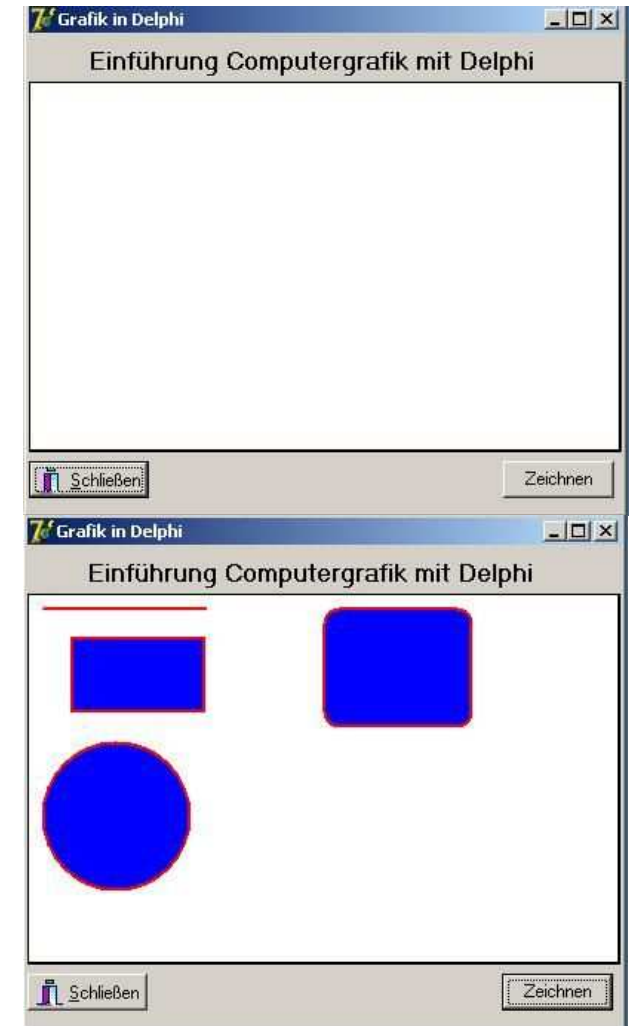


Einführung in die Grafikprogrammierung mit Delphi (Lösungen der Übungsaufgaben)

Übung 1: Zeichnen mit Delphi



```
procedure TForm1.FormCreate(Sender: TObject);
begin
  with Image1.Canvas do begin
    // weiße Leinwand zeichnen
    Pen.Color := clblack;
    Pen.Style := psSolid;
    Pen.Width := 2;
    Brush.Color := clwhite;
    Brush.Style := bsSolid;
    Rectangle(0,0,Image1.Width,Image1.Height);
  end;
end;
```

```
procedure TForm1.BZeichnenClick(Sender: TObject);
begin
  with Image1.Canvas do begin
    pen.color:=clred; // Pinselfarbe Rot
    pen.width:=2; // Pinselbreite 2 Pixel
    brush.color:=clblue; // Pinselfarbe blau
    moveto(10,10); // Stift an angegebene Position bewegen
    lineto(120,10); // Linie zeichnen
    rectangle(30,30,120,80); // Rechteck zeichnen
    ellipse(10,100,110,200); // Ellipse/Kreis zeichnen
    roundrect(200,10,300,90,20,20)
  end;
end;
```

Übung 2: Zeichnen eines einfachen Hauses

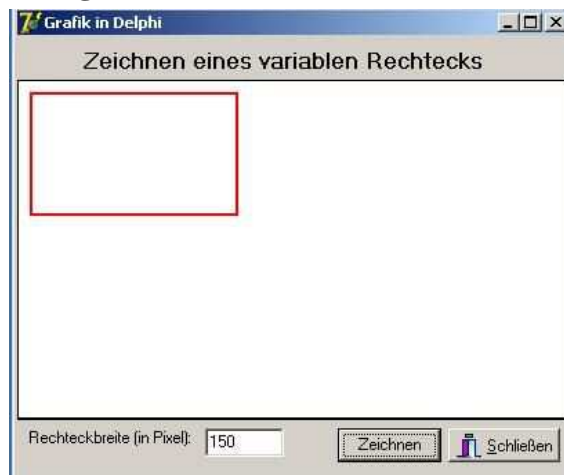


```

procedure TForm1.BZeichnenClick(Sender: TObject);
begin
  with Image1.Canvas do begin
    Pen.Color := clblue;
    Pen.Width := 2;
    Pen.Style := psSolid;
    Rectangle(100,100,300,225);
    // Dach zeichnen
    Pen.Color := clred; // rote Dachziegel
    MoveTo(100,100);
    LineTo(200,40);
    LineTo(300,100);
    // Dach fertig
    Pen.Color := clblue;
    Rectangle(120,120,150,160); // Fenster
    Rectangle(220,170,260,225); // Tuer
  end;
end;

```

Übung 3: Zeichnen eines Rechtecks mit variabler Breite

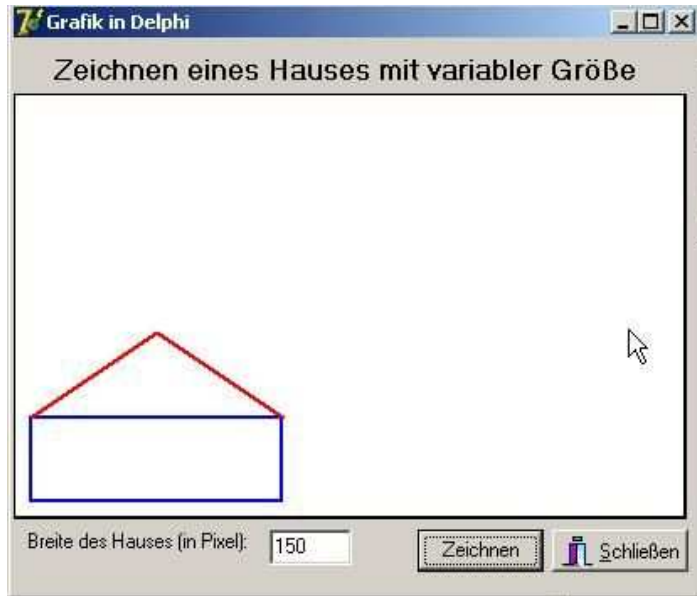


```

procedure TForm1.BZeichnenClick(Sender: TObject);
var breite: integer;
begin
  breite := StrToInt(Edit1.Text);
  with Image1.Canvas do begin
    Pen.Color := clred;
    Pen.Width := 2;
    Rectangle(10,10,10+breite,100);
  end;
end;

```

Übung 4: Haus mit variabler Größe

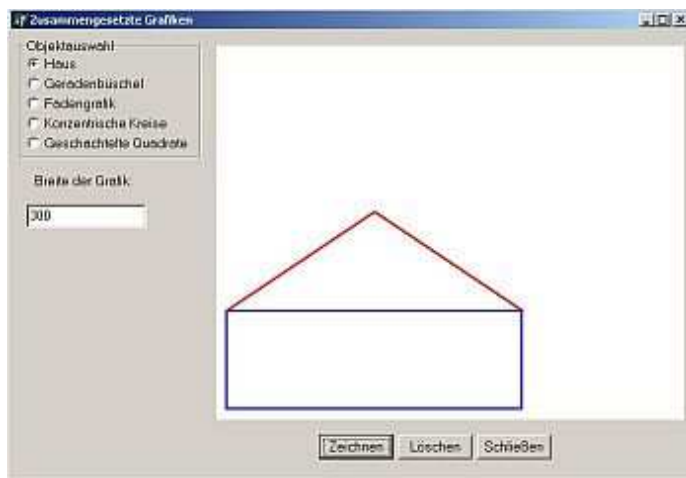


```

procedure TForm1.BZeichnenClick(Sender: TObject);
var
    ob, oh, xDach, yDach: integer;
begin
    ob := StrToInt(Edit1.Text);
    oh := round(ob/3);
    with Image1.Canvas do begin
        Pen.Color := clblue;
        Pen.Width := 2;
        Rectangle(10, Image1.Height-10-oh, 10+ob, Image1.Height-10);
        xDach := round((10+(10+ob))/2);
        yDach := round(Image1.Height-10-2*oh);
        Pen.Color := clRed;
        MoveTo(10, Image1.Height-10-oh);
        LineTo(xDach, yDach);
        LineTo(10+ob, Image1.Height-10-oh);
    end;
end;

```

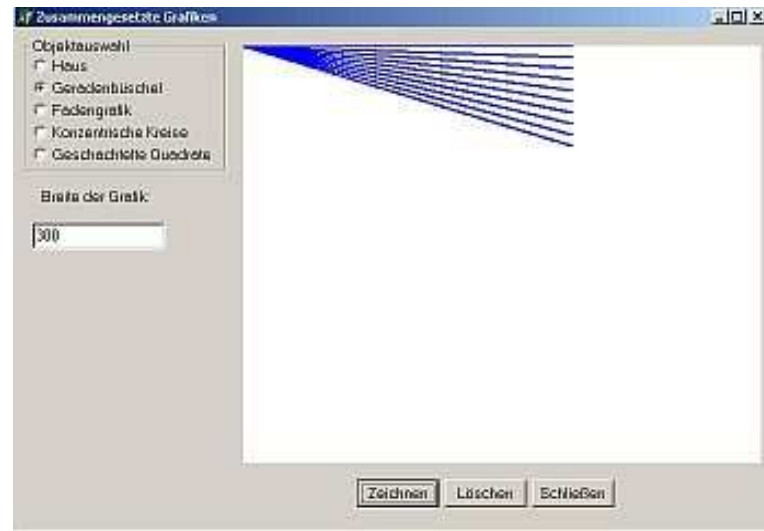
Übung 5: zusammengesetzte Grafiken



```

procedure hauszeichnen;
var
    oh, xDach, yDach: integer;
begin
    oh := round(ob/3);
    with Form1.Image1.Canvas do begin
        Pen.Color := clblue;
        Pen.Width := 2;
        Rectangle(10, Form1.Image1.Height-10-oh, 10+ob, Form1.Image1.Height-10);
        xDach := round((10+(10+ob))/2);
        yDach := round(Form1.Image1.Height-10-2*oh);
        Pen.Color := clRed;
        MoveTo(10, Form1.Image1.Height-10-oh);
        LineTo(xDach, yDach);
        LineTo(10+ob, Form1.Image1.Height-10-oh);
    end;
end;

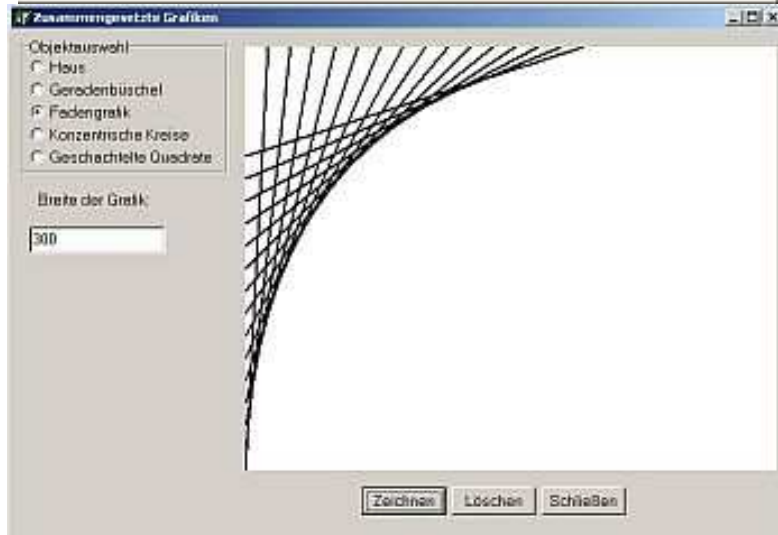
```



```

procedure geradenbueschel;
var
  i, j: integer;
begin
  j := 0;
  with Form1.Imagel.Canvas do
  begin
    Pen.Color := clblue;
    for i := 1 to 10 do
    begin
      MoveTo(0,0);
      LineTo(ob,j);
      j := j + 10;
    end;
  end;
end;

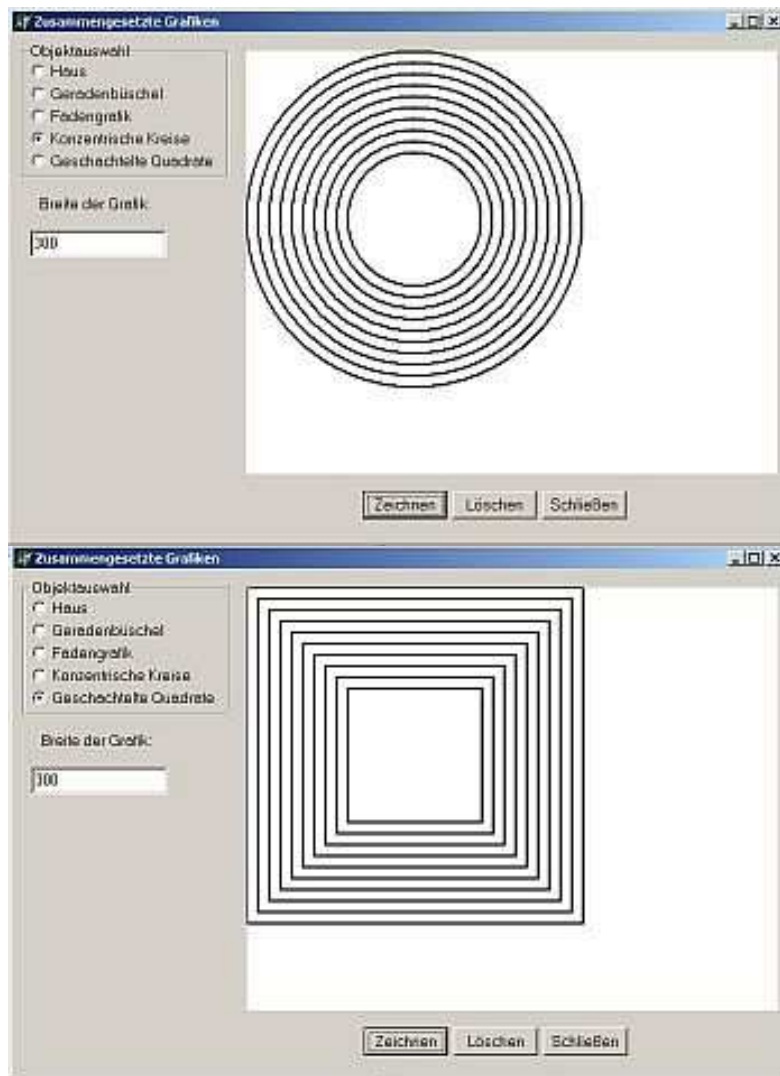
```



```

procedure fadengrafik;
var i,j: integer;
begin
  with Form1.Imagel.Canvas do
  begin
    i := Form1.Imagel.Height;
    j := 20;
    while j <= ob do
    begin
      MoveTo(0,i);
      LineTo(j,0);
      i := i-20;
      j := j+20;
    end;
  end;
end;

```



```

procedure kreise;
var
  x1,y1,x2,y2,i: integer;
begin
  x1 := 0; y1 := 0; x2 := ob; y2 := ob;
  with Form1.Imagel.Canvas do
    begin
      for i := 1 to 10 do
        begin
          Ellipse(x1,y1,x2,y2);
          x1 := x1+10; y1 := y1+10;
          x2 := x2-10; y2 := y2-10;
        end;
      end;
    end;

```

```

procedure quadrate;
var
  x1,y1,x2,y2,i: integer;
begin
  x1 := 0; y1 := 0; x2 := ob; y2 := ob;
  with Form1.Imagel.Canvas do
    begin
      for i := 1 to 10 do
        begin
          Rectangle(x1,y1,x2,y2);
          x1 := x1+10; y1 := y1+10;
          x2 := x2-10; y2 := y2-10;
        end;
      end;
    end;

```