

- 1 Für ein Break-Out-Spiel wird ein Kasten auf das Spielfeld gezeichnet. Wird dieser von einem Ball getroffen, erhöht sich die Variable „getroffen“ um 1. Ist der Kasten 3-mal getroffen worden, ist das Spiel zu Ende und das Spielfeld wird gelöscht. Untersuchen Sie, was in diesem Zusammenhang die Prozeduren „eins“ und „zwei“ machen.
- „eins“ zeichnet ein ausgefülltes gelbes Rechteck mit vorgegebenen Begrenzungen.
- „zwei“ löscht das gesamte Fenster, indem dieses mit der Farbe weiß übermalt wird.

```
type tKasten = class
    x1,y1,x2,y2:integer;
    Farbe:tColor;
    getroffen:integer;
    procedure eins;
    procedure zwei;
end;

var Kasten:tKasten;

procedure tKasten.eins;
begin
    Form1.Canvas.Pen.Color:=clYellow;
    Form1.Canvas.Brush.Style:=bsSolid;
    Form1.Canvas.Brush.Color:=clYellow;
    Form1.Canvas.Rectangle(Kasten.x1,Kasten.y1,Kasten.x2,Kasten.y2);
end;

procedure tKasten.zwei;
begin
    Form1.Canvas.Pen.Color:=clWhite;
    Form1.Canvas.Brush.Style:=bsSolid;
    Form1.Canvas.Brush.Color:=clWhite;
    Form1.Canvas.Rectangle(0,0,Form1.Width,Form1.Height);
end;
```

Schreiben Sie für folgende Aufgaben die entsprechende(n) Anweisung(en):

- a) Der x1-Wert von Kasten soll den Wert 45 erhalten,  
`Kasten.x1:=45;`
- b) y2 soll den doppelten Wert wie y1 erhalten.  
`Kasten.y2:=2*Kasten.y1;`
- c) Es soll überprüft werden, ob der Kasten 3-mal getroffen ist. Wenn das der Fall ist, soll das Spielfeld gelöscht werden.  
`if Kasten.getroffen=3 then Kasten.zwei;`
- 

- 2 Der Textinhalt der Komponente *Edit1* soll die Aufschrift auf der Komponente *Button1* werden.  
Schreiben Sie die entsprechende Anweisung.  
`Button1.Caption:=Edit1.Text;`

- 3 var Schublade : array [ 1..10 ] of integer;  
Schreiben Sie einen Anweisungsteil, durch den in jede der „Schubladen“ jeweils eine Zufallszahl von 100 (einschließlich) bis 200 (ausschließlich) eingefügt wird.

```
for i:=1 to 10 do Schublade[i]:=random(100)+100;
```

---

- 4 Addieren Sie alle ungeraden Zahlen (1, 3, 5, 7, ...) so lange, bis die Summe 25000 übersteigt. Schreiben Sie zu dieser Aufgabe einen Programmteil. In *Edit1* soll zum Schluss die letzte und damit größte addierte Zahl stehen.

```
Summe:=0;  
Zahl:=-1;  
repeat  
  Zahl:=Zahl+2;  
  Summe:=Summe+Zahl;  
until Summe>25000;  
Edit1.Text:=IntToStr(Zahl);
```

---

- 5 Ein Buchhändler hat seine Bücher folgendermaßen gekennzeichnet: 0 für neu, 1 für alt aber einwandfrei, 2 für alt mit leichten Mängeln und 3 für alt mit schweren Mängeln. Bücher der Kategorie 0 werden zum vollen Preis, der Kategorie 1 zum halben Preis, der Kategorie 2 zum 1/4 Preis und der Kategorie 3 zum 1/10 Preis verkauft. Der volle Preis steht in *Edit1*. In *Edit2* steht die Kategorie des Buchs. Schreiben Sie den Programmteil, der in *Edit3* den Verkaufs-Preis des Buches ausgibt.

```
k:=StrToInt(Edit2.Text);  
vollerPreis:=StrToFloat(Edit1.Text)  
case k of  
  0:VerkaufsPreis:=vollerPreis;  
  1:VerkaufsPreis:=vollerPreis/2;  
  2:VerkaufsPreis:=vollerPreis/4;  
  3:VerkaufsPreis:=vollerPreis/10;  
end;  
Edit3.Text:=FloatToStr(VerkaufsPreis);
```

---

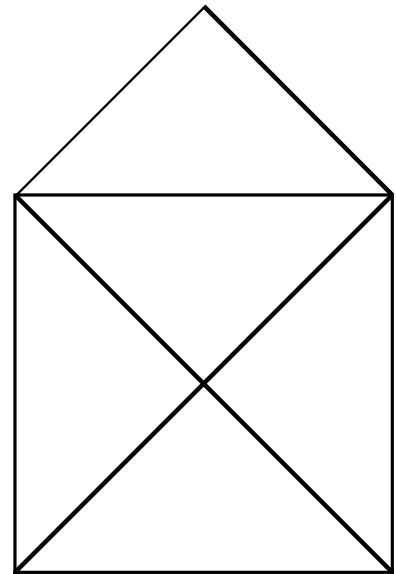
- 6 Wenn die Variable *Ende* den Wert *false* hat, soll die Prozedur *Weiter* aufgerufen werden, sonst soll das Programm beendet werden. Schreiben Sie die entsprechende Anweisung.

```
if not Ende  
  then Weiter  
  else application.terminate;
```

---

- 7 Lassen Sie den Computer auf der Form1 dieses Bild zeichnen.  
Der untere linke Punkt hat die Koordinaten (100/400).  
Höhe und Breite des Nikolaushauses sind 100.  
Das Dach oben ist rechtwinklig und gleichschenkelig.  
Schreiben Sie den Programmteil, der das Haus zeichnet.

```
with form1.canvas do begin  
  brush.style:=bsclear;  
  rectangle(100,300,200,400);  
  moveto(100,300);lineto(200,400);  
  moveto(100,400);lineto(200,300);  
  moveto(100,300);lineto(150,250);  
  moveto(150,250);lineto(200,300);  
end;
```



---

*Viel Erfolg bei der Arbeit !!!*