

Aufgabe 1: Gegeben ist das folgende Programm:

```

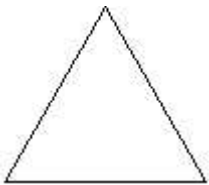
PR figur :n :Länge
  WH 3 [
    VW :Länge
    WENN :n>1 DANN (
      RE 90
      figur :n-1 :Länge/2
      LI 90
    )
    RW :Länge
    LI 90
  ]
  LI 90
ENDE

```

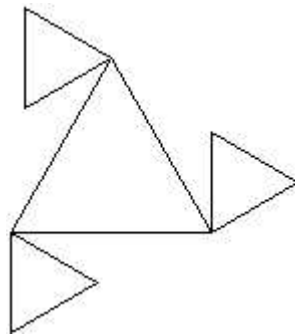
Zeichne das Bild, welches beim Aufruf der Prozedur mit **figur 4 80** entsteht. Ein *mm* der Zeichnung entspreche dabei einem *Pixel* in LOGO. Der Igel stehe am Anfang „mit dem Gesicht“ nach oben.

Aufgabe 2: Schau dir die folgenden Grafiken an:

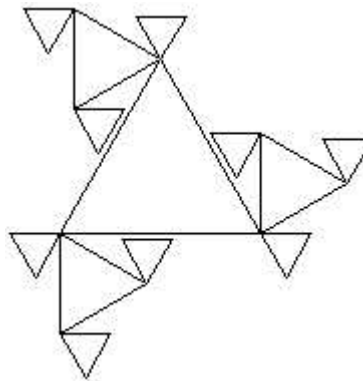
Rekursionsstufe 1



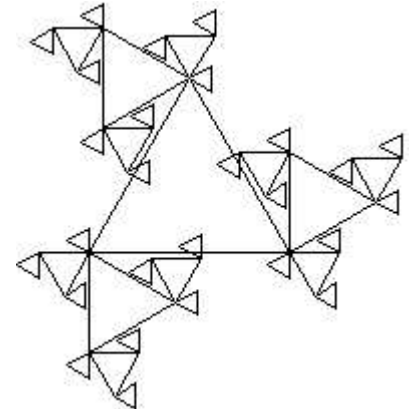
Rekursionsstufe 2



Rekursionsstufe 3

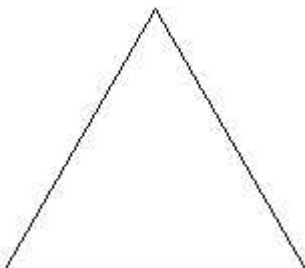


Rekursionsstufe 4

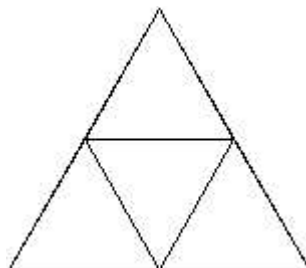


- Implementiere eine rekursive Prozedur **PR figur :n :Länge**, welche diese Grafiken rekursiv erzeugen kann.
- Eine leichte Abwandlung der Prozedur sollte die folgenden Bilder erzeugen. Wie sieht diese Abwandlung aus? Gib bitte das **vollständige** Programm an.

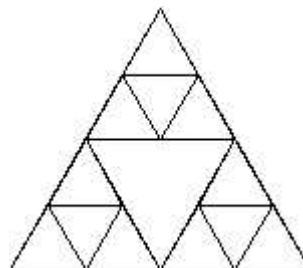
Rekursionsstufe 1



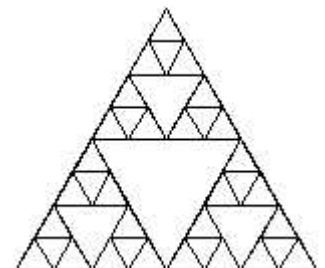
Rekursionsstufe 2



Rekursionsstufe 3



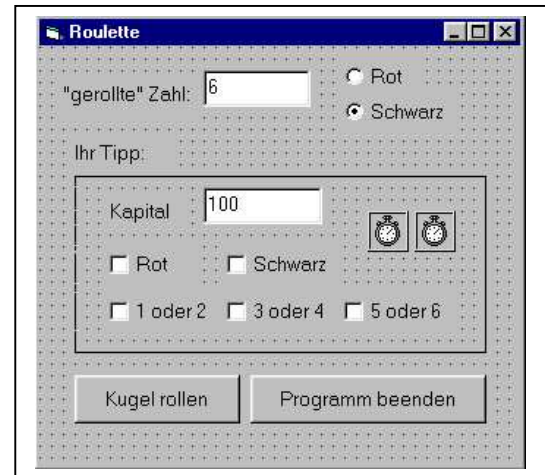
Rekursionsstufe 4



Bitte wenden ↵

Aufgabe 3: Im folgenden sollst du ein kleines Roulette-Spiel programmieren. Folgende Regeln gelten für das Programm:

- Der Spieler besitzt zu Anfang das Startkapital von 100 €.
 - Bevor ein neues Spiel beginnt, kann der Spieler über die Checkboxes eine Wette vornehmen. Jede Möglichkeit kostet 1 €.
 - Sobald der Spieler den „Kugel rollen“-Button drückt soll folgendes passieren:
 - 1) Die getätigte Auswahl ist nicht mehr änderbar (deaktiviert) und der Startbutton soll unsichtbar werden.
 - 2) Dem Spieler wird entsprechend seiner Wette der Einsatz vom Kapital abgezogen. Dieser kann zwischen 0 € (keine Wette) und 5 € (alle Häckchen gesetzt) betragen.
 - 3) Die „gerollte“ Zahl wird alle 20 Millisekunden neu „berechnet“. Dabei können nur Zahlen zwischen 1 und 6 auftreten. Außerdem sind alle geraden Zahlen schwarz und alle ungeraden Zahlen rot, was mittels der Optionsfelder angezeigt werden soll.
 - Nach 5 Sekunden stoppt die Kugel automatisch. Im Anschluss daran soll folgendes passieren:
 - 1) Für jede Farbe, die der Spieler richtig getippt hat, bekommt er 2 € gutgeschrieben.
 - 2) Für jede Zahlenwette (1 oder 2, 3 oder 4 bzw. 5 oder 6), die der Spieler richtig getippt hat, bekommt er 3 € gutgeschrieben.
 - 2) Die Wettenauswahl soll wieder änderbar (aktiviert) sein.
 - 3) Der Start-Button wird sichtbar, falls der Spieler noch über mindestens 3 € verfügt.
 - Drückt der Spieler den „Programm beenden“-Button, so wird das Programm beendet.
- a) Welche verschiedenen Dialogelemente wurden verwendet (insgesamt 8 verschiedene)? Gib jeweils einem Vertreter einen sinnvollen Namen. **Beispiel:** Es wurden zwei Timer verwendet. Einer der Timer könnte z. B. `Ti_Kugelstopp` heißen.
- b) Schreibe ein VisualBasic-Programm für dieses Spiel, indem Du die unten stehenden Routinen mit Programmzeilen füllst. Verwende dabei die in a) definierten Bezeichnungen für die Dialogelemente.



```
Private Sub B_Ende_Click()
    hier stehen Deine Programmzeilen
End Sub

Private Sub B_Start_Click()
    hier stehen Deine Programmzeilen
End Sub

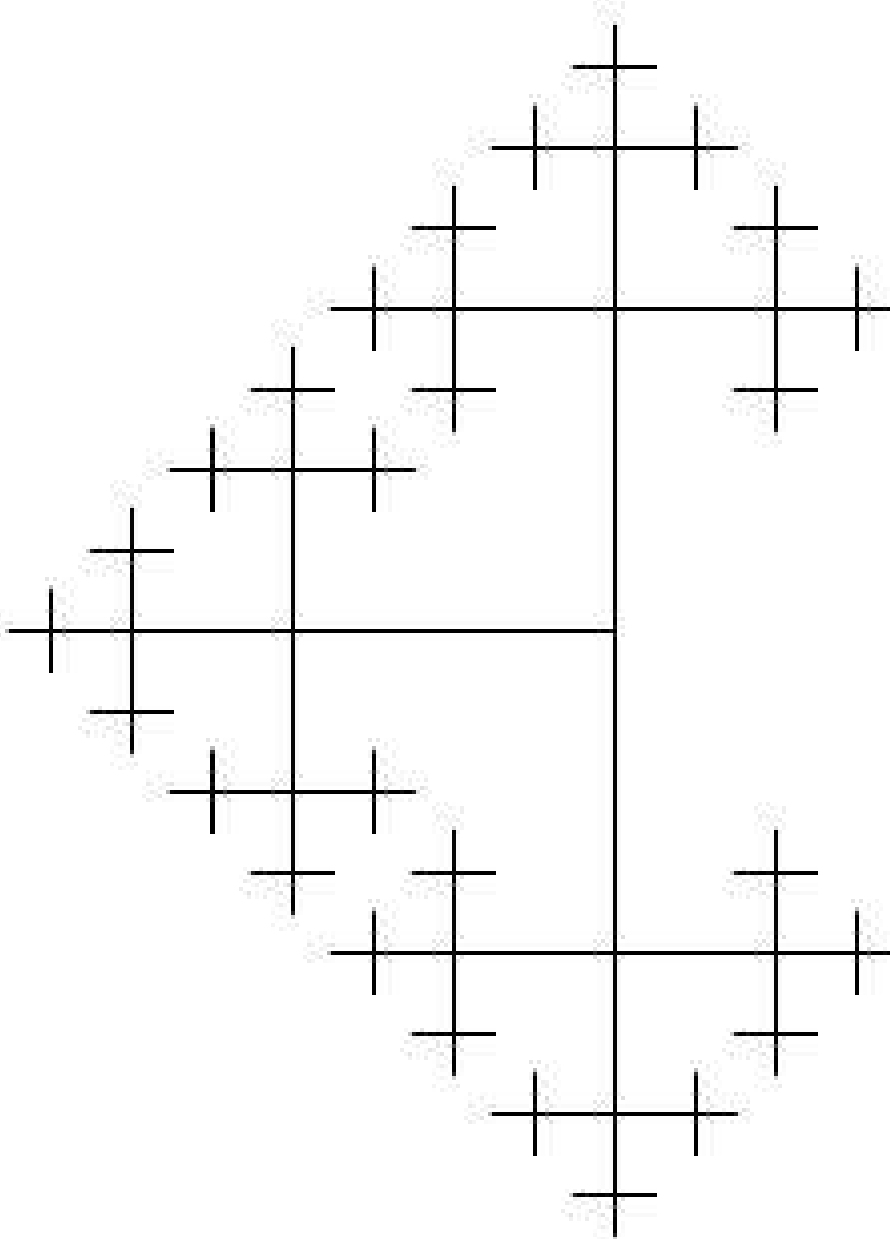
Private Sub Ti_Kugelanzeige_Timer()
    hier stehen Deine Programmzeilen
End Sub

Private Sub Ti_Kugelstopp_Timer()
    hier stehen Deine Programmzeilen
End Sub
```

Viel Erfolg!

LÖSUNGEN

Aufgabe 1: Das Bild sieht wie folgt aus:



Aufgabe 2: a) Implementierung z. B. so:

```
PR figur :n :länge
wh 3 [
  wenn :n>1 dann (
    re 90
    figur :n-1 :länge/2
    li 90
  )
  vw :länge
  links 120
]
```

ENDE

b) Die leichte Abwandlung:

```
PR figur :n :länge
wh 3 [
  wenn :n>1 dann (
    figur :n-1 :länge/2
  )
  vw :länge
  links 120
]
```

ENDE

Aufgabe 3: Die Implementierung könnte so oder so ähnlich aussehen:

```
Private Sub B_Ende_Click()
    End
End Sub

Private Sub B_Start_Click()
    If CB_Rot.Value = Checked Then T_Kapital.Text = T_Kapital.Text - 1
    If CB_Schwarz.Value = Checked Then T_Kapital.Text = T_Kapital.Text - 1
    If CB_1o2.Value = Checked Then T_Kapital.Text = T_Kapital.Text - 1
    If cb_3o4.Value = Checked Then T_Kapital.Text = T_Kapital.Text - 1
    If CB_5o6.Value = Checked Then T_Kapital.Text = T_Kapital.Text - 1

    CB_Rot.Enabled = False
    CB_Schwarz.Enabled = False
    CB_1o2.Enabled = False
    cb_3o4.Enabled = False
    CB_5o6.Enabled = False

    B_Start.Enabled = False
    Ti_Spiel.Enabled = True
    Ti_Kugel.Enabled = True
End Sub

Private Sub Ti_Kugel_Timer()
    T_Zahl.Text = Int(Rnd * 6) + 1
    If T_Zahl.Text = "1" Or T_Zahl.Text = "3" Or T_Zahl.Text = "5" Then
        O_Rot.Value = True
    Else
        O_Schwarz.Value = True
    End If
End Sub

Private Sub Ti_Spiel_Timer()
    Ti_Spiel.Enabled = False
    Ti_Kugel.Enabled = False
    If (CB_Rot.Value = Checked And O_Rot.Value = True) Or
        (CB_Schwarz.Value = Checked And O_Schwarz.Value = True) Then
        T_Kapital.Text = T_Kapital.Text + 2
    End If
    If (CB_1o2.Value = Checked And (T_Zahl.Text = "1" Or T_Zahl.Text = "2")) Then
        T_Kapital.Text = T_Kapital.Text + 3
    End If
    If (cb_3o4.Value = Checked And (T_Zahl.Text = "3" Or T_Zahl.Text = "4")) Then
        T_Kapital.Text = T_Kapital.Text + 3
    End If
    If (CB_5o6.Value = Checked And (T_Zahl.Text = "5" Or T_Zahl.Text = "6")) Then
        T_Kapital.Text = T_Kapital.Text + 3
    End If

    CB_Rot.Enabled = True
    CB_Schwarz.Enabled = True
    CB_1o2.Enabled = True
    cb_3o4.Enabled = True
    CB_5o6.Enabled = True

    If T_Kapital >= 3 Then B_Start.Enabled = True
End Sub
```

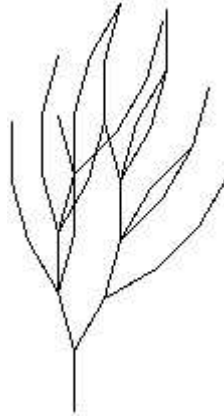
Nicht verwendet:

Aufgabe 3: Schreibe eine rekursive Prozedur, welche in den Rekursionsstufen 1 – 3 die folgenden Bilder zeichnet.

Rekursionsstufe 1



Rekursionsstufe 2



Rekursionsstufe 3

