```
Procedure dialog1;
(* vorhaben:
*)
(* vorgehen:
*)
CONST
(* im CONST block werden variablen deklariert, welche konstant sind,
*)
  {rechtecklaenge}
  rectLength  = 8;
  {abstand zwischen den rechtecken}
  distBetweenRects = 2;

VAR
  numbOfRect : INTEGER;
  deltaX, p1x, p1y, p2x, p2y : REAL;
  {zaehlervariablen}
  i : INTEGER;

BEGIN
(* FUNCTION   IntDialog (request  :STRING;default   :STRING) :REAL ;
--> function IntDialog  displays a dialog box which requests the user to enter a INTEGER value.
IntDialog automatically screens for valid numeric input.

was heisst das?
die funktion intdialog gibt einen integer-wert(ganzzahl) zurück.
die zurückgegebene zahl ist die umwandlung eines strings der über das dialogfeld eingegeben wird!

welche variablen muessen angegeben werden?
request   dein text als string, d.h. der text muss in anführungszeichen stehen!
default   default-wert als string.
*)
numbOfRect := IntDialog('Wieviele Rechtecke sollen gezeichnet werden?','8');

{zeichnen der rechtecke in einer schleife}
FOR i:= 1 TO numbOfRect DO BEGIN
  (*um wieviel muss das rechteck jeweils weiter rechts gezeichnet werden?
  --> die rechtecklaenge + den abstand zwischen den rechtecken !
  *)
  deltaX := rectLength + distBetweenRects;
  {linke untere ecke}
  p1x := i * deltaX;
  p1y := 0;
  {obere rechte ecke}
  p2x := p1x + rectLength;
  p2y := p1y + rectLength;

  {farbfüllung setzen, damit es etwas bunter ausssieht...}
  FillBack(random * 65535, random * 65535, random * 65535);
```

```
56    rect(p1x, p1y, p2x, p2y);
57
58  END;
59
60
61
62  END;
63  RUN(dialog1);
64
```

```
Procedure array1;
(* vorhaben:
es sollen 100 rechtecke in gleichmaessig verteilt in einer fläche gezeichnet werden.
jedes rechteck soll eine zufällige kantenlänge zwischen 3 und 10 einheiten haben.
jedes rechteck soll in abhängigkeit seiner kantenlänge rotiert werden.
*)
(* vorgehen:
  1. die kantenlänge der rechtecke in einem array speichern
  2. die rechtrecke zeichnen.
  3. die rechtecke rotieren
*)
CONST
(* im CONST block werden variablen deklariert, welche konstant sind,
d.h. sich im verlauf des skripts nicht ändern.
bei konstanten muss kein datentyp, z.b. string, integer oder real angegeben werden.
*)
  {max. rechtecklaenge}
  rectLength  = 10;
  {abstand zwischen den rechtecken}
  distBetweenRects = 2;
  {anzahl der rechtecke in X richtung}
  numbOfRectX = 10;
  {anzahl der rechtecke in Y richtung}
  numbOfRectY = 10;
VAR
  deltaX, deltaY, p1x, p1y, p2x, p2y, Cx, Cy : REAL;
  {zaehlervariablen}
  i, m, index : INTEGER;
  {array für dei kantenlänge}
  kanten_array : Array[0..99] OF Real;
  {array für dei kantenlänge}
  handle_array : Array[0..99] OF Handle;

BEGIN

{1.zufälliges setzen der kantenlänge der rechtecke}
FOR i:= 0 TO 99 DO BEGIN
  kanten_array[i] := 3 + random * 7;
END;

{2.zeichnen der rechtecke in zwei schleifen}
FOR m:= 0 TO numbOfRectY-1 DO BEGIN
  {die 'm' schleife ist für die positionierung in  y richtung verantwortlich}
  FOR i:= 0 TO numbOfRectX-1 DO BEGIN
    {die 'm' schleife ist für die positionierung in  x richtung verantwortlich}
    (*um wieviel muss das rechteck jeweils weiter rechts bzw. oben gezeichnet werden?
    --> delta = die rechtecklaenge + den abstand zwischen den rechtecken !
    *)
    {der index referenziert die position in dem array}
```

```
    index := m* 10 + i;

    deltaX := rectLength + distBetweenRects;
    deltaY := rectLength + distBetweenRects;
    {linke untere ecke}
    p1x := i * deltaX;
    p1y := m * deltaY;
    {obere rechte ecke}
    p2x := p1x + kanten_array[index];
    p2y := p1y + kanten_array[index];

    {farbfüllung setzen, damit es etwas bunter ausssieht...}
    FillBack(random * 65535, random * 65535, random * 65535);
    rect(p1x, p1y, p2x, p2y);

    {speichern eines handles auf das rechteck in einem array}
    handle_array[index] := LNewObj;
  END;
END;

{3.rotieren der rechtecke in abhängigkeit der kantenlänge um das rechteckzentrum}
FOR i:= 0 TO 99 DO BEGIN
  (*HCenter(h :HANDLE; VAR pX :REAL; VAR pY :REAL) ;
  --> Procedure HCenter returns the logical center point of the object specified in h.
  For most objects, this is the center of the bounding box.
  For circles, arcs, and round walls HCenter returns the arc center of the object.
  *)
  HCenter(handle_array[i],Cx, Cy);
  (*PROCEDURE   HRotat( h :HANDLE; centerX  :REAL; centerY :REAL; rotationAngle :REAL) ;
  --> Procedure HRotate rotates the referenced object about a coordinate point location. rotationAngle is in degrees.
  *)
  HRotate(handle_array[i],Cx, Cy,kanten_array[i] * 3);

END;


END;
RUN(array1);

```

```
Procedure dialog2_mouse;
(* vorhaben:
es sollen eine BESTIMMTE anzahl rechtecke in einer reihe gezeichnet werden.
diese BESTIMMTE anzahl soll über ein dialogfeld eingegeben werden können.
der startpunkt der reihe ist (0,0), der endpunkt der reihe in horizontaler richtung soll über mausklick festgelegt werden.
*)
(* vorgehen:
  1. dialog
  2. endposition der reihe durch mauseingabe abfragen
  3. zeichnen der rechtecke
*)
CONST
(* im CONST block werden variablen deklariert, welche konstant sind,
d.h. sich im verlauf des skripts nicht ändern.
bei konstanten muss kein datentyp, z.b. string, integer oder real angegeben werden.
*)
  {rechtecklaenge}
  rectLength  = 8;


VAR
  numbOfRect : INTEGER;
  deltaX, p1x, p1y, p2x, p2y, rowEndPoint, distBetweenRects : REAL;
  {zaehlervariablen}
  i : INTEGER;
  rowEndPointX,rowEndPointY : REAL;

BEGIN
(* FUNCTION   IntDialog (request  :STRING;default   :STRING) :REAL ;
--> function IntDialog  displays a dialog box which requests the user to enter a INTEGER value.
IntDialog automatically screens for valid numeric input.

was heisst das?
die funktion intdialog gibt einen integer-wert(ganzzahl) zurück.
die zurückgegebene zahl ist die umwandlung eines strings der über das dialogfeld eingegeben wird!

welche variablen muessen angegeben werden?
request   dein text als string, d.h. der text muss in anführungszeichen stehen!
default   default-wert als string.
*)
numbOfRect := IntDialog('Bitte geben Sie die Anzhal der Rechtecke ein und legen danach die Endposition der Reihe mit der
Muas fest?','8');

(*FUNCTION   MouseDown(VAR pX :REAL; VAR pY :REAL) :BOOLEAN ;
--> function MouseDown returns TRUE if a mouse down event has occurred within the active document window.

was heisst das?
die funktion mousedown gibt einen wahrwert zurück, wenn ein mausklick ausgelöst wurde.
die koordinaten der mausposition wird in den variablen pX und pY abgelegt.
```

```
(wenn innerhalb der funktionbeschreibung ein VAR vor der variablen steht wird in dieser variablen ein wert abgelegt!
vergleiche hierzu zum beispiel die funktion intdialog, hier werden werte als parameter uebergeben!)
*)


REPEAT
  message('Bitte legen Sie den Endpunkt der Rechteckreihe fest!');
  {das skript wird erst weiter ausgeführt, wenn ein mausklick ausgelöst wurde!}
UNTIL MouseDown(rowEndPointX,rowEndPointY);
message('rowEndPointX',rowEndPointX ,   'rowEndPointY' , rowEndPointY);

{Abstand zwischen den Rechtecken berechnen}
distBetweenRects := (rowEndPointX - (numbOfRect* rectLength))/numbOfRect;

{zeichnen der rechtecke in einer schleife}
FOR i:= 1 TO numbOfRect DO BEGIN
  (*um wieviel muss das rechteck jeweils weiter rechts gezeichnet werden?
  --> die rechtecklaenge + den abstand zwischen den rechtecken !
  *)
  deltaX := rectLength + distBetweenRects;
  {linke untere ecke}
  p1x := i * deltaX;
  p1y := 0;
  {obere rechte ecke}
  p2x := p1x + rectLength;
  p2y := p1y + rectLength;

  {farbfüllung setzen, damit es etwas bunter ausssieht...}
  FillBack(random * 65535, random * 65535, random * 65535);
  rect(p1x, p1y, p2x, p2y);

END;



END;
RUN(dialog2_mouse);

```