

# Top-Down Beispiels-Programm in COMAL

Um Problemlösungen für Computer zu entwickeln, hat sich das Prinzip der schrittweisen Verfeinerung (Top-Down-Methode, strukturierte Programmierung) durchgesetzt und bewährt. Ein Problem wird schrittweise in einzelne Abschnitte untergliedert, die weiter verfeinert werden, bis eine Stufe erreicht ist, für die ein Äquivalent in der verwendeten Programmiersprache existiert. Die Programmiersprache dient dann dazu, die in Umgangssprache ausformulierte Problemlösung auf den Computer zu übertragen und dort ablaufen zu lassen.

Die Programmiersprache COMAL unterstützt diese Vorgehensweise in hervorragender Weise, weil es möglich ist die umgangssprachlichen Bezeichnungen als Prozedur-, Funktions- oder Variablenbezeichner zu verwenden, so dass lesbarer und verständlicher Programmcode erzeugt wird. Der so entstehende Programmcode hat zunächst sehr viele umgangssprachliche Bestandteile, die in immer weiter verfeinerten Prozeduren und Funktionen Verwendung finden, bis man für eine zunächst noch umgangssprachliche Formulierung eine Anweisung der Programmiersprache COMAL setzen kann. Das Programm ist fertig, wenn man alle Prozeduren soweit verfeinert hat.

Da es bei der Entwicklung eines COMAL-Programms i.a. fast von selbst zu einer Baumstruktur der Prozeduren kommt und da das Programm bei entsprechender Vorgehensweise gut lesbar ist, zeigen Testläufe meist sehr schnell, in welchem Teil eventuell noch Fehler aufgetreten sind.

Besonders für kleinere Programmprojekte im Schulbereich ist COMAL deshalb immer noch gut geeignet. Spaghetti-Code ist zwar auch unter COMAL möglich, wird aber nicht unterstützt. Vereinbarungen von Bezeichnern sind nicht als Selbstzweck erforderlich, sondern können erfolgen. Bei größeren Projekten müssen sie sein - allein schon, um die Übersicht zu behalten.

Auch wenn COMAL häufig als BASIC-Ableger bezeichnet wird, stellt es doch den Übergang zu PASCAL und Delphi nicht, da die verwendeten Schreibweisen in wesentlichen Teilen gleich sind.

COMAL ist allerdings in manchen Dingen einfacher, aber nicht weniger durchdacht. Das Semikolon als Trennzeichen zwischen Anweisungen entfällt am Ende einer Zeile, ebenfalls "begin" und "end" als Umrahmung von einfachen Strukturblocken. Dies kommt der Arbeit in der Sekundarstufe I sehr entgegen.

Nachfolgend die COMAL Befehlsliste für den Anfang:

## **Turtlegrafik**

Befehl	Beschreibung
use turtle	Schaltet Turtlegrafik ein (Package turtle wird geladen)
fullscreen	Grafik und Text auf dem ganzen Bildschirm
home	Turtle auf Anfangsposition (0,0)
clear	Grafikbildschirm löschen
nowrap	"nicht "wickeln"" (Im wrap-Modus wird eine Turtlespur, die über den Rand hinausgehen müsste, an der entgegengesetzten Bildschirmseite fortgesetzt.)"
pendown	Stift nach unten (Turtle zeichnet)
penup	Stift nach oben (Turtle zeichnet nicht)
forward(10)	10 Schritte vorwärts
right(30)	Rechtsdrehen um 30°
left(50)	Linksdrehen um 50°
arc(20,30)	Bogen rechts(Radius 20 Schritte, Winkel 30°)
arc(50,45)	Bogen links(Radius 50 Schritte, Winkel 45°)
circle(80,50,25)	Kreis um M(80,50) mit Radius von 25 Schritten
circle(xcor,ycor,30)	Kreis um Turtleposition mit Radius von 30 Schritten

<code>pencolor(0)</code>	Stiftfarbe auf Hintergrundfarbe
<code>pencolor(15)</code>	Stiftfarbe auf weiß
<code>window(-160,160,-120,120)</code>	Ändert auf meinem Laptop den voreingestellten Grafikbildschirm so, dass Kreise nicht als Ellipsen dargestellt werden. Voreingestellt ist das Koordinatensystem in x-Richtung von -160 bis 160 und in y-Richtung von -100 bis 100.
<code>moveto(70,-30)</code>	Bewegung zur Position (70,-30)
<code>drawto(70,-30)</code>	Linie zeichnen zur Position(70,-30)
<code>fill(21,80)</code>	füllt Bereich um Position (21,80) mit Stiftfarbe

Hinweis: Einige Anweisungen benötigen absolute Koordinaten. Im Allgemeinen sollte man beim Entwickeln von Programmen zum Zeichnen bestimmter Figuren auf diese verzichten und möglichst mit relativ zur Turtleposition und -lage arbeitenden Anweisungen arbeiten, weil dann der entwickelte Programmcode für solche Figuren auch nach Drehungen, Verschiebungen oder anderen Abbildungen ohne Probleme weiterverwendet werden kann.

## Sonstiges

<code>//</code>	Alles nach <code>//</code> in einer Zeile stehende wird vom Computer nicht bearbeitet. In Programmen meistens für Kommentare benutzt, funktioniert aber auch im Direktmodus, zB. um zu verhindern, dass der Computer versucht, am Ende einer Zeile noch stehengebliebene Grafikteile zu interpretieren.
<code>bye</code>	COMAL beenden
<code>"load""xyz.cml"""</code>	lädt das Comalprogramm xyz
<code>"save""abc"""</code>	"speichert Programm als <code>""abc.cml""</code> auf Festplatte"

## Direktmodus:

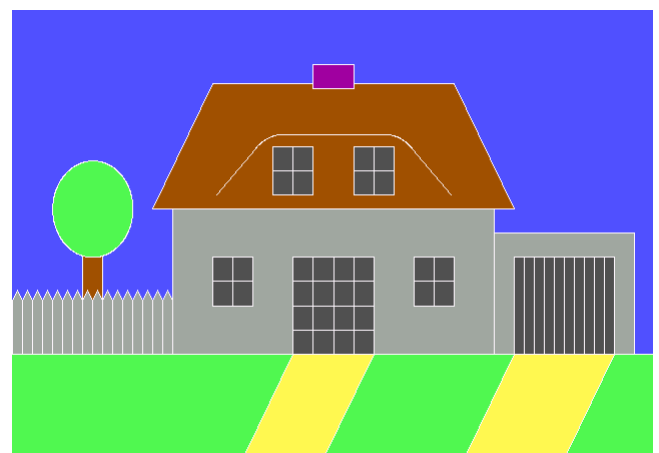
<code>forward(30) ENTER</code>	wird sofort ausgeführt.
--------------------------------	-------------------------

## Programmmodus:

<code>60 forward(30) ENTER</code>	wird nicht ausgeführt, sondern als Programmzeile im Arbeitsspeicher gespeichert. Der Arbeitsspeicher wird beim Ausschalten des Computers gelöscht. Ein Programm ist eine Folge von Anweisungen, hier nach Zeilennummern geordnet, die der Computer ausführt, wenn man RUN eingibt.
-----------------------------------	---

Im Beispielprogramm, welches ein Bild (Haus mit Garten) zeichnet, wird ein Verzweigungspfad (fett gedruckt) vom groben bis zu immer feineren Begriffen deutlich, bis man auf Programmiersprachebene endet. Das Programm lässt sich sicher noch eleganter anlegen (z.B. bei der Parameterübergabe), hat sich aber im Unterricht so ergeben und zeigt das Top-Down-Prinzip sehr deutlich.

Nachfolgend das COMAL Programm hierzu:



## TOP-DOWN-Programm

[hausmi2.cml]

```
0010 USE turtle
0020 pendown; nowrap; hideturtle
0030 fullscreen
0050 bild
0060 END
0070
```

```

0080 PROC bild
0090  grundlinie
0100  haus
0110  garage
0120  baum
0130  wege
0140  zaun
0145  farbe
0150 ENDPROC bild
0160
0170 PROC grundlinie
0180  moveto(-160,-50)
0190  right(90)
0200  forward(320)
0210  left(90)
0220 ENDPROC grundlinie
0230
0240 PROC haus
0250  umriß
0260  dachgaube(-58,16)
0270  fensterunten
0280  tür
0290  schornstein
0300 ENDPROC haus
0310
0320 PROC garage
0330  moveto(80,-50)
0340  rechteck(70,50)
0345  färben(xcor+1,ycor+1,7)
0350  moveto(90,-50)

0360  LOOP 10 TIMES    // Garagentor
0370    rechteck(5,40)
0375    färben(xcor+1,ycor+1,8)
0380    right(90)
0390    forward(5)
0400    left(90)
0410  ENDLOOP
0420 ENDPROC garage
0430
0440 PROC baum
0450  moveto(-125,-50)
0460  forward(40)
0470  moveto(-115,-50)
0480  forward(40)
0490  moveto(-120,-10)
0500  right(90)
0510  arc1(20,360)
0520  left(90)
0530 ENDPROC baum
0540
0550 PROC wege
0560  left(150)
0570  moveto(-20,-50)
0580  forward(70)
0590  moveto(20,-50)
0600  forward(70)
0610  moveto(90,-50)
0620  forward(70)
0630  moveto(140,-50)
0640  forward(70)
0650  right(150)
0660 ENDPROC wege
0670
0680 PROC zaun

```

```

0690  moveto(-85,-50)
0700  LOOP 16 TIMES
0710    zaunpfahl
0720    left(90)
0730    forward(5)
0740    right(90)
0750  ENDLOOP
0760 ENDPROC zaun
0770
0780 PROC umriß
0790  moveto(-90,10)
0800  right(30)
0810  forward(60)
0820  right(60)
0830  forward(50)
0840  moveto(10,ycor)
0850  forward(50)
0860  right(60)
0870  forward(60)
0880  right(120)
0890  forward(180)
0900  right(90)
0910  moveto(-80,-50)
0920  forward(60)
0930  moveto(80,-50)
0940  forward(60)
0950 ENDPROC umriß
0960
0970 PROC dachgaube(posx,posy)
0980  moveto(posx,posy)
0990  right(45)
1000  forward(27)
1010  arc1(20,45)
1020  forward(50)
1030  arc1(20,45)
1040  forward(27)
1050  left(135)
1051  moveto(posx+28,posy); fenster
1052  moveto(posx+68,posy); fenster
1060 ENDPROC dachgaube
1070
1080 PROC fensterunten
1090  moveto(-60,-30); fenster
1100  moveto(40,-30); fenster
1130 ENDPROC fensterunten
1140
1150 PROC tür
1160  moveto(-20,-30); fenster
1170  moveto(-20,-50); fenster
1180  moveto(0,-30); fenster
1190  moveto(0,-50); fenster
1200 ENDPROC tür
1210
1220 PROC schornstein
1230  moveto(-10,60)
1240  rechteck(20,10)
1250 ENDPROC schornstein
1260
1270 PROC zaunpfahl
1280  forward(22)
1290  right(30)
1300  forward(5)
1310  right(120)
1320  forward(5)
1330  right(30)

```

```

1340 forward(22)
1350 right(90)
1360 forward(5)
1370 right(90)
1375 färben(xcor+1,ycor+1,7)
1380 ENDPROC zaunpfahl
1390
1400
1410 PROC rechteck(br,hö)
1420 LOOP 2 TIMES
1430 fd(hö); rt(90); fd(br); rt(90)
1440 ENDLOOP
1450 ENDPROC rechteck
1460
1470 PROC fenster
1480 rechteck(20,20)
1482 färben(xcor+1,ycor+1,8)
1485 move(0,10); draw(20,0)
1486 move(-10,10); draw(0,-20)
1490 ENDPROC fenster

```

```

1500
1510 PROC farbe
1520 färben(-100,-70,10) // Rasen links
1530 färben(20,-95,10) // Rasen Mitte
1540 färben(-120,0,10) // Baumkrone
1550 färben(155,-90,10) // Rasen rechts
1560 färben(0,95,9) // Himmel
1570 färben(0,0,7) // Haus
1572 färben(-120,-15,6) // Baumstamm
1573 färben(0,-55,14) // Weg zur Haustür
1574 färben(120,-55,14) // Weg zur Garage
1575 färben(0,50,6) // Dach
1576 färben(0,63,5) // Schornstein
1580 ENDPROC farbe
1590
1600 PROC färben(xpos,ypos,farbe)
1610 pencolor(farbe) // nur noch
1620 fill(xpos,ypos) // COMAL-
1630 pencolor(15) // Anweisungen
1640 ENDPROC färben

```