



Hardware und Software

Hardware:

Alle Computer bestehen aus folgenden Grundbausteinen:

Eingabegeräte (Tastatur, Maus, Scanner, Joystick,)

Zentraleinheit (Rechenwerk, Steuerwerk, Arbeitsspeicher, ...)

Ausgabegeräte (Monitor, Drucker, Interface,)

Neben dem Hauptspeicher der Zentraleinheit (Arbeitsspeicher) benötigt man **Speicher**, mit deren Hilfe man Informationen (Daten) sichern kann:

USB-Stick mit USB-Port

Festplatte mit Festplattenlaufwerk

Diskette mit Diskettenlaufwerk

CDs mit CD-ROM-Laufwerk

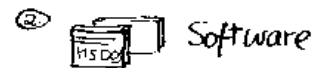


Man unterscheidet 2 Arten von Speichern:

Nur-Lese-Speicher : ROM (Read Only Memory)

z. B. für das Steuerprogramm des Computers, CDs.

Der Speicherinhalt lässt sich nicht verändern



Schreib-Lese-Speicher : RAM (Random Access Memory)

Hauptspeicher, Disketten, USB-Stick, Festplatte.

Daten können gelöscht und geändert werden.



Software:

Was bisher beschrieben wurde, sind die Geräte, die so genannte **Hardware**. Mit der Hardware eines Computers alleine kann man nichts anfangen. Damit ein Computer arbeiten kann, benötigt man **Software**. Software ist das Programm, die Arbeitsmethode. **Anwenderprogramme** (z. B. Textverarbeitungsprogramme) sind Programme, die der Benutzer für festgelegte Aufgaben benutzen kann (Schreiben, Zeichnen, Spiele, ...). Wenn du den Computer für eigene Aufgaben (Lernprogramme, Berechnungen, Spiele, ..) benutzen willst, brauchst du eine **Programmiersprache**, die diese Aufgaben dem Computer verständlich macht. Computer verstehen nur diese eigens für sie geschaffenen **Computersprachen**.

Es gibt zahlreiche Programmiersprachen: Comal, Logo, BASIC, TURBOPASCAL, C++, Delphi, Java und viele mehr.

Du benutzt in diesem Kurs sogenannte Turtle-Befehle der Programmiersprache **COMAL** und **LOGO**. COMAL wurde 1973 in Dänemark entwickelt und ist sehr einfach zu erlernen.

Beim Programmieren erfährst du, dass der Computer ein „nimmermüdes Arbeitstier“ ist und bekommst eine Vorstellung, wie Profiprogramme entstehen.

Mit deinen Programmen kannst du Bilder erstellen, die du mit anderen Programmen oder von Hand nicht oder nur mühsam zeichnen könntest.



Zeichnen mit MAC-TURTLE

In diesem Kurs wollen wir mit dem Zeichnen beginnen. Wie bei jeder Sprache musst du auch bei einer Computersprache bestimmte Vokabeln (Befehle) und Regeln (Syntax) erlernen.

DirektModus:

Im Direktmodus kannst du sofort die Wirkung des Befehls beobachten. Gleichzeitig wird jeder Befehl in einer Programmliste mitgeschrieben.

ProgrammModus:

Die Befehle werden in **einzelne** Zeilen geschrieben und mit Klick auf <RUN> ausgeführt. Die Befehle werden in der Reihenfolge einzeln in der Reihenfolge (sequentiell) ausgeführt. Du kannst die Programmliste beliebig verändern.

Mit *Datei-Speichern* kannst du deine Befehlszeilen speichern

Mit *Datei-Öffnen* kannst du gespeicherte Programme öffnen und weiterbearbeiten.

Relative Befehle:

Die Befehle ***forward(x)***, ***back(x)***, ***left(x)***, ***right(x)***, ***kreis(x)***, ***bogenr(2,90)***, ***bogenl (2,90)*** und ***schreib(Text)*** bezeichnet man als relative Befehle, weil der Zeichenstift immer vom letzten Standort weiterzeichnet.

Absolute Befehle:

Die Befehle ***richtung(x)***, ***drawto(x,y)***, ***richtungXY(x,y)*** und ***moveto(x,y)*** bezeichnet man als absolute Befehle, weil der Zeichenstift unabhängig vom letzten Standort bewegt wird.

Bitte beachte: Dezimalbrüche (Kommazahlen) musst du immer mit Dezimalpunkt schreiben!

Beispiel: nicht 4,7 sondern 4.7

Wiederholungen (Schleifen):

Alle Befehle, die zwischen ***repeat(10)*** ***end*** stehen, werden wiederholt. Die Anzahl der Wiederholungen wird in der Klammer angegeben. Es sind auch geschachtelte Wiederholungen möglich (siehe Beispiele).

Prozeduren:

Ein gut strukturiertes Programm besteht aus einzelnen Prozeduren die bestimmte Teilaufgaben ausführen. Befehle in Prozeduren werden nur ausgeführt, wenn die Prozedur mit dem Prozedurnamen aufgerufen wird. Mit Prozeduren kann man also neue Befehle programmieren.

Der **Beginn einer Prozedur** besteht aus dem Befehl ***proc*** gefolgt von einem beliebigen ***namen*** (z.B. *proc quadrat*). Der Name darf nicht mit bestehenden Befehlsnamen identisch sein (*kreis*, *forward* o.ä.).

Das Ende der Prozedur besteht aus dem Befehl ***endproc***.

Prozeduren können auch aus anderen Prozeduren heraus aufgerufen werden (siehe Beispiele).

Einschränkungen:

Bei MacTurtle handelt es sich um eine Programmierumgebung, die in die Grundlagen des Programmierens einführen soll.

Es sind nur einfach geschachtelte Wiederholungen möglich, es gibt keine geschachtelten Prozeduren. Es gibt keine Verzweigungen (*if* *then* *else*).

**Befehle für den Programmmodus von MacTurtle**

Befehl	Abkürzung	Bedeutung
forward(4) vor(4)*	fd(4)	die Turtle (Malstift) bewegt sich 4 cm (Einheiten) vorwärts u. zeichnet
back(3), zurück(3)*	bk(3)	die Turtle bewegt sich rückwärts und zeichnet
right(90) rechts(90)*, dreh(90)*	rt(90)	dreht die Turtle um 90° weiter nach rechts
left(60), links(60)*, dreh(-60)*	lt(60)	dreht die Turtle um 60° weiter nach links
kreis(3)*	-----	zeichnet einen Kreis mit dem Radius 3 cm (Einheiten)
arcr(2,90), bogenr(2,90)*	br(2,90)*	zeichnet einen Kreisbogen nach rechts mit einem Radius von 2 Einheiten und einem Winkel von 90°
arcl(2,90), bogenl(2,90)*	bl(2,90)*	zeichnet einen Kreisbogen nach links mit einem Radius von 2 Einheiten und einem Winkel von 90°
schreib(Hallo)*		Schreibt Hallo an der aktuellen Turtleposition in der Turtlerichtung
plot(5,4), punkt(5,4)*	-----	zeichnet einen Punkt (Kreuz) auf die Koordinate (5 4)
moveto(2.5,6.3)	mt(2.5,6.3)*	setzt die Turtle auf den Punkt (2.5 6.3) ohne zu zeichnen
drawto(2,5)	dt(2,5)*	zeichnet eine Linie zum Punkt (2 5)
setheading(270), richtung(270)*, richtungXY(3,4)*	seth(270)	dreht die Turtle auf 270° (West), unabhängig von dem vorher eingestellten Winkel. die Turtle zeigt auf den Punkt (3 4)
repeat(10), wiederhole(10)*	-----	die folgenden Befehle (bis ende) werden 10 mal wiederholt
end, ende*	-----	ende der Wiederholung
clear	cl*	löscht die Zeichenfläche (Radiergummi)
penup	pu	die Turtle wird hochgenommen. (Sie bewegt sich danach ohne zu zeichnen)
pendown	pd	die Turtle wird heruntergesetzt. Sie zeichnet wieder.
pencolor(yellow)	pc(zufall)	die Turtle zeichnet mit der angegebenen Farbe. Es können auch Zahlen von 0 bis 15 (oder 0-16 Mio). pc(zufall): es wird eine Farbe zufällig ausgewählt.
backgroundcolor(red)	bc(11)	die Hintergrundfarbe wird festgelegt. Es können auch Zahlen von 0 bis 15 (oder 0-16 Mio)
penwidth(3)	pw(3)	Setzt Linienbreite auf 3 Pixel
fill	----	färbt geschlossene Fläche mit aktueller Hintergrundfarbe aus. (aktuelle Turtleposition, Linienbegrenzung: aktuelle Vordergrundfarbe)
fill(red) oder fill(9)		färbt geschlossene Fläche mit angegebener Farbe.
fill(x,y)	----	färbt geschlossene Fläche mit aktueller Hintergrundfarbe an der Position x,y aus
fill(x,y,farbe)	----	färbt geschlossene Fläche mit gewählter Hintergrundfarbe an der Position x,y aus
hideturtle	ht	Turtle wird unsichtbar
showturtle	st	Turtle wird sichtbar
q1 (q2,q3, q4, q1234)		wählt den 1. (2., 3., 4. oder alle) Quadranten als Zeichenfläche
löschen		löscht die Zeichenfläche

Variable

a:=2	----	weist der Variablen a den Wert 2 zu. (Achte auf den Doppelpunkt!)
fd(3*a)	----	zeichnet eine Linie mit 6 Einheiten
a:+3 oder a:-3	----	vergrößert (verkleinert) den Wert der Variablen a um 3 (-> a=5 ; a=-1)
x:=a^3	----	weist der Variablen x den Wert 8 zu
z:=random(1,6) z:=zufall(1,6)	z:=rnd(1,6)	weist der Variablen z eine Zufallszahl zwischen 1 und 6 zu (würfeln)
y:=3x+2		es können beliebige Formeln als Zuweisung gewählt werden

Prozeduren

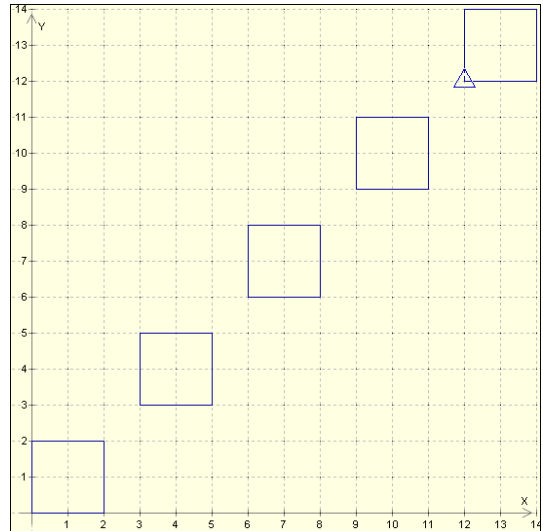
proc quadrat		Beginn einer Prozedur (Teilprogramm) mit dem Namen 'quadrat'
endproc		Ende der Prozedur 'quadrat' (Geschachtelte Prozeduren sind nicht möglich)
quadrat		Die Prozedur mit dem Namen 'quadrat' wird ausgeführt



Programmbeispiele I

//Qadaratmuster

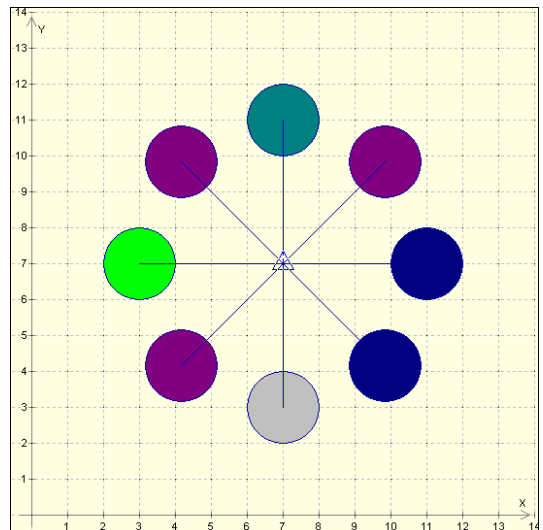
```
q1
x:=0
y:=0
repeat(5)
  mt(x,y)
  repeat(4)
    fd(2)
    rt(90)
  end
  x:+3
  y:+3
end
```



// Prozeduren
Schildmühle

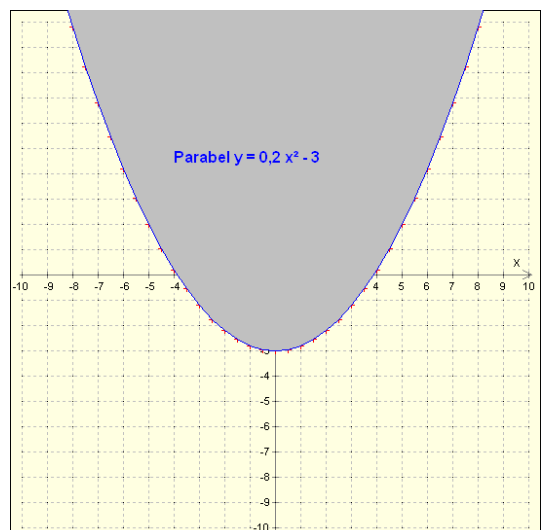
```
proc Schild
  fd(4)
  kreis(1)
  farbe:=zufall(1,10)
  fill(farbe)
  bk(4)
endproc
```

```
proc Schildmühle
  n:=zufall(4,20)
  repeat(n)
    Schild
    rt(360/n)
  end
endproc
```



//Quadratische Funktion

```
q1234 //alle Quadranten
x:=-10
repeat(40)
  plot(x,0.2x^2-3)
  x:+0.5
  drawto(x,0.2x^2-3)
end
fill(1,1,silver)
rt(90)
schreib(-4,5,Parabel: y = 0,2 x^2 - 3)
```





Programmbeispiele II

<pre>löschen x:=0 y:=0 seite:=2 anzahl:=5 siedlung ht proc quadrat repeat(4) fd(Seite) rt(90) end endproc proc dreieck repeat(3) fd(Seite) rt(120) end endproc proc baum pw(seite*10) pc(brown) fd(seite) pw(1) pc(green) lt(90) bogenr(seite/3,360) rt(90) pu fd(seite/3/2) fill(green) bk(seite/3/2) bk(seite) pd pc(Darkblue) endproc</pre>	<pre>proc haus quadrat pu fd(seite) pd rt(30) dreieck lt(30) pu bk(seite) pd endproc proc Straße repeat(Anzahl) mt(x,y) haus pu lt(90) fd(seite/4) rt(90) pd baum lt(90) pu bk(seite/4) rt(90) x:=x+seite+seite/1.5 y:=y+seite+seite/2 end endproc proc Siedlung repeat(10) seite:=zufall(1,20)/10 anzahl:=1 straße end endproc</pre>	
--	---	--



Programme mit Wiederholungen

Schreibe Programme, die folgende Bilder zeichnen:
