

i n p u t f o r ... n e x t
i f ... t h e n ... e n d i f

PROGRAMMIEREN MIT QBASIC

c l s g o t o p r i n t
 r n d a \$ =
 e n d r u n

Themenwochenkurs für Schülerinnen und
Schüler des 7. - 8. Schuljahrs

Inhalt

Seite

1. Manual.....	3
2. Aufgaben.....	7
3. Arbeitsblätter.....	11
3.1 Befehle.....	11
3.2 Farbcode.....	12
3.3 Zahlen runden.....	13
3.4 Würfelzahlen.....	14
3.5 Play-Befehl – Musik erzeugen.....	15
3.6 Draw – Grafiken erzeugen.....	16

1. Manual

Basic (Beginners all-purpose symbolic instruction code) ist eine Programmiersprache, welche in vielen Dialekten angeboten wird. Weil die meisten Schlüsselbegriffe dem gebräuchlichen Englisch sehr nahe kommen, ist Basic einfach zu lernen. Wir verwenden "QBasic 4.5", das kostenlos vom Internet herunter geladen werden darf und als deutsche Version vorliegt.

Die Internet-Adresse "<http://www.antonis.de>" ist eine wahre Fundgrube für QBasic (zum Offline-Surfen für 5€ auch als CD-Rom erhältlich). QBasic-Fans treffen sich auf der Seite "<http://www.qbasic.de>"

QBasic wird mit der Datei "QB.EXE" gestartet. Es erscheint ein zweiteiliges Arbeitsfenster mit Menuleiste, Programmierfeld (für die Eingabe oder Bearbeitung von Programmen) und Direktfeld (zum unmittelbaren Ausprobieren einzelner Befehle).

Ein Programm ist eine Liste von Befehlen, welche der Computer der Reihe nach erledigt. Wichtig ist, dass die Wörter richtig geschrieben und dass "Kleinigkeiten" wie Punkte, Kommas, Anführungszeichen usw. genau beachtet werden. Auf Gross- und Kleinschreibung kommt es hingegen nicht an, QBasic setzt alle Anweisungen automatisch in Grossbuchstaben um. Normalerweise brauchen wir für jeden Befehl eine Zeile (kommen mehrere Befehle auf eine Zeile, so müssen sie mit Doppelpunkt getrennt werden).

Basic-Programme erhalten Namen (bis zu 8 Buchstaben) mit dem Suffix .BAS (z. Bsp. RECHTECK.BAS). Nach dem Laden können sie mit der Tastenkombination [Umschalttaste]+[F5] gestartet werden. Ein laufendes Programm lässt sich (notfalls) mit [Ctrl]+[Pause/Break] beenden!

QBasic bietet eine enorme Anzahl verschiedener Befehle an, die wir zu Programmen zusammenstellen können. Nachfolgend eine Auswahl (wer mehr wissen will, kann auch im Menu "Hilfe" nachschauen):

Allgemeines

:	Trennt Befehle, die auf der gleichen Zeile stehen.
CLEAR	Setzt alle Variablen auf 0 beziehungsweise "".
CLS	Leert den Bildschirm (clear screen).
REM	Bemerkungen (remarks), werden vom Computer nicht beachtet.
END	Beendet das Programm.

Variablen

Variablen bezeichnen Speicherplätze entweder für Zahlen oder für Buchstaben. Sie werden in der Regel mit Buchstaben oder Wörtern bezeichnet.

x	Zahlenvariablen (für Dezimalbrüche mit max. 7 Stellen):	
	y, z1, Zins, Resultat2, umfang,...	
x\$	Stringvariablen (für Zeichenketten, Wörter) müssen auf \$ enden:	
	y\$, z1\$, Name\$, ausgabe\$,...	
LET	Zuweisung	
	LET x = 4.5	Der Variablen x wird der Wert 4,5 zugewiesen.
	LET y = y+1	Der Inhalt des Speichers y wird um 1 erhöht.
	LET a\$ = "ja"	Das Wort "ja" wird gespeichert.

Für Zuweisungen (aber auch für Bildschirm-Ausgaben) muss Text zwischen Anführungs- und Schlusszeichen gesetzt werden.

Hinweis: Der LET-Befehl ist fakultativ, das heisst "x=5" bedeutet dasselbe wie "LET x=5".

Mathematische Operationen und Funktionen

+ - * / Grundoperationen (plus, minus, mal, durch)
^ Potenz (hoch)
SQR(x) Quadratwurzel von x
INT(x) Integerfunktion: liefert die nächst kleinere ganze Zahl von x
Dezimalbrüche müssen mit Punkt statt Komma eingegeben werden!

Vergleiche

=	gleich	<>	ungleich
<	kleiner	<=	kleiner oder gleich (höchstens)
>	grösser	>=	grösser oder gleich (mindestens)

Zeiten und Zufall

SLEEP x Unterbricht das Programm für x Sekunden (das Warten kann durch Betätigung irgend einer Taste vorzeitig abgebrochen werden).

TIMER Zeigt die seit Mitternacht verfloßenen Sekunden an (die Systemuhr macht Schritte von etwa 0,056 s).

Computer können keine echten Zufallszahlen liefern, sie produzieren nur eine recht unregelmässige Zahlenfolge (Pseudo-Zufallszahlen). Mit dem Befehl RANDOMIZE TIMER wird mindestens mit Hilfe der Systemuhr der Einstieg in diese fixe Produktion zufällig gewählt.

RND Gibt eine Zufallszahl zwischen 0 und 0,9999999
LET x = INT(RND*6)+1 Erzeugt Würfelzahlen!

Ein- und Ausgaben

INPUT Tastatur-Eingabe (wobei das Programm unterbrochen wird)
INPUT x Der Computer erwartet eine Zahleneingabe.
INPUT x\$ Der Computer erwartet eine Texteingabe.
INPUT-Anweisungen können auch mit Text versehen werden:
INPUT "Gib eine Zahl ein:", x (Statt Komma kann auch ein Semi kolon gesetzt werden: ergibt ein Fragezeichen.)

INKEY Liest ein Zeichen von der Tastatur (ohne Programmunterbruch).

PRINT Bildschirm-Ausgabe
PRINT x Der Wert von x erscheint auf dem Bildschirm
PRINT "Wow" Das Wort "Wow" erscheint auf dem Bildschirm
PRINT x\$ Die in x\$ gespeicherte Zeichenkette erscheint.
PRINT Nichts erscheint: erzeugt eine Leerzeile.
Zwischen aufeinander folgende PRINT-Befehle (oder nach PRINT-Befehlen) können Kommas oder Semi kolons gesetzt werden:
PRINT x, : PRINT y vereinfacht: PRINT x,y
PRINT x; : PRINT y vereinfacht: PRINT x;y

Text-Bildschirm

Der normale Bildschirm für Textausgabe (Screen 0) ist in 25 Zeilen zu je 80 Spalten unterteilt.

PRINT TAB(x) Die Ausgabe erfolgt ab der x. Spalte der aktuellen Zeile.

PRINT TAB(25) "Das gibt ", x

LOCATE x, y Der Cursor springt in die x. Zeile und dort an die y. Spalte

LOCATE 5, 15

Bildschirm und Schrift lassen sich auch anders färben (als Code dienen die Zahlen von 0 bis 15):

COLOR x, y x gibt die Farbe des Texts an (Vordergrund), y gibt die Farbe des Bildschirms an (Hintergrund).

COLOR 4, 1 Erzeugt rote Schrift auf blauem Grund.

Sprünge

Für Sprünge im Programmablauf müssen Marken gesetzt werden, entweder eine Zahl oder ein Wort mit Doppelpunkt auf einer Programmzeile.

GOTO Sprungbefehl (der Ablauf wird nach der angegebenen Marke fortgesetzt):

GOTO 100 Das Programm springt zur Marke "100" und fährt dort weiter.

Bedingungen

Oft wird die Ausführung eines Programnteils von Bedingungen abhängig gemacht:

IF THEN Bedingter Befehl: nur wenn ... zutrifft, soll der nachfolgende Befehl ausgeführt werden.

IF x < 1 THEN y = y+1

IF antwort\$ = "n" THEN PRINT "Tschüss"

IF a > 100 THEN GOTO Ziel

Bedingte Befehle können mehrere Zeilen lang sein. Sie werden dann mit ENDIF abgeschlossen:

IF ... THEN

...

...

ENDIF

Zu IF...THEN gehört noch ELSE (andernfalls). Damit lassen sich Programmverzweigungen realisieren:

IF ... THEN

...

ELSE

...

ENDIF

Schleifen

Eine Schleife ist die Wiederholung eines Programnteils. Mit den folgenden Befehlen lassen sich Schleifen programmieren:

WHILE WEND so lange als ... zutrifft wird der Programnteil bis zum Schleifenende WEND wiederholt

WHILE a\$ = "j"

...

WEND

Will man die Anzahl der Schleifen zum Voraus festsetzen, verwendet man:

FOR NEXT Mit jedem Durchgang wird der Zähler n um 1 erhöht bis zum Zielwert. Dann geht das Programm bei der Zeile nach NEXT weiter.

FOR n = 1 TO 45

...

NEXT n

In FOR-NEXT-Schleifen können auch andere Schritte verlangt werden:

FOR n=0 TO 100 STEP 2 zählt 0, 2, 4, 6, ..., 100

FOR n=100 TO 0 STEP -1 zählt 100, 99, 98, 97, ..., 0

EXIT FOR Erlaubt, die Schleife vorzeitig zu verlassen.

Beachte: Zu jedem WHILE gehört ein WEND, zu jedem FOR ein NEXT!

Töne

SOUND x, y Erzeugt einen Ton, wobei x die Tonhöhe (Frequenz in Hz) angibt und y die Dauer (Anzahl Zeiteinheiten von 56 ms) des Tons bestimmt.

Weil SOUND über Zahlenwerte gesteuert wird, lassen sich durch Veränderungen (Berechnungen) verschiedenste Effekte erzielen. Für Musik bietet sich hingegen PLAY an, weil die Töne wie bei einem Klavier gegeben sind:

PLAY a\$ durch die Zeichenkette a\$ wird die Tonfolge bestimmt:

Oktave (01, 02, ...)

Länge (L1 ganze Note, L2 halbe Note, L4 Viertelnote, ...)

Höhe (Töne der Oktave: C, D, E, F, G, A, B) mit Halbtonschritten + und - (C+ für cis, D- für des ...)

PLAY "01 L4 CDEF L2 GG"

Der PLAY-Befehl kann weiter mit Pausen (P1 ganze Pause, P2 halbe Pause, P4 Viertelpause, ...), Spielart (legato, staccato) und Tempo ausgeschmückt werden!

Grafik

Um zeichnen zu können, muss zuerst ein Grafik-Bildschirm erstellt werden (es gibt verschiedene), der Text-Bildschirm (Screen 0) ist nicht grafikfähig.

SCREEN x Wechselt zum gewünschten Bildschirm
SCREEN 0 Text
SCREEN 9 Grafik

Im Grafikmodus ist die Fläche in Pixel (Punkte) unterteilt: Screen 9 ist 640 Pixel lang und 350 Pixel breit. Jede Stelle kann (ähnlich wie im Koordinatensystem) durch zwei natürliche Zahlen angegeben werden, allerdings ist der Nullpunkt oben links!

0, 0 linke obere Ecke des Grafik-Bildschirms
320, 175 Bildschirmmitte
640, 350 rechte untere Ecke des Grafik-Bildschirms

Für Textausgaben stehen in Screen 9 weiterhin 15 Zeilen zu je 80 Spalten zur Verfügung. Er bietet zudem 16 Farben an, die sich über die Zahlenwerte von 0 bis 15 einstellen lassen.

PSET Zeichnet Punkte, wobei natürlich noch Stelle und Farbe angegeben werden müssen:
PSET (320, 175), 4 ergibt einen roten Punkt in der Mitte des Bildschirms.

LINE Zeichnet Linien oder Rechtecke:
LINE (0, 0) - (640, 350), 2 ergibt eine grüne Diagonale über den Bildschirm (von oben links nach unten rechts).
LINE (50, 100) - (100, 200), 2, B ergibt ein grünes Rechteck mit (50, 100) als obere linke Ecke und (100, 200) als untere rechte (B steht für Box).
LINE (50, 100) - (100, 200), 2, BF ergibt das gleiche Rechteck ganz grün gefüllt (BF steht für "Box Filled").

CIRCLE Zeichnet Kreise und Ellipsen:
CIRCLE (320, 175), 50, 4 ergibt einen roten Kreis mit einem Radius von 50 Pixeln in der Mitte des Bildschirms.
CIRCLE (590, 300), 50, 4, , , 0.8 ergibt in der rechten unteren Ecke des Bildschirms eine rote Ellipse mit einem grossen Radius (horizontal) von 50 Pixeln und einem kleinen (vertikal) von $50 \cdot 0,8 = 40$ Pixeln.
Mit dem PAINT-Befehl lässt sich ein vorher gezeichneter Kreis ganz färben:
CIRCLE (200, 100), 50, 2 Ergibt einen grün gefüllten Kreis
PAINT (200, 100), 2 mit $r = 50$ und $M(200/100)$.

DRAW Zeichnet Polygonzug, der über eine Zeichenkette gesteuert wird:
Mx, y Bewegt den Zeichenstift zu (x/y)
B Hebt den Stift (Präfix)
Cx Zeichenfarbe
Richtungen: Mit den Buchstaben L, R, U, D, E, F, G, H und einer Zahl wird angegeben, wie viele Pixel der Zeichenstift in welche Richtung gehen soll
 H U E
 L • R
 G D F
Px, y Füll- und Randfarbe
DRAW "C4 BM200, 50 R50 U30 M200, 50" zeichnet ein Dreieck mit den Ecken (200/50), (250/50) und (250/20).

Empfehlenswerte Literatur

Thomas Antoni: Das QBasic-Kochbuch

Wilfried Furrer: QBasic

Claus-Dieter Volko: The real Adok's way to QBasic

Die Scripts befinden sich auf der QBasic CD-Rom von www.antoni.s.de.

2. Aufgaben

1. Mit dem folgenden Programm kannst du die Potenz zweier Zahlen berechnen:

```
INPUT a
INPUT b
LET c = a^b
PRINT c
END
```

Versuchen wir schrittweise, die Bedienerfreundlichkeit zu heben (du kannst das alte Programm ergänzen):

```
CLS
INPUT "Wie heisst die Basis "; a
INPUT "Wie heisst der Exponent "; b
LET c = a^b
PRINT
PRINT "Der Potenzwert beträgt: ", c
END
```

Damit wir das Programm nicht nach jeder Rechnung wieder neu starten müssen, packen wir es in eine Schleife ein:

```
w$ = "j"
WHILE w$ = "j"
CLS
INPUT "Wie heisst die Basis "; a
INPUT "Wie heisst der Exponent "; b
LET c = a^b
PRINT
PRINT "Der Potenzwert beträgt: ", c
PRINT
INPUT "Noch eine Aufgabe (j/n) "; w$
WEND
END
```

Zuletzt fügen wir noch ein paar Bemerkungen bei und machen, dass sich das Programm am Schluss verabschiedet:

```
REM "POTENZ. BAS"
REM Berechnung von Potenzen
REM Februar 2005
w$ = "j"
WHILE w$ = "j"
CLS
INPUT "Wie heisst die Basis "; a
INPUT "Wie heisst der Exponent "; b
LET c = a^b
PRINT
PRINT "Der Potenzwert beträgt: ", c
PRINT
INPUT "Noch eine Aufgabe (j/n) "; w$
WEND
PRINT
PRINT "Tschüss !"
END
```

Markiere bei jeder Version, was neu dazu gekommen ist. Verstehst du alle Befehle?

Speichere die letzte Version. Wie wird ein Programm gespeichert, wie wird es geladen?

2. Ein Programm zu Rechtecksberechnungen: Schreibe das Programm ab, speichere es dann unter dem angegebenen Namen und starte es.

```
REM "RECHTECK. BAS"
REM Februar 2005
CLS : PRINT
```

```

PRINT "Rechtecksberechnungen"
PRINT "*****"
PRINT
INPUT "Wie lang (in cm) ist das Rechteck "; a
INPUT "Wie breit (in cm) ist das Rechteck "; b
u = (a+b)*2
f = a*b
PRINT
PRINT "Umfang = "; u; "cm"
PRINT "Fläche = "; f; "cm2"
END

```

Ändere das Programm "RECHTECK.BAS" so ab, dass der Benutzer gefragt wird, ob er noch eine weitere Aufgabe lösen wolle. Wenn ja, dann werden neue Seiteneingaben gefordert.

Ausbau: Wenn wir von Länge, Breite, Umfang und Fläche zwei Grössen eingeben, dann werden die andern beiden berechnet.

3. Schreibe ein Programm für Kreisberechnungen: Eingabe des Radius ergibt Umfang und Fläche. Zusatz: Sektorberechnungen
4. Umrechnungen von Franken in Euro (und umgekehrt). Der Benutzer wird zunächst nach dem aktuellen Kurs gefragt und dann, ob er Franken in Euro oder Euro in Franken umrechnen wolle. Je nach Antwort wird ein Franken- oder Eurobetrag verlangt und in die andere Währung umgerechnet.
5. In den USA wird die Temperatur in Fahrenheit angegeben, bei uns in Celsius. Es gilt:

$0^{\circ}\text{C} = 32^{\circ}\text{F}$ Schmelzpunkt von Wasser
 $100^{\circ}\text{C} = 212^{\circ}\text{F}$ Siedepunkt von Wasser

 Entwickle ein Programm für die Umwandlung von Celsius in Fahrenheit (und umgekehrt).
6. Runden:
 - a) Was bewirkt die INT-Funktion? Probiere sie aus!
 - b) Vergleiche $\text{INT}(x)$ mit $\text{INT}(x+0.5)$.
 - c) Was bewirkt die Funktion $\text{INT}(x*10)/10$?
 - d) Was bewirkt die Funktion $\text{INT}(x*10+0.5)/10$?
 - e) Wie wird eine Zahl auf Zehntel, Hundertstel, Tausendstel,... gerundet?
 - f) Wie könnte ein Betrag auf 5 Rp. genau gerundet werden?
7. Entwirf ein kleines Programm, das
 - a) die geraden Zahlen von 0 bis 100
 - b) die ungeraden Zahlen < 50
 - c) die Zahlen der Zwölferreihe
 - d) die Quadratzahlen bis 15^2
 produziert.
8. Programmierkniiffe:
 - a) Wie kann ich prüfen, ob eine Division ganzzahlig aufgeht?
 - b) Wie produziere ich eine Zufallszahl zwischen 0 und n?
 - c) Wie produziere ich eine Zufallszahl zwischen m und n?
 - d) Wie lässt sich ein Programm kurz unterbrechen?
9. Ist 2341 eine Primzahl? Der Computer weiss das zwar auch nicht, aber er kann blitzschnell rechnen und rasch nachschauen, ob die Division durch 2, 3, 4, 5, ... aufgeht. Bis zu welchem Divisor musst du ihn arbeiten lassen? Die Division durch die geraden Zahlen (ausser 2) kann man sich auch sparen!

Ausbau: Bruchkürzungs-Programm
10. Bruchrechnen: Nach Eingabe zweier Brüche und einer Operation liefert der Computer das Resultat (vielleicht sogar gekürzt).

11. Nicht aufgehende Divisionen durch n liefern periodische Dezimalbrüche, wobei die Periode höchstens (n-1) Stellen aufweisen kann. Eine schöne Arbeit für den Computer!
12. Die Kreiszahl Pi lässt sich auf verschiedene Arten bestimmen. Eine interessante Art hat Gottfried Wilhelm Leibniz 1673 gefunden:

$$= 4 * (1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots)$$
 Entwickle ein Programm, das den Pi-Wert aus den ersten n Brüchen ergibt. Hinweis: Verwende statt der üblichen Single-Variablen (für Dezimalbrüche mit 7 Stellen) so genannte Double-Variablen (16 Stellen). Kennzeichnung mit #: x# , y# , z# , ...
13. Zinsrechnen: Nach Eingabe von Kapital, Zinssatz und Zeit (Datum der Einlage, Datum des Abhebens) soll der Zins (auf 5 Rp. gerundet) ausgegeben werden. Vielleicht werden automatisch 35% Verrechnungssteuer abgezogen, falls der Bruttozins Fr. 50.- oder mehr beträgt.
14. TIMER gibt die momentane Anzahl Sekunden an, die seit Mitternacht verfloßen sind. Entwickle ein Programm, das die aktuelle Uhrzeit in h, min und sec angibt. Ausbau: Digital-Uhr
15. Ein guter Würfel sollte alle Augenzahlen gleich häufig zeigen (bei genügend grosser Anzahl Würfe). Wie gut ist der QBasic-Würfel? – Lass den Computer viele Male würfeln und gleichzeitig zählen, wie oft er eine Eins, eine Zwei, ... gewürfelt hat.
16. Rechentrainer: Du wählst den Zahlenbereich (z. Bsp. Zahlen von 1 bis 25) und die Operation, der Computer stellt dir dann eine Serie von Kopfrechnungen und korrigiert deine Resultate.
17. Ein Ratespiel: Der Computer würfelt eine Zahl, der Spieler versucht sie zu erraten. Wie gross ist die Trefferchance (theoretisch, praktisch)?
18. Ein anderes Ratespiel: Der Computer wählt eine Zufallszahl zwischen 0 und n. Der Spieler soll mit möglichst wenigen Versuchen diese Zahl erraten, indem er das mögliche Intervall einschränkt ("Die gesuchte Zahl liegt zwischen ... und ...").
19. Böse Eins: Ein Spieler würfelt ein oder mehrere Male nacheinander und addiert die Augenzahlen. Wenn er eine 1 würfelt, so ist die erhaltene Summe verloren und er muss den Würfel dem Gegner geben. Natürlich darf er den Würfel weitergeben bevor er eine Eins würfelt. Dann kann er die Summe zu seinen bereits gewürfelten Punkten hinzugeben. Gewonnen hat, wer zuerst 100 Punkte erreicht.
 Probiere das Spiel zunächst aus und entwickle dann ein Programm dafür.
20. Das folgende Programm erzeugt "Pseudo-Zufallszahlen":

```

INPUT "Anzahl Werte: "; n
x = TIMER/1000
x = x - INT(x)
FOR i=1 to n
  x = x+1
  x = x^6
  x = x - INT(x)
  PRINT x,
NEXT i
END

```

 Kontrolliere mit der Würfelstatistik (Nr. 15), ob es sich um einen guten Generator von "Zufallszahlen" handelt. Weshalb liefert er sicher keine echten Zufallszahlen?
21. Ein Ton-Experiment:

```

x = 3200 : y = 1
FOR n = 1 TO 50

```

```

      x = x * 0.95
      y = y * 1.04
      SOUND x, y
NEXT n
FOR n = 1 TO 50
  x = x * 1.04
  y = y * 0.95
  SOUND x, y
NEXT n
END

```

Mach eigene Versuche und schreibe das interessanteste Resultat als Programm auf (speichern nicht vergessen!).

22. Hörtest: Bis zu welcher Frequenz hörst du einen Ton? Der Test soll von tiefen Tönen nach oben verlaufen und dann von sehr hohen, unhörbaren Tönen zu tieferen.

23. Bring dem Computer bei, ein Lied zu singen (z. Bsp. "Frère Jacques")!

24. Mit dem folgenden Programm kannst du Klavier spielen:

```

CLS : CLEAR
PRINT "Klavier spielen"
PRINT "-----"
PRINT
PRINT "Drücke die Tasten 1, 2, 3, ... 8"
PRINT "(beenden mit q)"
WHILE t$ <> "q"
  t$ = INKEY$
  IF t$ = "1" THEN PLAY "01 L4 C"
  IF t$ = "2" THEN PLAY "01 L4 D"
  IF t$ = "3" THEN PLAY "01 L4 E"
  IF t$ = "4" THEN PLAY "01 L4 F"
  IF t$ = "5" THEN PLAY "01 L4 G"
  IF t$ = "6" THEN PLAY "01 L4 A"
  IF t$ = "7" THEN PLAY "01 L4 B"
  IF t$ = "8" THEN PLAY "02 L4 C"
WEND
END

```

25. Untersuche den Befehl COLOR: Welche Zahl erzeugt welche Farbe?

26. Ein Roboter geht eine Anzahl Pixel nach rechts, dann eine (andere) Anzahl Pixel nach unten, dann eine gewisse Anzahl Pixel nach links und dann noch etwas nach oben. Diese Bewegung wiederholt er einige Male und zeichnet dabei seine Spur auf den Bildschirm.

27. Bring einen Kreis dazu, quer über den Bildschirm zu laufen! Vielleicht ändert er dabei sogar seine Grösse und er quietscht dazu ...

28. Den Zufall sichtbar machen: Entwickle ein Programm, das
 a) zeigt, wie Schneeflocken auf dem Bildschirm landen! Oder
 b) zeigt, wie es regnet! Oder
 c) auf dem Bildschirm einen wirren Linienzug erzeugt! Oder...

3.1 Befehle

+	-	plus, minus
*	/	mal, durch
^		hoch
=	<>	gleich, ungleich
<	>	kleiner, grösser
<=	>=	höchstens, mindestens
a		Variable für Zahlen
a\$		Variable für Text (Zeichenkette, String)
:		trennt Befehle auf der gleichen Zeile
CIRCLE (a, b), c, d		Kreis mit $M(a/b)$ und Radius c, Farbe d
CIRCLE (a, b), c, d, , , e		Ellipse mit $r_2 = r_1 \cdot e$
CLEAR		leert alle Speicher
CLS		leert den Bildschirm
COLOR a, b		Textfarbe a auf Hintergrundfarbe b
DRAW		zeichnet Polygonzug
END		beendet ein Programm
ENDIF		beendet einen bedingten Befehl
EXIT FOR		Verlassen einer Zählschleife
FOR...NEXT (... STEP)		Schleife (mit gezählten Durchläufen)
GOTO a		Sprung zur Marke a (oder a\$)
IF...THEN (... ELSE) (... END IF)		Befehlsbedingung
INKEY		Tastatureingabe ohne Programmstop
INPUT		Tastatureingabe mit Programmstop
INT(a)		Integerfunktion
LET		Zuweisung (Zahl oder Text in Speicher)
LINE (a, b) - (c, d), e		Linie von (a/b) zu (c/d), Farbe e
LINE (a, b) - (c, d), e, B		Rechteck zwischen (a/b) und (c/d), Farbe e
LINE (a, b) - (c, d), e, BF		gefülltes Rechteck
LOCATE a, b		Cursorsprung nach Zeile a, Spalte b
PAINT (a, b), c		Kreisfüllung mit Farbe c
PLAY a\$		Ton der Tonleiter
PRINT		BildschirmAusgabe
PRINT TAB (a)		BildschirmAusgabe ab Spalte a
PSET (a, b), c		Punkt bei (a/b), Farbe c
RANDOMIZE TIMER		richtet den Zufallszahlengenerator ein
REM		Bemerkung
RND		liefert eine Zufallszahl
SCREEN a		Text- oder Grafikbildschirm
SLEEP a		Unterbruch während a Sekunden
SOUND a, b		Ton mit Frequenz a Hz während $b \cdot 0,056$ Sek
SQR(a)		Quadratwurzel
TIMER		Zeit (Anzahl Sekunden nach Mitternacht)
WHILE...WEND		Schleife (abhängig von Bedingung)

[Ctrl] + [Pause/Break]
 [Umschalttaste] + [F5]

Programm- Abbruch
 Programm- Start

3. 2 Farbcode

Das folgende kleine Programm ermöglicht dir, den Farbcode zu untersuchen:

```
REM "FARBCODE. BAS"
REM Februar 2005
CLS
FOR n = 0 TO 15
  m = n+1
  IF m = 16 THEN m = 0
  COLOR n, m
  PRINT "Das ist Schriftfarbe"; n; "auf Hintergrund"; m
NEXT n
END
```

0	8
1	9
2	10
3	11
4	12
5	13
6	14
7	15

Mit diesem Code kann die Farbe auch bei den grafischen Befehlen PSET, LINE, CIRCLE, PAINT und DRAW bestimmt werden.

3. 3 Zahlen runden

Die Integer-Funktion schneidet bei positiven Dezimalbrüchen einfach den Teil nach dem Komma ab und lässt die ganze Zahl vor dem Komma stehen, sie liefert also die nächst tiefere ganze Zahl. Beim Runden auf Einer muss ab 0,5 aber die nächst höhere ganze Zahl genommen werden. Man erreicht das durch Addition von 0,5 vor dem Abschneiden:

x	INT(x)	x + 0.5	INT(x + 0.5)
1, 214
2, 438
3, 542
4, 751
5, 975

Wenn wir auf Zehntel runden wollen, so nehmen wir das Zehnfache der gegebenen Zahl, runden auf Einer und dividieren das Resultat durch 10:

x	x*10+0.5	INT(x*10+0.5)	INT(x*10+0.5)/10
1, 214
2, 438
3, 542
4, 751
5, 975

Wie lassen sich Dezimalbrüche auf Hundertstel, Tausendstel, ... runden?

3. 4 Würfel zahlen

Bei vielen Spielen muss auch der Computer würfeln können. Mit Hilfe der Zufalls-Funktion RND (random) produzieren wir aus Dezimalbrüchen < 1 die Würfelzahlen 1, 2, 3, 4, 5, 6:

Befehl	Ausgabe	von	bis
RND	Dezimalbrüche	0,00000000	0,99999999
RND*6	Dezimalbrüche	0,00000000	0,99999999
INT(RND*6)	natürliche Zahlen	0	5
INT(RND*6)+1	natürliche Zahlen	1	6

Probiere das folgende Programm aus:

```
CLS
RANDOMIZE TIMER
FOR n=1 TO 600
    w = INT(RND*6)+1
    PRINT w;
NEXT n
END
```

Was unterscheidet einen guten Würfel von einem schlechten? Welche Bedingungen muss die produzierte Zahlenfolge erfüllen, damit sie dem Resultat eines guten Würfels entspricht?

3. 5 PLAY-Befehl – Musik erzeugen

Der PLAY-Befehl wird über eine Zeichenkette (einen String) gesteuert, wobei Normaleinstellungen nicht erwähnt werden müssen. Elemente dieses Befehls-Strings sind:

MF	Musik im Vordergrund: Die Folgebefehle des Programms werden erst erledigt, wenn der PLAY-Befehl ausgeführt ist (Normaleinstellung).
MB	Musik im Hintergrund: Folgebefehle werden bereits während der Ausführung des Play-Befehls erledigt.
01, 02, ... 06	Oktave, aus der die nachfolgenden Noten stammen.
<	Wechsel zur nächst tieferen Oktave.
>	Wechsel zur nächst höheren Oktave.
L1, L2, ... L64	Länge der nachfolgenden Noten: L1 ganze Noten, L2 halbe Noten, L4 Viertelnoten (Normaleinstellung), L8 Achtelnoten usw.
T32, T33, ... T255	Tempo der nachfolgenden Noten: Anzahl Viertelnoten pro Minute (Normaleinstellung: T128)
P1, P2, ... P64	Pause zwischen zwei Noten: P1 ganze Pause, P2 halbe Pause, P4 Viertelpause (Normaleinstellung) usw.
MS	Nachfolgende Noten staccato spielen.
ML	Nachfolgende Noten legato spielen.
MN	Nachfolgende Noten normal spielen (Normaleinstellung).
C, D, E, F, G, A, B	Noten einer Oktave (c, d, e, f, g, a, h)
+	Vorangehende Note um einen Halbtonschritt erhöhen.
-	Vorangehende Note um einen Halbtonschritt herabsetzen.
N0, N1, ... N84	Eine der 85 Noten aus den vorhandenen 7 Oktaven
Beispiel:	PLAY "ML T160 L2 01 GDEC P2 < G > DEC" lässt den Big-Ben-Schlag ertönen.

3. 6 DRAW-Befehl – Grafiken erzeugen

Zum DRAW-Befehl gehört eine Zeichenkette (ähnlich wie zu PLAY). Mit diesem String wird ein gedachter Zeichenstift über den Bildschirm dirigiert, der eine Spur hinterlässt oder auch vom Zeichenblatt abgehoben werden kann. Elemente des Befehls-Strings sind:

B	Hebt den Zeichenstift für die nachfolgende Bewegung.
Mx, y	Bewegt den Zeichenstift zum Punkt (x/y).
Rn	Bewegt den Zeichenstift um n Pixel nach rechts (right).
Ln	Bewegt den Zeichenstift um n Pixel nach links (left).
Un	Bewegt den Zeichenstift um n Pixel nach oben (up).
Dn	Bewegt den Zeichenstift um n Pixel nach unten (down).
En	Bewegt den Zeichenstift um n Pixel nach rechts oben.
Fn	Bewegt den Zeichenstift um n Pixel nach rechts unten.
Gn	Bewegt den Zeichenstift um n Pixel nach links unten.
Hn	Bewegt den Zeichenstift um n Pixel nach links oben.
Schema:	
H U E	
L • R	
G D F	
An	Dreht das Zeichenblatt unter dem Stift um $n \cdot 90^\circ$ im Uhrzeigersinn ($n = 1, 2, 3$).
TAn	Dreht das Zeichenblatt unter dem Stift um n° .
Cn	Bestimmt die Zeichenfarbe ($n = 0, 1, 2, \dots, 15$).
Pm, n	Bestimmt die Füllfarbe m und die Randfarbe n einer Figur.
Beispiel:	DRAW "C2 BM280, 80 R50 F30 D50 G30 L50 H30 U50 E30" zeichnet mit grüner Linie ein Achteck.

---=== Ende ===---