VOSS

DAS SCHULBUCH ZU ATARI®

600XL/800XL

ABBUC EDITION 2022 (p) GBXL

Bearbeitungsstand: 6. Juni 2Ø22

Erstellt mit Open Office 4.1.12 unter Linux Mint 2Ø.3.

Danksagung
Mit Ablauf des Jahres 2Ø13 wurden der Zeitschriften- und der Buchverlag von DATA Becker geschlossen; zum 31. März 2Ø14 wurde der gesamte Geschäftsbetrieb aufgegeben. Es gibt keinen Rechtsnachfolger.

Der Autor *Prof. Dr. Werner Voß* gab auf Bitte des ABBUC e.V. im März 2Ø22 die Zustimmung zur digitalen Aufarbeitung und Veröffentlichung.

Dafür recht herzlichen Dank.

Dank geht ebenfalls an *CharlieChaplin*, der das originale Buch zur Verarbeitung bereitstellte.

Anmerkung
Cover und Layout des Buches entsprechen dem des Originaldrucks von 1984. Trotz notwendiger Korrekturen und Anpassungen konnte das Werk weitestgehend seitengleich digitalisiert werden und erlaubt so den Abgleich zum originalen Buch aus 1984.

Das Buch erschien ursprünglich bei DATA Becker:

ISBN 3-89Ø11-Ø57-6

Copyright (C) 1984 DATA BECKER GmbH
Merowingerstr. 3Ø
4ØØØ Düsseldorf

Re-Edit by GBXL for ABBUC e. V. (p) 2Ø22

Wichtiger Hinweis!

Die in diesem Buch wiedergegebenen Schaltungen, Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden.

Alle Schaltungen, technische Angaben und Programme in diesem Buch wurden von den Autoren mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. DATA BECKER sieht sich deshalb gezwungen, darauf hinzuweisen, dass weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für die Mitteilung eventueller Fehler ist der Autor jederzeit dankbar.

Gliederung Seite
==========

Ø. Vorwort 1

1. Grundelemente der Programmiersprache BASIC 7

1.1 Vorbemerkung 7
1.2 Grundbegriffe, die man kennen sollte 8
1.3 Zur Funktionsweise von Rechnern .. 12
1.4 Problemanalyse 14
1.5 Ergebnisausgabe 17
1.6 Wertzuweisungen 22
1.7 Informationseingabe 26
1.8 Programmverzweigungen 28
1.9 Programmschleifen 32
1.1Ø Die Benutzung externer Speicher .. 34
1.11 Ergänzungen 36

2. Mathematik 36

2.1 Vorbemerkung 39
2.2 Der Satz des Pythagoras 41
2.3 g.g.T. und k.g.V. 5Ø
2.4 Primzahlenprüfung 59
2.5 Quadratische Gleichung 68
2.6 Euler'sche Zahl 74
2.7 Prozentrechnung 8Ø
2.8 Würfel 85
2.9 Probleme der Rechengenauigkeit ... 92

Seite

3. Chemie 95

3.1 Zusätzliche BASIC-Anweisungen .. 95
3.2 Das Wassermolekül 1Ø3
3.3 Reaktionsgleichung 1Ø9
3.4 Stöchiometrisches Rechnen 116
3.5 Das Periodensystem der Elemente 122

4. Physik 131

4.1 Vorbemerkung 131
4.2 Graphik-Programmierung 132
4.3 BASIC-Programme 136
4.4 Die Federwaage 144
4.5 Der Satz des Archimedes 151
4.6 Pendelbewegung 157
4.7 Optische Abbildung 163
4.8 Kräfteparallelogramm 168
4.9 Das Ohm'sche Gesetz 173

5. Sprachen 179

5.1 Vorbemerkung und BASIC-Ergänzungen 179
5.2 Englische unregelmäßige Verben . 182
5.3 Französisch-Vokabeln 189
5.4 Englisch-Vokabeltest 196
5.5 Geheimschrift 2Ø3

6. Biologie/Ökologie 211

6.1 Vorbemerkung 211
6.2 Ungebremstes Wachstum 212
6.3 Gebremstes Wachstum 217

Seite

6.4 Umweltverschmutzung 223

7. Chemie 231

7.1 Vorbemerkung 231
7.2 Historische Jahreszahlen 232
7.3 Die Hauptstädte der Länder 238
7.4 Die Bevölkerungsentwicklung in verschiedenen Nationen 245

8. Wirtschaft 255

8.1 Vorbemerkung 255
8.2 Zinsrechnung 256
8.3 Hypothekentilgung 262
8.4 Arithmetisches Mittel 269
8.5 Häufigkeitsverteilung 273
8.6 Die Wirtschaftskraft der Bundesländer 282

9. Mathematik II 291

9.1 Vorbemerkung 291
9.2 Gitter 293
9.3 Die Gerade 296
9.4 Der Kreis 299
9.5 Die Sinuslinie 303

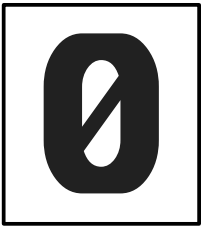
Seite

9.6 Mathem. Tafel 1 306
9.7 Mathem. Tafel 2 309
9.8 Mathem. Tafel 3 313
9.9 Ellipse 316
9.10 Gauß'sche Normalverteilung 320
9.11 Regression 324
9.12 Permutationen 332
9.13 Kombinationen 336
9.14 Lotto 342
9.15 Binomialverteilung 347
9.16 Zentrales Grenzwerttheorem 358
9.17 Wahlen 376

10. Ausblick 381

Stichwortverzeichnis 383

Vorwort



Computer - wie der ATARI - sind nicht nur zum Spielen da. So faszinierend fertige Programme auch sein mögen, - Fußballspiele, Weltraumkriege, Autorennen, Schachpartien, Kämpfe mit seltsamen Ungeheuern usw. - noch aufregender ist es, selbst zu programmieren: Wer zum ersten Mal ein Problem so aufbereitet hat, daß das in den Rechner eingegebene Programm dieses Problem in Sekundenschnelle und ohne Fehler tatsächlich löst, der kann sich freuen. Ein solcher Erfolg nämlich zeigt, daß der Computer sinnvolle Arbeit leisten kann, d.h. er kann uns von der Last mühsamer "Kopfarbeit" befreien - und das ist auch sein wesentlicher Verwendungszweck; dies macht die außerordentlich rasch zunehmende Bedeutung dieser Rechner aus.

In diesem Buch soll deshalb gezeigt werden, wie man als Schüler den Rechner, speziell den ATARI, dazu einsetzen kann, wesentliche Probleme aus dem Bereich des schulischen Alltags zu lösen.

Dies bedeutet also, daß mit diesem Buch vier Ziele gleichzeitig verfolgt werden:

1. Typische Schulprobleme, beispielsweise aus der Mathematik, aus der Physik und aus anderen Fächern sollen aufgegriffen und die Wege zu ihrer Lösung sollen diskutiert werden.

2. Die Beschreibung der Problemlösungswege soll dem Leser zeigen, wie man ganz allgemein Probleme anpacken muß, damit ein Rechner sie dann für uns lösen kann.

3. Der Leser wird erkennen, wie die Programmiersprache BASIC verwendet werden kann, um diejenigen Computerprogramme zu schreiben, die wir zur Lösung der oben genannten Probleme benötigen.

4. Auf diese Weise erhält zum guten Schluß der Leser eine Sammlung von Programmen, die er nutzbringend im Schulalltag verwenden kann.

Es versteht sich, daß der Nutzen dieses Buches für den Leser sich insbesondere dann erschließt, wenn er die vorgeführten Programme selbst am Rechner ausprobiert und wenn er auch Alternativprogramme oder Programmveränderungen erprobt.

Im Bereich des Programmierens und des Rechnereinsatzes gilt nämlich wie in vielen anderen Lebensbereichen auch:

Übung macht den Meister !

Im einzelnen werden in den Kapiteln dieses Buchs die folgenden Problembereiche angesprochen:

1. Aufgaben aus der Mathematik
2. Probleme aus der Chemie
3. Physikalische Aufgabenstellungen
4. Aufgaben aus dem Sprachunterricht
5. Aufgaben aus dem Bereich Biologie/Ökologie
6. Probleme aus Erdkunde und Geschichte
7. Wirtschaftsprobleme
8. Sonstige Aufgabenstellungen

Diesen inhaltlichen Kapiteln werden einige Ausführungen zu den Grundlagen und Bedingungen des Rechnereinsatzes beigefügt, und wir werden auch die Grundelemente der Programmiersprache BASIC besprechen.
Der Leser, der schon über hinreichende BASIC–Kenntnisse verfügt, mag die entsprechenden Ausführungen getrost überblättern.
Zu den Computerprogrammen, die in den folgenden Kapiteln entwickelt oder vorgestellt werden, ist folgende Anmerkung erforderlich:

Zum einen handelt es sich um Programme, die bestimmte Probleme lösen sollen (Beispiel: Programm zur Ermittlung eines "größten, gemeinsamen Teilers").

Diese Programme können mit unterschiedlichen Ausgangsdaten "gefüttert" werden und dienen dann immer dem gleichen Zweck. Zum anderen aber gibt es auch reine Trainingsprogramme: Ein Programm zum Beispiel, das zum Erlernen von Englisch–Vokabeln verwendet werden kann, löst keine Probleme, sondern es "trainiert" nur.
Allerdings gilt auch bei diesen Programmen, daß sie allgemeiner eingesetzt werden können, wenn sie mit unterschiedlichem Ausgangsmaterial versorgt werden.

Zu diesem Buch werden auch eine Programmkassette und eine Programm-Diskette vorbereitet, wo sich alle Programme, die hier besprochen werden, wiederfinden, so daß der Leser, der diese Programme benutzen will, sich von der Last des Eintippens befreien kann. Diese Programmsammlung verzichtet allerdings auf die kleinen Beispielprogramme im folgenden Kapitel, die nur dazu dienen, die Grundelemente der Programmiersprache BASIC zu illustrieren.

Die folgende Anmerkung ist für die weiteren Ausführungen besonders wichtig:

Wir haben in den Programmen, die Schulprobleme lösen sollen, bewußt darauf verzichtet, mit fortgeschritteneren und anspruchsvolleren BASIC-Sprachelementen zu arbeiten. Vielmehr haben wir uns bemüht, bei der Erstellung der Programme nur einfachere und grundlegende Sprachelemente zu benutzen. Dies hat den Zweck, den Leser - auch wenn er noch Anfänger sein sollte - nicht durch programmiertechnische Fragen zu überfordern, sondern ihm die Konzentration auf die sachlichen Probleme zu ermöglichen.

Diese Einschränkung hat zur Folge, daß manche Programme etwas unbeholfen wirken mögen – aber sie funktionieren, und zwar mit einfachsten Mitteln; darauf kam es uns zunächst an.

Bei komplizierteren Problemen in den späteren Kapiteln wird dann allerdings auch der Einsatz weiterer Sprachelemente erforderlich, die dann erst von Fall zu Fall besprochen werden.

Weiterhin möge der Leser insbesondere beachten, daß manche der vorgestellten Programme nur exemplarischen Charakter haben:

Wenn z.B. in einem Vokabel—Abfrageprogramm nur 1Ø Vokabeln zur Verfügung stehen, so muß der Benutzer, wenn er dieses Programm konkret nutzen will, zunächst die Vokabelliste verlängern. An der grundsätzlichen Struktur des Programms - und auf die kommt es ja an — ändert sich dadurch natürlich nichts.

1

Kapitel 1: Grundlagen der Programmiersprache BASIC

1.1 Vorbemerkung

In diesem Kapitel werden neben einigen allgemeinen Grundbegriffen aus der Datenverarbeitung, die später wieder auftauchen, die wichtigsten Grundelemente der Programmiersprache BASIC dargestellt und kurz erläutert.
Derjenige Leser, der schon hinreichend mit BASIC vertraut ist, kann dieses Kapitel überblättern.

Leser aber, die noch gar keine Programmiererfahrung haben, werden die Ausführungen in diesem Kapitel für etwas zu kurzgefaßt ansehen. Man kann sich aber mit dem Hinweis trösten, daß in den späteren Kapiteln bei der praktischen Verwendung dieser BASIC–Sprachbestandteile deren Verwendung, Funktionsweise und Wirkung dann auf jeden Fall deutlich werden wird.
So gesehen, enthält dieses Kapitel eher nur einen vorläufigen Überblick.

1.2 Grundbegriffe, die man kennen sollte

Das Wort "Computer" bedeutet wörtlich übersetzt "Rechner". Ein moderner Computer kann aber mehr als bloß rechnen. Er kann beispielsweise auch mit Buchstaben, mit Worten und Texten umgehen; er kann graphische Darstellungen erzeugen, er kann Musik produzieren, er kann Steuerungsaufgaben übernehmen u.ä. Dies bedeutet also, daß die Rechner, die heute schon für ein paar Hundert Mark erworben werden können, nicht nur die Rechenkapazitäten bereitstellen wie z.B. die Großrechner der sechziger Jahre, die damals unerschwinglich teuer waren, sondern sie sind viel universeller einsetzbar.

Daten: Unter "Daten" kann man im weitesten Wortsinn "Informationen" verstehen. Im einzelnen können dies z.B. sein :

- Ziffern und Zahlen
- Werte
- Buchstaben
- Symbole (Sonderzeichen)
- Worte und Texte

Datenverarbeitung: Unter "Datenverarbeitung" versteht man dann alle Prozeduren (häufig rechnerischer Art), derartige Daten zu erfassen, zu speichern, auszuwerten oder zu analysieren und Ergebnisse auszugeben.

Führt man Datenverarbeitung im obigen Sinn nicht mehr per Hand, sondern mit Hilfe von Computern durch, dann ist es sinnvoll, auch die folgenden Begriffe zu kennen:

Bit: Unter einem "Bit" (binary information digit) versteht man die kleinste Informationseinheit (s.o.: "Daten" = "Informationen").
Die kleinste Informationseinheit nennt man ein Bit deshalb, weil in einem Bit nur zwei Informationsinhalte vorstellbar sind. Aus der Sicht des Computerherstellers gesprochen, heißt dies: Ein Bit kann nur zwei Informationsinhalte speichern.
Diese beiden Informationsinhalte werden üblicherweise bezeichnet mit Ø und 1.

Byte: Ein Byte ist eine Zusammenfassung einer Folge von Bits.
Üblicherweise bilden acht Bits ein Byte. Ein Byte ist somit in der Lage, eine Folge von 8 Nullen und/oder Einsen zu speichern.

Symbol: Wir unterscheiden drei Gruppen von Symbolen:

1. numerische Symbole = Ziffern
2. alphabetische Symbole = Buchstaben
3. sonstige Symbole = Sonderzeichen

Im allgemeinen wird jedes dieser Symbole im Rechner dargestellt als Folge von 8 Nullen und/oder Einsen, d.h. ein Byte (s.o.) nimmt ein Symbol auf.

Feld: Eine Folge von Symbolen (z.B. Worte oder Zahlen) bilden ein Feld.

Beispielsweise besteht das Wort PRINT aus 5 Symbolen, belegt mithin 5 Byte im Rechner, und da diese 5 Symbole zusammengehören, bilden sie ein Feld.

Entsprechendes gilt z.B. für die Zahl 178. Sie wird in einem dreistelligen Feld gespeichert (man sieht, daß es auch einstellige Felder geben kann).

Variable: Eine Variable ist eine veränderliche Größe oder - präziser gesprochen - ein Name, unter dem verschiedene Werte (s.u.) angesprochen werden können.
Beispielsweise stehen hinter dem Variablennamen "Körpergröße" die Werte 178, 185, 167 usw., hinter dem Namen "Kinderzahl" z.B. die Werte 1, 2, 0, 4 usw., hinter dem Namen "Ort" z.B. die Werte "Dortmund", "Bochum", "Kassel" und dergl.

Wert: Wie aus der vorangegangenen Begriffsklärung schon zu entnehmen ist, verstehen wir unter dem Begriff "Wert" eine bestimmte Ausprägung einer gegebenen Variablen. Es muß sich dabei nicht notwendigerweise um Zahlen handeln, sondern wie das obige Beispiel der Variablen "Ort" zeigt, können als Werte auch Worte oder Texte auftreten können.

Die Werte belegen jeweils ein Feld im Rechner (s.o.). Man spricht in diesem Zusammenhang auch von einer Speicherstelle.

<u>String</u>: Einen Text, der z.B. als Variablenausprägung auftreten kann (s.o.), nennt man "String", allgemein gesprochen handelt es sich um eine Zeichenkette.

<u>Array</u>: Die Gesamtheit aller Ausprägungen <u>einer</u> Variablen (Zahlen oder Worte) nennt man "Array".

<u>Daten-satz</u>: Mehrere zusammengehörige Felder nennt man einen Datensatz.
Beispielsweise bilden alle vorhandenen Angaben über eine Person (Name, Geschlecht, Alter, Wohnort etc.) einen Datensatz. In diesem Beispiel wäre ein Datensatz also eine Folge zusammengehöriger Zahlen und/oder Worte.
Aber auch eine Folge nur von Worten kann als Datensatz bezeichnet werden, z.B. die <u>Programm-anweisung</u>:

```
PRINT "OTTO", "SUSI", "OLAF"
```

Der Datensatz wird auch als "Record" bezeichnet.

<u>Datei</u>: Mehrere zusammengehörige Datensätze bilden eine Datei. Beispielsweise erhalten wir eine Datei, wenn wir die Datensätze aller Beschäftigten eines Unternehmens zusammenstellen. Genauso würden wir eine Datei erhalten, wenn wir alle Programmzeilen (alle Programmanweisungen) eines Programms insgesamt betrachten.

Eine Datei wird auch als "<u>File</u>" bezeichnet.

1.3 Zur Funktionsweise von Rechnersystemen

Es wurde schon erläutert, was unter "Datenverarbeitung" verstanden werden soll. Daraus geht hervor, daß eine Datenverarbeitungsanlage die folgenden Tätigkeiten (Funktionen) ausführen muß:

1. Informationsaufnahme (Dateneingabe)
2. Informationsspeicherung
3. Verarbeiten von Daten im engeren Wortsinn (z.B. Rechnen oder Umstellen von Texten und dergl.
4. Ausgabe von Ergebnissen

E	Eingabe
V	Verarbeitung
A	Ausgabe

Informationseingabe, -verarbeitung und -ausgabe machen eine Reihe organisatorischer Schritte notwendig, die wir hier nicht zu besprechen brauchen, weil sie der Rechner im allgemeinen selbständig erledigt. Verantwortlich dafür ist das sog. Betriebssystem.
Der Rechner muß also von uns Informationen erhalten, damit ein Datenverarbeitungsprozeß in Gang kommen kann.
Er gibt seinerseits Informationen zurück, beispielsweise errechnete Ergebnisse.

Diejenigen Informationen, die wir ihm geben müssen, lassen sich in drei Informationsgruppen einteilen:

1. Zu verarbeitende Daten

2. Programmanweisungen
   (Arbeitsschritte, die notwendig sind, um die eingegebenen Daten den gewünschten Auswertungsprozeduren zu unterziehen)

3. Kommandos
   (Mitteilungen und Anweisungen an das Betriebssystem des Rechners; s.o.)

Entsprechend den unterschiedlichen Teilaufgaben, die der Rechner erledigen muß, unterteilen wir:

1. Eingabegeräte (z.B. Tastatur)
2. Verarbeitungsbereich (die sog. Zentraleinheit)
3. Ausgabegeräte (Bildschirm oder z.B. Drucker)

Darüber hinaus gibt es sog. externe (oder periphere) Speicher. Sie dienen dazu, Programme oder Datenbestände auf Dauer aufzubewahren. In der Regel handelt es sich dabei um Tonband-Kassetten oder um sog. Disketten, die in entsprechenden Laufwerken bespielt bzw. gelesen werden können.

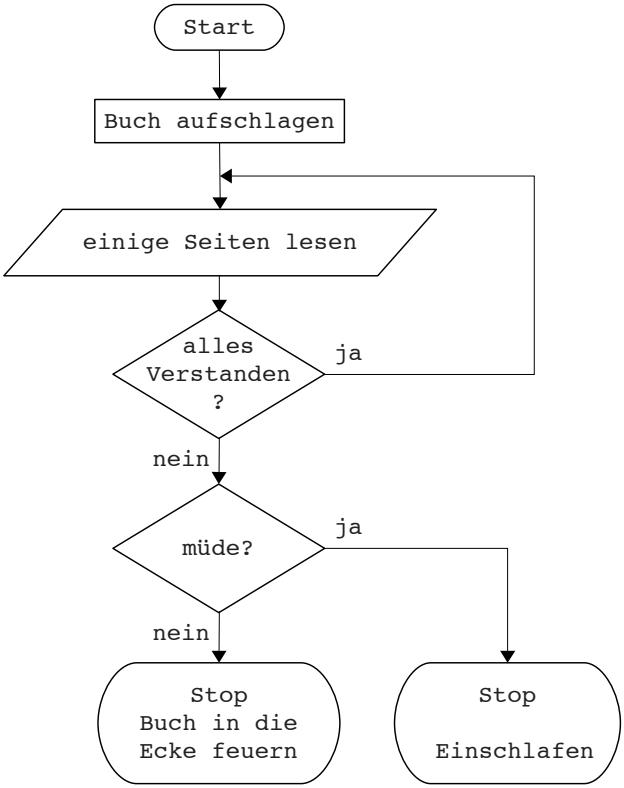
1.4 Problemanalyse

Wenn ein bestimmtes Problem mit Hilfe eines Computers gelöst werden soll, so ist dies nur dann möglich (der Leser möge diese Anmerkung immer beachten), wenn der Computerbenutzer den Lösungsweg vorher schon gefunden hat. Das heißt, daß die Problemlösung an sich, trotz aller modernen Rechner, immer noch nach wie vor nur im Kopf und nicht in der Maschine erledigt werden kann; der Rechner entlastet uns dann nur von der Aufgabe, den Lösungsweg Schritt für Schritt abzuarbeiten.

Damit ein Rechner nun diese Abarbeitung leisten kann, muß ihm der eigentliche Lösungsweg als Folge einzelner Anweisungen Schritt für Schritt vorgezeichnet werden. Dies wiederum macht es erforderlich, daß ein zu lösendes Problem zunächst gedanklich in eine logische Folge von Einzelschritten zerlegt wird - wir bezeichnen diesen Vorgang als Problemanalyse.

Häufig ist es bei einer derartigen Problemanalyse hilfreich, wenn man die Folge von Arbeitsschritten graphisch in Form eines sog. Flußdiagramms darstellt.

Dies zeigt anschaulich das folgende Beispiel:



Quelle : W. VOSS : BASIC leicht und schnell gelernt am alphaTronic PC, Darmstadt 1984, (Heim-Verlag), Seite 8.

In den folgenden Kapiteln werden wir auf dieses Instrument der "Problemzergliederung" zurückgreifen, weil auf diese Weise das Programmieren, d.h. die Erstellung der Anweisungsfolge für den Computer, wesentlich erleichtert wird und logische Programmfehler eher vermieden werden können.

1.5 Ergebnisausgabe

Eine Folge von Anweisungen, die einen Rechner veranlassen, ein vorgegebenes Problem selbständig zu lösen, nennt man ein Programm . Es muß in einer Sprache geschrieben werden, die der Rechner verstehen kann.
Eine solche Sprache ist die Programmiersprache BASIC.

BASIC = Beginner's
All-Purpose
Symbolic
Instruction
Code

Jedes BASIC-Programm besteht aus einer Folge von Anweisungen (Statements), die dem Rechner zeilenweise eingegeben werden. Eine Zeile nennen wir Satz.

Regel 1: Jeder Satz in BASIC benötigt eine Satznummer.

Regel 2: Jeder Satz muß mit der RETURN-Taste abgeschlossen werden.

Regel 3: In einem Satz können mehrere Statements stehen, die dann durch Doppelpunkte voneinander zu trennen sind.

Statement 1:

```
nn    END
```

Das END-Statement beschließt das BASIC-Programm.

Statement 2:

```
                  leer
                  Zahl
nn PRINT          Rechnung
                  String
                  Variablenname
```

Das PRINT-Statement bringt Ergebnisse auf den Bildschirm. Man unterscheidet fünf Möglichkeiten:

1. leer : Ergibt eine Leerzeile.
2. Zahl : Die Zahl wird ausgegeben.
3. Rechnung : Das Ergebnis der Rechnung wird ausgegeben.
4. String : Der String wird ausgegeben (ein String ist eine Zeichenkette, die in Anführungszeichen einzuschliessen ist).
5. Variablenname : Der Inhalt des Speicherfeldes mit diesem Namen wird ausgegeben.

Das Wort PRINT darf durch ein ? abgekürzt werden.

Das folgende einfache BASIC-Programm benutzt alle diese Möglichkeiten:

```
10 PRINT 3
20 PRINT
30 PRINT 4.8/2
40 ?
50 ? "SUSI"
60 ?
70 PRINT X
80 END
```

Ein solches BASIC-Programm wird erst abgearbeitet, wenn wir das Kommando RUN eingeben:

Kommando 1:

RUN

Für das obige Beispiel erhalten wir nach dem Kommando RUN auf dem Bildschirm die folgenden Ergebnisse:

```
3
2.4
SUSI
0
```

Regel 4: Kommandos erhalten keine Satznummer.

Wollen wir nach dem Ergebnisausdruck unser Programm auf dem Bildschirm noch einmal sehen, benötigen wir das folgende Kommando:

Kommando 2:

```
LIST
```

Zum PRINT-Statement sind noch einige Anmerkungen erforderlich:

Die verschiedenen Möglichkeiten, die in Statement 2 genannt wurden, können in einem PRINT-Statement mehrfach und auch gemischt verwendet werden.
Sie sind dann durch Kommata oder durch Strichpunkte voneinander zu trennen.

Regel 5: Werden die verschiedenen Ausdrücke im PRINT-Statement durch Kommata getrennt, wird der Ausdruck zu Beginn der nächsten Bildschirmzone fortgesetzt; wird das Semikolon als Trenner benutzt, geht es unmittelbar weiter.

Regel 6: Jeder PRINT-Befehl bewirkt in der Regel auch einen Zeilenvorschub. Steht im Statement nur das Wort PRINT, so erfolgt nur ein Zeilenvorschub.

Regel 7: Ein Komma oder ein Semikolon am Ende eines PRINT-Statements unterdrückt den Zeilenvorschub.

Das PRINT-Statement erzeugt Ergebnisausdrucke, die am linken Bildschirmrand beginnen. Will man dies ändern, kann man sich der Anweisung POSITION bedienen:

```
Statement 3:
      nn  POSITION S,Z
```

Diese Anweisung rückt den Cursor auf Spalte S und Zeile Z des Bildschirms und nachfolgende Ausdrucke beginnen dort.

Dabei kann 3 Werte zwischen Ø und 39 und Z Werte zwischen Ø und 23 annehmen.

Man hat sich also den Bildschirm aufgeteilt zu denken in 4Ø Spalten und 24 Zeilen, wobei die Zeile Ø ganz oben und die Spalte Ø ganz links ist.

Auf den Begriff des Variablennamens, der im PRINT-Statement ja auch schon eine Rolle spielte und auf Einzelheiten der ausführbaren Rechnungen, kommen wir im folgenden Abschnitt zu sprechen.

Will man vor der Eingabe eines neuen Programms ein eventuell schon vorhandenes, altes und nun nicht mehr benötigtes Programm löschen, benötigt man das folgende Kommando, welches den Inhalt des Programmspeichers löscht:

```
Kommando 3:
            NEW
```

1.6 Wertzuweisungen

Für viele Zwecke ist es – wie man noch sehen wird – sehr sinnvoll, wenn Speicherplätze im Rechner mit Namen versehen werden können, damit dann unter diesen Namen unterschiedliche Werte gespeichert werden können.

Zunächst genügt es, zwei Typen von Variablen zu unterscheiden:

1. Numerische Variablen
2. Stringvariablen

Numerische Variablen können als Werte Zahlen annehmen, also z.B.:

3.5 4 -17.Ø1 7 18. -3 Ø.Ø4 Ø

-7.5 2Ø etc.

(statt des Dezimalkommas wird beim Programmieren ein Punkt verwendet).

String-Variablen haben als Werte Symbolfolgen, die in Anführungszeichen einzuschließen sind, also z.B.:

"OTTO" "ØØ" "DORTMUND" "A-15" usw.

Regel 8: Die Namen numerischer Variablen bestehen aus einem oder zwei alphanumerischen Zeichen, wobei das erste ein Buchstabe sein muß.
Die Namen von String-Variablen werden zusätzlich mit $ versehen.

noch Regel 8:	Beim ATARI dürfen die Namen auch länger sein (Benutzerhandbücher konsultieren!).

Beispiele:

1. Numerische Variablen: X B1 NN

2. String-Variablen : F$ C2$ ZZ$

Nach dieser Begriffsklärung können wir das LET-Statement besprechen, welches der Wertzuweisung dient, so daß für einen gesamten Programmablauf dieser Wert dann zur Verfügung steht:

Statement 2:

nn LET Var.name = Zahl / Rechnung / String / Var.name

Das Wort LET darf auch entfallen.

Mit diesem Statement kann also, wie auch übrigens schon mit dem PRINT-Statement, gerechnet werden.
Die Rechenoperatoren sind die folgenden:

+	Addition
-	Subtraktion
*	Multiplikation
/	Division
^	Potenzierung

Die Rangfolge der Operatoren entspricht der aus der Schulmathematik bekannten; bei Bedarf kann durch Klammersetzung davon abgewichen werden.

Mit dem LET-Statement können also auch schon kompliziertere Anweisungen geschrieben werden. Wenn man sich dabei vertippt, besteht die einfachste Korrektur darin, den fehlerhaften Satz per RETURN-Taste zu schließen und ihn darunter einfach neu zu schreiben.
Soll ein Satz ganz gelöscht werden, so gibt man einfach seine Satznummer ein plus RETURN-Taste.

Soll ein Satz nachträglich in ein Programm eingefügt werden, so wird dies durch Vergabe passender "Zwischen-Satznummern" erreicht.

Viele Rechnungen können durch die sog. eingebauten Funktionen erleichtert werden. Die wichtigsten dieser Funktionen sind die folgenden:

Allgemeine Schreibweise:

```
Var.name = Funktionsname (Argument)
```

Wichtige Funktionen

Name	Aufgabe
SIN COS	berechnet die Winkelfunktionswerte, wobei als Argument Einheiten des Kreisparameter s ~ zu verwenden ist
LOG EXP	natürlicher Logarithmus Umkehrung dazu
SQR	Quadratwurzel
ABS	Absolutwert
INT	größter, in einem reellen Wert vorhandener, ganzzahliger Teil
RND	Zufallszahlenerzeugung
CHR$	Ausgabe eines ASCII- Code-Zeichens

Beispiele zu diesen Funktionen sind hier entbehrlich. Sie finden sich zur Genüge in den folgenden Kapiteln.

1.7 Informationseingabe

Zur Informationseingabe eignet sich hervorragend das folgende Statement:

Statement 5:
nn INPUT Var.liste

Dieses Statement sieht recht kompliziert aus, ist aber leicht zu verstehen:

Gelangt der Rechner an dieses Statement, unterbricht er die Programmabarbeitung und produziert ein Fragezeichen auf dem Bildschirm. Er erwartet dann vom Benutzer so viele Werte (durch Kommata zu trennen), wie Variablennamen in der Variablenliste genannt sind (also mindestens einen Wert, wenn nur ein Name genannt ist).

Beispiele:

INPUT X	Der Rechner erwartet eine Zahl, die dann im Feld X gespeichert wird.
INPUT A,B,C	Der Rechner erwartet drei Zahlen.

INPUT N$: Der Rechner erwartet einen String (z.B. "OTTO"), der im Feld N$ gespeichert wird.

Allerdings gilt, daß jede Stringvariable zunächst dimensioniert werden muß (siehe S. 31).

Regel 9: Die Variablennamen in der Variablenliste des INPUT-Statements werden durch Kommata getrennt. Die einzugebenden Werte müssen genauso getrennt werden.

Beispiel:

```
10 PRINT "BITTE VIER WERTE : ";
20 INPUT A,B,C,D
30 LET S=A+B+C+D
40 LET AM=S/4
50 PRINT "MITTELWERT = ";AM
60 END
```

Dieses Programm berechnet offenbar den Durchschnitt aus beliebigen vier einzugebenden Zahlen.

1.8 Programmverzweigungen

Solange in einem BASIC–Programm keine Verzweigungen auftreten (sog. Sprünge), erfolgt die Abarbeitung in der Reihenfolge der vergebenen Satznummern.
Will man von dieser Reihenfolge abweichen, so sind Programmsprünge erforderlich.
Wir unterscheiden unbedingte Sprünge von bedingten Sprüngen.

Betrachten wir zunächst den bedingten Sprung, der mit Hilfe des folgenden Statements möglich ist:

Statement 6:

```
                                  Anweisung
      nn  IF  Bedingung  THEN <
                                  Satznummer mm
```

Dieses Statement veranlaßt das Betriebssystem des Rechners zu einer Abfrage: Wird die Bedingung, die nach dem Befehlswort IF steht, erfüllt, so wird der hinter dem THEN stehende Befehl ausgeführt bzw. es wird zum Satz mit der Nummer mm verzweigt.

Ist die Bedingung hingegen nicht erfüllt, so geht die Programmsteuerung zu dem Satz über, der nach dem IF-Statement folgt.

Regel 10: Ist die Bedingung im IF-Statement erfüllt, so wird die Anweisung hinter THEN ausgeführt bzw. zu der dort stehenden Satznummer verzweigt.
Ist die Bedingung nicht erfüllt, wird der nächste Satz bearbeitet.

Regel 11: Stehen hinter dem IF-Statement noch weitere Statements im gleichen Satz, so werden diese nur dann ausgeführt, wenn die Bedingung erfüllt ist.

Beispiel:

```
10 I=1
20 Q=I*I
30 W=SQR(I)
40 PRINT I,Q,W
50 I=I+1
60 IF I<=20 THEN 20
70 END
```

Dieses Programm bestimmt offensichtlich für alle ganzen Zahlen von 1 bis 20 (jeweils im Feld I zu finden), die Quadratzahlen (im Feld Q; Satz 20) und die Quadratwurzeln (im Feld W; Satz 30) und druckt die jeweils zusammengehörigen drei Werte zeilenweise auf dem Bildschirm.

Entscheidend dabei ist SatZ 60. Er besagt, daß das Programm immer wieder zum Satz 20 zurückkehren soll, solange im Feld I nach der Erhöhung des Inhalts dieses Feldes jeweils um 1 (Satz 50) ein Wert steht, der kleiner als 21 ist.

Solange diese Bedingung erfüllt ist, wird das getan, was hinter dem THEN steht (nämlich Rücksprung ach Satz 20); ist sie nicht mehr erfüllt, wird der dem IF-Statement folgende Satz (Satz 70) bearbeitet.

Die Alternative zum bedingten Sprung ist der unbedingte Sprung:

Statement 7:

```
nn  GOTO  mm
```

Erreicht die Programmsteuerung dieses Statement, so erfolgt ein Sprung zum Satz mit der Nummer mm. Für das nächste Beispielprogramm benötigen wir noch das folgende Statement:

Statement 8:

```
DIM  Var.name (Zahl1),Var.name(...
```

Beispiel:

```
DIM X(20)
```

Dieses Statement hält für die Variable X nicht nur einen, sondern 21 Plätze frei (Platz 0 bis 20).

Bei Stringvariablen muß für jedes Symbol des Strings ein Feld freigehalten werden, also z.B.:

```
DIM A$(5)
```

hält für einen String fünf Felder frei, so daß ein String mit der Maximallänge von fünf Symbolen eingegeben werden kann.
Immer wenn mit Stringvariablen gearbeitet wird, muß ein derartiges DIM-Statement vorgegeben werden.

```
10 PRINT CHR$(125)
15 DIM A$(1)
20 PRINT "BITTE WERT EINGEBEN :
";
30 INPUT X
40 N=N+1
50 S=S+X
60 PRINT "NOCH EIN WERT (J/N) ";
70 INPUT A$
80 IF A$="J" THEN 20
90 PRINT "MITTELWERT = ";S/N
100 END
```

Dieses Programm bestimmt für eine beliebige Zahl einzugebender Werte das arithmetische Mittel. Wenn ein Wert gegeben wurde, fragt der Rechner den Benutzer, ob noch einer einzugeben ist. Antwortet der Benutzer mit "J" (für "Ja"), so erfolgt ein Sprung von Satz 8Ø zu Satz 2Ø und der nächste Wert wird angefordert.
Ist hingegen kein weiterer Wert vorhanden, d.h. antwortet der Benutzer beim INPUT-Statement in Satz 7Ø nicht mit "J", so erfolgt die weitere Programmabarbeitung ab Satz 9Ø (Berechnung und Ausgabe des Mittelwerts).

Das Statement

```
PRINT CHR$(125)
```

löscht den Bildschirm

1.9 Programmschleifen

Häufig ist es sinnvoll, daß bestimmte Programmteile mehrfach zu durchlaufen sind. Zu diesem Zweck benötigt man sog. Schleifen (loops). Solche Schleifen können mit denjenigen Statements, die wir bisher besprochen haben, leicht erzeugt werden, wie das Beispiel des vorangegangenen Programms zeigt.

Mit den folgenden Statements geht die Schleifenkonstruktion einfacher und es werden zudem zusätzliche Möglichkeiten eröffnet.

```
Statement 9:  nn  FOR Laufvarname = Anfangswert

                  TO  Endwert (STEP Schrittweite)
```

Das folgende Statement gehört notwendig dazu:

```
Statement 1Ø:  nn  NEXT Laufvarname
```

("Lauf varname" steht dabei als Abkürzung für:

"Name einer Laufvariablen").

Die Laufvariable, die in diesen beiden Statements auftaucht, muß eine reelle Variable sein. Anfangswert, Endwert und Schrittweite können Zahlen, Variablen oder arithmetische Ausdrücke sein.

"Gelangt der Rechner an das FOR-Statement, so wird zunächst die Laufvariable gleich dem Anfangswert gesetzt und der Programmteil zwischen FOR und NEXT wird mit diesem Wert durchlaufen. Bei Erreichen des NEXT wird nun die Laufvariable auf den nächsten Wert, nämlich Anfangswert + Schrittweite, gesetzt und wieder derselbe Programmteil abgearbeitet, jetzt mit dem 2. Wert der Laufvariablen.

Dies erfolgt solange, bis bei Erreichen des NEXT durch weitere Addition der Schrittweite der Endwert überschritten wird. Dann wird der nach NEXT folgende Satz bearbeitet.

"STEP Schrittweite" kann aus dem FOR-Befehl weggelassen werden, wenn die Schrittweite gleich eins sein soll." (PRUST, 1982, S. 64).

Das folgende Programm druckt 1Ø mal "GUTEN TAG" untereinander:

```
10 FOR I=1 TO 10
20 PRINT "GUTEN TAG"
30 NEXT I
40 PRINT
50 PRINT "ENDE"
60 END
```

1.1Ø Die Benutzung externer Speicher

Gelungene Programme möchte man gern auf Dauer aufheben, so daß man bei Bedarf wieder darauf zurückgreifen kann (entsprechend gilt, daß man ab und zu gern schon vorhandene Programme anderer Verfasser benutzen möchte). Dies wird ermöglicht, wenn man Tonbandkassetten oder Disketten benutzt. Dabei muß man die folgenden Kommandos kennen.

Zum Laden eines Programms von einer Diskette benötigen wir:

Kommando 4:

```
LOAD "D1: Name des Programms"
```

Zum Speichern eines Programms auf Diskette benötigen wir:

Kommando 5:

```
SAVE "D1: Name des Programms"
```

(Es wird hier vorausgesetzt, daß nur ein Diskettenlaufwerk angeschlossen ist, deshalb D1)

Will man feststellen, welche Programme sich auf einer Diskette befinden, so kann man folgendermaßen vorgehen:

1. Laden der Betriebssystem-Diskette ("Master-Diskette")

2. Eintippen des Kommandos DOS (DOS = Disk Operating System = Betriebssystem)
   Der Rechner gibt daraufhin ein sog. Menü aus, auf dem sich an erster Stelle das Kommando FILES befindet (FILES = Dateien)

3. Es genügt, nun den Buchstaben F einzutippen

4. Dann ist die zu prüfende Diskette einzulegen

5. Danach ist viermal die RETURN-Taste zu betätigen.
   Daraufhin erscheint auf dem Bildschirm ein Inhaltsverzeichnis der Diskette

6. Nach der Rückkehr ins Menü ist die RESET-Taste zu drücken. Dann kann normal weitergearbeitet werden (z.B. Laden eines Programms).

Nähere Einzelheiten sind der Betriebsanleitung des Diskettenlaufwerks zu entnehmen.

1.11 Ergänzungen

Ein paar wenige, nützliche Ergänzungen mögen dieses zusammenfassende Übersichtskapitel beschließen.

Es sieht ganz gut aus, wenn der Bildschirm bei der Programmabarbeitung zunächst gelöscht wird, bevor die ersten Ergebnisse ausgegeben werden. Dies wird ermöglicht mit der Anweisung:

```
nn  PRINT CHR$(125)
```

Häufig ist es sinnvoll, einen Programmlauf zu stoppen, beispielsweise dann, wenn man eine sog. Endlosschleife programmiert hat. Betätigt man die Taste BREAK, so hält die Programmabarbeitung an.

Schließlich ist anzumerken, daß mit dem Statement

```
Statement 11:  nn  STOP
```

der Programmablauf vom Programm selbst unterbrochen werden kann. Soll das unterbrochene Programm weiter abgearbeitet werden, benötigen wir das Kommando:

```
Kommando 5:
            CONT
```

Schließlich sei ergänzend angemerkt, daß es häufig sehr nützlich ist, in ein Programm erklärende Anmerkungen (Überschriften, Kommentare, Erläuterungen usw.) einzufügen. Dies kann mit dem folgenden Statement geschehen:

Statement 12: nn REM Text

Dieses Statement hat keinen Einfluß auf die Programmabarbeitung, erscheint aber in jeder Programmliste, die erstellt wird.

Diese Ausführungen mögen genügen, um auch dem Anfänger eine Vorstellung von den Grundelementen der Programmiersprache BASIC zu vermitteln. So wird er in die Lage versetzt, die Bausteine der folgenden Anwendungsprogramme zu verstehen. Sollten dennoch Verständnisprobleme bleiben, so gilt der folgende Ratschlag:

A U S P R O B I E R E N

Beim Ausprobieren von Programmen und beim Versuch, gedanklich nachzuvollziehen, warum der Rechner genau das tut, was er tut, lernt man am meisten.

2

Kapitel 2: Mathematik

2.1 Vorbemerkung

Die Mathematik ist derjenige Bereich, der uns am ehesten einfällt, wenn die Frage gestellt wird, wofür denn überhaupt Computer eingesetzt werden. Deshalb wollen wir mit diesem Bereich die inhaltlichen Ausführungen beginnen. Wir werden später aber sehen, daß die Mathematik nur ein Anwendungsbereich unter vielen ist.

Allerdings stehen wir nun, wie auch in den folgenden Kapiteln, vor dem Problem, welcher Art die Beispiele sein sollen, die hier zweckmäßigerweise aufzunehmen sind. Die Mathematik umfaßt ja bekanntlich ein sehr weites Spektrum unterschiedlicher Schwierigkeitsgrade - beginnend vielleicht bei den Grundrechenarten und bei der Differential- und der Integralrechnung noch lange nicht endend.

Wir wollen das hier so handhaben, daß wir Probleme "mittleren Schwierigkeitsgrades" aufgreifen, also uns etwa an denjenigen Aufgabenstellungen orientieren, wie sie zum Beispiel in der gymnasialen Mittelstufe, also in den

Jahrgangsstufen 8 bis 10 auftreten. Sie eignen sich sehr gut dazu, die Möglichkeiten von Rechnern zu demonstrieren, ohne daß wegen des eventuell zu hohen Komplexitätsgrades die Darstellung der behandelten Probleme zu viel Platz in Anspruch nehmen würde.

Absicht dieses Buches ist es ja nicht, den Leser durch die Anwendungsbeispiele zu überfordern, sondern er soll erkennen, was Rechner leisten. Dies ist auch mit einfach strukturierten Fragestellungen möglich, die dem Schulalltag auch der jüngeren Leser angepaßt sind.

Für die Programme in diesem und in den folgenden Kapiteln gilt die folgende wichtige Anmerkung:

> Bei den Programmen wird darauf verzichtet, Eingabefehler der Benutzer "abzufangen"!

Wenn also zum Beispiel bei einem bestimmten mathematischen Problem nur positive Zahlen verwendet werden können (so etwa beim Wurzelziehen) , so ist im entsprechenden Programm nicht vorgesehen, denn Benutzer dann zu informieren, wenn er irrtümlich eine negative Zahl eingegeben hat.

Mit einem sehr einfachen Beispiel "mittleren Schwierigkeitsgrades" wollen wir nun beginnen.

2.2 Der Satz des Pythagoras

Am berühmten Lehrsatz des Pythagoras kann in anschaulicher Weise gezeigt werden, wie die einzelnen Beispiele gegliedert werden.

Wir werden jeweils die folgenden Arbeitsschritte vollziehen:

Arbeitsschritte:

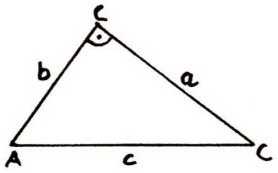
1. SCHRITT : Vorstellung des Problems
2. SCHRITT : Problemanalyse
3. SCHRITT : Flußdiagramm
4. SCHRITT : Programm
5. SCHRITT : Variablenliste
6. SCHRITT : Programmbeschreibung
7. SCHRITT : Ergebnisse

Wenden wir uns nun, dieser Einteilung entsprechend, dem hier genannten Problem zu:

1. SCHRITT : Vorstellung des Problems

Der Lehrsatz des Pythagoras besagt, daß im rechtwinkligen Dreieck die Länge der Grundseite (Hypotenuse) sich ergibt als Quadratwurzel aus der Summe der beiden Kathetenquadrate.

Dies illustriert die folgende Abbildung:



$$c = \sqrt{a^2 + b^2}$$

2. SCHRITT : Problemanalyse

Bei diesem Problem ist die Problemanalyse sehr einfach:

Man muß sich dabei, wie bei allen folgenden Problemen auch, daran erinnern, daß ganz allgemein ein Datenverarbeitungsprozeß nach dem schon vorgestellten

E V A - Prinzip

abläuft.

E	=	Eingabe
V	=	Verarbeitung
A	=	Ausgabe

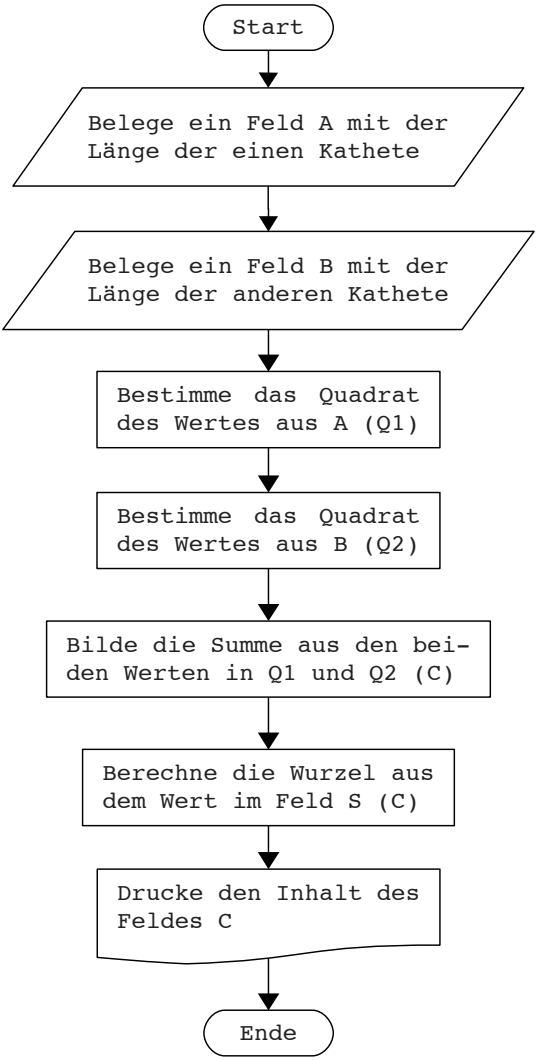
Wir müssen bei der Problemanalyse also zunächst festlegen, welche Informationen der Rechner von uns bekommen muß, damit das gestellte Problem gelöst werden kann.

Die notwendigen Eingabeinformationen sind offensichtlich die Längen der beiden Kathetenseiten a und b.

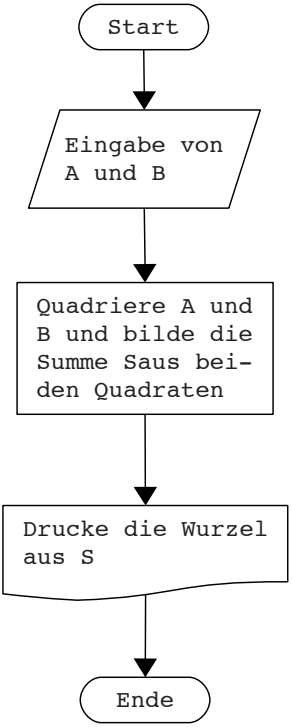
Im Verarbeitungsschritt sind diese beiden Werte zu quadrieren; es ist die Summe dieser beiden Quadrate zu bilden und aus dieser Summe muß dann die Quadratwurzel gezogen werden.

Der Ausgabeschritt besteht lediglich darin, das errechnete Ergebnis (die Länge der Hypothenuse c) auf dem Bildschirm auszugeben.

3. SCHRITT : Flußdiagramm



Kürzere Version des Flußdiagramms



4. SCHRITT : Programm

```
10 REM M1-PYTHRGORAS
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BESTIMMUNG DER HYPOTHENU-"
40 PRINT " SE EINES RECHTWINKLIGEN DREIECKS."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF.DR.W.VOSS, 1984"
50 PRINT
60 PRINT "ERSTE KATHETE  : ";:INPUT A
70 PRINT "ZWEITE KATHETE : ";:INPUT B
80 Q1=A*A
90 Q2=B*B
100 S=Q1+Q2
110 C=SQR(S)
120 PRINT :PRINT :PRINT "HYPOTHENUSE C = ";C
130 END
```

Kürzere Programmversion

```
10 bis 50 : wie oben

60 PRINT "ZWEI KATHETENWERTE : ";:INPUT A,B
70 C=SQR(A*A+B*B)
80 PRINT "HYPOTHENUSE C = ";C:END
```

5. SCHRITT : Variablenliste

A : Erste Kathete
B : Zweite Kathete
C : Hypothenuse
Q1 : Quadrat der ersten Kathete A
Q2 : Quadrat der zweiten Kathete B
S : Summe der Kathetenquadrate

6. SCHRITT : Programmbeschreibung

Bei der Beschreibung beziehen wir uns auf die erste, die längere Programmversion:

Satz 1Ø : Kommentar

Satz 2Ø : Der Bildschirm wird freigemacht

Satz 3Ø-5Ø: Titelausdruck auf dem Bildschirm

Satz 6Ø : Für die erste Dreiecks-Kathete wird ein Wert angefordert und im Feld A gespeichert

Satz 7Ø : Entsprechend zweite Kathete für Feld B

Satz 8Ø : Berechnung des ersten Kathetenquadrats

Satz 9Ø : Berechnung des zweiten Kathetenquadrats

Satz 1ØØ : Bestimmung der Summe der beiden Katheten-quadrate

Satz 11Ø : Bestimmung der Wurzel aus der Quadratsumme

Satz 12Ø : Ausdruck des Ergebnisses auf dem Bildschirm

Satz 13Ø : Beendigung des Programms

7. SCHRITT : Ergebnisse

Nach dem Starten des Programms (erste Programmversion) erscheint auf dem Bildschirm die Überschrift :

PROGRAMM ZUR BESTIMMUNG DER HYPOTHENU-
SE EINES RECHTWINKLIGEN DREIECKS.

PROF.DR.W.VOSS, 1984

Zwei Zeilen tiefer erhalten wir die erste Anforderung:

ERSTE KATHETE : ?

Geben wir daraufhin zum Beispiel den Wert 3 ein, so reagiert der Rechner mit:

ZWEITE KATHETE : ?

Geben wir beispielsweise den Wert 4 ein, so antwortet der Rechner mit:

HYPOTHENUSE C = 5

Dieses Programm eignet sich also dazu, für beliebige Eingaben die Hypothenuse rechtwinkliger Dreiecke auszurechnen. In diesem Sinn ist es also ganz allgemein gehalten.

Wir haben dieses Problem hier sehr ausführlich behandelt, um die Vorgehensweise bei den einzelnen Beispielen im Detail zu illustrieren. Sicherlich werden wir uns bei den folgenden Beispielen daraufhin kürzer fassen können, zumindest wenn sie so einfach sind wie der Lehrsatz des Pythagoras.

2.3 g.g.T. Und k.g.V.

1. SCHRITT : Vorstellung des Problems

Für viele algebraische Rechnungen ist es, insbesondere beim Auftreten von Brüchen, sinnvoll, den größten gemeinsamen Teiler (g . g.T .) und das kleinste gemeinsame Vielfache (k.g.V.) verschiedener Zahlen zu kennen.

Beim g.g.T. handelt es sich um den Wert, durch den zwei Zahlen ohne Rest dividiert werden können , ohne daß es eine größere Zahl gibt, für die dies möglich wäre.

Beim k.g . V. handelt es sich um die Zahl, die durch zwei Ausgangszahlen ohne Rest dividiert werden kann, ohne daß es eine kleinere Zahl gibt , für die dies gilt.

Haben wir zum Beispiel die beiden Zahlen Z1 = 30 und z2 = 40, so ist der g.g.T. offenbar 10 und das k.g.V. hat den Wert 120. (Der Leser möge dies durch Kopfrechnen überprüfen).

2. SCHRITT : Problemanalyse

Wie findet man nun bei zwei gegebenen Zahlen den g.g.T. und das k.g.V.?

Prüfen wir einmal anhand der beiden Zahlen 3Ø und 4Ø, wie sinnvollerweise vorzugehen ist, wenn man den g.g.T. finden will:

1. Schritt : 4Ø : 3Ø = 1 Rest 1Ø

2. Schritt : 3Ø : 1Ø = 3 Rest Ø

In diesem Beispiel finden wir also den g.g.T. folgendermaßen:

1. Schritt : Division der größeren durch die kleinere Zahl; Notieren des Rests.

2. Schritt : Division der kleineren Zahl durch den Rest; Notieren des neuen Rests.

Wird dabei der neue Rest null, so ist der Divisor, der zu diesem Nullrest führte, der gesuchte g.g.T.

Dazu noch ein zweites Beispiel zur Verdeutlichung:

Feld 1 : Z1 (größere Zahl)	Feld 2 : Z2 (kleinere Zahl)	Division	Rest
78	42	1	36
42	36	1	6
36	6	6	Ø

Der g.g.T. von 78 und 42 ist also 6.

Das Berechnungsschema, wie es die obige Tabelle zeigt, läßt sich leicht in ein Flußdiagramm umsetzen.

Zunächst wenden wir uns aber noch dem kleinsten gemeinsamen Vielfachen (k.g.V.) zu:

Das k.g.V. von zwei Zahlen Z1 und Z2 kann man dadurch erhalten, daß man das Produkt dieser beiden Zahlen durch den g.g.T. dividiert.

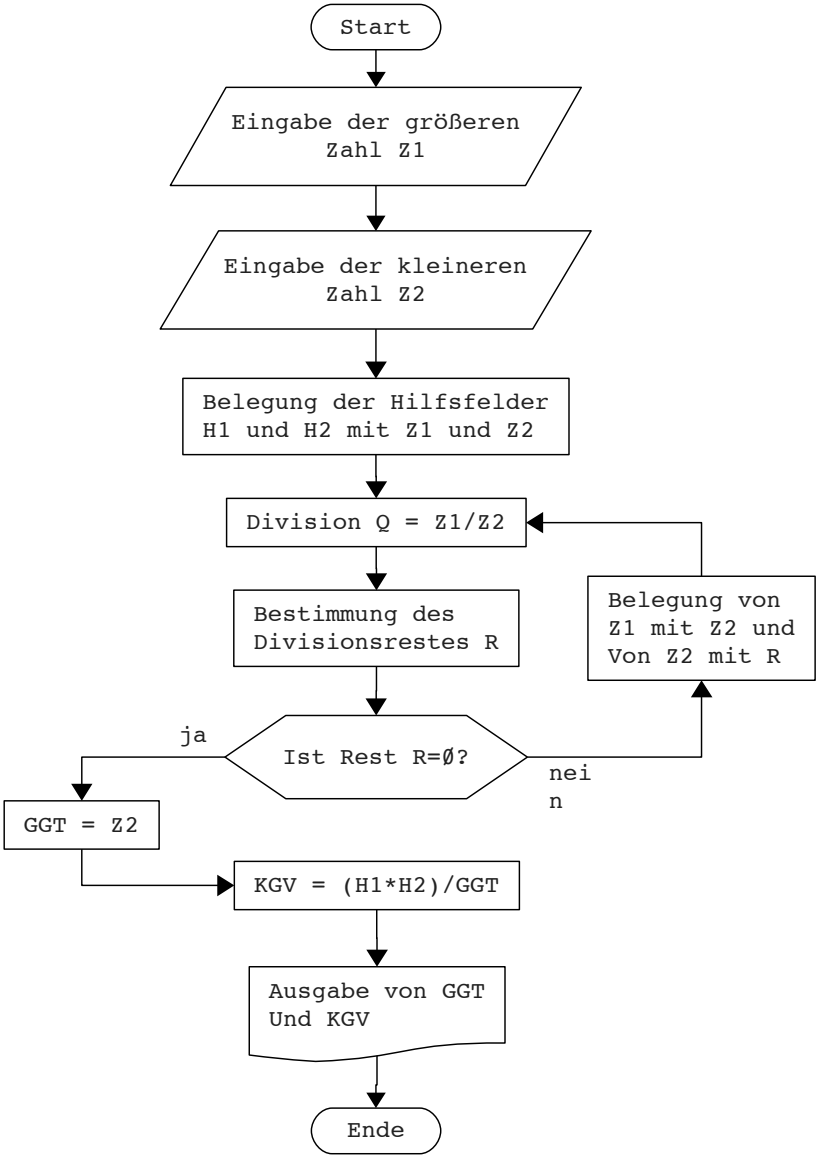
Für die beiden Zahlen Z1 = 78 und Z2 = 42 gilt also:

$$\text{k.g.V.} = \frac{Z1 * Z2}{\text{g.g.T.}} = \frac{78 * 42}{6} = 546$$

Wenn wir in einem BASIC-Programm diese Berechnung durchführen wollen, gemäß der Übersichtstabelle oben, aber fortwährend die Feldbesetzungen verändern, müssen <u>vor</u>

der Berechnung des g.g.T. die einzugebenden Werte Z1 und Z2 in zwei Hilfsfeldern H1 und H2 "zwischengespeichert" werden, um für die Berechnung des k.g.V. dann zur Verfügung zu stehen.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM M2-GGT UND KGV
20 PRINT CHR$(125)
30 PRINT " PROGRAMM ZUR BESTIMMUNG DES G.G.T. "
40 PRINT "UND DES K.G.V. AUS ZWEI EINGEGEBENEN"
42 PRINT "                ZAHLEN."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF.DR. W.VOSS, 1984"
50 PRINT
60 PRINT "GROESSERE ZAHL : ";:INPUT Z1
70 PRINT :PRINT
80 PRINT "KLEINERE ZAHL  : ";:INPUT Z2
90 H1=Z1:H2=Z2
100 Q=Z1/Z2
110 R=Z1-Z2*INT(Q)
120 IF R>0 THEN Z1=Z2:Z2=R:GOTO 100
130 GGT=Z2
140 KGV=(H1*H2)/GGT
150 PRINT :PRINT :PRINT "G.G.T.  =  ";GGT
160 PRINT :PRINT :PRINT "K.G.V.  =  ";KGV
170 PRINT :PRINT "ENDE DER BERECHNUNG":END
```

5. SCHRITT : Variablenliste

GGT	:	Größter gemeinsamer Teiler
H1	:	Hilfsfeld 1
H2	:	Hilfsfeld 2
KGV	:	Kleinstes gemeinsames Vielfaches
Q	:	Quotient aus beiden Zahlen Z1 und Z2
R	:	Divisionsrest
Z1	:	Größere Zahl
Z2	:	Kleinere Zahl

6. SCHRITT : Programmbeschreibung

Satz 10-50 : Überschrift und Erläuterungen.

Satz 60 : Eingabe der ersten Zahl.

Satz 70 : Zwei Leerzeilen.

Satz 80 : Eingabe der zweiten Zahl.

Satz 90 : Belegung der beiden Hilfsfelder H1 und H2 mit den eingegebenen Werten, damit diese später noch zur Verfügung stehen (siehe 140).

Satz 100 : Bestimmung des Quotienten aus beiden eingegebenen Zahlen.

Satz 110 : Bestimmung des Divisionsrestes durch Benutzung der INT-Funktion.

Satz 120 : Ist dieser Rest größer als null, wird das Feld Z1 mit der zweiten Zahl Z2 und das Feld Z2 mit dem Rest R belegt und das Programm kehrt zurück zu Satz 100.

Satz 130 : Ist hingegen der Rest gleich null, so steht im Feld Z2 der g.g.T., der in das Feld GGT übertragen wird.

Satz 14Ø : Gemäß unserer Vorüberlegungen bestimmt sich dann das k.g.V. aus dem Produkt der beiden Zahlen (die noch in den Feldern H1 und H2 zu finden sind), dividiert durch den g.g.T.

Satz 15Ø-16Ø : Ausdruck der Ergebnisse mit einigen Leerzeilen dazwischen.

Satz 17Ø : Beendigung des Programms

7. SCHRITT : Programmergebnisse

Nach dem Kommando RUN meldet das Programm auf einem "geräumten" Bildschirm:

```
 PROGRAMM ZUR BESTIMMUNG DES G.G.T.
UND DES K.G.V. AUS ZWEI EINGEGEBENEN
               ZAHLEN.


        PROF.DR. W.VOSS, 1984

GROESSERE ZAHL : ?
```

Geben wir nach Erscheinen dieses Fragezeichens, das ja bekanntlich eine Programmunterbrechung anzeigt, bei-

spielsweise den Wert 78 ein, so fordert das Programm daraufhin die zweite Zahl an :

KLEINERE ZAHL : ?

Geben wir zum Beispiel die Zahl 42 ein, so bringt der Rechner die folgenden Ergebnisse auf den Bildschirm:

G.G.T. = 6

K.G.V. = 546

ENDE DER BERECHNUNG

2.4 Primzahlenprüfung

1. SCHRITT : Vorstellung des Problems

Es soll ein Programm entwickelt werden, welches für eine beliebige eingegebene, positive, ganze Zahl prüft, ob diese eine Primzahl ist oder nicht.

Bekanntlich handelt es sich bei einer Primzahl um eine Zahl, die nur durch 1 oder durch sich selbst ohne Rest teilbar ist.

2. SCHRITT : Problemanalyse

Um zu prüfen, ob eine beliebige Zahl eine Primzahl ist, muß sie durch alle Teiler, die größer als 1, aber kleiner als sie selbst sind, geteilt werden. Sofern bei diesen Divisionen ein Rest auftritt, bzw. wenn das Divisionsergebnis nicht ganzzahlig ist, muß durch einen weiteren Teiler dividiert werden.

Erzielt man bei einer dieser Divisionen ein Ergebnis ohne Rest, so ist die zu prüfende Zahl offenbar keine Primzahl.

Wird bei allen denkbaren Divisionen hingegen ein Rest beobachtet, so ist schließlich festzustellen, daß die zu prüfende Ausgangszahl eine Primzahl ist.

Die Teiler, die im einzelnen durchprobiert werden müssen, beginnen beim Wert 2 und können bei der Hälfte der Ausgangszahl enden; größere Teiler brauchen nicht probiert zu werden, weil dann auf gar keinen Fall mehr ein ganzzahliges Divisionsergebnis erzielt werden kann.

Beispiel : Ausgangszahl X = 25

1. Teiler : 2	25/2 = 12 Rest 1
2. Teiler : 3	25/3 = 8 Rest 1
3. Teiler : 4	25/4 = 6 Rest 1
4. Teiler : 5	25/5 = 5 Rest Ø
	===

25 ist <u>keine</u> Primzahl.

Beispiel : Ausgangszahl X = 7

1. Teiler : 2 7/2 = 3 Rest 1
2. Teiler : 3 7/3 = 2 Rest 1

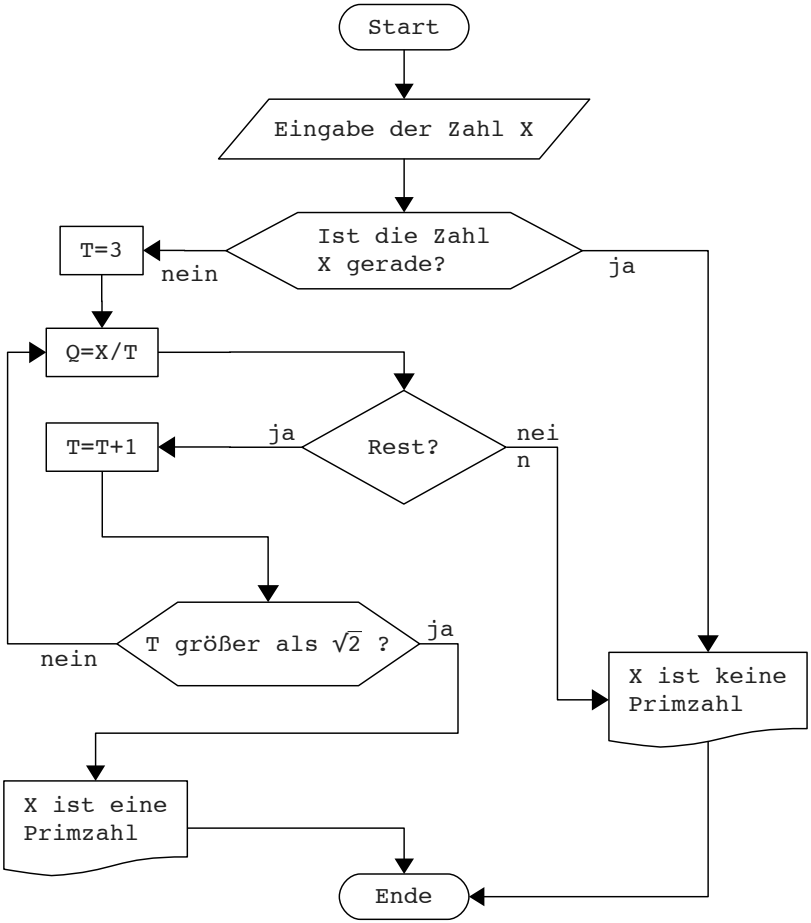
Ein weiterer Teiler braucht nicht mehr probiert zu werden, weil das Divisionsergebnis nicht mehr ganzzahlig werden kann.

Da bei diesem Beispiel kein Rest Ø aufgetreten ist, gilt also :

7 ist eine Primzahl.

Diese beiden Beispiele zeigen, daß wir Schritt für Schritt den Teiler, mit dem Wert 2 beginnend, um je 1 erhöhen müssen, bis maximal der Wert X/2 erreicht ist.

3. SCHRITT : Flußdiagramm



Dieses Beispiel zeigt übrigens sehr deutlich, daß bei sorgfältiger gedanklicher Durchdringung des Problems das Programm wesentlich vereinfacht und damit auch beschleunigt werden kann:

Man kann schon von vornherein feststellen, daß eine gerade Zahl keine Primzahl sein kann, denn sie ist ja auf jeden Fall durch 2 teilbar. Deshalb können gerade Zahlen schon zu Beginn des Prüfprogramms als "Nicht-Primzahlen" ausgeschieden werden.

Bei den verbleibenden, den ungeraden Zahlen braucht für eine beliebige Zahl X der Teiler offenbar nicht bis X/2 zu laufen, sondern es genügt, wenn wir bis maximal $\sqrt{X}$ die Divisionen probieren. Wenn bis dahin keine Division ohne Rest möglich war, dann wird dies auch bei größeren Teilern nicht mehr möglich sein.

Der Leser überlege sorgfältig, warum diese Überlegung richtig ist.

4. SCHRITT : Programm

```
10 REM M3-PRIMZAHL
20 PRINT CHR$ (125)
30 PRINT "PROGRAMM ZUR PRUEFUNG, OB EINE EINGE-"
40 PRINT " GEBENE ZAHL EINE PRIMZAHL IST ODER"
42 PRINT "                NICHT."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF.DR.W.VOSS, 1984"
50 PRINT
60 PRINT "BITTE EINE GANZE ZAHL : ";:INPUT X
70 T=2
80 IF X/2=INT(X/2) THEN 140
90 Q=X/T
100 IF Q=INT(Q) THEN 140
110 T=T+1
120 IF T<=SQR(X) THEN 90
130 PRINT :PRINT X;" IST EINE PRIMZAHL":GOTO 150
140 PRINT :PRINT X;" IST KEINE PRIMZAHL, SONDERN"
145 PRINT "DURCH ";T;" TEILBAR."
150 PRINT :PRINT "ENDE DER BERECHNUNG":END
```

5. SCHRITT : Variablenliste

Q = Quotient aus der Zahl X und dem Teiler T.

T = Teiler

X = Einzugebende Zahl, die daraufhin geprüft wird, ob sie Primzahl ist oder nicht.

6. SCHRITT : Programmbeschreibung

Satz 1Ø-5Ø : Überschrift, Räumung des Bildschirms und Titelausdruck auf dem Bildschirm.

Satz 6Ø : Anforderung der zu prüfenden Zahl.

Satz 7Ø : Belegung des Teilerfeldes T mit 2.

Satz 8Ø : Wenn X durch 2 ohne Rest teilbar sein sollte (dann ist X/2 = INT(X/2)), dann weiter bei Satz 14Ø; wenn nicht, weiter bei Satz 9Ø.

Satz 9Ø : Division von X durch T (ergibt Q).

Satz 1ØØ : Wenn bei der Division kein Rest aufgetreten ist (dann ist Q = INT(Q)), dann weiter bei Satz 14Ø, andernfalls weiter bei Satz 11Ø.

Satz 11Ø : Erhöhung des Teilers T um 1.

Satz 12Ø : Solange nach dieser Erhöhung T noch kleiner ist als $\sqrt{X}$, geht es zurück zum Satz 9Ø, andernfalls geht es bei 13Ø weiter.

Satz 13Ø : Wird der Satz 13Ø erreicht, dann deshalb, weil keine Division ohne Rest vollzogen werden konnte. Dies bedeutet, daß X eine Primzahl sein muß.

Dieses Ergebnis wird ausgegeben und es

erfolgt ein Sprung zum Satz 15Ø.

Satz 14Ø : und 145 — Satz 14Ø wird nur in dem Fall erreicht, in dem eine Division ohne Rest gelungen ist.

Dies bedeutet aber dann, daß X keine Primzahl ist, sondern (zumindest) durch den Wert im Feld T teilbar ist.

Dies wird ausgegeben.

Satz 15Ø : — Beendigung des Programms.

7. SCHRITT : Ergebnisse

Auf die Wiederholung des Titelausdrucks wollen wir nun hier und bei den folgenden Beispielen verzichten.

Nachdem dieser Titelausdruck erfolgt ist, fordert das programm die zu prüfende Zahl an und speichert sie im Feld X (auch auf die Wiedergabe der entsprechenden Anforderungstexte soll hier und bei den folgenden Beispielen verzichtet werden).

Geben wir z.B. die Zahl 13 ein, so meldet der Rechner:

13 IST EINE PRIMZAHL

ENDE DER BERECHNUNGEN

Geben wir hingegen die Zahl 27 ein, so meldet der Rechner:

```
27 IST KEINE PRIMZAHL, SONDERN
DURCH 3 TEILBAR

ENDE DER BERECHNUNGEN
```

2.5 Quadratische Gleichung

1. SCHRITT : Vorstellung des Problems

Das Lösen quadratischer Gleichungen gehört zu den wesentlichen Aufgabenstellungen der Mittelstufenmathematik und bereitet auf komplexere Aufgaben vor.

Die allgemeine Form einer quadratischen Gleichung, die nach der Unbekannten x aufzulösen ist, lautet:

$$(1) \qquad ax^2 + bx + c = \emptyset$$

Dividiert man diese Gleichung durch den Koeffizienten a, so erhält man:

$$(2) \qquad X^2 + px + q = \emptyset$$

Die Lösung führt zu zwei Werten x1 und x2 gemäß der folgenden Lösungsformel:

$$(3) \qquad x_{1/2} = -p/2 \pm \sqrt{p^2/4 - q}$$

2. SCHRITT : Problemanalyse

Je nach Konstellation der Koeffizienten a, b und c führt die Lösungsformel (3) zu unterschiedlichen Ergebnissen.

Beispielsweise erhalten wir Resultate außerhalb des reellen Zahlenbereichs, wenn der Wert unter der Wurzel negativ wird. Dieser Wert heißt Diskriminante (D).

Zu Sonderlösungen gelangt man auch z.B. dann, wenn der Koeffizient a = Ø ist; dann liegt aber auch eigentlich keine quadratische Gleichung mehr vor.

Deshalb betrachten wir im folgenden nur die "echten" Fälle ($a \neq Ø$) und zugleich die im reellen Bereich lösbaren Fälle ($D \geq Ø$). Bei allen anderen Fällen soll das Programm eine Abbruchmeldung erzeugen.

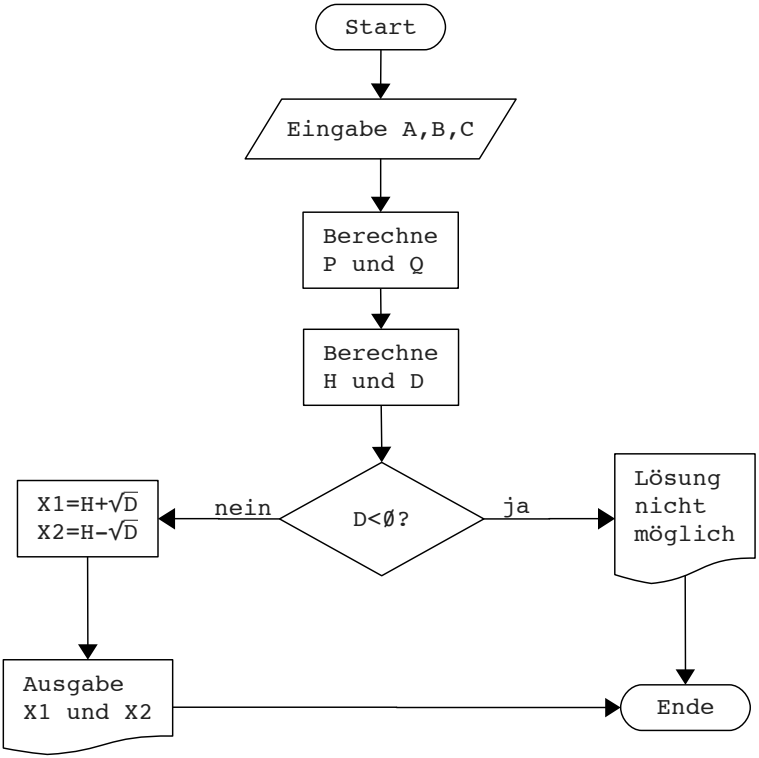
Zur Lösung des Problems verwenden wir einfach die Formel (3) in der abgewandelten Form (4):

$$(3) \qquad x_{1/2} = -p/2 \pm \sqrt{p^2/4 - q}$$

$$(4) \qquad x_{1/2} = H \pm \sqrt{D}$$

Das entsprechende BASIC-Programm muß also zunächst die Größen H und D bestimmen und kann dann, in Abhängigkeit davon, ob $D \geq Ø$ oder $D < Ø$, die Lösungen bzw. eine Abbruchmeldung ausgeben.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM M4-QUADRATISCHE GLEICHUNG
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR LOESUNG EINER QUADRATI-"
40 PRINT "           SCHEN GLEICHUNG."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF.DR.W.VOSS, 1984"
50 PRINT
60 PRINT "PARAMETER DES QUADR.GLIEDS A : ";:INPUT A
70 PRINT "PARAMETER DES LIN. GLIEDS  B : ";:INPUT B
80 PRINT "KONSTANTES GLIED           C : ";:INPUT C
90 P=B/A:Q=C/A
100 H=P/2:D=P*P/4-Q
110 IF D<0 THEN PRINT "LOESUNG NICHT MOEGLICH":GOTO 150
120 X1=H+SQR(D):X2=H-SQR(D)
130 PRINT :PRINT "X1 = ";X1
140 PRINT :PRINT "X2 = ";X2
150 PRINT :PRINT "ENDE DER BERECHNUNGEN":END
```

5. SCHRITT : Variablenliste

A = Koeffizient des quadratischen Teils
B = Koeffizient des linearen Teils
C = Konstante in der quadratischen Gleichung

D = Diskriminante
H = -P/2
P = B/A
Q = C/A

X1 = erste Lösung
X2 = zweite Lösung

6. SCHRITT : Programmbeschreibung

Satz 1Ø-5Ø : Kommentar und Überschriftausgabe.

Satz 6Ø-8Ø : Eingabe der Koeffizienten der quadratischen Gleichung (in der Form der Gleichung (1)).

Satz 9Ø : Bestimmung von P und Q gemäß Gleichung (2).

Satz 1ØØ : Bestimmung der Hilfsgröße H und der Diskriminante D.

Satz 11Ø : Wenn D kleiner als null ist, so erfolgt eine Abbruchmeldung auf dem Bildschirm und ein Sprung zu Satz 15Ø.

Satz 12Ø : Ist die Diskriminante hingegen nicht negativ, so werden die beiden Lösungen X1 und X2 berechnet.

Satz 13Ø und 14Ø : Beide Lösungen werden auf dem Bildschirm ausgegeben.

Satz 15Ø : Beendigung des Programms.

7. SCHRITT : Ergebnisse

Geben wir z.B. ein:

A = 4,
B = 4,
C = -24

Der Leser sollte auch andere Werte erproben!

geht es also um die quadratische Gleichung:

$$4x^2 + 4x - 24 = 0 ,$$

so erzeugt das Programm als Ergebnisausgabe:

X1 = 2
X2 = -3
ENDE DER BERECHNUNGEN

2.6 Euler'sche Zahl

1. SCHRITT : Vorstellung des Problems

Die Euler'sche Zahl, abgekürzt e, auch Wachstumskonstante genannt, spielt bei mathematischen Reihenentwicklungen, die inhaltlich als Wachstums- oder Fortpflanzungsketten betrachtet werden können, eine wichtige Rolle.

Sie dient darüberhinaus als Basis der sog. Natürlichen Logarithmen und wird uns im Kapitel über wirtschaftliche Problemstellungen wieder begegnen.

Wenn man den Wert dieser Zahl e vergessen hat, kann man mit Hilfe eines geeigneten BASIC-Programms diesen Wert ausrechnen, sofern man weiß, wie die Zahl e zustandekommt.

Es handelt sich bei e um den Grenzwert der folgenden Größe, wenn n gegen unendlich strebt:

$$e = (1 + 1/n)^n \quad \text{für n gegen unendlich.}$$

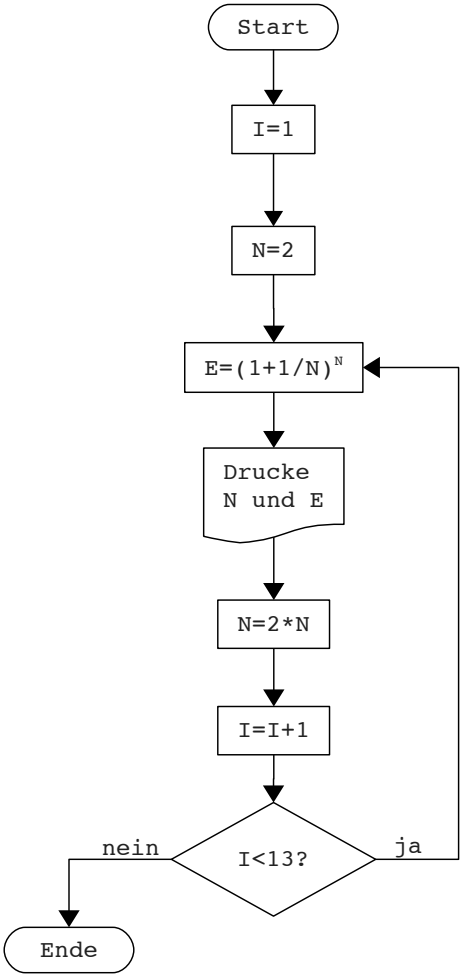
2. SCHRITT : Problemanalyse

Die Problemanalyse ist bei dieser Fragestellung sehr einfach, denn es ist lediglich die vorangegangene Formel zu programmieren.

Dieses Programm kann dann für alternative, größer werdende Werte von n durchprobiert werden. Da der Wert e erst für n gegen unendlich erreicht wird, empfiehlt es sich, n rasch wachsen zu lassen, z.B. dadurch, daß man n von Rechendurchgang zu Rechendurchgang verdoppelt.

Dies verlangt allerdings, daß man mit einem n-Wert beginnt, der größer als 1 ist (also z.B. mit n = 2), damit bei den Verdoppelungen n nicht beim Wert 1 verharrt.

3. SCHRITT : Flußdiagramm



Es wird davon ausgegangen, daß 12 "Runden" zu durchlaufen sind.

4. SCHRITT : Programm

```
10 REM M5-EULER
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BESTIMMUNG DER ZAHL   E."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF. DR. W. VOSS, 1984"
50 PRINT
60 N=2
70 PRINT "   N","   E":PRINT
80 FOR I=1 TO 12
90 E=(1+1/N)^N
100 PRINT N,E
110 N=N*2
120 NEXT I
130 PRINT "ENDE":END
```

5. SCHRITT : Variablenliste

E = Euler'sche Zahl (Näherungswert)
I = Schleifenzählindex
N = Parameter in der Berechnungsformel für die Euler'sche Zahl

6. SCHRITT : Programmbeschreibung

Satz 1Ø-5Ø : Überschrift.

Satz 6Ø : Belegung des Parameters N mit dem Startwert 2.

Satz 7Ø : Ausgabe einer Tabellenüberschrift und einer Leerzeile.

Satz 8Ø bis 12Ø : Schleife über 12 Runden. In jeder Runde wird E berechnet, werden N und E ausgegeben und wird N verdoppelt.

Satz 13Ø : Ende des Programms.

7. SCHRITT : Ergebnisse

Dieses Programm erzeugt eine Tabelle, in der anwachsenden N-Werten die berechneten E-Werte gegenübergestellt werden.

Man sieht bei dieser Gegenüberstellung, daß sich e einem festen Wert annähert.

Wollte man diesem Grenzwert noch näher kommen, dann müßte das Programm noch mehr Schleifendurchläufe erledigen.

Man müßte also den Satz 8Ø beispielsweise wie folgt ändern:

```
8Ø FOR I = 1 TO 3Ø
```

Es ist allerdings dabei zu beachten, daß bei zu hohen Werten für n die Rechenungenauigkeiten des Computers zu Fehlern führen kann.

Die sich durch obiges Programm ergebende Tabelle sieht folgendermaßen aus:

N	E
2	2.24999997
4	2.4414Ø621
8	2.56578451
16	2.63792848
32	2.67699Ø16
64	2.69734499
128	2.7Ø773919
256	2.71299187
512	2.71562546
1Ø24	2.71695Ø25
2Ø48	2.71761466
4Ø96	2.71794529

ENDE

2.7 Prozentrechnung

1. SCHRITT : Vorstellung des Problems

Beim Prozentrechnen spielt die Dreisatzrechnung eine entscheidende Rolle. Sie beantwortet zum Beispiel die Frage, welchen Anteil ein bestimmter Betrag B an einer Gesamtgröße G ausmacht. Ein typisches Beispiel kann diese Aufgabenstellung gut verdeutlichen:

Ein Rechnungsbetrag beläuft sich auf netto DM 450.--. Der Kunde muß zusätzlich die sog. Mehrwertsteuer bezahlen, die derzeit 14 % des Nettorechnungsbetrages ausmacht. Somit ergibt sich also:

Bruttobetrag = Nettobetrag + Steuerbetrag

Die Rechenaufgabe, die sich hier stellt, besteht also darin, den Steuerbetrag und danach den Bruttobetrag zu bestimmen, wenn der Nettobetrag und der Steuer-Prozentsatz gegeben sind.

Natürlich könnte man genauso gut nach dem Nettobetrag fragen, wenn z.B. der Steuersatz und der Bruttobetrag gegeben sind.

2. SCHRITT : Problemanalyse

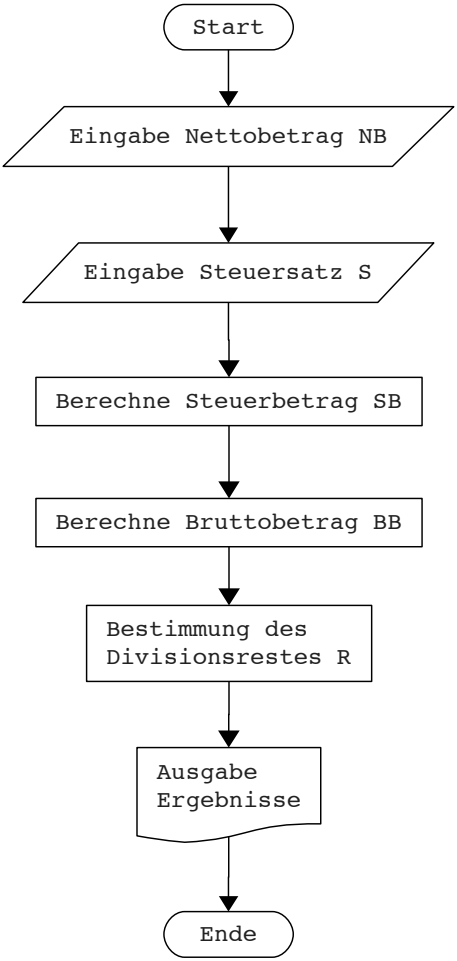
Das oben genannte Beispiel kann mit der Dreisatzrechnung folgendermaßen gelöst werden (und zeichnet uns so den Lösungsweg vor):

Nettobetrag = 100 % entsprechen 450.- DM
Steuerbetrag = 14 % entsprechen x DM

$$x = \frac{450 * 14}{100} = 63.- \ (DM)$$

Damit wird der Lösungsweg deutlich : Wir müssen in ein allgemein gehaltenes Programm den Nettobetrag eingeben und dazu den Steuersatz (oder allgemein gesprochen, den Prozentsatz, der beliebig sein kann, wenn das Programm generell einsetzbar sein soll).

3. SCHRITT : Flußdiagramm



3. SCHRITT : Programm

```
10 REM M6-PROZENT
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BESTIMMUNG VON PROZENT-"
40 PRINT "                WERTEN."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF.DR.W.VOSS, 1984"
50 PRINT
60 PRINT "NETTOBETRAG      : ";:INPUT NB
70 PRINT :PRINT "PROZENTSATZ      : ";:INPUT SS
80 SB=NB*SS/100
90 BB=NB+SB
95 PRINT :PRINT :PRINT :PRINT
100 PRINT " NETTOBETRAG     : ";NB
110 PRINT :PRINT "PROZENTBETRAG   : ";SB
120 PRINT :PRINT "BRUTTOBETRAG    : ";BB
130 PRINT :PRINT "ENDE":END
```

5. SCHRITT : Variablenliste

BB = Bruttobetrag
NB = Nettobetrag
SB = Prozentbetrag
SS = Prozentsatz

6. SCHRITT : Programmbeschreibung

Satz 1Ø-5Ø : Überschrift.

Satz 6Ø-7Ø : Eingabe von Nettobetrag und Prozentsatz.

Satz 8Ø-9Ø : Berechnung von Prozentbetrag und Brutto-betrag.

Satz 95 : Vier Leerzeilen.

Satz 1ØØ bis 12Ø : Ausgabe der Ergebnisse auf dem Bildschirm.

Satz 13Ø : Ende des Programms.

7. SCHRITT : Ergebnisse

Gibt man beispielsweise als Nettobetrag 2ØØ und als Prozentsatz den Wert 15 ein, so meldet das Programm:

```
NETTOBETRAG    :  2ØØ
PROZENTBETRAG  :  3Ø
BRUTTOBETRAG   :  23Ø
ENDE
```

2.8 Würfel

1. SCHRITT : Vorstellung des Problems

Es soll ein Programm vorgestellt werden, welches 1ØØ mal "würfelt" und den Durchschnitt dieser Augenzahlen ausgibt.

Man kann sich zu diesem Zweck der Funktion RND (RND = random = Zufall) bedienen, die Zufallszahlen zwischen Ø bis unter 1 erzeugt.

Zufallszahlen nennt man solche Zahlen, die in unregelmäßiger und unsystematischer Folge auftreten.

2. SCHRITT : Problemanalyse

Es genügt, beim ATARI bei der Benutzung der Funktion RND das Argument Ø vorzugeben.

Der Leser sollte diese Funktion einfach einmal erproben, indem er beispielsweise eingibt:

```
PRINT RND (Ø)
```

und noch einmal:

```
PRINT RND (Ø)
```

oder noch besser:

```
FOR I=1 TO 1Ø : PRINT RND(Ø):NEXT I
```

und danach die RETURN-Taste.

Auf dem Bildschirm erscheinen Zufallszahlen, die alle zwischen Ø bis unter 1 liegen.

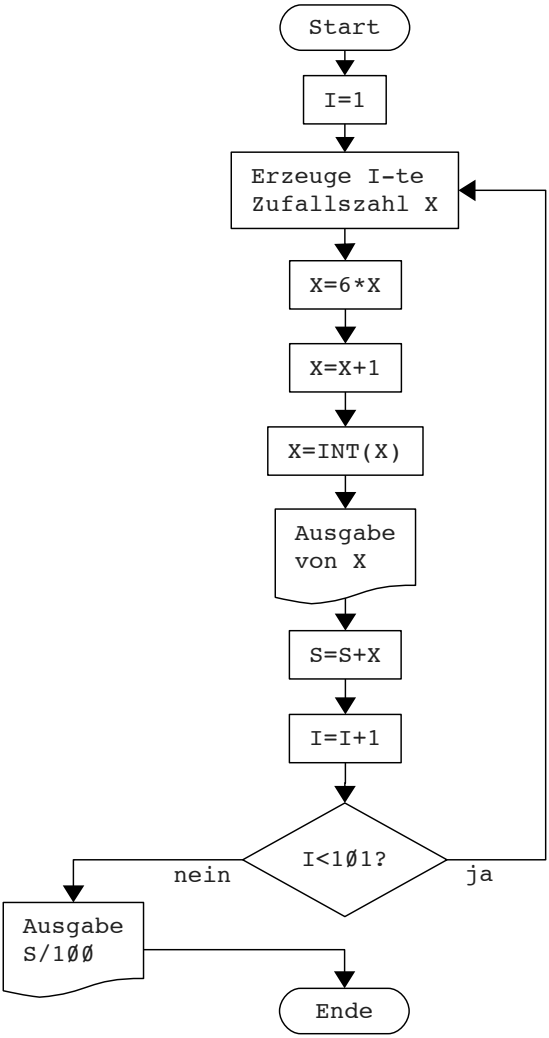
Wir hätten nun aber gern als Ergebnisse 1ØØ ganzzahlige Werte zwischen 1 und 6 - eben Würfel-Augenzahlen.

Das folgende Schema verdeutlicht, wie man aus Zufallszahlen aus dem Wertebereich zwischen Ø bis unter 1 ganze Zahlen zwischen 1 und 6 erzeugen kann:

Verarbeitung	**Ergebnis**
RND-Funktion	zwischen Ø bis unter 1
*6	zwischen Ø bis unter 6
1	zwischen 1 bis unter 7
INT-Funktion	ganzzahlig zwischen 1 bis 6

Damit ist der Problemlösungsweg vorgezeichnet. Da schließlich noch der Durchschnitt gewünscht wird, sind alle Ergebnisse aufzuaddieren und die entstehende Summe durch 1ØØ zu teilen.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM M7-WUERFEL
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR ERZEUGUNG VON 100 WUER-"
40 PRINT "FELWUERFEN UND DER DURCHSCHNITTLICHEN"
50 PRINT "            AUGENZAHL."
60 PRINT :PRINT :PRINT
70 PRINT "        PROF.DR.W.VOSS, 1984":PRINT
80 I=1
90 X=RND(0)
100 X=X*6:X=X+1:X=INT(X)
110 PRINT X;" ";
120 S=S+X
130 I=I+1
140 IF I<101 THEN 90
150 AM=S/100
160 PRINT :PRINT :PRINT :PRINT
170 PRINT "MITTELWERT = ";AM
180 PRINT :PRINT "ENDE DER AUSGABE":END
```

5. SCHRITT : Variablenliste

AM = Mittelwert

I = Laufindex (Rundenzähler)

S = Summe aller Würfelergebnisse

X = Zufallsergebnis

6. SCHRITT : Programmbeschreibung

Satz 10-70 : Überschrift.

Satz 80 : Vorgabe des Rundenzählers.

Satz 90 -100 : Erzeugung einer Zufallszahl und Transformierung in einen ganzzahligen Wert zwischen 1 und 6.

Satz 110 : Ausgabe.

Satz 120 : Erhöhung der Summe S um das "Würfelergebnis" X.

Satz 130 : Erhöhung des Rundenzählers I um 1.

Satz 140 : Solange I kleiner als 101 ist, geht es zurück zu Satz 90.

Satz 150 -170: Berechnung des Mittelwerts und Ausgabe.

Satz 150 : Ende des Programms.

7. SCHRITT : Ergebnisse

Wenn wir dieses Programm starten, so erzeugt der Rechner nach der üblichen Überschrift 100 zufällige Würfelergebnisse und gibt diese zeilenweise aus (der Leser überlege, warum eine zeilenweise Ausgabe erfolgt).

Danach wird der Durchschnitt aus den 100 Ergebnissen berechnet und ausgegeben, der - wenn der Zufallszahlengenerator in Ordnung ist - in der Nähe von 3.5 liegen muß.

Da die Ergebnisse zufallsabhängig sind, können sie sinnvollerweise hier nicht dargestellt werden.

2.9 Probleme der Rechengenauigkeit

Ein Computer wie der ATARI reserviert für jede reelle Zahl, mit der gerechnet wird, nur eine bestimmte Zahl von Bytes. Dies bedeutet, daß er nur mit beschränkter Genauigkeit arbeiten kann.

Man sieht dies sehr deutlich, wenn man in dem folgenden Programm "große" Werte vorgibt, und diese sich wieder ausdrucken läßt:

```
10 REM ZAHLENGENAUIGKEIT
20 A=12345678
30 B=123456789
40 C=1234567891
50 D=12345678912
60 ?A: ?B: ?C: ?D
70 END
```

Als Ergebnisse erhalten wir:

```
12345678
123456789
1234567890
1.23456789E+10
```

Bekommt der Rechner also Zahlen mit mehr als 10 Ziffern (Beispiel D), dann wandelt er sie bei der Ausgabe in die sog. Exponentialdarstellung um und druckt z.B. für D:

1.23456789E+1Ø

Dies ist zu lesen als:

$1.23456789 * 1Ø^{9} = 1.23456789 * 1\ ØØØ\ ØØØ\ ØØØ$

Dieses Produkt wiederum ergibt:

123456789Ø

Der Rechner hat also beim Wert für D die letzte Stelle "unterschlagen"; er kann sie nicht mehr berücksichtigen.

Allerdings wird, wie man an entsprechenden Beispielen zeigen kann, bei diesem "Unterschlagen" nicht einfach nur "abgeschnitten", sondern korrekt gerundet.

Das Hauptproblem dabei ist nun folgendes:

Wenn wir beispielsweise zum Wert D in obigem Programm den Wert 1 addieren, etwa in der neuen Programmzeile 55:

```
55 E = D + 1
```

und auch diesen Wert E ausdrucken lassen:

```
65 PRINT E
```

so erkennen wir keinen Unterschied in den Werten für D und E - der Rechner rechnet falsch !

Auf dieses Problem muß man immer dann achten, wenn man zum Beispiel sehr groBe mit sehr kleinen Zahlen (und umgekehrt) verrechnen will.

3

Kapitel 3: Chemie

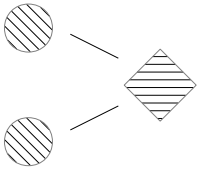
3.1 Zusätzliche BASIC-Anweisungen

Will man Programme aus dem Bereich der Chemie programmieren, so kann man hervorragend die Graphik- und Farbmöglichkeiten des ATARI ausnutzen - etwa zur graphischen Darstellung von Molekülen oder von chemischen Verbindungen.

Beispielsweise kann man das Wassermolekül

H_2O ,

das ja bekanntlich aus zwei Wasserstoffatomen (H) und einem Sauerstoffatom (Ø) zusammengesetzt ist, anschaulich folgendermaßen darstellen:



Um solche Darstellungen erzeugen zu können, benötigt man zusätzlich Anweisungen. Diese beziehen sich auf die Verwendung spezieller Symbole (auf die Möglichkeiten der sog. hochauflösenden Graphik kommen wir erst in einem späteren Kapitel zu sprechen).

Zur Anfertigung solcher und ähnlicher Zeichnungen eignet sich hervorragend der Graphikzeichen-Vorrat des ATARI (siehe Bedienungshandbuch).

Diese Zeichen, die unter bestimmten ASCII-Codezahlen zu finden sind (oabei sind auch noch einige spezielle Sonderzeichen), sind aufrufbar mit der früher kurz erwähnten Funktion

CHR$ (Argument)

Das Argument dieser Funktion ist die ASCII-Codezahl des gewünschten Symbols.

Um alle Symbole des ASCII-Codes sichtbar zu machen, kann man das folgende Programm benutzen:

```
10 FOR I=0 TO 255
15 IF I=28 OR I=29 OR I=125 THEN 30
17 PRINT I,CHR$(I)
20 IF I/20=INT(I/20) THEN STOP
30 NEXT I
40 END
```

Gibt der Benutzer zum Beispiel ein:

```
PRINT CHR$ (2Ø)
```

so erscheint auf dem Bildschirm ein Symbol, das einem Kügelchen ähnelt.

In entsprechender Weise erhalten wir nach der Eingabe von

```
PRINT CHR$ (16Ø)
```

ein helles Quadrat (das uns schon längst bekannte Symbol des Cursors) Für spätere Aufgabenstellungen sind weiterhin die folgenden Möglichkeiten interessant:

PRINT CHR$(6) ergibt ⟋

PRINT CHR$(7) ergibt ⟍

PRINT CHR$(18) ergibt ▬

PRINT CHR$(96) ergibt ♦

PRINT CHR$(253) läßt einen Signalton (bell) erklingen

Baut man derartige Anweisungen in Programme ein, so kann man Graphiken erzeugen (wenn auch noch recht grobe Graphiken).

Diese Möglichkeiten werden durch das folgende einfache Programm kurz angedeutet:

```
20 PRINT CHR$(125)
30 POSITION 10,10
40 FOR I=1 TO 20
50 PRINT CHR$(160);
60 NEXT I
70 PRINT
80 FOR I=1 TO 5
85 POSITION 10,I+10
90 PRINT CHR$(160)
100 NEXT I
110 POSITION 10,16
120 FOR I=1 TO 20
130 PRINT CHR$(160);
140 NEXT I
145 PRINT
150 FOR I=1 TO 5
160 POSITION 29,I+10
170 PRINT CHR$(160)
180 NEXT I
190 POSITION 16,13
200 PRINT "E N D E"
210 END
```

Weiterhin benötigen wir in diesem Kapitel und in den meisten der folgenden auch einige Anweisungen, die es uns erlauben, größere Informationsmengen geschickter als dies per LET- oder INPUT-Statement möglich wäre, in den Rechner zu bekommen.

Dazu eignet sich besonders gut das DATA-Statement in Verbindung mit dem READ-Statement.

Statement 13:

```
nn DATA Wert1,Wert2,Wert3, ...
```

Statement 14:

```
nn READ  Variable1,Variable2,Variable3, ...
```

Beispiel:

```
10 REM DATENEINGABE
20 DATA 4600,4630,8000,2000
30 READ P1,P2,P3,P4
40 PRINT P1:PRINT P2:PRINT P3:PRINT P4
50 END
```

Der Leser erprobe dieses Programm.

Bei großen Datenbeständen empfiehlt es sich, nur einen Variablennamen zu verwenden, der dann aber mehrere Werte erfaßt. Dies gelingt durch Verwendung indizierter Variablen, wie man sie ja zum Beispiel aus der Mathematik kennt. Üblicherweise schreibt man in der Mathematik:

x_1, X_2, X_3, ... allgemein x_i

Dies geht auch in entsprechender Weise in der Programmiersprache BASIC:

X(1), X(2), X(3), ... allgemein X(I)

(generell: Variablenname(Indexname))

Bei Verwendung solcher indizierten Variablen muß aber dem Rechner zu Beginn des entsprechenden Programms mitgeteilt werden, wieviele Speicherplätze für die jeweilige Variable freigehalten werden müssen.

Dies leistet das DIM-Statement:

<u>Statement 15</u>:

```
nn DIM Var.name1(Zahl1),Var.name2(Zahl2), ...
```

Beispielsweise hält das Statement

```
10 DIM X(20)
```

für die Variable X 21 Speicherplätze frei.

Es handelt sich dabei um die Plätze X(Ø), X(1), X(2), X(3), ...·, X(2Ø).

Ergänzend ist darauf hinzuweisen, daß zum Beispiel auch doppelte Indizierungen möglich sind:

Das Statement

```
1Ø DIM Z(2,3)
```

hält für die Variable Z 3*4 = 12 Plätze frei, die man sich in drei Zeilen und vier Spalten aufgeteilt vorstellen kann:

Z(Ø,Ø)	Z(Ø,1)	Z(Ø,2)	Z(Ø,3)
Z(1,Ø)	Z(1,1)	Z(1,2)	Z(1,3)
Z(2,Ø)	Z(2,1)	Z(2,2)	Z(2,3)

Die allgemeine Schreibweise lautet in diesem Fall:

```
Z(I,J)
```

Der erste Index (hier I) ist der sog. Zeilenindex, der zweite (hier J) der Spaltenindex.

Zum Abschluß muß erwähnt werden, daß mit den READ-Statements eines Programms immer in Abfolge auf die vorhandenen DATA-Statements zugegriffen wira. Dies bedeutet, daß ein zweites READ-Statement dort in den vorhandenen DATA-Statements fortfährt zu lesen, wo das vorangegangene READ-Statement aufgehört hat.

Häufig ist es nun allerdings sinnvoll, daß mit einem zweiten (oder sonst folgenden) READ-Statement die DATA-Liste wieder von Anfang an gelesen wird.

Um dies zu ermöglichen, benötigt man ein weiteres Statement:

Statement 16: nn RESTORE

Dieses Statement bewirkt, daß der "Lesezeiger" wieder an den Anfang des ersten DATA-Statements gestellt wird, so daß die Daten mit dem folgenden READ-Statement wieder von Anfang an gelesen werden.

3.2 Das Wassermolekül

1. SCHRITT : Vorstellung des Problems

Es soll mit Hilfe eines BASIC-Programms ein graphisches Abbild des Wassermoleküls (H_2O) erzeugt werden (siehe auch Abschnitt 3.1). Die Wasserstoffatome sollen z.B. als gelbe Kügelchen, das Sauerstoffatom als rotes Kügelchen präsentiert werden.

Mit einem derartigen Programm hätte man dann einen Grundbaustein auch für komplexere Programme graphischer Darstellungen dieser Art zur Verfügung.

2. SCHRITT : Problemanalyse

Wenn man das Bild, welches durch das Programm entstehen soll, schon vor Augen hat, dann ist in diesem Beispiel die Problemanalyse sehr einfach:

Es soll genau das Bild erzeugt werden, das wir schon zu Beginn des Abschnitts 3.1 gezeichnet haben.

Wir erkennen, daß dieses Bild sich aus vier Zeilen aufbaut:

1. Zeile: Ein Kügelchen und ein Schrägstrich von links oben nach rechts unten.

2. Zeile: Ein Karo (nach rechts versetzt)

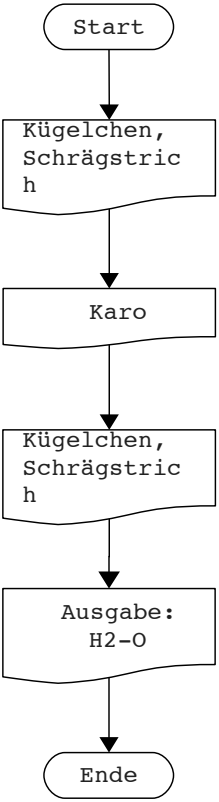
3. Zeile: Ein Kügelchen unter dem der ersten Zeile und ein Schrägstrich von links unten nach rechts oben.

4. Zeile: Text: H2-O.

Das Kügelchen erzeugen wir, wie schon zuvor, mit dem ASCII-Codezeichen mit der Nummer 20 (mit Hilfe der CHR$-Funktion), das Karo mit der Nummer 96.

Entsprechend hat der Schrägstrich von links oben nach rechts unten (\) die Codezahl 7 und der umgekehrte Schrägstrich (/) die Codezahl 6.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM C1-H20
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR GRAPHISCHEN DARSTELLUNG"
40 PRINT"    DES WASSERMOLEKUELS H2-O."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF.DR.W.VOSS, 1984"
50 PRINT
60 POSITION 16,11:PRINT CHR$(20)
70 POSITION 18,12:PRINT CHR$(7)
80 POSITION 20,13:PRINT CHR$(96)
90 POSITION 16,15:PRINT CHR$(20)
100 POSITION 18,14:PRINT CHR$(6)
110 PRINT :PRINT :PRINT "                H - O"
115 POSITION 17,18:PRINT "2"
120 PRINT :PRINT "ENDE":END
```

5. SCHRITT : Variablenliste

In diesem Programm werden keine Variablen benutzt. Deshalb entfällt hier dieser Arbeitsschritt.

6. SCHRITT : Programmbeschreibung

Satz 10-50 : Überschrift.

Satz 60 : Ausgabe eines Kügelchens in Spalte 16, Zeile 11.

Satz 70 : Ausgabe eines Schrägstrichs in Spalte 18, Zeile 12.

Satz 80 : Ausgabe eines Karos in Spalte 20, Zeile 13.

Satz 90 : Ausgabe eines Kügelchens in Spalte 16, Zeile 15.

Satz 100 : Ausgabe eines Schrägstrichs in Spalte 18, Zeile 14.

Satz 110 : Ausgabe von H — O nach zwei Leerzeilen.

Satz 115 : Ausgabe einer tiefergestellten 2.

Satz 120 : Ende des Programms.

7. SCHRITT : Ergebnisse

Die Bildschirmausgabe dieses Programms entspricht dem Bild, das wir in Abschnitt 3.1 schon vorgestellt haben. Eine weitere Präsentation ist deshalb hier entbehrlich.

3.3 Reaktionsgleichung

1. SCHRITT : Vorstellung des Problems

Auch chemische Reaktionen lassen sich anschaulich graphisch darstellen, wie am Beispiel der für die Stahlerzeugung notwendigen Prozesse gezeigt werden kann.

Die Stahlerzeugung in den industrialisierten Ländern beginnt damit, daß im Hochofen zunächst Roheisen produziert wird. Diese Produktion geht in folgenden Schritten vor sich:

1. Schritt: Erhitzte Luft wird mit Koks zu Kohlenmonoxid verbrannt:

$$2\ C + O_2 \longrightarrow 2\ CO$$

Koks	Sauerstoff	Kohlenmonoxid

2. Schritt: Das heiße Kohlenmonoxid (CO) steigt im Hochofen auf und reagiert dort mit Eisenoxid (Fe_2O_3):

$$Fe_2O_3 + 3\ CO \longrightarrow 2\ Fe + 3\ CO_2$$

Eisen-oxid	Kohlen-monoxid	Eisen	Kohlen-dioxid

Damit steht dann Roheisen für die weitere Verwendung zur Verfügung.

Es soll nun ein BASIC-Programm vorgestellt werden, welches diese chemischen Reaktionsbeziehungen veranschaulicht.

Bei der Darstellung der Atome und der Moleküle bedienen wir uns der gleichen graphischen Möglichkeiten wie im Beispiel zuvor.

2. SCHRITT : Problemanalyse

Mit der obigen Beschreibung und unter Rückgriff auf die Überlegungen im vorangegangenen Beispiel ist im Grunde die Problemanalyse schon geleistet, so daß weitere Ausführungen an dieser Stelle entbehrlich sind.

3. SCHRITT : Flußdiagramm

Auch das Flußdiagramm ist hier nicht mehr erforderlich, wobei die gleichen Gründe maßgeblich sind wie im Schritt zuvor.

4. SCHRITT : Programm

```
10 REM C2-CHEM.REAKTION
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR DARSTELLUNG EINER CHEMI-"
40 PRINT "              SCHEN REAKTION."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF.DR.W.VOSS, 1984"
60 PRINT :PRINT :PRINT
70 POSITION 6,9:PRINT CHR$(20)
80 POSITION 22,10:PRINT CHR$(20);CHR$(96)
90 POSITION 9,11:PRINT "+  ";CHR$(96);CHR$(96)
100 POSITION 17,11 :PRINT "="
110 POSITION 22,12:PRINT CHR$(20);CHR$(96)
120 POSITION 6,13:PRINT CHR$(20)
130 POSITION 5,16:PRINT "2 C +  O    =   2 CO"
140 POSITION 13,17:PRINT "2"
150 PRINT :PRINT :PRINT
155 PRINT "         BITTE CONT EINGEBEN" :STOP
```

```
160 PRINT CHR$(125)
170 POSITION 29,2:PRINT CHR$(96)
180 POSITION 22,3:PRINT CHR$(160);"      ";CHR$(20)
190 POSITION 29,4:PRINT CHR$(96)
200 POSITION 6,6:PRINT CHR$(96);"      ";CHR$(20);CHR$(96)
210 POSITION 5,7:PRINT CHR$(160)
220 POSITION 29,7:PRINT CHR$(96)
230 POSITION 6,8:PRINT CHR$(96);"  +  ";CHR$(20)
240 POSITION 13,8:  PRINT CHR$  (96) ;"   ="
250 POSITION 25,8:PRINT "+  ";CHR$(20)
260 POSITION 5,9:PRINT CHR$(160)
270 POSITION 29,9:PRINT CHR$(96)
280 POSITION 6,10:PRINT CHR$(96)
290 POSITION 12,10:PRINT CHR$(20);CHR$(96)
300 POSITION 29,12:PRINT CHR$(96)
310 POSITION 22,13:PRINT CHR$(160)
320 POSITION 28,13:PRINT CHR$(20)
330 POSITION 29,14:PRINT CHR$(96)
340 POSITION 3,17:PRINT "FE O  + 3 CO  =  2 FE + 3 CO"
350 POSITION 5,18:PRINT "2 3"
360 POSITION 31,18:PRINT "2"
370 PRINT :PRINT "ENDE":END
```

5. SCHRITT : Variablenliste

Es werden keine Variablen benutzt.

6. SCHRITT : Programmbeschreibung

Satz 1Ø-6Ø : Überschrift.

Satz 7Ø : Ausgabe eines Kügelchens in Spalte 6, Zeile 9.

Satz 8Ø : Ausgabe eines Schrägstrichs in Spalte 18, Zeile 12.

Satz 9Ø : Ausgabe eines Kügelchens, daneben eines Karos in Spalte 22/23, Zeile 1Ø.

Satz 1ØØ : Ausgabe eines =-Zeichens in Spalte 17, Zeile 11.

Satz 11Ø : Wie Satz 8Ø, nun aber in Zeile 12.

Satz 12Ø : Wie Satz 7Ø, nun in Zeile 13.

Satz 13Ø -14Ø : Ausgabe des Textes:
$2\ C + O_2 = 2\ CO$

Satz 15Ø -16Ø : Abwarten und Löschen des Bildschirms.

Satz 17Ø -33Ø : Ausgabe des graphischen Abbilds der zweiten Reaktionsgleichung.

Satz 34Ø -35Ø : Ausgabe der zweiten Reaktionsgleichung.

Satz 37Ø : Ende des Programms.

7. SCHRITT : Ergebnisse

Das vorgestellte Programm erzeugt nacheinander zwei graphische Bilder auf dem Bildschirm; das zweite Bild erscheint, wenn der Programmbenutzer das Kommando CONT eingegeben hat.

Es dürfte entbehrlich sein, diese beiden Bilder hier vorzustellen, insbesondere auch deshalb, weil hier keine farbliche Darstellung möglich ist. Sinnvoller ist es deshalb, dieses Programm direkt an einem Farbfernseher zu erproben.

3.4 Stöchiometrisches Rechnen

1. SCHRITT : Vorstellung des Problems

Beim stöchiometrischen Rechnen geht es darum, die Stoffmengen zu quantifizieren, die bei chemischen Reaktionen umgesetzt werden. Es könnte damit also zum Beispiel die folgende Frage beantwortet werden:

Wieviel Gramm Schwefel sind notwendig, um 100 Gramm Zink vollständig zu Zinksulfid umzusetzen? (Schwefel = S; Zink = Zn; Zinksulfid = ZnS)

Atomgewicht von Schwefel : 32.1
Atomgewicht von Zink : 65.4

2. SCHRITT : Problemanalyse

Zur Lösung des gestellten Problems ist von der zuständigen Reaktionsbeziehung auszugehen:

$$Zn + S \longrightarrow ZnS$$

Wenn wir die sog. Molmassen des gesuchten und des gegebenen Stoffs aufeinander beziehen, erhalten wir:

32.1 g Schwefel sind notwendig, um 65.4 g Zink umzusetzen (s.o.).

Nun gilt offenbar:

$$x/100 = 32.1/65.4$$

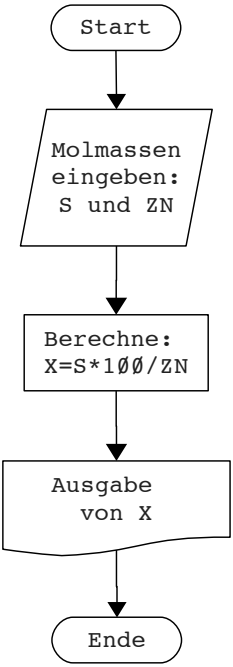
oder

$$x = 3210/65.4$$

Dabei ist x der gesuchte Wert.

3. SCHRITT : Flußdiagramm

Das Flußdiagramm ist bei dieser Fragestellung wieder sehr einfach:



4. SCHRITT : Programm

```
10 REM C3-CHEM.RECHNEN
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM STOECHIOMETRISCHEN RECH-"
40 PRINT "                NEN."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF.DR.W.VOSS, 1984"
50 DIM A$(10),B$(10)
60 PRINT :PRINT "BEISPIEL :":PRINT :PRINT
70 PRINT "ZINK + SCHWEFEL = ZINKSULFID :"
80 PRINT :PRINT " ZN + S          =    ZN S":PRINT
90 PRINT :PRINT "WIEVIEL GRAMM SCHWEFEL SIND NOTWEN-"
100 PRINT "DIG, UM 100 G ZINK VOLLSTAENDIG ZU"
110 PRINT "ZINKSULFID UMZUSETZEN ?":PRINT :PRINT
120 PRINT "1. ELEMENT : ";:INPUT A$
130 PRINT "2. ELEMENT : ";:INPUT B$
140 PRINT :PRINT :PRINT
150 PRINT "ATOMGEWICHT VON ";A$
152 POSITION 30,22:PRINT " : ";:INPUT A
155 PRINT "ATOMGEWICHT VON ";B$
160 POSITION 30,22:PRINT " : ";:INPUT B
170 X=B*100/A:PRINT :PRINT :PRINT
180 PRINT "BENOETIGTE MENGE VON ";B$;" : "
185 PRINT :PRINT "      ";X;" G"
190 PRINT :PRINT "ENDE DER BERECHNUNG":END
```

5. SCHRITT : Variablenliste

A = Atomgewicht Stoff A
A$ = Bezeichnung Stoff A

B = Atomgewicht Stoff B
B$ = Bezeichnung Stoff B

X = Gesuchte Menge

6. SCHRITT : Programmbeschreibung

Satz 10-110 : Überschrift und Beispielangabe mit der in diesem Programm zu behandelnden Fragestellung.

Satz 50 : Dimensionieren der Strings.

Satz 120-130 : Eingabe der Namen der beiden Stoffe.

Satz 140 : 3 Leerzeilen.

Satz 150-160 : Ausgabe der Namen der Stoffe und Anforderung der Atomgewichte dieser Stoffe.

Satz 170 : Berechnung der gesuchten Menge und 3 Leerzeilen.

Satz 180-185 : Ausgabe des Ergebnisses (zuzüglich einiger Leerzeilen, auf die aber zukünftig nicht immer gesondert aufmerksam gemacht werden soll).

Satz 190 : Ende des Programms.

7. SCHRITT : Ergebnisse

Nach den erläuternden Erklärungen fordert das Programm den Namen des ersten Elements an:

Geben wir z.B. ein "Schwefel", so fordert das Programm den Namen des zweiten Elements, das in die Rechnung eingehen soll, an:

Geben wir z.B. ein "Zink", so fordert das Programm die Atomgewichte für Schwefel und Zink an.

Wenn wir daraufhin eingeben die Werte 32.1 und 65.4, dann liefert das Programm das folgende Ergebnis:

```
BENOETIGTE MENGE VON ZINK :

     2Ø3.738317 G

ENDE DER BERECHNUNG
```

Es versteht sich, daß bei anderen chemischen Prozessen (z.B. auch dann, wenn mehr als zwei Stoffe an den in Frage stehenden Reaktionen beteiligt sind), das obige Programm unter Umständen sehr grundlegend verändert werden muß.

Hier kam es nur darauf an, beispielhaft ein derartiges Programm einmal vorzustellen.

3.5 Das Periodensystem der Elemente

1. SCHRITT : Vorstellung des Problems

Ein Rechner kann nicht nur dafür eingesetzt werden, Rechnungen durchzuführen, wie das zum Beispiel im vorangegangenen Abschnitt der Fall war, oder anschauliche graphische Abbilder zu erzeugen (wie im Abchnitt davor), sondern er kann - gerade im schulischen Bereich - als "Auskunftei" benutzt werden.

Am Beispiel des Periodensystems der Elemente kann verdeutlicht werden, wie ein solches "Nachschlageprogramm", das gewissermaßen auch als Trainingsprogramm verstanden werden kann, aussehen könnte.

Beispielsweise könnte der Rechenbenutzer daran interessiert sein, nach Eingabe eines Elementnamens (z.B. Schwefel) vom Rechner wesentliche Informationen über dieses chemische Element zu erfahren.

Solche Informationen sind etwa:

1. Chemisches Zeichen,
2. Ordungszahl im Periodensystem,
3. Atomgewicht,
4. Wichte,
5. Schmelzpunkt,
6. Siedepunkt,
7. Gruppenzugehörigkeit (I-VIII,Ø),
8. Nebengruppenzugehörigkeit (a,b)
9. Elektronenschalen (K-Q)

2. SCHRITT : Problemanalyse

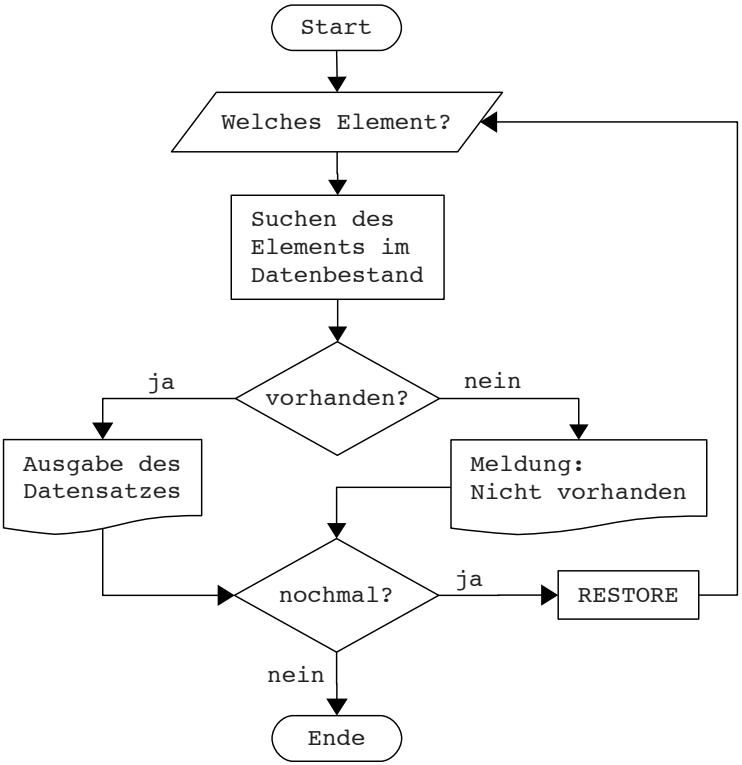
pie Entwicklung eines entsprechenden Computerprogramms, das die interessierenden Auskünfte bereitstellen kann, bereitet keine besonderen Schwierigkeiten.

Der Benutzer gibt über I NPUT an, über welches Element er Informationen wünscht und das Programm stellt dann den gesamten Datensatz für dieses Element zur Verfügung, indern es ihn über READ aus DATA-Statements herausgreift.

Schließlich kann abgefragt werden, ob für ein weiteres chemisches Element diese Informationen gewünscht werden. Wenn ja, muß über RESTORE der Datenbestand wieder zugänglich gemacht werden.

3. SCHRITT : Flußdiagramm

Entsprechend einfach wie die Problemanalyse im vorangegangenen Abschnitt ist auch das Flußdiagramm:



4. SCHRITT : Programm

```
10 REM C4-ELEMENTE
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR ANGABE WESENTLICHER "
40 PRINT "EIGENSCHAFTEN CHEMISCHER ELEMENTE."
45 PRINT :PRINT :PRINT
47 PRINT "         PROF.DR.W.VOSS, 1984"
60 N=17:REM ANZAHL DER AUFGENOMMENEN ELEMENTE
70 DIM E$(20),AB$(2),G$(4),N$(1),S$(1)
75 DIM A$(1),B$(20)
77 PRINT :PRINT :PRINT
80 PRINT "WELCHES ELEMENT (NAME BITTE) :"
90 PRINT :INPUT B$
100 RESTORE
110 FOR I=1 TO N
120 READ E$,AB$,O,G,W,S1,S2,G$,N$,S$
130 IF E$=B$ THEN GOTO 160
140 NEXT I
150 PRINT "ELEMENT NICHT IM BESTAND":GOTO 290
160 PRINT CHR$(125)
170 PRINT E$:POSITION 18,1:PRINT AB$
180 PRINT :PRINT
190 PRINT "ORDNUNGSZAHL  :  ";O
200 PRINT "GRUPPE        :  ";G$
210 PRINT "NEBENGRUPPE   :  ";N$
220 PRINT "SCHALE        :  ";S$
230 PRINT :PRINT
240 PRINT "ATOMGEWICHT   :  ";G
250 PRINT "WICHTE        :  ";W
260 PRINT
270 PRINT "SCHMELZPUNKT  :  ";S1
280 PRINT "SIEDEPUNKT    :  ";S2
290 PRINT :PRINT :PRINT "EIN ANDERES ELEMENT (J/N) ";
295 INPUT A$
300 IF A$="J" THEN PRINT CHR$(125):GOTO 80
310 PRINT :PRINT "ENDE DER AUSGABE":END
```

```
500 DATA ALUMINIUM,AL,13,26.97,2.7,658,2270,III,B,M
510 DATA ANTIMON,SB,51,121.76,6.67,630,1440,V,B,O
520 DATA BARIUM,BA,56,137.36,3.5,704,1537,II,A,P
530 DATA BLEI,PB,82,207.21,11.34,327,1690,IV,B,P
540 DATA CHLOR,CL,17,35.457,1.557,-100,-34,VII,B,M
550 DATA EISEN,FE,26,55.85,7.86,1525,2450,VIII,0,N
560 DATA FLUOR,F,9,19,.0017,-218,-187,VII,B,L
570 DATA GOLD,AU,79,197,19.25,1063,2677,I,B,P
580 DATA HELIUM,HE,2,4.003,.000018,-272,-269,0,0,K
590 DATA JOD,J,53,126.92,4.942,114,184,VII,B,O
600 DATA KADMIUM,CD,48,112.41,8.64,321,770,II,B,O
610 DATA NATRIUM,NA,11,22.991,.971,98,880,I,A,M
620 DATA PHOSPHOR,P,15,30.98,1.83,44,280,V,B,M
630 DATA SAUERSTOFF,O,8,16,0.0014,-218.7,-182.97,VI,B,L
640 DATA SCHWEFEL,S,16,32.066,2.07,112,444,VI,B,M
650 DATA WASSERSTOFF,H,1,1.008,.00009,-262,-252.78,I,A,K
660 DATA ZINK,ZN,30,65.38,7.14,419,906,II,B,N
```

Anzumerken ist, daß sich im Datenbestand dieses Programms nur 17 Elemente befinden. Es ist aber ohne Schwierigkeiten zu erkennen, wie das Programm verändert werden müßte, wenn man mehr als 17 oder vielleicht alle bekannten chemischen Elemente berücksichtigen wollte.

5. SCHRITT : Variablenliste

A$ = Antwortstring (Ja/Nein)
AB$ = Elementabkürzung
B$ = Speicherfeld zur Aufnahme des vom Benutzer gewünschten Elements
E$ = Elementname
G = Atomgewicht
G$ = Elementgruppe
I = Laufindex
N = Anzahl der Elemente
N$ = Nebengruppe
O = Ordnungszahl
S1 = Schmelzpunkt
S2 = Siedepunkt
W = Wichte

6. SCHRITT : Programmbeschreibung

Satz 1Ø-5Ø : Überschrift.

Satz 6Ø : Angabe der Elementanzahl (es sind in dieses Programm nicht alle chemischen Elemente aufgenommen, sondern nur 17 Stück. Ggf. muß also Satz 6Ø verändert werden, falls die DATA-Statements verlängert werden sollten).

Satz 7Ø-75 : Dimensionierungen.

Satz 77-9Ø : Anforderung des Elements, über das Informationen gewünscht werden.

Satz 11Ø-14Ø : Einlesen der Datensätze.

Satz 13Ø : Wenn das gewünschte Element gefunden wird, Sprung nach 16Ø.

Satz 15Ø : Satz 15Ø wird nur erreicht, wenn das gewünschte Element nicht im Datenbestand ist; dann Sprung nach 29Ø.

Satz 16Ø-28Ø : Ausgabe der gewünschten Informationen nach Löschen des Bildschirms.

Satz 29Ø-295 : Abfrage, ob eine weitere Auskunft gewünscht wird;
wenn ja, Löschen des Bildschirms und Sprung nach 8Ø;
wenn nein, weiter bei 21Ø.

Satz 31Ø : Ende des Programms.

Satz 5ØØ ff Datenbestand in DATA-Statements

7. SCHRITT : Ergebnisse

Geben wir z.B. als gewünschtes Element das Element Schwefel ein, so antwortet der Rechner:

```
SCHWEFEL        S


ORDNUNGSZAHL  : 16
GRUPPE        : VI
NEBENGRUPPE   : B
SCHALE        : M


ATOMGEWICHT   : 32.Ø66
WICHTE        : 2.Ø7

SCHMELZPUNKT  : 112
SIEDEPUNKT    : 444


EIN ANDERES ELEMENT (J/N) ?
```

4

Kapitel 4: Physik

4.1 Vorbemerkung

Der Bereich der Schulphysik bietet ähnliche Problemstellungen, wie wir sie auch schon teilweise im Bereich der Mathematik vorgefunden haben (siehe Kapitel 2). Allerdings kommen hier nun häufig einige Besonderheiten hinzu, die es nahelegen, wieder von den Graphikmöglichkeiten des Rechners Gebrauch zu machen.

Deshalb soll - über das hinaus, was im vorangegangenen Kapitel schon über die Methoden der Graphikprogrammierung gesagt wurde - in diesem einleitenden Abschnitt noch einmal auf die Graphik-BASIC-Anweisungen etwas näher eingegangen werden.

4.2 Graphik-Programmierung

Verwendet man den ATARI zur Erstellung von Graphiken, so sind zwei Möglichkeiten zu unterscheiden:

1. Normalgraphik (auch Blockgraphik genannt)

2. Hochauflösende Graphik

Um die eine oder die andere Art der graphischen Darstellung benutzen zu können , ist es notwendig, den Bildschirmmodus zu ändern, der bislang nur im sog. Textmodus, also zur Ausgabe von Texten benutzt wurde.

Wir benötigen hier das folgende Statement (das übrigens im Direktmodus auch als Kommando benutzt werden kann):

Statement 17: nn GRAPHICS Zahl

An der Stelle "Zahl" in diesem Statement kann ein Wert zwischen Ø und 15 auftauchen, wobei sehr unterschiedliche Wirkungen bezüglich der Bildschirmaufteilung (und der Farbgebung , falls ein Farbmonitor angeschlossen ist) zu beobachten sind.

Für unsere Zwecke genügt es, die folgenden drei Möglichkeiten zu unterscheiden:

1. GRAPHICS Ø : schaltet (zurück) in den Textmodus.

2. GRAPHICS 3: Umschalten in den Blockgraphikmodus .

3. GRAPHICS 8: Umschalten in den hochauflösenden Graphikmodus.

Im Blockgraphikmodus hat man sich den Bildschirm aufgeteilt zu denken in 4Ø Spalten (Ø bis 39) und 2Ø Zeilen (Ø bis 19) zuzüglich vier Textzeilen am unteren Bildschirmrand.

Zeile Ø ist ganz oben, Spalte Ø ganz links auf dem Bildschirm.

Im hochauflösenden Graphikmodus hat man sich in entsprechen der Weise den Bildschirm aufgeteilt zu denken in 32Ø Spalten und 16Ø Zeilen.

Im Blockgraphikmodus stehen vier Farben zur Verfügung. Sie werden mit folgendem Statement erzeugt:

Statement 18: nn COLOR Zahl

Es gilt folgende Wertetabelle:

Zahl	Farbe
Ø	Hintergrundfarbe
1	orange
2	hellgrün
3	dunkelblau

Faktisch stehen also drei Farben zum Zeichnen zur Verfügung.

In der hochauflösenden Graphik stehen nur zwei Farben zur Verfügung, von denen eine der Hintergrundfarbe entspricht, so daß faktisch nur einfarbig gezeichnet werden kann.

Will man nun auf dem Bildschirm einen Punkt zeichnen, so dient dazu die folgende Anweisung:

Statement 19: nn PLOT S,Z

Dabei steht S für die Spalte und Z für die Zeile, in die gezeichnet werden soll.

Der Leser erprobe das folgende Programm:

```
10 GRAPHICS 3
20 COLOR 2
30 PLOT 19,10
40 END
```

Das Ergebnis ist ein hellgrünes Quadrat auf der Mitte des Bildschirms. *)

Entsprechend einfach ist das Ziehen von geraden Linien mit dem folgenden Statement:

Statement 20: nn DRAWTO S,Z

* Anmerkung: Die Zeichenfunktionen verwenden die zuletzt ausgewählte Farbe. Nach dem Start des Rechners ist das die Hintergrundfarbe (= COLOR 0). Zeichnen mit schwarzer Farbe auf schwarzem Hintergrund führt zu einem unsichtbaren Ergebnis.

4.3 BASIC-Programme

So zieht beispielsweise das Statement

```
DRAWTO 3Ø,1Ø
```

einen geraden Strich von der jeweiligen Position des Graphik-Cursors ausgehend bis zum Koordinatenpunkt (3Ø,1Ø).

Betrachten wir nun einige Programmbeispiele zunächst für den Blockgraphikmodus:

Das folgende Programm zieht einen orangefarbenen senkrechten Strich in die Mitte des Bildschirms

```
10 GRAPHICS 3
20 COLOR 1
30 PLOT 19,0
40 DRAWTO 19,15
50 END
```

Der Leser möge daran denken, daß er nach Beendigung des Programms durch das Kommando

```
GRAPHICS Ø
```

wieder in den gewohnten Textmodus zurückkehren kann.

Entsprechend zeichnet das folgende Programm drei verschiedenfarbige waagrechte Balken:

```
10 GRAPHICS 3
20 FOR I=1 TO 3
30 COLOR I
40 Z=3*I
50 PLOT 0,Z
60 DRAWTO 20,Z
70 NEXT I
80 END
```

Nun einige Programmbeispiele für die hochauflösende Graphik.

Das folgende Programm zeichnet ein Achsenkreuz

```
10 GRAPHICS 8
20 COLOR 1
30 PLOT 20,150:DRAWTO 200,150
40 PLOT 20,150:DRAWTO 20,40
50 END
```

Wenn man sich genau überlegt, welche Möglichkeiten diese hochauflösende Graphik bietet, so sieht man sofort, daß sie beispielsweise sehr gut dazu geeignet ist, geometrische Figuren zu erzeugen.

Das folgende Programm zeichnet ein Rechteck:

```
10 GRAPHICS 8
20 COLOR 1
30 PLOT 20,150:DRAWTO 200,150
40 DRAWTO 200,40
50 DRAWTO 20,40
60 DRAWTO 20,150
70 END
```

Damit nähern wir uns schon den Möglichkeiten der "Computerkunst", wie das folgende Programm verdeutlicht:

Dieses Programm zeichnet fünf verschiedene Rechtecke, wobei deren Lage, ihre Länge und ihre Breite zufällig ausgewählt wird - allerdings wird dabei darauf geachtet, daß der zulässige Bildschirmbereich nicht verlassen wird.

```
10 GRAPHICS 8
15 COLOR 1
20 FOR I=1 TO 5
30 S=RND(1)*210
35 S1=S
40 Z=RND(1)*45
45 Z1=Z
50 L=RND(1)*100
60 B=RND(1)*100
70 PLOT S,Z
80 S=S+L
90 IF S>319 THEN S=319
100 DRAWTO S,Z
110 Z=Z+B
120 IF Z>159 THEN Z=159
130 DRAWTO S,Z
140 S=S-L
150 IF S<0 THEN S=0
160 DRAWTO S,Z
170 Z=Z-B
180 IF Z<0 THEN Z=0
190 DRAWTO S1,Z1
200 NEXT I
210 END
```

Genauso gut wie in der hochauflösenden Graphik Linien parallel zu den Bildschirmzeilen und Bildschirmspalten gezogen werden können, ist es aber auch möglich, mit dem DRAWTO-Statement schräge Linien zu ziehen.

Das folgende Programm zeichnet ein Rechteck und seine Diagonalen:

```
10 GRAPHICS 8
15 COLOR 1
20 PLOT 20,20
30 DRAWTO 20,100
40 DRAWTO 150,100
50 DRAWTO 150,20
60 DRAWTO 20,20
70 DRAWTO 150,100
80 PLOT 150,20
90 DRAWTO 20,100
100 END
```

Nach dem gleichen Prinzip ist das folgende Programm aufgebaut, welches eine Schraffur auf dem Bildschirm erzeugt:

```
10 GRAPHICS 8
15 COLOR 1
20 FOR I=1 TO 200 STEP 20
30 PLOT I,10
40 J=I+50
50 DRAWTO J,100
60 NEXT I
70 END
```

Diese Beispiele mögen genügen, um hier einleitend die Möglichkeiten der hochauflösenden Graphik zu illustrieren.

Wenden wir uns abschließend noch einmal der Normalgraphik zu, bei der wir ja auch mit unterschiedlichen Farben arbeiten können.

Das folgende Programm erzeugt mit Hilfe der PLOT-Anweisung ein Zufalls-Farbmuster (dem Leser wird empfohlen, das Programm mehrfach zu starten):

```
10 GRAPHICS 3
20 FOR S=10 TO 30
30 FOR Z=5 TO 15
40 F=INT(RND(1)*3+0.5)
50 COLOR F
60 PLOT S,Z
70 NEXT Z
80 NEXT S
90 END
```

Auch das letzte Programm in dieser kleinen Beispielsammlung erinnert an erste Ansätze geometrischer "Computerkunst". Es zeichnet drei beliebige Rechtecke in drei verschiedenen Farben:

```
10 GRAPHICS 3
20 FOR I=1 TO 3
30 COLOR I
40 SA=INT(RND(1)*29+0.5)
50 SE=SA+INT(RND(I)*10+0.5)
60 ZA=INT(RND(1)*12+0.5)
70 ZE=ZA+INT(RND(I)*7+0.5)
80 FOR Z=ZA TO ZE
90 FOR S=SA TO SE
100 PLOT S,Z
110 NEXT S
120 NEXT Z
130 NEXT I
140 END
```

4.4 Die Federwaage

1. SCHRITT : Vorstellung des Problems

Wenn man Werkstücke unterschiedlicher Gewichte an eine Federwaage hängt, so wird diese unterschiedlich weit ausgedehnt, d.h. sie reagiert in unterschiedlicher Weise auf die verschiedenen Gewichtskräfte.

Es gilt dabei ein Proportionalitätsgesetz, das besagt, daß z.B. bei Verdopplung der Gewichtskraft auch die erzielte Verlängerung der Federwaage sich verdoppelt.

Es gibt also eine sog. Federkonstante, nämlich den Quotienten aus Gewichtskraft Fund Federverlängerung s, d.h. es gilt:

$$D = F/s = \text{konstant}$$

Wenn beispielsweise D=20 ist, heißt dies, daß eine Gewichtskraft von 100 (cN) zu einer Federverlängerung von 5 cm, eine Gewichtskraft von 200 (cN) zu einer Federverlängerung von 10 cm führt etc.

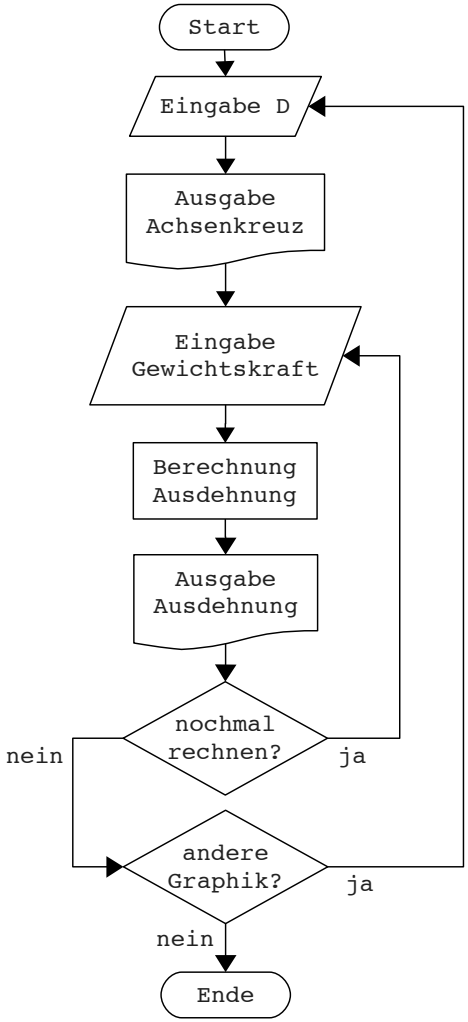
2. SCHRITT : Problemanalyse

Der oben geschilderte Zusammenhang läßt sich (z.B. für D = 2Ø) folgendermaßen graphisch darstellen:

Das hier zu entwickelnde BASIC-Programm soll für alternative Werte der Federkonstanten D das entsprechende graphische Bild auf dem Bildschirm ausgeben, so daß also die folgenden Programmteile erforderlich werden:

- Erzeugung eines Achsenkreuzes,
- Eingabe eines alternativen D-Wertes,
- Ausgabe der Konstanzlinie (siehe obige Abbildung),
- Beantwortung der Frage nach der Ausdehnung der Federwaage bei Eingabe einer beliebigen Gewichtskraft.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM P1-FEDERWAAGE
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR GRAPHISCHEN VERANSCHAU-"
40 PRINT "LICHUNG DES FEDERWAAGEN-GESETZES."
45 PRINT :PRINT : PRINT
50 PRINT "          PROF.DR.W.VOSS, 1984"
52 DIM A$(1)
55 PRINT :PRINT :PRINT :PRINT
60 PRINT "WERT DER FEDERKONSTANTEN D : ";
65 INPUT D
70 GRAPHICS 8
75 COLOR 1
80 PLOT 50,150:DRAWTO 200,150
90 PLOT 50,150:DRAWTO 50,50
110 PLOT 50,150
120 B=D/10
130 FOR A=50 TO 200
140 E=150-B*(A-50)
150 IF E<=50 THEN GOTO 170
160 PLOT A,E
170 NEXT A
180 FOR I=1 TO 1000:NEXT I
190 GRAPHICS 0
200 PRINT "KRAFT F           : ";: INPUT F
210 FW=F/D
220 PRINT : PRINT "FEDERKONSTANTE   : ";D
230 PRINT : PRINT "FEDERWEG         : ";FW
240 PRINT :PRINT :PRINT
250 PRINT "NOCH EINE RECHNUNG (J/N) ";
260 INPUT A$
270 IF A$="J" THEN PRINT :PRINT :GOTO 200
280 PRINT :PRINT :PRINT
290 PRINT "ANDERE GRAPHIK     (J/N) ";
300 INPUT A$
310 IF A$="J" THEN GOTO 55
320 PRINT :PRINT :PRINT "ENDE":END
```

5. SCHRITT : Variablenliste

A = Spalte für das Zeichnen der Konstanzlinie

A$ = Antwortstring (Ja/Nein)

B = D/1Ø (Größe zur Normierung der Bildschirmausgabe)

D = Federkonstante

E = Zeile für das Zeichnen der Konstanzlinie

F = Kraft

FW = Federweg
I = Laufindex

6. SCHRITT : Programmbeschreibung

Satz 1Ø-5Ø : Überschrift.

Satz 52 : Dimensionieren von A$.

Satz 6Ø-65 : Eingabe der Federkonstanten D.

Satz 7Ø-9Ø : Zeichnen des Achsenkreuzes
Ursprung: 5Ø,15Ø.

Satz 11Ø-17Ø : Zeichnen der Konstanzlinie. Dabei werden Koordinatenschnittpunkte außerhalb der Fläche, die das Achsenkreuz einschließt, nicht gezeichnet.

Satz 18Ø-19Ø : Warteschleife und Rückkehr in den Textmodus.

Satz 2ØØ : Anforderung einer Kraft F.

Satz 21Ø : Berechnung des Federweges.

Satz 220-240 : Ausgabe der Ergebnisse.

Satz 250-270 : Abfrage, ob noch eine Rechnung;

wenn ja, zurück nach 200,
wenn nein, weiter bei 280.

Satz 290-310 : Abfrage, ob eine andere Graphik;

wenn ja, zurück nach Satz 55 nach Löschen des Bildschirms,
wenn nein, weiter bei Satz 320.

Satz 320 : Ende des Programms.

7. SCHRITT : Ergebnisse

Nach der Eingabe der Input-Information erzeugt das Programm eine Graphik, wie sie schematisch auf einer der vorangegangenen Seiten schon vorgestellt wurde.

Nach einer Pause fordert dann das Programm eine Kraft F an und berechnet den dazugehörigen Federweg. Ein Rechenbeispiel dazu dürfte sich an dieser Stelle erübrigen (vergl. dazu 1. Schritt, wo sich ein entsprechendes Beispiel ja findet).

4.5 Der Satz des Archimedes

1. SCHRITT : Vorstellung des Problems

Der Satz des Archimedes lautet:

Der Auftrieb, den ein in eine Flüssigkeit getauchter Körper erfährt, ist gleich dem Gewicht der verdrängten Flüssigkeit.

Dieser Satz beschreibt die bekannte Erfahrung, daß es zunehmender Kraft bedarf, um zum Beispiel einen leeren Eimer mit dem Boden nach unten unter Wasser zu drücken.

Ein immer tiefer in eine Flüssigkeit eintauchender Körper erfährt einen zunehmenden scheinbaren Gewichtsverlust, der durch den wachsenden Auftrieb vorgetäuscht wird.

Es soll nun ein Programm vorgestellt werden, welches auf der Grundlage dieser Beziehungen für einen unbekannten Körper, der in eine bekannte Flüssigkeit eingetaucht wird (z.B. in Wasser), Volumen und Wichte (spezifisches Gewicht) bestimmt, nachdem sein Gewicht an der Luft bekannt ist.

2. SCHRITT : Problemanalyse

Wie bei jeder konkreten Problemstellung, ist auch hier nach dem folgenden bewährten Muster zu verfahren:

Das heißt, wir überlegen zunächst, welche Eingabeinformationen der Rechner benötigt, wenn er das oben formulierte Problem lösen soll. Er benötigt folgendes:

E1. Gewicht des unbekannten Körpers in der Luft;

E2. Scheinbares Gewicht des Körpers in der Flüssigkeit, in die er eingetaucht wird;

E3. Wichte der Flüssigkeit, die zur Überprüfung verwendet wird.

Wenn diese Informationen vorliegen, dann kann im Verarbeitungsschritt bestimmt werden:

V1. Auftrieb und Gewicht der verdrängten Flüssigkeitsmenge;

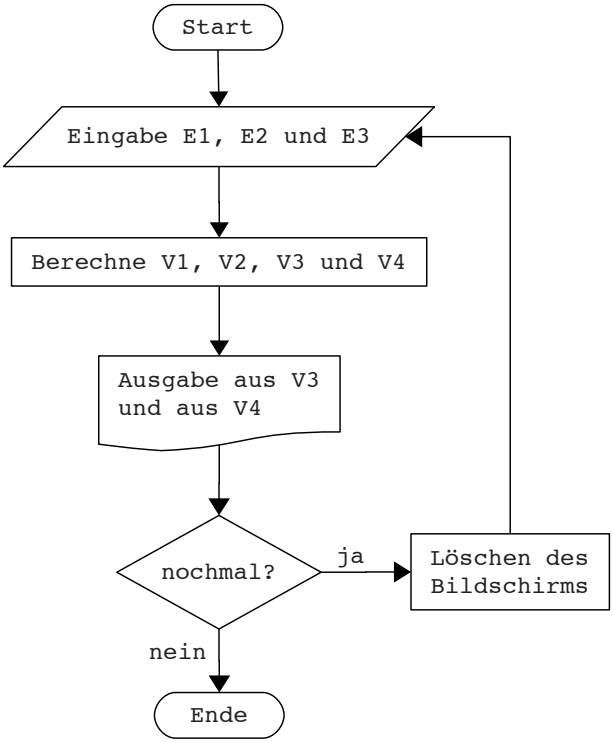
V2. Volumen dieser Flüssigkeitsmenge;

V3. Volumen des verdrängenden, zu prüfenden Körpers;

V4. Wichte des zu prüfenden, eingetauchten Körpers.

Als Ausgabe sind schließlich die in den letzten Punkten V3 und V4 erzielten Ergebnisse vorzusehen.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM P2-ARCHIMEDES
15 DIM A$(1)
20 PRINT CHR$(125)
30 PRINT "     DAS GESETZ DES ARCHIMEDES"
40 PRINT :PRINT
50 PRINT "        PROF.DR.W.VOSS,  1984"
55 PRINT :PRINT :PRINT
60 PRINT "MIT DEM SATZ DES ARCHIMEDES KOENNEN"
65 PRINT "VOLUMEN UND WICHTE EINES UNBEKANNTEN"
70 PRINT "KOERPERS BESTIMMT WERDEN":PRINT
75 PRINT "EINZUGEBEN SIND DIE FOLGENDEN INFOR-"
80 PRINT "MATIONEN : ":PRINT :PRINT
85 PRINT "     - GEWICHT DES KOERPERS AN DER"
90 PRINT "       LUFT  (GL)":PRINT
95 PRINT "     - SCHEINBARES GEWICHT DES KOER-"
100 PRINT "       PERS IN DER FLUESSIGKEIT  (GS)"
110 PRINT :PRINT "     - WICHTE DER FLUESSIGKEIT  (WF)"
140 PRINT :PRINT "GL  = ";:INPUT GL
150 PRINT :PRINT "GS  = ";:INPUT GS
160 PRINT :PRINT "WF  = ";:INPUT WF
170 GV=GL-GS
175 VV=GV/WF
180 VK=VV
190 WK=GL/VK
200 PRINT CHR$(125)
210 PRINT "ZU PRUEFENDER KOERPER : ":PRINT
220 PRINT :PRINT "     VOLUMEN       : ";VK
230 PRINT :PRINT "     WICHTE        : ";WK
240 PRINT :PRINT :PRINT "NOCHMAL (J/N) ";:INPUT A$
250 IF A$="J" THEN PRINT CHR$(125):GOTO 75
260 PRINT :PRINT :PRINT "ENDE":END
```

5. SCHRITT : Variablenliste

A$ = Stringvariable für die Antworten Ja/Nein

GL = Gewicht des Körpers an der Luft
GS = Scheinbares Gewicht des Körpers in der Flüssigkeit
GV = Scheinbarer Gewichtsverlust

VK = Volumen der verdrängten Flüssigkeit
VV = Volumen des Körpers

WF = Wichte der Flüssigkeit
WK = Wichte des Körpers

6. SCHRITT : Programmbeschreibung

Satz 10-130 : Überschrift und erläuternde Informationen und Dimensionieren von A$.

Satz 140-160 : Anforderung der Input-Informationen.

Satz 170-190 : Verarbeitungsschritte (Berechnungen).

Satz 200-230 : Ausgabe der Ergebnisse.

Satz 110-170 : Abfrage, ob weitere Berechnung (zurück nach 75), bzw. Beendigung des Programms.

7. SCHRITT : Ergebnisse

Geben wir zum Beispiel die folgenden Input-Informationen:

E1: Gewicht des Körpers = 7Ø g;
E2: Scheinbares Gewicht des Körpers in der Prüfflüssigkeit (zu messen mit einer Federwaage) = 4Ø g;
E3: Wichte der Prüfflüssigkeit = 1 (Wasser),

dann erzeugt das Programm die folgenden Ergebnisse:

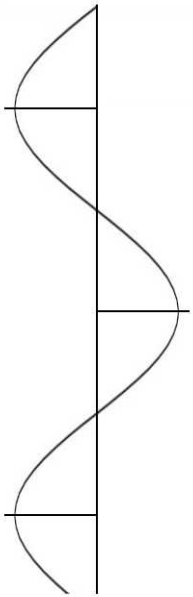
```
ZU PRUEFENDER KOERPER :

     VOLUMEN        : 3Ø

     WICHTE         : 2.33333333

NOCHMAL (J/N) ?
```

4.6 Pendelbewegung

1. SCHRITT : Vorstellung des Problems

Die Pendelbewegung ist ein wichtiger physikalischer Grundtatbestand, der in der Lehre vom Schall (Akustik), in der Lehre vom Licht (Optik) und in der Elektrizitätslehre eine wichtige Rolle spielt.

Pendelbewegungen folgen einer Sinusschwingung, wie es die nebenstehende Skizze zeigt.



Es soll nun ein Programm vorgestellt werden, welches für alternative Amplituden (A) und für alternative Periodenlängen (P), die ihrerseits die sog. Frequenz (F) einer Schwingung ausmachen, eine solche Sinusschwingung graphisch veranschaulicht.

2. SCHRITT : Problemanalyse

Für das geplante Programm müssen wir als Input-Informationen die Amplitude der Schwingung und ihre Periodenlänge vorgeben.

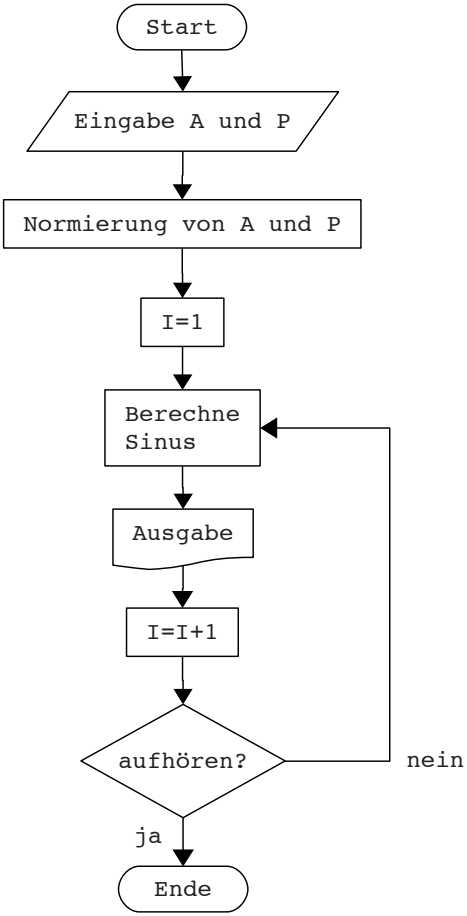
Das Programm kann dann veranlassen, daß - wie in obiger Skizze - die Sinusschwingung von oben nach unten über den Bildschirm "läuft" (was zu einem einfacheren Programm führt, als wenn diese Schwingung horizontal ausgegeben werden sollte).

Wesentlicher Inhalt des Programms muß es dabei sein, Amplitude und Periodenlänge so zu normieren, daß der Bildschirm zur Darstellung der Schwingung optisch gut ausgenutzt wird.

Dies kann erreicht werden, indem wir die Mittellinie der Schwingung in die Spalte 20 legen und die Amplitude, also den größtmöglichen Ausschlag z.B. auf den Maximalwert 18 ausrichten.

Wir wählen hier also - wie aus der gerade genannten Bildschirmaufteilung hervorgeht - nicht die hochauflösende Graphik (dazu siehe Kap. 9), sondern stellen die Schwingung mit Hilfe von PRINT CHR$(20) dar.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM P3-PENDEL
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR SIMULATION EINER PENDEL-"
40 PRINT "              BEWEGUNG."
45 PRINT :PRINT
50 PRINT "          PROF.DR.W.VOSS, 1984"
52 PRINT :PRINT :PRINT
54 PRINT "ZUM ABBRECHEN BREAK-TASTE DRUECKEN"
60 PRINT :PRINT "AMPLITUDE DER SCHWINGUNG : ";
65 INPUT A
70 PRINT "PERIODENLAENGE            : ";
72 INPUT P
75 IF A>=18 THEN A=18
80 PRINT CHR$(125)
90 I=1
100 J=(2*(22/7)/P)*I
110 Y=SIN(J)
120 S=20+Y*A
130 FOR K=1 TO S
140 PRINT CHR$(20);
150 NEXT K
160 PRINT
170 I=I+1:GOTO 100
```

5. SCHRITT : Variablenliste

A = Amplitude der Schwingung

I = Laufindex

J = Transformierte Größe : J=(2*pi/p)*I
(J dient der Normierung der Periodenlänge)

K = Laufindex, der die Länge der pro Bildschirmzeile auszugebenden Punktstäbe mit Hilfe der Größe S bestimmt

P = Periodenlänge der Schwingung

S = Bildschirmspeicherstelle, an der die jeweilige Bildschirmzeile endet

Y = SIN(J)

6. SCHRITT : Programmbeschreibung

Satz 1Ø-54 : Überschrift und Erläuterungen.

Satz 6Ø-72 : Anforderung der Input-Informationen.

Satz 75 : Beschränkung der Schwingungsamplitude auf den Maximalwert 18.

Satz 8Ø : Löschen des Bildschirms.

Satz 9Ø : Beginn Ausdruckschleife (I = Zeilenzählindex).

Satz 1ØØ : Normierung von I nach Maßgabe der Periodenlänge P zur Hilfsgröße J.

Satz 11Ø : Bestimmung des dazugehörigen Sinuswerts.

Satz 12Ø : Bestimmung der Bildschirmzeilen-Position des jeweiligen Sinuswertes unter Berücksichtigung der Tatsache, daß der Nullwert in der 2Ø. Spalte liegen soll.

Satz 13Ø-15Ø : Schleife zum Auffüllen der Zeile bis zum jeweiligen Sinuswert mit "Kügelchen".

Satz 16Ø-17Ø : Erzwingen eines Zeilenvorschubs und Ausgabe der nächsten Zeile durch Rücksprung zu Satz 1ØØ nach Erhöhung von I um 1.

7. SCHRITT : Ergebnisse

Die durch dieses Programm erzeugten Ergebnisse entsprechen der Skizze, die im ersten Arbeitsschritt vorgestellt wurde, mit dem Unterschied, daß die Bereiche zwischen dem linken Bildschirmrand und der Sinuslinie mit hellen "Kügelchen" gefüllt werden.

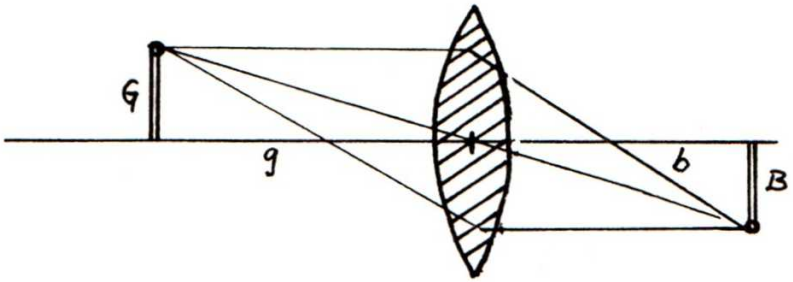
Der Leser kann sich leicht selbst ausdenken, wie das Programm verändert werden müßte, wenn man dieses "Auffüllen" unterdrücken wollte, also nur diejenigen Positionen z.B. durch "Kügelchen" optisch markiert, die tatsächlich auf der Sinuslinie liegen.

Das hier vorgestellte Programm ist ein Endlosproqramm, das nur durch Betätigung der BREAK -Taste abgebrochen werden kann. Auch dies ist aber leicht abänderbar.

4.7 Optische Abbildung

1. SCHRITT : Vorstellung des Problems

Mit einer konvexen Linse kann - so wie es die folgende Abbildung schematisch zeigt - das Abbild eines beliebigen Gegenstandes erzeugt werden:



Nach diesem Modell funktioniert der Photoapparat, aber auch z.B. das menschliche Auge.

Es soll nun ein Programm vorgestellt werden, welches bei vorgegebener Größe des Gegenstandes, Entfernung des Gegenstandes und Brennweite der konvexen Linse die Bildgröße bzw. den Abbildungsmaßstab bestimmt.

2. SCHRITT : Problemanalyse

Das zu erstellende Programm benötigt an Input-Informationen:

1. Gegenstandsgröße,
2. Gegenstandsentfernung,
3. Brennweite der Linse.

Zur Lösung des Problems benötigen wir die sog. Linsengleichung:

$$1/f = 1/g + 1/b$$

wobei:
f = Brennweite
g = Gegenstandsweite
b = Bildweite

Löst man diese Beziehung nach b auf, so erhält man:

$$b = \frac{1}{(1/f - 1/g)}$$

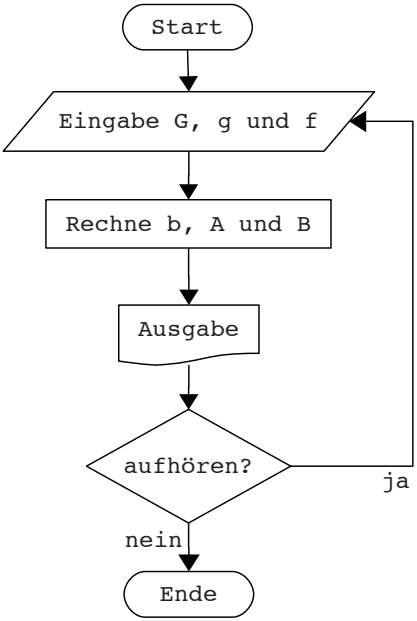
Weiterhin gilt, daß der Abbildungsmaßstab A sich ergibt zu:

$$A = B/G = b/g$$

Wenn nun beispielsweise der abzubildende Gegenstand die Größe G = 20 cm hat, so ist mit Hilfe der obigen Beziehungen auch die Bildgröße bestimmbar, nämlich:

$$B = (b*G)/g \text{ , weil } B/G = b/g$$

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM P4-LINSE
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BERECHNUNG VON BILDWEITE"
40 PRINT "BILDGROESSE UND ABBILDUNGSMASSTAB BEI"
50 PRINT "        EINER KONVEXEN LINSE."
55 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
65 DIM A$(1)
70 PRINT :PRINT "EINZUGEBEN SIND DIE FOLGENDEN DATEN"
80 PRINT :PRINT "      - GEGENSTANDSGROESSE : ";
85 INPUT G
90 PRINT "      - GEGENSTANDSWEITE   : ";
95 INPUT GW
100 PRINT "      - BRENNWEITE         : ";
105 INPUT F
110 PRINT CHR$(125)
120 BW=1/(1/F-1/GW)
130 A=BW/GW
140 B=(G*BW)/GW
145 BW=INT(BW*100+0.5)/100
146 A=INT(A*100+0.5)/100
147 B=INT(B*100+0.5)/100
150 PRINT "AUSGANGSDATEN : ":PRINT
160 PRINT "      BRENNWEITE           : ";F
170 PRINT "      GEGENSTANDSGROESSE   : ";G
180 PRINT "      GEGENSTANDSWEITE     : ";GW
190 PRINT :PRINT :PRINT "ERGEBNISSE : ":PRINT
200 PRINT :PRINT "ABBILDUNGSMASSTAB        : ";A
210 PRINT :PRINT "BILDGROESSE              : ";B
220 PRINT :PRINT "BILDWEITE                : ";BW
225 PRINT :PRINT :PRINT
230 PRINT "NOCH EINE RECHNUNG (J/N) ";:INPUT A$
240 IF A$="J" THEN PRINT CHR$(125):GOTO 70
250 PRINT :PRINT "ENDE DER BERECHNUNGEN":END
```

5. SCHRITT : Variablenliste

A = Abbildungsmaßstab
A$ = Stringvariable (Ja/Nein)
B = Bildgröße
BW = Bildweite
F = Brennweite
G = Gegenstandsgröße
GW = Gegenstandsweite

6. SCHRITT : Programmbeschreibung

Satz 1Ø-7Ø : Überschrift und Erläuterungen.

Satz 8Ø-1Ø5 : Anforderung der Input-Informationen.

Satz 11Ø-14Ø : Beschränkung der Schwingungsamplitude auf den Maximalwert 18.

Satz 145-147 : Runden auf 2 Dezimalstellen.

Satz 15Ø-225 : Ausgabe der Ergebnisse.

Satz 23Ø-24Ø : Abfrage, ob weitere Berechnung gewünscht (dann zurück nach Satz 7Ø nach Löschen des Bildschirms).

Satz 25Ø : Beendigung des Programms.

7. SCHRITT : Ergebnisse

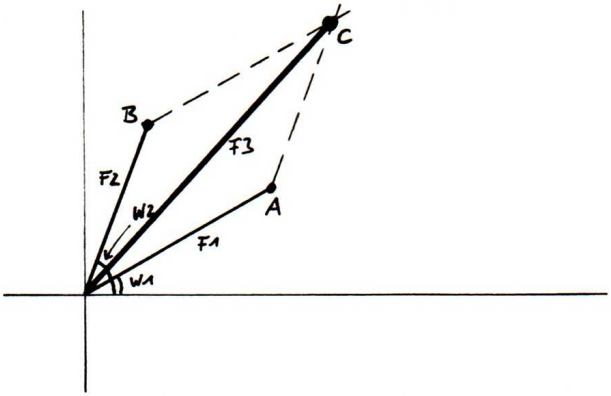
Geben wir z.B. ein: G = 2Ø cm, GW = 3ØØ cm, F = 15 cm, so erhalten wir:

```
AUSGANGSDATEN :
     BRENNWEITE            : 15
     GEGENSTANDSGROESSE    : 2Ø
     GEGENSTANDSWEITE      : 3ØØ
ERGEBNISSE :
     ABBILDUNGSMASSTAB         : Ø.Ø5
     BILDGROESSE               : 1.Ø5
     BILDWEITE                 : 15.79
NOCH EINE RECHNUNG (J/N) ?
```

4.8 Kräfteparallelogramm

1. SCHRITT : Vorstellung des Problems

Wenn zwei Kräfte an einem gemeinsamen Punkt ansetzen, so ergibt sich eine gemeinsame resultierende Kraft als Diagonale des sog. Kräfteparallelogramms, wie die folgende Skizze verdeutlicht:



Im folgenden Programm soll für die beliebige Konstellation von je zwei Kräften das Kräfteparallelogramm graphisch ausgegeben und die resultierende Kraft F3 berechnet werden.

2. SCHRITT : Problemanalyse

Der Ansatzpunkt der beiden Kräfte soll in die Mitte des Bildschirms gelegt werden (160,80).

Einzugeben sind die Kräfte F1 und F2, die zunächst so umzurechnen sind, daß F1 + F2 nicht größer als 320 wird, damit auch bei ungünstigster Konstellation der Platz des Bildschirms ausreicht.

Einzugeben sind weiterhin die Winkel W1 und W2, in denen die Kräfte F1 und F2 am Ansatzpunkt angreifen.

Mit Hilfe des Satzes des Pythagoras und der Benutzung der Winkelbeziehungen im rechtwinkligen Dreieck können dann Spalten und Zeilen von A und B und daraus dann auch die Koordinaten von C berechnet werden (siehe Skizze).

Ist die Lage von C bekannt, kann wiederum aus F1 und F2 die interessierende Kraft F3 berechnet werden (Kosinussatz).

3. SCHRITT : Flußdiagramm

Auf ein Flußdiagramm kann hier verzichtet werden.

4. SCHRITT : Programm

```
10 REM P5-PARALLELOGRAMM
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR VERANSCHAULICHUNG DES "
40 PRINT "KRAEFTE-PARALLELOGRAMMS BEI ZWEI AN-"
50 PRINT "        GREIFENDEN KRAEFTEN."
55 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 SU=160:ZU=80
80 PRINT :PRINT :PRINT :PRINT
90 PRINT "ERSTE KRAFT    F1          : ";
95 INPUT F1
100 PRINT "ANSATZWINKEL   F1 (0-360) : ";
105 INPUT W1
107 PRINT :PRINT
110 PRINT "ZWEITE KRAFT   F2          : ";
115 INPUT F2
120 PRINT "ANSATZWINKEL   F2 (0-360) : ";
125 INPUT W2
127 W=W1-W2
130 W1=W1*((22/7)/180):W2=W2*((22/7)/180)
135 H=F1:G=F2
140 F1=(F1/(F1+F2))*75:F2=(F2/(H+F2))*75
150 Z1=80-F1*SIN(W1)
155 T=SIN(2*W1)/(1+COS(2*W1))
160 S1=((80-Z1)/T)+150
165 GRAPHICS 8:COLOR 1
170 PLOT SU,ZU
180 DRAWTO S1,Z1
190 Z2=80-F2*SIN(W2)
195 T=SIN(2*W2)/(1+COS(2*W2))
200 S2=((80-Z2)/T)+150
210 PLOT SU,ZU
220 DRAWTO S2,Z2
230 S3=S2+(S1-SU)
240 Z3=Z2+(Z1-ZU)
250 PLOT SU,ZU
260 DRAWTO S3,Z3
265 W=180-W:W=(W/180)*(22/7)
270 F3=SQR(G*G+H*H-2*G*H*COS(W))
280 FOR I=1 TO 2000:NEXT I:GRAPHICS 0
290 PRINT "F1 = ";H;"      ";"F2 = ";G
300 PRINT "RESULTIERENDE KRAFT = ";F3
310 END
```

5. SCHRITT : Variablenliste

F1 = erste Kraft
F2 = zweite Kraft
F3 = dritte Kraft
G = Hilfsgröße (=F2)
H = Hilfsgröße (=F1)
SU = Spalte Ursprung
S1 = Spalte Punkt A
S2 = Spalte Punkt B
S3 = Spalte Punkt C
T = Tangens
W = W1-W2
W1 = erster Ansatzwinkel
W2 = zweiter Ansatzwinkel
ZU = Zeile Ursprung
Z1 = Zeile Punkt A
Z2 = Zeile Punkt B
Z3 = Zeile Punkt C

6. SCHRITT : Programmbeschreibung

Satz 1Ø-6Ø : Überschrift.

Satz 7Ø : Ursprungskoordinaten.

Satz 8Ø-125 : Eingabe der Ausgangsinformationen.

Satz 127 : Bestimmung von W = W1-W2.

Satz 130 : Umrechnen der Winkel in Einheiten des Kreisparameters.

Satz 135 : Speichern der eingegebenen Kraftangaben.

Satz 140 : Normierung der beiden Kräfte zur Bildschirmausnutzung.

Satz 150-160 : Bestimmung der Koordinaten von A.

Satz 165-180 : Zeichnen von F1.

Satz 190-200 : Koordinaten von B.

Satz 210-220 : Zeichnen von F2.

Satz 230-240 : Koordinaten von C.

Satz 250-260 : Zeichnen von F3.

Satz 265-270 : Berechnen von F3 (Kosinussatz).

Satz 280 : Warteschleife.

Satz 290-300 : Ausgabe der Ergebnisse.

Satz 310 : Beendigung des Programms.

7. SCHRITT : Ergebnisse

Auf die Darstellung von Ergebnissen kann hier - angesichts der Skizze aus dem 1. Schritt – verzichtet werden.

4.9 Das Ohm'sche Gesetz

1. SCHRITT : Vorstellung des Problems

In Stromleitern, in denen bei steigender Spannung die Stromstärke proportional zunimmt, gilt das Ohm'sche Gesetz.

Es besagt in Formelschreibweise:

$$\frac{\text{Spannung}}{\text{Stromstärke}} = \text{konstant} = R$$

Die Größe R nennt man den elektrischen Widerstand des betreffenden Stromleiters.

Im folgenden Programm soll für jeweils zwei eingegebene Werte (Spannung, Stromstärke, Widerstand) die jeweils dritte Größe berechnet werden.

2. SCHRITT : Problemanalyse

Das Programm, um das es hier geht, bietet unter mathematischen Gesichtspunkten überhaupt keine Schwierigkeiten, weil ja im Prinzip nur ein einziger Rechenschritt erforderlich ist.

Die einzige Schwierigkeit ist programmlogischer Art und besteht darin, daß die Eingabe aus drei unterschiedlichen Varianten bestehen kann, denen sich dann unterschiedliche Rechenprozeduren anschließen.

Diese Eingaben können sein:

1. Spannung und Stromstärke;
2. Spannung und Widerstand;
3. Stromstärke und Widerstand.

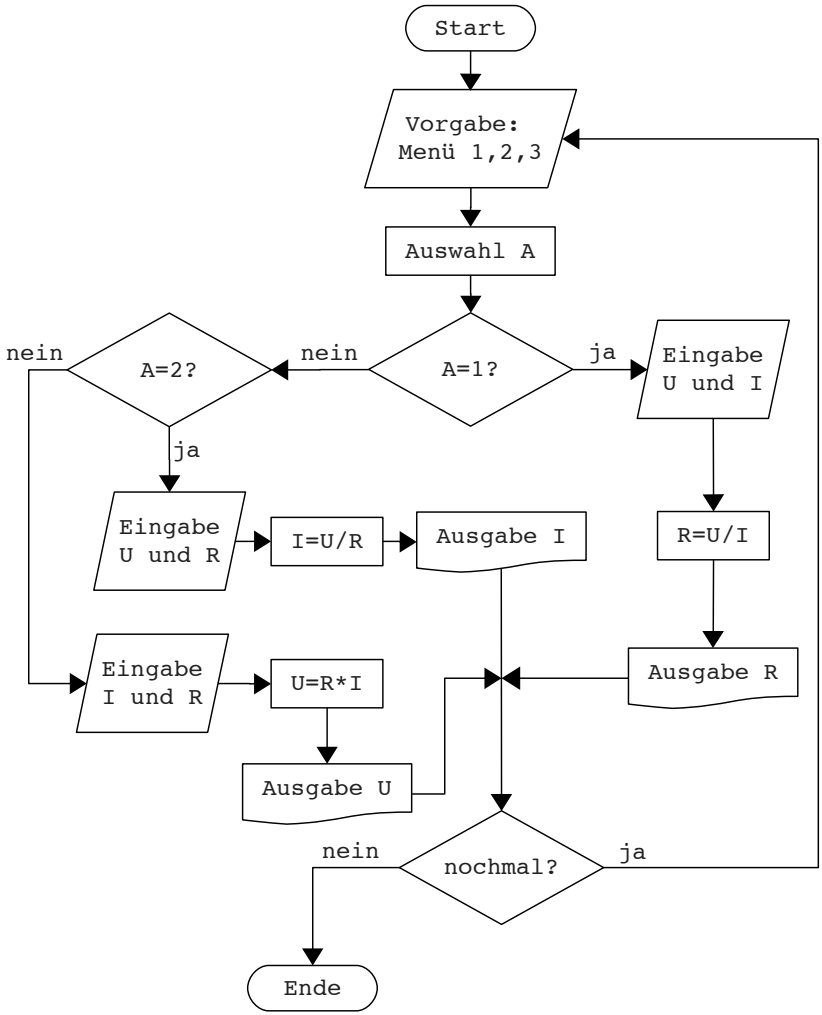
Dies gibt uns Gelegenheit, die sog. Menütechnik vorzuführen:

Es geht dabei darum, dem Programmbenutzer ein "Menü" zur Auswahl auf dem Bildschirm vorzugeben, aus dem dieser dann die gewünschte Variante auszuwählen hat. Je nach Auswahl kommen dann unterschiedliche, wenn auch hier sehr einfache Rechenprozeduren in Gang.

Dieses Menü würde hier aus drei Positionen bestehen (s.o.).

Der Problemlösungsweg selbst wird aus dem folgenden Flußdiagramm deutlich.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM P6-0HMSCHES GESETZ
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR DARSTELLUNG DER BEZIE-"
40 PRINT "HUNGEN ZWISCHEN STROMSTAERKE,SPANNUNG"
50 PRINT "UND WIDERSTAND (OHMSCHES GESETZ)."
55 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
65 DIM A$(1)
66 PRINT :PRINT :PRINT
70 PRINT :PRINT "MIT WELCHEN VORGABEN WIRD GEARBEITET?"
80 PRINT :PRINT "      SPANNUNG UND STROMSTAERKE    (1)"
90 PRINT :PRINT "      SPANNUNG UND WIDERSTAND      (2)"
100 PRINT :PRINT "      STROMSTAERKE UND WIDERSTAND  (3)"
105 PRINT :PRINT
110 PRINT "BITTE ZIFFER EINGEBEN ";:INPUT A
120 PRINT CHR$(125)
130 IF A>1 THEN 190
140 PRINT "SPANNUNG         : ";:INPUT U
150 PRINT "STROMSTAERKE     : ";:INPUT I
160 R=U/I:R=INT(R*100+0.5)/100
170 PRINT :PRINT "WIDERSTAND      : ";R
180 GOTO 300
190 IF A=3 THEN 250
200 PRINT "SPANNUNG         : ";:INPUT U
210 PRINT "WIDERSTAND       : ";:INPUT R 2
220 I=U/R:I=INT(I*100+0.5)/100
230 PRINT :PRINT "STROMSTAERKE    : ";I
240 GOTO 300
250 PRINT "WIDERSTAND       : ";:INPUT R
260 PRINT "STROMSTAERKE     : ";:INPUT I
270 U=R*I:U=INT(U*100+0.5)/100
280 PRINT : PRINT "SPANNUNG       : ";U
285 PRINT :PRINT :PRINT
290 PRINT :PRINT :PRINT
300 PRINT "NOCH EINE RECHNUNG (J/N) ";:INPUT A$
310 IF A$="J" THEN PRINT CHR$(125):GOTO 70
320 PRINT :PRINT "ENDE DER BERECHNUNGEN":END
```

5. SCHRITT : Variablenliste

A = Auswahlvariable (Menü)
A$ = Stringvariable (Ja, Nein)

I = Stromstärke
R = Widerstand
U . Spannung

6 . SCHRITT : Programmbeschreibung

Satz 10-66 : Überschrift und Erläuterungen.

Satz 70-105 : Vorgabe des Menüs.

Satz 110 : Menüauswahl.

Satz 120-130 : Wenn A größer als 1 ist (A = Menüvariable) geht es nach Löschen des Bildschirms weiter bei Satz 190.

Satz 140-150 : Eingaben gemäß Auswahl A = 1.

Satz 160-170 : Berechnungen und Ergebnisausgabe.

Satz 140 : Normierung der beiden Kräfte zur Bildschirmausnutzung.

Satz 180 : Sprung zum Satz 300.

Satz 165-180 : Zeichnen von F1.

Satz 190-240 : Entsprechende Berechnungen wie oben; nun aber für A = 2.

Satz 25Ø-29Ø : Entsprechende Berechnungen; nun für A = 3.

Satz 3ØØ-31Ø : Abfrage, ob noch eine Berechnung gewünscht wird.

Wenn ja, Löschen des Bildschirms und zurück zum Satz 7Ø.

Satz 32Ø : Beendigung des Programms.

7. SCHRITT : Ergebnisse

Wählen wir bei der Menüauswahl z.B. die Ziffer 1, so werden vom Programm Spannung und Stromstärke angefordert. Geben wir daraufhin z.B. für die Spannung den Wert 22Ø an und für die Stromstärke den Wert 11, so erhalten wir als Ergebnisausdruck:

WIDERSTAND : 2Ø

NOCH EINE RECHNUNG (J/N)?

5

Kapitel 5: Sprachen

5.1 Vorbemerkung und BASIC-Ergänzungen

Auch beim Erlernen oder beim Üben von Fremdsprachen können Rechner wirkungsvolle Hilfe leisten. Dabei geht es jetzt nicht darum, komplette Sprachunterrichtsprogramme (zum Beispiel zum Erlernen von Englisch) vorzustellen, weil dies den Rahmen und den Anspruch dieses Buches weit übersteigen würde - außerdem kann man solche "Komplettprogramme" schon zu relativ günstigen Preisen käuflich erwerben. Vielmehr interessiert hier, mit geringem Aufwand einfache Programme zu erstellen, die den sprachlernenden Schüler bei seinen Bemühungen unterstützen können.

In Frage kommen hier in erster Linie solche Programme, die das Erlernen von Vokabeln erleichtern, oder die Übersetzungen bereitstellen, um so das Nachschlagen in Wörterbüchern zu ersparen.

Für derartige Programme ist es nun allerdings notwendig, einige zusätzliche BASIC-Anweisungen kennenzulernen.

Häufig ist es sinnvoll, daß bestimmte Programmsegmente aus dem eigentlichen Hauptprogramm ausgelagert werden, um dann von diesem aus mehrfach angesprungen zu werden.

Solche ausgelagerten Teilprogramme nennt man Unterprogramme (subroutines). Der Sprung vom Hauptprogramm in das Unterprogramm erfordert das folgende Statement:

Statement 21: nn GOSUB S,Z

Dieses Statement bewirkt einen Sprung in das Unterprogramm, das mit der Satznummer mm beginnt.

Ist dann das Unterprogramm abgearbeitet, muß ein Rücksprung in das Hauptprogramm erfolgen. Zu diesem Zweck wird das Unterprogramm mit dem folgenden Statement beschlossen :

Statement 22: nn RETURN

Dieses Statement bewirkt, daß ein Rücksprung an diejenige Stelle des Hauptprogramms erfolgt, die direkt hinter der entsprechenden GOSUB-Anweisung folgt.

Die Wirkungsweise dieser beiden Statements wird in den folgenden Programmen deutlich werden, so daß hier auf Demonstrationsbeispiele verzichtet werden kann.

5.2 Englische unregelmäßige Verben

1. SCHRITT : Vorstellung des Problems

In einem ersten Beispiel soll ein Programm vorgestellt werden, welches für ein eingegebenes deutsches Verb die englische Übersetzung und die dazugehörigen unregelmäßigen Stammformen angibt.

Beispielsweise soll der Rechner auf die Eingabe des Verbs

gehen

mit den Meldungen

go went gone (gehen)

antworten.

Um den Programmieraufwand nicht zu hoch werden zu lassen, wollen wir uns dabei auf 20 wichtige unregelmäßige Verben beschränken. Der Leser wird sofort erkennen, wie er das Programm in Eigenarbeit ergänzen muß, um einen kompletten Überblick über alle englischen unregelmäßigen Verben zu erhalten.

2. SCHRITT : Problemanalyse

In diesem Programm müssen deutsche Verben, englische Übersetzungen und Stammformen bereitgestellt werden.

Bei sehr umfangreichen Informationsbeständen ist es sinnvoll, die Ausgangsinformationen nicht bei jedem Programmstart neu zu lesen, sondern nur die gewünschte Information aus einer Textdatei einer entsprechend vorbereiteten Diskette interaktiv einzulesen.

Diese Möglichkeit, auf die fortgeschrittenere Programmierer sicherlich zurückgreifen werden - zumindest sofern sie über ein Diskettenlaufwerk verfügen – soll hier aber nicht besprochen werden.

Wir lassen also bei jedem Programmstart vom Benutzer angeben, für welches Verb er Informationen wünscht. Daraufhin wird vom Programm der Datenbestand durchsucht, bis das betreffende Verb gefunden ist, so daß dann der entsprechende Datensatz ausgegeben werden kann.

Findet er bei diesen Vergleichen in seinem Datenbestand das eingegebene Verb nicht, dann deshalb, weil es

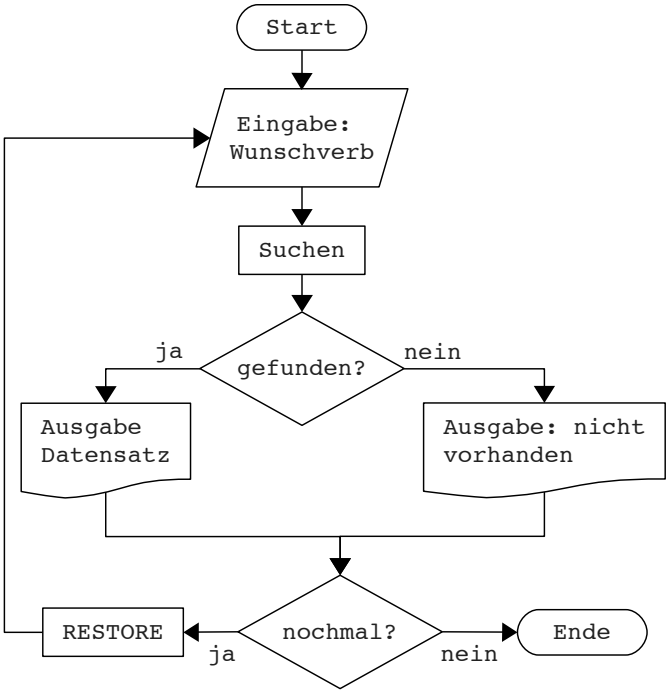
- entweder im Datenbestand noch nicht vorhanden ist (wir haben ja vorläufig nur 20 Verben aufgenommen),

- oder, weil es sich bei dem eingegebenen Verb nicht um ein unregelmäßiges Verb handelt.

In beiden Fällen, die aber bei dem folgenden Programm wegen der Nicht-Vollständigkeit der Ausgangsinformationen nicht voneinander trennbar sind, ist eine Meldung auszugeben.

Weiterhin soll das Programm dem Benutzer die Gelegenheit geben, ein weiteres Verb abzufragen.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM S1-ENGL.UNREGELM.VERBEN
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR AUSGABE VON ENGLISCHEN"
40 PRINT "       UNREGELMAESSIGEN VERBEN"
50 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT :PRINT
80 PRINT "DIE ZAHL DER VORHANDENEN VERBEN IST"
90 PRINT "IN DIESEM PROGRAMM BESCHRAENKT.":PRINT
100 PRINT "ERWEITERUNGEN SIND MOEGLICH, WENN DIE"
110 PRINT "LISTE DER DATA-STATEMENTS VERLAENGERT"
120 PRINT "WIRD (SATZ 500 FF.) UND WENN IN SATZ"
130 PRINT "160 DIE ANGABE FUER N ERHOEHT WIRD."
132 PRINT :PRINT :PRINT
135 PRINT "ZUR FORTSETZUNG BITTE CONT EINGEBEN !":STOP
140 DIM A$(1),D$(15),E1$(15),E2$(15),E3$(15),V$(15)
155 PRINT CHR$(125)
160 N=20
200 PRINT "BITTE DAS DEUTSCHE VERB, DESSEN "
210 PRINT "ENGLISCHE STAMMFORMEN GEWUENSCHT"
220 PRINT "WERDEN, EINGEBEN: ";:INPUT V$
225 RESTORE
230 FOR I=1 TO N
240 READ D$,E1$,E2$,E3$:IF D$=V$ THEN 275
250 NEXT I
255 PRINT :PRINT :PRINT
260 PRINT "VERB IM DATENBESTAND NICHT VORHANDEN"
270 PRINT "BZW. NICHT UNREGELMAESSIG.":GOTO 310
275 PRINT :PRINT :PRINT
280 PRINT "VERB : ";D$:PRINT
290 PRINT "ENGLISCHE STAMMFORMEN : ":PRINT
294 PRINT "        ";E1$
296 PRINT "        ";E2$
298 PRINT "        ";E3$
300 PRINT :PRINT :PRINT
310 PRINT "NOCHMAL (J/N) ";:INPUT A$
320 IF A$="J" THEN PRINT CHR$(125) :GOTO 200
330 PRINT :PRINT :PRINT "ENDE":END
500 DATA SEIN,BE,WAS,BEEN
502 DATA HABEN,HAVE,HAD,HAD
504 DATA BEGINNEN,BEGIN,BEGAN,BEGUN
506 DATA BRINGEN,BRING,BROUGHT,BROUGHT
508 DATA KOMMEN,COME,CAME,COME
510 DATA TUN,DO,DID,DONE
512 DATA ESSEN,EAT,ATE,EATEN
```

```
514 DATA KAUFEN,BUY,BOUGHT,BOUGHT
516 DATA FUEHLEN,FEEL,FELT,FELT
518 DATA FINDEN,FIND,FOUND,FOUND
520 DATA GEHEN,GO,WENT,GONE
522 DATA VERLIEREN,LOOSE,LOST,LOST
524 DATA MACHEN,MAKE,MADE,MADE
526 DATA LESEN,READ,READ,READ
528 DATA LAUFEN,RUN,RAN,RUN
530 DATA SAGEN,SAY,SAID,SAID
532 DATA SEHEN,SEE,SAW,SEEN
534 DATA SITZEN,SIT,SAT,SAT
536 DATA SPRECHEN,SPEAK,SPOKE,SPOKEN
538 DATA NEHMEN,TAKE,TOOK,TAKEN
```

5. SCHRITT : Variablenliste

```
A$  =  Stringvariable für Eingaben
D$  =  deutsches Verb
E1$ =  englisches Verb, Form 1
E2$ =  englisches Verb, Form 2
E3$ =  englisches Verb, Form 3
I   =  Laufvariable
N   =  Anzahl der Datensätze
V$  =  gewünschtes Verb
```

6. SCHRITT : Programmbeschreibung

Satz 1Ø-32 : Überschrift, Erläuterungen und Hinweise, wie der Datenbestand aktualisiert werden kann.

Satz 135 : Programm in "Warteposition". Die Programmausführung geht erst weiter, wenn der Benutzer CONT eingibt.

Satz 14Ø : Dimensionierung.

Satz 155 : Löschen des Bildschirms.

Satz 16Ø : Angabe der Zahl der Verben.

Satz 2ØØ : Anforderung des gewünschten Verbs.

Satz 225 : Zurücksetzen für ev. Zusatzläufe.

Satz 23Ø-25Ø : Suche nach dem richtigen Datenbestand. Wird er gefunden, dann weiter bei 275; wird er nicht gefunden, weiter bei 255.

Satz 255 : Meldung, daß Verb im Datenbestand nicht vorhanden; dann weiter bei Satz 31Ø.

Satz 275-3ØØ : Ausgabe der gesuchten Informationen.

Satz 31Ø-32Ø : Entsprechende Berechnungen; nun für A = 3.

Satz 3ØØ-31Ø : Abfrage, ob weitere Ausgabe gewünscht; wenn ja, Löschen des Bildschirms und zurück zu Satz 2ØØ.

Satz 33Ø : wenn nein, Ende des Programms.

Satz 5ØØ-538 : Daten.

7. SCHRITT : Ergebnisse

Gibt man beispielsweise auf die Anforderung des Rechners aus Satz 2ØØ ein:

```
GEHEN
```

so antwortet der Rechner:

```
VERB : GEHEN

ENGLISCHE STAMMFORMEN :

       GO
       WENT
       GONE


NOCHMAL (J/N) ?
```

Es ist deutlich zu sehen, wie das Programm geändert werden müßte, wenn die Liste der unregelmäßigen Verben komplettiert werden soll: In den Sätzen 5ØØff. sind weitere DATA-Statements anzuhängen und Satz 16Ø muß ausgetauscht werden mit der Angabe der aktuellen Zahl eingegebener Verben.

5.3 Französisch-Vokabeln

1. SCHRITT : Vorstellung des Problems

In diesem Beispiel soll nun gezeigt werden, wie der Rechner als Vokabelheft-Ersatz Verwendung finden kann. Auf ein einzugebendes französisches Wort antwortet er mit dem entsprechenden deutschen Wort oder umgekehrt (es gilt hier die gleiche Anmerkung wie in Abschnitt 5.2 zur interaktiven Nutzung eines Diskettenspeichers).

Um das Programm wieder nicht zu groß werden zu lassen, beschränken wir uns auf nur zehn Vokabeln. Dieses Programm dient also nur als Beispiel. Wenn es echt genutzt werden soll, müßte sich der Leser die Mühe machen, seinen gesamten deutsch-französischen Vokabelvorrat zunächst dem Rechner einzugeben.

An den Funktionsprinzipien des Programms ändert sich aber dadurch nichts.

2. SCHRITT : Problemanalyse

Bei diesem Problem können wir uns kurz fassen:

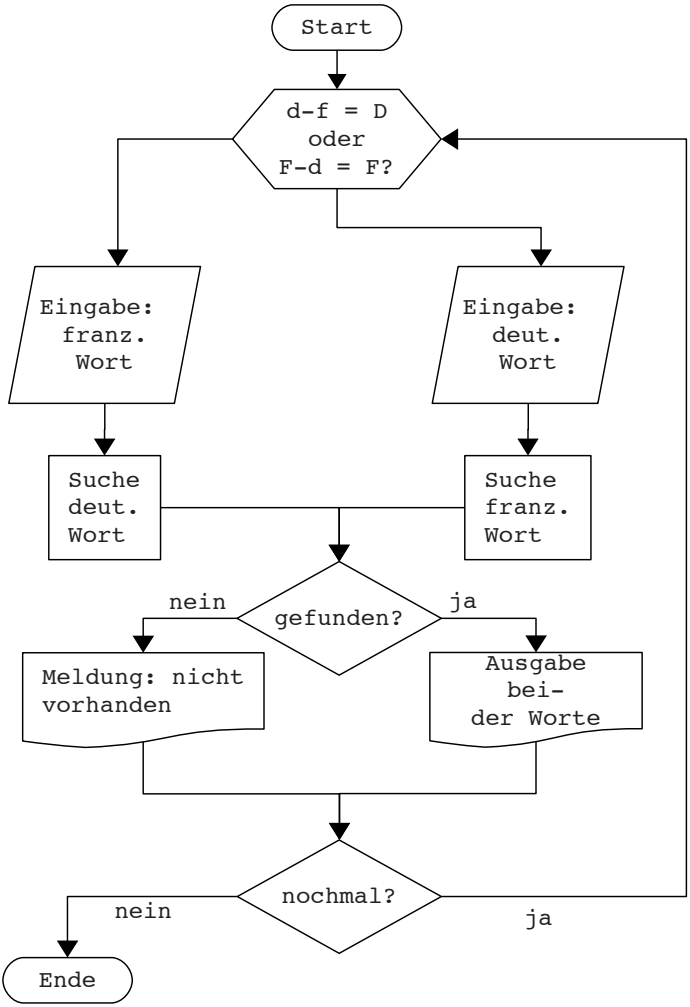
Gibt der Benutzer ein deutsches Wort ein, so muß der Rechner durch Vergleich mit allem ihm zur Verfügung stehenden deutschen (und damit paarweise verbundenen französischen Worten) das entsprechende deutsch-französische Wortpaar aus seinem Datenbestand heraussuchen und ausgeben. Entsprechend ist auch im umgekehrten Fall zu verfahren.

Damit beide "Übersetzungsrichtungen" in nur einem Programm bewältigt werden können, muß dem Rechner zunächst mitgeteilt werden, ob deutsch-französische oder ob französisch-deutsche Übersetzung gewünscht wird.

Falls das eingegebene Wort nicht übersetzt werden kann, weil es in der (hier sehr kurzen) Vokabelliste (noch) nicht vorhanden ist, muß wieder eine entsprechende Meldung erfolgen.

Weiterhin sollte dem Benutzer Gelegenheit geboten werden, im gleichen Programmlauf eine weitere Übersetzung zu erhalten, wenn er dies wünscht.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM S2-VOKABELN DEUTSCH-FRANZ.
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM NACHSCHLAGEN VON VOKABELN"
40 PRINT :PRINT "DEUTSCH-FRANZOESISCH ODER UMGEKEHRT."
45 PRINT :PRINT
50 PRINT "        PROF.DR.W.VOSS, 1984"
60 PRINT :PRINT :PRINT
70 PRINT "IM DATENBESTAND DIESES PROGRAMMS BE-"
80 PRINT "FINDEN SICH NUR 10 VOKABELN.":PRINT
90 PRINT "SOLL DAS PROGRAMM AUSGEWEITET WERDEN,"
100 PRINT "SO MUESSEN IN 500 FF. WEITERE DATA-"
110 PRINT "STATEMENTS ANGEFUEGT UND DER WERT"
120 PRINT "FUER N IN SATZ 160 GEAENDERT WERDEN."
125 PRINT :PRINT :PRINT
130 PRINT "ZUR FORTSETZUNG BITTE CONT EINGEBEN !":STOP
150 PRINT CHR$(125)
160 N=10
170 DIM A$(1),W$(15),D$(15),F$(15)
200 PRINT "WELCHE UEBERSETZUNG WIRD GEWUENSCHT ?"
205 PRINT :PRINT :PRINT
210 PRINT "     DEUTSCH-FRANZOESISCH  (1)"
220 PRINT :PRINT "     ODER":PRINT
230 PRINT "     FRANZOESISCH-DEUTSCH  (2)"
240 PRINT :PRINT "BITTE 1 ODER 2 EINGEBEN : ";
242 INPUT Z
245 PRINT CHR$(125)
247 RESTORE
250 IF Z=2 THEN 350
255 PRINT :PRINT
260 PRINT "DEUTSCHES WORT BITTE ";:INPUT W$
270 FOR I=1 TO N
280 READ D$,F$:IF D$=W$ THEN 320
290 NEXT I
295 PRINT :PRINT
300 PRINT "GESUCHTES WORT IM DATENBESTAND NICHT"
310 PRINT "VORHANDEN.":GOTO 450
320 PRINT CHR$(125)
325 PRINT "DEUTSCH          : ";D$:PRINT
330 PRINT "FRANZOESISCH     : ";F$
340 GOTO 450
350 PRINT :PRINT :PRINT
355 PRINT "FRANZOESISCHES WORT BITTE ";:INPUT W$
357 RESTORE
360 FOR I=1 TO N:READ D$,F$:IF F$=W$ THEN 400
```

```
370 NEXT I
375 PRINT :PRINT
388 PRINT "GESUCHTES WORT IM DATENBESTAND NICHT"
390 PRINT "VORHANDEN.":GOTO 450
400 PRINT CHR$(125)
405 PRINT "FRANZOESISCH   : ";F$:PRINT
410 PRINT "DEUTSCH        : ";D$
450 PRINT :PRINT :PRINT "NOCHMAL (J/N) ";:INPUT A$
460 IF A$="J" THEN PRINT CHR$(125):GOTO 200
470 PRINT :PRINT "ENDE DER AUSGABE":END
500 DATA KAUFEN,ACHETER
502 DATA HABEN,AVOIR
504 DATA GEHEN,ALLER
506 DATA SEHEN,VOIR
508 DATA TRINKEN,BOIRE
510 DATA MACHEN,FAIRE
512 DATA LESEN,LIRE
514 DATA ESSEN,MANGER
516 DATA NEHMEN,PRENDRE
518 DATA RAUCHEN,FUMER
```

5. SCHRITT : Variablenliste

A$ = Stringvariable für Eingaben

D$ = deutsche Vokabeln
F$ = französische Vokabeln

I = Laufindex
N = Anzahl der Datensätze

W$ = gewünschtes Wort
Z = Kennziffer bei der Menü-Auswahl

6. SCHRITT : Programmbeschreibung

Satz 1Ø-12Ø : Überschrift und Erläuterungen etc.

Satz 13Ø-15Ø : Warten und Löschen des Bildschirms.

Satz 16Ø-17Ø : Angabe der Anzahl der Datensätze und Dimensionierung.

Satz 19Ø-242 : Löschen des Bildschirms und Menüvorgabe.

Satz 245-247 : Löschen und Restaurieren.

Satz 25Ø : Wird 2 eingegeben (französisch-deutsch), dann weiter bei Satz 35Ø.

Satz 26Ø : Eingabe eines deutschen Wortes.

Satz 27Ø-29Ø : Suche nach dem entsprechenden Datensatz.

Satz 3ØØ-31Ø : Ist das gesuchte Wort im Datenbestand (noch) nicht vorhanden, weiter bei 45Ø.

Satz 32Ø-33Ø : Ausgabe der Ergebnisse.

Satz 34Ø : Sprung nach Satz 45Ø.

Satz 35Ø-41Ø : Entsprechende Vorgehensweise für die Eingabe eines französischen Wortes.

Satz 45Ø-47Ø : Beendigung des Programms mit Wiederholungsmöglichkeit.

Satz 5ØØ-518 : Daten.

7. SCHRITT : Ergebnisse

Geben wir nach der Menüvorgabe die Ziffer 1 ein und entscheiden uns damit für die deutsch-französische Ausgabe, so fordert der Rechner von uns ein deutsches Wort an.

Geben wir z.B. daraufhin das Wort haben ein, so antwortet der Rechner:

DEUTSCH : HABEN

FRANZOESISCH : AVOIR

NOCHMAL (J/N) ?

5.4 Englisch-Vokabeltest

1. SCHRITT : Vorstellung des Problems

Ähnlich wie im vorangegangenen Abschnitt sollen - ausgehend von einem bestimmten Vokabelvorrat - Übersetzungen bereitgestellt werden. Im Gegensatz aber zum vorhergehenden Programm soll der Benutzer aufgefordert werden, zu einem vom Rechner zufällig ausgewählten deutschen Wort das korrekte englische Wort einzusetzen.

Es wird hier also ein Programm vorgestellt, das so funktioniert wie früher das Vokabellernen mit zugehaltener Fremdwortspalte im Vokabelheft vor sich ging.

Es leuchtet unmittelbar ein, daß ein entsprechendes Programm auch für Englisch-Deutsch bzw. für andere Fremdsprachen verwendet werden könnte.

Im Interesse der Reduzierung des Programmieraufwandes beschränken wir uns hier wieder auf die exemplarische Darstellung anhand von nur zehn Vokabeln. Eine Untergruppe davon wird jeweils vom Rechner zufällig ausgewählt und dem Benutzer wird nach jeder Serie mitgeteilt, wie hoch seine Erfolgsquote war.

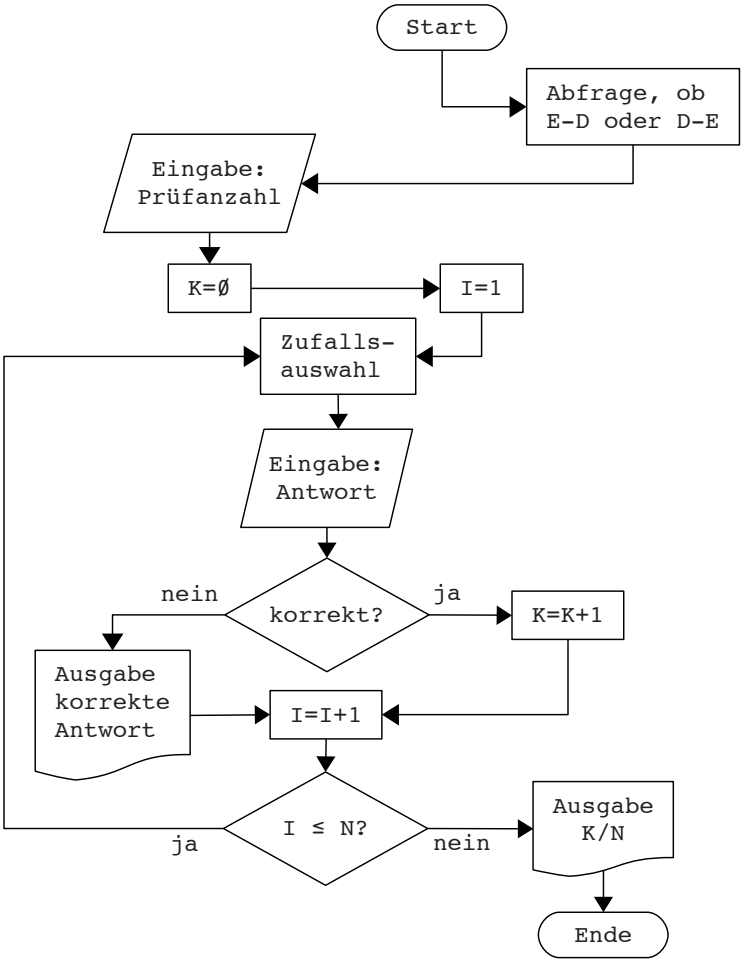
2. SCHRITT : Problemanalyse

Ausgehend von einem vorgegebenen Datenbestand, muß das Programm eine Zufallsauswahl treffen.

Auf jede vorgegebene Vokabel antwortet der Benutzer. Die Anzahl der richtigen Antworten wird in Beziehung gesetzt zur Anzahl der angeforderten Antworten.

Besondere Schwierigkeiten tauchen bei der Problemanalyse des weiteren nicht auf.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM S3-VOKABELTEST
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM ABFRAGEN VON VOKABELN."
35 PRINT
40 PRINT "HIER : "
45 PRINT "DEUTSCH/ENGLISCH ODER UMGEKEHRT"
47 PRINT :PRINT
50 PRINT "        PROF. DR. W. VOSS, 1984"
60 PRINT :PRINT
70 PRINT "IM DATENBESTAND DIESES PROGRAMMS BE-"
80 PRINT "FINDEN SICH NUR 10 VOKABELN." :PRINT
90 PRINT "SOLL DAS PROGRAMM AUSGEWEITET WERDEN,"
100 PRINT "SO MUESSEN IN 1000 FF. WEITERE DATA-"
110 PRINT "STATEMENTS ANGEFUEGT UND DER WERT"
120 PRINT "FUER N IN STATEMENT 190 GEAENDERT"
125 PRINT "WERDEN."
130 PRINT :PRINT :PRINT
140 PRINT "ZUR FORTSETZUNG BITTE CONT EINGEBEN":STOP
150 PRINT CHR$(125): PRINT "MOMENT BITTE"
160 N=10
170 DIM A$(1),D$(15),E$(15),W$(15),H$(15)
185 ZZ=1
190 PRINT CHR$(125)
200 PRINT "DEUTSCH/ENGLISCH     (1)"
210 PRINT :PRINT :PRINT "ODER"
215 PRINT :PRINT
220 PRINT "ENGLISCH/DEUTSCH     (2)"
225 PRINT :PRINT
230 PRINT "BITTE 1 ODER 2 EINGEBEN"
240 PRINT "                      ";:INPUT Z
247 PRINT :PRINT :PRINT
250 PRINT "WIEVIELE VOKABELN SOLLEN GEPRUEFT"
260 PRINT "WERDEN ";:INPUT A
270 GOSUB 2000:REM PRUEFUNG
420 PRINT :PRINT "ENDE DER AUSGABE":END
1000 DATA LAUFEN,RUN,SCHREIBEN,WRITE
1010 DATA DRUCKEN,PRINT,GEHEN,GO
1020 DATA WENN,IF,UNTERPROGRAMM,SUBROUTINE
1030 DATA RECHNER,COMPUTER,ZEICHEN,CHARACTER
1040 DATA BILDSCHIRM,SCREEN,TASTE,KEY
```

```
2000 PRINT CHR$(125)
2002 RESTORE
2005 K=0
2010 I=1
2020 R=INT(RND(1)*N+1)
2022 FOR J=1 TO R:READ D$,E$:NEXT J
2025 IF Z<>ZZ THEN GOSUB 3000:REM TAUSCH
2030 ? D$
2035 PRINT :PRINT "ANTWORT :  ":PRINT
2037 INPUT W$
2038 PRINT :PRINT :PRINT
2040 IF W$=E$ THEN K=K+1:PRINT "SEHR GUT !":GOTO 2070
2045 PRINT :PRINT :PRINT
2050 PRINT "LEIDER FALSCH !":PRINT
2060 PRINT "RICHTIG MUSS ES HEISSEN :  ";E$
2070 PRINT :PRINT :PRINT :I=I+1
2080 IF I<=A THEN RESTORE :GOTO 2020
2090 KA=(K/A)*100:KA=INT(KA*100+0.5)/100:PRINT
2100 PRINT "ANTEIL KORREKTER ANTWORTEN :  ";KA;" %"
2110 RETURN
3000 REM UP TAUSCH
3010 H$=D$:D$=E$:E$=H$
3020 RETURN
```

5. SCHRITT : Variablenliste

A	=	Anzahl der zu prüfenden Vokabeln
A$	=	Antwortstring
D$	=	deutsche Vokabeln
E$	=	englische Vokabeln
H$	=	Hilfsfeld beim Tausch in UP 3000
I	=	Laufindex
J	=	Laufindex
K	=	Anzahl korrekter Antworten
KA	=	Anteil korrekter Antworten
N	=	Anzahl der Vokabeln im Datenbestand
R	=	Zufallszahl
W$	=	Antwort des Benutzers
Z	=	Ziffernantwort im Menü
ZZ	=	Hilfsgröße (auf 1 gesetzt) Tausch Deutsch-Englisch

6. SCHRITT : Programmbeschreibung

Satz 10-125 : Überschrift, Erläuterungen und Hinweise auf die eventuelle Verlängerung des Programms.

Satz 130-150 : Warten, Löschen des Bildschirms.

Satz 160-170 : Angabe der Zahl der Vokabeln im Datenbestand, Dimensionieren.

Satz 185 : Belegung der Hilfsvariablen ZZ mit 1.

Satz 190-240 : Löschen des Bildschirms, Ausgabe des Menüs und Anforderung der Benutzerauswahl.

Satz 250-260 : Anforderung der Anzahl der zu testenden Vokabeln.

Satz 420 : Beendigung des Programms.

Satz 1000 ff : Daten.

Satz 2000-2110: UP Zufallsauswahl:

2002 : Restaurieren.

2005-2010 : Zähler K und I auf Anfangspositionen.

2020-2022 : Zufallszahl und Lesen.

2025 : Eventueller Tausch.

2030-2038 : Ausgabe der Vokabel und Anforderung der Antwort.

2040 : Bei korrekter Antwort erfolgt Sprung nach 2070.

2050-2060 : Fehlermeldung und Ausgabe der korrekten Antwort.

2070-2080 : Nächste Vokabel.

2090-2100 : Ausgabe des Erfolgsanteils.

2110 : Rücksprung

7. SCHRITT : Ergebnisse

Ergebnisse brauchen hier nicht vorgestellt zu werden. Was im Laufe des Programmablaufs an Ergebnissen ausgegeben wird, ergibt sich unmittelbar aus der Programmbeschreibung (siehe insbesondere Satz 2040 bis Satz 2100).

5.5 Geheimschrift

1. SCHRITT : Vorstellung des Problems

Nach den vorangegangenen Beispielen, die ja sehr "schultypisch" waren, wollen wir nun der Abwechslung halber einmal eine Aufgabe vorstellen, die eher spielerischen Charakter aufweist – die Entwicklung einer Geheimschrift.

Es soll also ein Programm vorgestellt werden, das beliebig einzugebende Sätze so verwandelt, daß sie ein in den Verschlüsselungscode nicht Eingeweihter nicht mehr lesen kann.

Es geht dabei einfach darum, die Symbole, die in der normalen Schreibweise benutzt werden, gemäß einer festzulegenden Umwandlungsregel in andere Symbole, die sich der Benutzer des Programms jeweils selbst ausdenken kann, zu transformieren.

2. SCHRITT : Problemanalyse

Die Problemanalyse bei dieser Aufgabenstellung ist nicht sehr kompliziert:

Wir müssen lediglich dem gebräuchlichen Zeichensatz (A, B, C, ·.· , incl. der Sonderzeichen, die wir benutzen wollen) den "Geheim-Zeichensatz" gegenüberstellen. Beides kann mit geeigneten Stringvariablen geschehen.

Danach muß ein beliebig eingegebener Satz Symbol für Symbol verschlüsselt werden.

Damit dies möglich wird, müssen wir zwei Funktionen benutzen, über die bisher noch nicht gesprochen wurde:

Die erste Funktion dient dazu, die Länge eines Strings festzustellen, also die Zahl der Symbole (incl. Der Leerzeichen), aus denen er sich zusammensetzt.

Die zweite Funktion dient dazu, einzelne Symbole aus einem gegebenen String "herauszubrechen".

Die erstgenannte Funktion erfordert den Aufruf LEN (LEN = length = Länge).

Der Benutzer gebe beispielsweise im Direktmodus ein:

```
PRINT LEN ("COMPUTER")
```

und der Rechner antwortet mit:

```
8
```

Mit dieser Funktion wurde also festgestellt, daß der String "Computer" aus 8 Symbolen besteht.

Die zweite Funktion wird folgendermaßen benutzt:

Teilstring = Ursprungsstring (Zahl 1, Zahl 2)

Im Teilstring wird dann derjenige Teil des Ursprungs-strings gespeichert, der mit dem Symbol Nr. Zahl 1 beginnt und mit dem Symbol Nr. Zahl 2 endet.

Beispiel:

```
10 DIM A$(8),B$(8)
20 A$ = "COMPUTER"
30 B$ = A$(4,7)
40 PRINT B$
```

Dieses Programm druckt den Inhalt des Feldes B$ aus, welches aus der Symbolfolge

PUTE

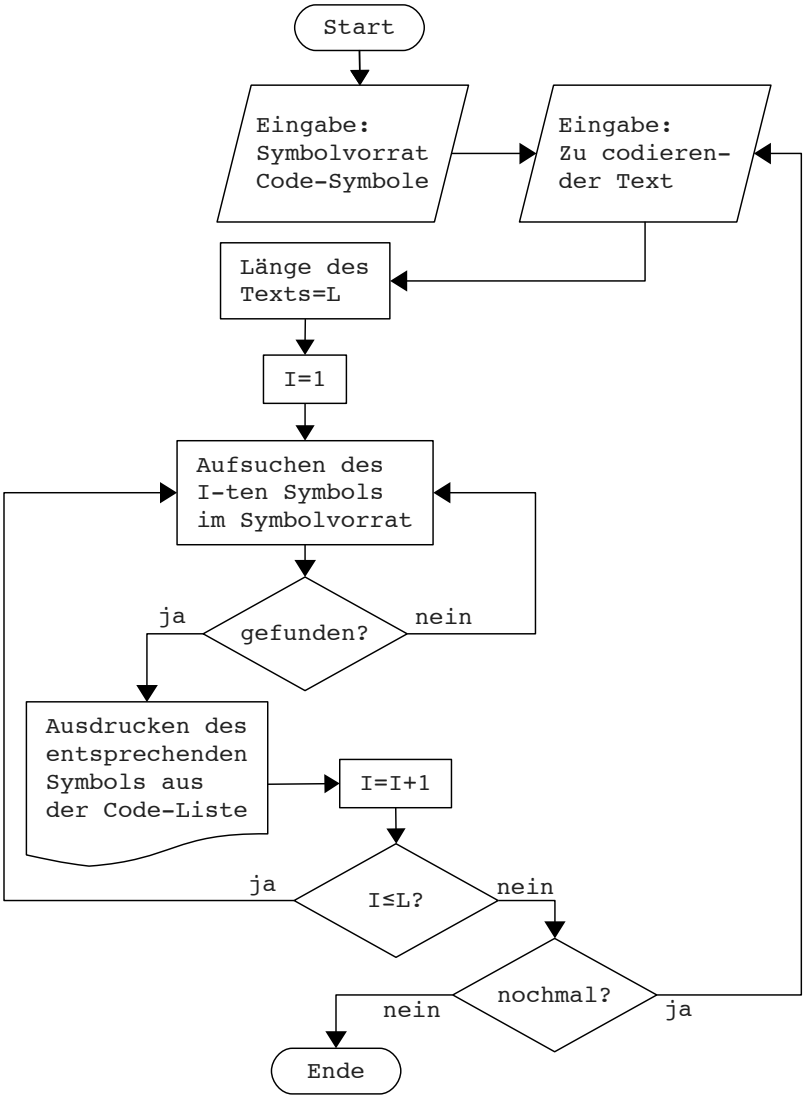
besteht.

Entsprechend können auch einzelne Symbole herausgegriffen werden, wenn Zahl 1 = Zahl 2, also z.B.:

```
30 B$ = A$(4,4)
```

Ergebnis: P

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM S4-GEHEIMSCHRIFT
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM VERSCHLUESSELN VON BELIE-"
40 PRINT "           BIGEN TEXTEN."
50 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT
80 PRINT "SOLL DIE VERSCHLUESSELUNG VERAENDERT"
90 PRINT "WERDEN, MUSS SATZ 210 VERAENDERT WER-"
100 PRINT "DEN."
110 PRINT :PRINT "SOLL DIE VERSCHLUESSELUNG GEHEIM"
120 PRINT "BLEIBEN, MUSS DER STRING C$ UEBER EIN"
130 PRINT "INPUT-STATEMENT BELEGT WERDEN, DESSEN"
140 PRINT "INHALT NICHT VEROEFFENTLICHT WERDEN"
150 PRINT "DARF."
160 PRINT :PRINT "ZUR FORTSETZUNG BITTE CONT EINGEBEN":STOP
190 DIM A$(500),B$(30),C$(30),Z$(1)
200 B$="ABCDEFGHIJKLMNOPQRSTUVWXYZ ,."
210 C$="QWERTYUIOPASDFGHJKL ZX,C.VBNM"
220 PRINT CHR$(125)
230 PRINT "BITTE, DEN ZU VERSCHLUESSELNDEN TEXT"
240 PRINT "EINGEBEN: ":PRINT :PRINT
250 INPUT A$
260 L=LEN(A$)
265 PRINT :PRINT :PRINT
270 FOR I=1 TO L
280 FOR J=1 TO LEN(B$)
290 IF A$(I,I) <> B$(J,J) THEN 320
300 PRINT C$(J,J);
310 GOTO 330
320 NEXT J
330 NEXT I
340 PRINT :PRINT :PRINT
350 PRINT "NOCHMAL (J/N) ";:INPUT Z$
360 IF Z$="J" THEN PRINT CHR$(125) :GOTO 230
370 PRINT :PRINT : PRINT
380 PRINT "ENDE DER AUSGABE":END
```

5. SCHRITT : Variablenliste

A$ = zu verschlüsselnder Text
B$ = Symbolvorrat
C$ = Code-Zeichen
I = Laufvariable
J = Laufvariable
L = Länge des zu verschlüsselnden Texts
Z$ = Antwortstring (J/N)

6. SCHRITT : Programmbeschreibung

Satz 10-150 : Titelausdruck und Erläuterungen.

Satz 160 : Programmunterbrechung.

Satz 190 : Dimensionierungen.

Satz 200 : Vorgabe des Symbolvorrats.

Satz 220-250 : Löschen des Bildschirms und Anforderung des zu verschlüsselnden Textes.

Satz 260 : Bestimmung der Länge dieses Textes.

Satz 270-330 : Codierung.

270 : Beginn der I-Schleife (für alle Symbole des Textes).

280 : Beginn der J-Schleife (für den Symbolvorrat)

290 : Prüfung, ob das jeweilige Textsymbol einem Symbol des Vorrats entspricht;

wenn nein, weiter bei 320,

wenn ja, Ausgabe des entsprechenden Zeichens der Verschlüsselung.

310 : Sprung zum Satz 330, d.h. Behandlung des nächsten Symbols des Textes.

320 : Prüfung des nächsten Symbols aus dem Symbolvorrat.

330 : Nächstes Symbol des Textes.

Satz 340 : Abfrage, ob eine weitere Verschlüsselung gewünscht wird.

Satz 370-380 : Beendigung des Programms.

7. SCHRITT : Ergebnisse

Die Ergebnisse dieses Programms hängen davon ab, welche Codierungsregel vorgegeben wird (siehe Satz 210: C$) und welcher Text zu verschlüsseln ist (siehe Satz 250: A$).

Geben wir bei dem vorliegenden Programm z.B. als zu verschlüsselnden Text WERNER VOSS ein, so erhalten wir:

,TKFTKBXGLL

6

Kapitel 6: Biologie/Ökologie

6.1 Vorbemerkung

Der Rechnereinsatz bei biologisch-ökologischen Problemen bezieht sich insbesondere darauf, Entwicklungs- und Wachstumsvorgänge zu simulieren (natürlich könnten aber auch Abfrageprogramme nach dem Muster des vorangegangenen Kapitels entwickelt werden). Diese Entwicklungen können dann auf dem Bildschirm tabellarisch oder auch graphisch ausgegeben werden.

6.2 Ungebremstes Wachstum

1. SCHRITT : Vorstellung des Problems

Eine Population mit dem Anfangsbestand XØ wachse mit einer jährlichen Wachstumsrate von R Prozent (R sei positiv). Wenn dieses Wachstum nicht an äußere Grenzen stößt - was aber auf lange Sicht natürlich sehr unwahrscheinlich ist - dann stellt sich ein sog. exponentielles Wachstum ein.

Der Ablauf eines solchen sehr einfachen Wachstumsprozesses soll durch ein BASIC-Programm simuliert werden.

2. SCHRITT : Problemanalyse

Für das entsprechende Programm müssen der Anfangsbestand und die jährliche Zuwachsrate eingegeben werden.

Nach einem Jahr erhält man dann den Bestand X1:

X1 = XØ + XØ*R/1ØØ = XØ*(1 + R/1ØØ)

Nach zwei Jahren ergibt sich X2:

X2 = X1*(1 + R/1ØØ) = XØ*(1+R/1ØØ)*(1+R/1ØØ)

Daraus folgt:

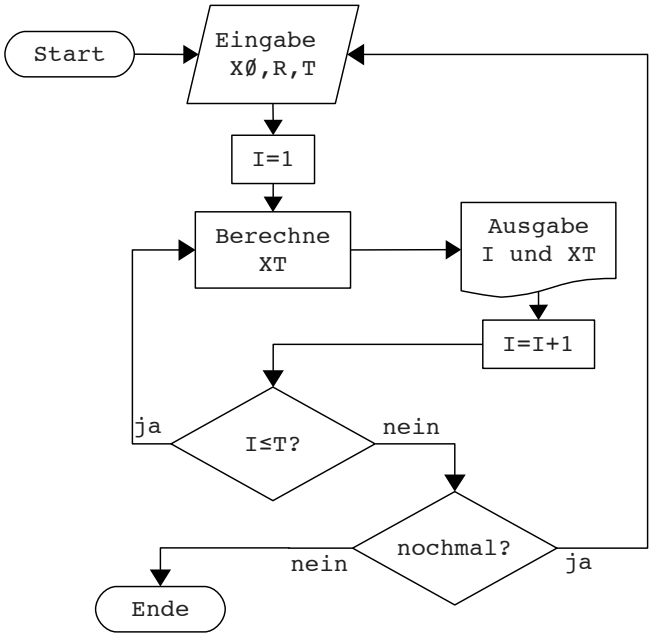
$$X2 = x0*(1 + R/100)^2$$

Generell gilt deshalb nach T Jahren

$$XT = x0*(1 + R/100)^T$$

Diese Formel zeigt den Rechenalgorithmus für das entsprechende BASIC-Programm.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM B1-WACHSTUM
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR TABELLARISCHEN DARSTEL-"
40 PRINT"    LUNG UNGEBREMSTEN WACHSTUMS."
45 PRINT :PRINT :PRINT
50 PRINT "         PROF.DR.W.VOSS, 1984"
55 DIM A$(1)
60 PRINT :PRINT :PRINT :PRINT
70 PRINT "ANFANGSBESTAND      : ";
75 INPUT X0:PRINT
80 PRINT "WACHSTUMSRATE IN %  : ";
85 INPUT R:PRINT
90 PRINT "ANZAHL DER JAHRE    : ";
95 INPUT T:PRINT CHR$(125)
100 PRINT "NACH ... JAHREN    BESTAND"
105 PRINT :PRINT
110 I=1
120 XT=X0*(1+R/100)^I:XT=INT(XT*100+0.5)/100
130 PRINT "      ";I;"                ";XT
140 IF I/10=INT(I/10) THEN PRINT :PRINT "BITTE CONT
    EINGEBEN":STOP
145 I=I+1
150 IF I<=T THEN GOTO 120
160 PRINT :PRINT :PRINT "NOCHMAL (J/N) ";:INPUT A$
170 IF A$="J" THEN PRINT CHR$(125):GOTO 70
180 PRINT :PRINT "ENDE DER AUSGABE":END
```

5. SCHRITT : Variablenliste

A$ = Antwortstring (Ja,Nein)

I = Laufindex

R = Wachstumsrate in %
T = Zeit

XØ = Anfangsbestand
XT = Bestand zum Zeitpunkt T

6. SCHRITT : Programmbeschreibung

Satz 1Ø-6Ø : Überschrift.

Satz 7Ø-95 : Vorgabe der Input-Informationen, die das Programm benötigt.

Satz 1ØØ-1Ø5 : Ausgabe einer Tabellenüberschrift.

Satz 11Ø : Vorgabe der ersten Periode.

Satz 12Ø : Berechnung des jeweiligen Bestandes.

Satz 13Ø : Ausgabe.

Satz 14Ø : Nach jeder zehnten Periode wird das Programm unterbrochen (also dann, wenn I durch 1Ø ohne Rest teilbar ist).
Fortsetzung ist nur mit dem Kommando CONT möglich, deshalb ein entsprechender Hinweis.

Satz 145 : Übergang zur nächsten Periode.

Satz 15Ø : Solange I kleiner gleich T ist, ist das Programm fortzusetzen (Satz 12Ø).

Satz 16Ø-18Ø : Beendigung des Programms, es sei denn, es wird ein zusätzlicher Programmlauf gewünscht (dann nach Löschen des Bildschirms zurück zu Satz 7Ø).

7. SCHRITT : Ergebnisse

Geben wir z.B. als Anfangsbestand den Wert 5ØØ ein, als Wachstumsrate den Wert 4 (%) und als Zeitdauer den Wert 8 (z.B. Jahre), so erzeugt das Programm die folgende Ausgabe:

```
NACH ... JAHREN    BESTAND


1                 52Ø
2                 54Ø.8
3                 562.43
4                 584.93
5                 6Ø8.33
6                 632.66
7                 657.97
8                 684.28


NOCHMAL (J/N) ?
```

6.3 Gebremstes Wachstum

1. SCHRITT : Vorstellung des Problems

Exponentielles Wachstum, wie es im vorangegangenen Abschnitt vorgestellt wurde, gibt es in der Realität praktisch nicht oder nur in sehr beschränkten Zeitabschnitten. Es muß vielmehr mit hemmenden Effekten gerechnet werden, die in der Regel um so stärker werden, je größer die Werte der interessierenden Variablen schon geworden sind.

Man denke beispielsweise daran, daß eine exponentiell wachsende Bevölkerung sehr rasch an Ernährungsgrenzen stößt, die dem weiteren Wachstum immer größere Hemmnisse auferlegen.

Eine solcherart "gedämpfte" Entwicklung soll im folgenden BASIC-Programm simuliert werden.

2. SCHRITT : Problemanalyse

Die Problemanalyse führt hier zu einem ähnlichen Lösungsweg wie im vorangegangenen Abschnitt:

Zunächst gilt auch hier, daß Ausgangswerte benötigt werden:

XØ = Ausgangsbestand
RØ = Jährliche Zunahmerate (in %) zu Beginn des Entwicklungsprozesses

Im Gegensatz zum Beispiel zuvor gehen wir jetzt aber davon aus, daß die jährliche Zunahmerate nicht konstant bleibt, sondern um so kleiner wird, je größer die Population schon geworden ist.

Da die Population im Zeitablauf wächst, können wir in einem sehr einfachen Denkmodell also einfach davon ausgehen, daß die Zuwachsrate mit fortschreitender Zeit absinkt. Beispielsweise können wir die Zuwachsrate bei Verdopplung der Zeit halbieren, bei Vervierfachung der Zeit auf ein Viertel zurückgehen lassen usw.

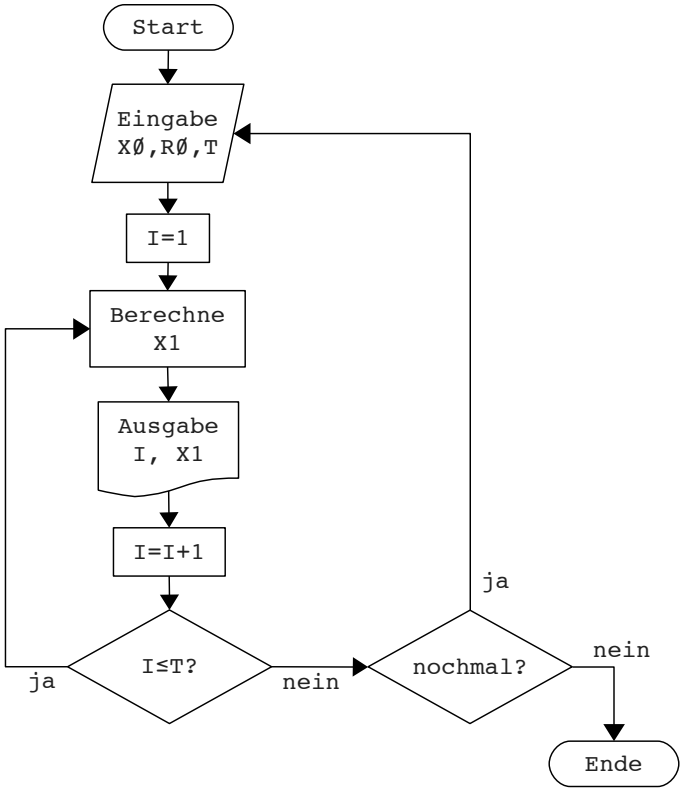
Natürlich lassen sich auch andere mathematische Modelle vorstellen, um diesen Bremsvorgang zu simulieren - darauf kommt es aber ja in diesem Zusammenhang nicht entscheidend an.

Den so beschriebenen Rechenalgorithmus können wir mathematisch folgendermaßen fassen:

$$X1 = X\emptyset * (1 + R/I)$$

Das heißt, der Bestand der folgenden Periode X1 berechnet sich aus dem der vorhergehenden Periode x16 so wie im letzten Beispiel; die Wachstumsrate aber wird durch den Laufindex I dividiert, so daß sie um so kleiner wird, je höher I wird, d.h. je weiter die Zeit schon fortgeschritten ist.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM B2-GEBREMSTES WACHSTUM
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR TABELLARISCHEN DARSTEL-"
40 PRINT "LUNG EINES GEBREMSTEN WACHSTUMSPRO-"
45 PRINT "                ZESSES."
47 PRINT :PRINT :PRINT
50 PRINT "         PROF.DR.W.VOSS, 1984"
55 DIM A$(1)
60 PRINT :PRINT :PRINT :PRINT :PRINT
70 PRINT "ANFANGSBESTAND        : ";
75 INPUT X0:PRINT
80 PRINT "W.RATE IM 1. JAHR     : ";
85 INPUT R:PRINT
90 PRINT "WIEVIELE JAHRE        : ";
95 INPUT T:PRINT CHR$(125)
100 PRINT "NACH ... JAHREN    BESTAND    ZUWACHS"
102 PRINT ,,,"(X)"
105 PRINT :PRINT
110 I=1
120 X1=X0+X0*(R/I)/100:X1=INT(X1*100+0.5)/100
140 Z=(X1-X0)/X0:Z=Z*100:Z=INT(Z*100+0.5)/100
150 PRINT "      ";I,,X1,Z
160 X0=X1:I=I+1:IF I<=T THEN GOTO 120
180 PRINT :PRINT "NOCHMAL ( J/N) ";:INPUT A$
190 IF A$="J" THEN PRINT CHR$(125):GOTO 70
200 PRINT :PRINT "ENDE DER AUSGABE":END
```

5. SCHRITT : Variablenliste

A$ = Antwortstring (Ja, Nein)
I = Laufindex (Anzahl der Perioden)
R = Wachstumsrate zu Beginn des Prozesses
T = Zeitdauer des Prozesses

X0 = Anfangsbestand
X1 = Bestand nach einer Periode
Z = Bestandsveränderung

6. SCHRITT : Programmbeschreibung

Satz 1Ø-6Ø : Überschrift und Erläuterungen.

Satz 7Ø-95 : Anforderung der Ausgangsdaten und Löschen des Bildschirms.

Satz 1ØØ-1Ø5 : Ausgabe einer Tabellenüberschrift.

Satz 11Ø : Vorgabe der ersten Periode.

Satz 12Ø-15Ø : Berechnung des Periodenbestands, des Zuwachses und Ausgabe der entsprechenden Werte nach Rundung auf zwei Dezimalstellen.

Satz 16Ø : Belegung von XØ mit X1 und Fortführung der Berechnungen, solange I nach Erhöhung um 1 kleiner als der vorgegebene Zeitwert T bleibt.

Satz 16Ø-18Ø : Abfrage, ob noch eine Berechnung gewünscht (wenn ja, zurück nach 7Ø) und Beendigung des Programms.

7. SCHRITT : Ergebnisse

Geben wir zum Beispiel als Anfangsbestand den Wert 5ØØ ein, und als Zuwachsrate im ersten Jahr den Wert 4 % und schließlich die Angabe, daß der Prozeß 8 Jahre dauern soll, so erhalten wir:

NACH ... JAHREN	BESTAND	ZUWACHS (%)
1	52Ø	4
2	53Ø.4	2
3	537.47	1.33
4	542.84	1
5	547.18	Ø.8
6	55Ø.83	Ø.67
7	553.98	Ø.57
8	556.75	Ø.5

NOCHMAL (J/N) ?

6.4 Umweltverschmutzung

1. SCHRITT : Vorstellung des Problems

Wachsende Populationen hemmen in der Regel ihre weitere Entwicklung selbst, nämlich dadurch, daß sie ihre Lebensbedingungen in zunehmendem Maße belasten.

Typisches Beispiel ist die zunehmende Umweltverschmutzung während der Industrialisierungsphasen.

Also auch hier tritt eine "Wachstumsbremse" zutage - ähnlich wie im vorangegangenen Abschnitt - die das ungebremste Wachstum, wie wir es in Abschnitt 6.2 kennengelernt haben, modifiziert.

Im Gegensatz aber nun zum Beispiel im vorangegangenen Abschnitt, wirkt diese "Bremse" nicht in Abhängigkeit von der Populationsgröße selbst, sondern indirekt in Abhängigkeit von einer Variablen, die ihrerseits proportional mit der Populationsgröße zusammenhängt.

2. SCHRITT : Problemanalyse

Die Population in der Periode 1 ergibt sich aus der der Periode Ø jetzt nicht mehr als

$$X1 = XØ + XØ * R/1ØØ$$

sondern als:

$$X1 = XØ + XØ * (R/1ØØ - P*BØ)$$

Dabei ist BØ die Umweltbelastung zum Zeitpunkt Ø, die über einen geeigneten Parameter P (z.B. in der Größenordnung P = Ø.Ø1) die Populationszuwächse bremst.

Natürlich brauchen wir jetzt aber auch eine funktionale Beziehung, welche die Größe B1 (Umweltbelastung in der Periode 1) in Abhängigkeit von BØ und XØ erklären kann.

Als Beispiel einer solchen Beziehung mag gelten:

$$B1 = BØ + A * XØ$$

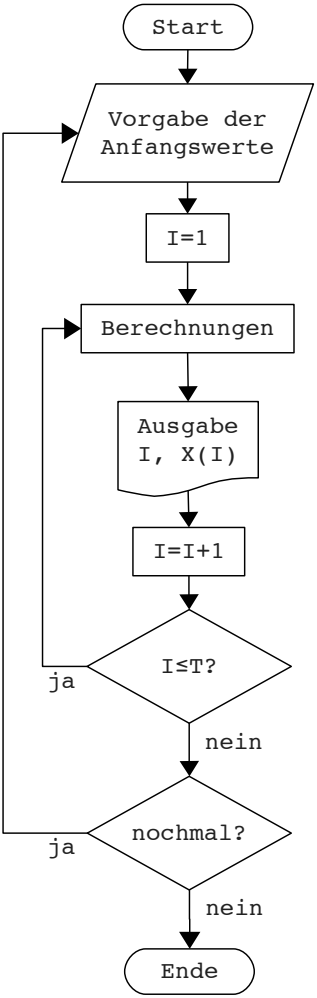
Von einer Belastungsgröße von z.B. BØ = Ø ausgehend, wächst die Belastung in Abhängigkeit von der erreichten Populationsgröße nach Maßgabe eines geeignet festzusetzenden Parameters A (z.B. A = Ø.Ø15).

Somit gilt also der folgende Algorithmus:

(1) $B1 = BØ + A*XØ$

(2) $X1 = XØ + XØ * (R/1ØØ - P*B1)$

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM B3-UMWELT
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR TABELLARISCHEN SIMULA-"
40 PRINT "TION EINES WACHSTUMSPROZESSES, DER"
50 PRINT "DURCH SEINE EIGENE DYNAMIK GEBREMST."
55 PRINT "                WIRD."
57 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
65 DIM A$(1)
70 PRINT :PRINT :PRINT :PRINT
80 PRINT "DIESES PROGRAMM BENOETIGT DIE FOLGEN-"
90 PRINT "DEN INPUT-INFORMATIONEN :":PRINT
100 PRINT "ANFANGSBESTAND            :  ";
105 INPUT X0:PRINT
110 PRINT "WIEVIELE JAHRE            :  ";
112 INPUT T:PRINT
115 PRINT "JAEHRL. WACHSTUMSRATE (%):  ";
117 INPUT R:PRINT
119 PRINT "FAKTOR UMWELTBELASTUNG   :  ";
120 INPUT P
130 PRINT CHR$(125)
140 PRINT "ZEIT","BESTAND","ZUWACHS","BELASTUNG"
145 PRINT :PRINT
150 I=1:A=0.015:B0=0
160 B1=B0+A*X0
170 X1=X0+X0*(R/100-P*B0)
175 IF X1<=0 THEN PRINT "POPULATION VERNICHTET":GOTO 210
180 Z=(X1-X0)/X0:Z=Z*100
185 X1=INT(X1*100+0.5)/100:B1=INT(B1*100+0.5)/100
187 Z=INT(Z*100+0.5)/100
190 PRINT "  ";I,X1,Z,B1
200 I=I+1:IF I<=T THEN B0=B1:X0=X1:GOTO 160
210 PRINT :PRINT "NOCHMAL ( J/N) ";:INPUT A$
220 IF A$="J" THEN PRINT CHR$(125):GOTO 80
230 PRINT :PRINT "ENDE DER AUSGABE":END
```

5. SCHRITT : Variablenliste

A = Faktor populationsbedingte Umweltbelastung
A$ = Stringvariable (Ja, Nein)
BØ = Umweltbelastung Anfang
B1 = Umweltbelastung nächste Periode
I = Laufindex
P = Faktor Umweltbelastung
R = Jährliche Wachstumsrate
T = Zeit
XØ = Population Anfang
X1 = Population nächste Periode
Z = Populationszuwachs

6. SCHRITT : Programmbeschreibung

Satz 1Ø-7Ø : Überschrift und Erläuterungen.

Satz 8Ø-12Ø : Anforderung der Input-Informationen.

Satz 13Ø-145 : Löschen des Bildschirms und Ausgabe einer Tabellenüberschrift.

Satz 15Ø : Vorgabe weiterer Ausgangsparameter.

Satz 16Ø-17Ø : Berechnungen für die folgende Periode.

Satz 175 : Wenn die neu berechnete Population kleiner oder gleich null wird, ist eine entsprechende Meldung auszugeben und Sprung zum Satz 21Ø.

Satz 18Ø : Berechnung von Z.

Satz 185-187 : Runden auf zwei Dezimalstellen.

Satz 19Ø : Ausgabe der Periodenergebnisse.

Satz 2ØØ : Erhöhung des Laufindex I;
solange I kleiner als T bleibt, wird xØ mit X1 und BØ mit B1 belegt und das Programm kehrt zurück zum Satz 16Ø.

Satz 21Ø-23Ø : Abfrage, ob eine weitere Berechnung gewünscht wird (wenn ja, Löschen des Bildschirms und zurück zu Satz 8Ø) und Beendigung des Programms.

7. SCHRITT : Ergebnisse

Geben wir als Ausgangsbestand z.B. den Wert 1ØØØ, als Zeithorizont 8 Jahre, als jährliche Wachstumsrate 5 % und als Faktor der Umweltbelastung z.B. den Wert Ø.Ø1 ein, so erhalten wir die folgenden Angaben:

ZEIT	BESTAND	ZUWACHS	BELASTUNG
1	1Ø5Ø	5	15
2	945	-1Ø	3Ø.75
3	7Ø1.66	-25.75	44.93
4	421.49	-39.93	55.45
5	2Ø8.85	-5Ø.45	61.77
6	9Ø.29	-56.77	64.9
7	36.21	-59.9	66.25
8	14.Ø3	-61.25	66.79

NOCHMAL (J/N) ?

7

Kapitel 7: Erdkunde/Geschichte

7.1 Vorbemerkung

Es können in diesem Kapitel ähnliche Programme erstellt werden wie im Kapitel "Sprachen", also vor allem Test- und Abfrageprogramme. Auch hier werden wir uns dabei auf exemplarische Datenbestände beschränken, d.h. es wird nur gezeigt, wie die Programme aufgebaut werden müssen und wie sie dann funktionieren.

Sollen sie praktisch genutzt werden, müssen erst, wie unschwer zu sehen ist, die kompletten Datenbestände erzeugt und eingegeben werden. Dazu müssen die DATA-Statements ergänzt und die jeweilige Vorgabe von N entsprechend geändert werden.

7.2 Historische Jahreszahlen

1. SCHRITT : Vorstellung des Problems

Es soll ein Programm vorgestellt werden, welches dazu benutzt werden kann, das Erlernen historischer Jahreszahlen zu erleichtern und die erlernten Daten zu trainieren.

Zu diesem Zweck stellt der Rechner dem Benutzer historische Ereignisse vor und verlangt von ihm, daß er die korrekte Jahreszahl dieses geschichtlichen Ereignisses angibt. Es bleibt dabei dem Benutzer überlassen zu entscheiden, wieviele Fragen er sich pro Testserie vom Rechner stellen lassen möchte.

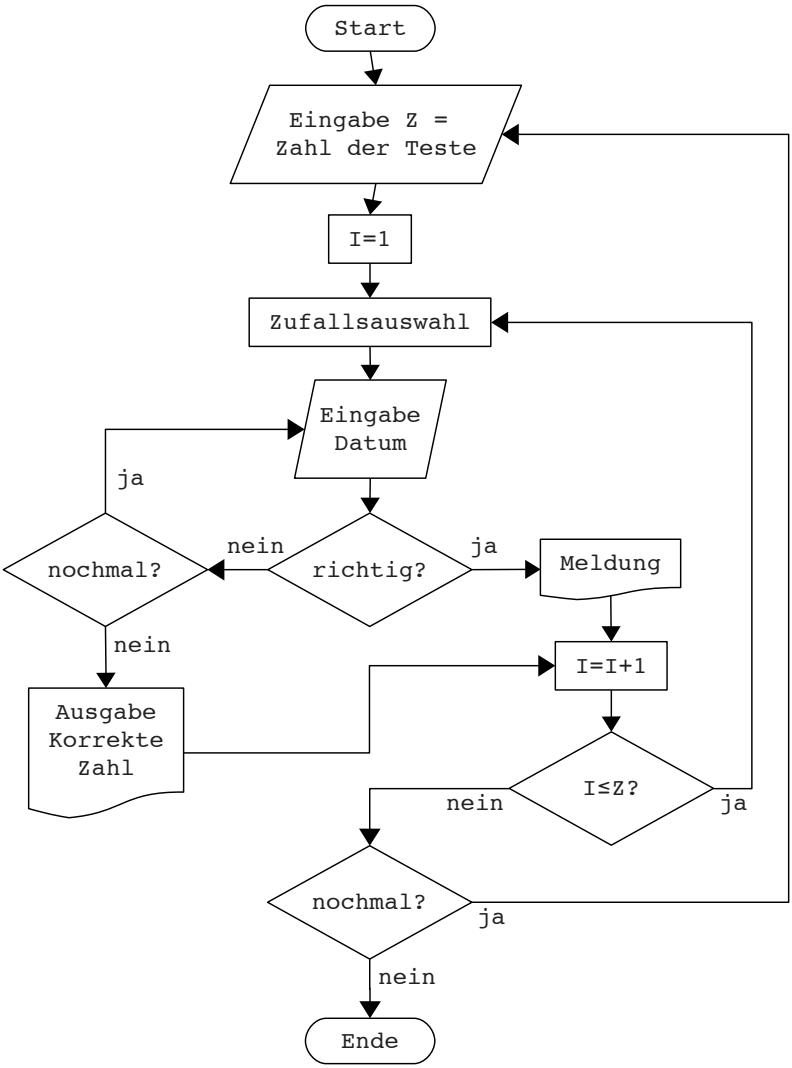
Gibt der Benutzer die korrekte Antwort, so reagiert der Rechner mit Lob; erhält er hingegen keine korrekte Antwort, so soll er den Benutzer fragen, ob dieser noch einen Versuch wagen möchte. Wird dies abgelehnt, soll das Programm dafür sorgen, daß die korrekte Antwort auf dem Bildschirm erscheint.

2. SCHRITT : Problemanalyse

Die Vorstellung des Problems hat schon hinreichend verdeutlicht, wie bei diesem Programm im einzelnen vorzugehen ist:

Wir benötigen einen Informationseingabeteil, der Benutzer muß gefragt werden, wieviele Abfragen er über sich ergehen lassen möchte und das Programm muß die jeweils gegebenen Antworten prüfen. Desweiteren tauchen keine besonderen Schwierigkeiten auf.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM E1-JAHRESZAHLEN
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM ABFRAGEN HISTORISCHER"
35 PRINT "                DATEN.  "
37 PRINT :PRINT :PRINT
40 PRINT "         PROF.DR.W.VOSS, 1984"
50 PRINT :PRINT :PRINT :PRINT
60 PRINT "DIESES PROGRAMM BEZIEHT SICH NUR AUF"
77 PRINT "10 AUSGEWAEHLTE HISTORISCHE DATEN"
80 PRINT "(SIEHE DATA-STATEMENTS IN 500 FF.)":PRINT
90 PRINT "SOLLEN MEHR DATEN VERWENDET WERDEN,"
100 PRINT "SO MUESSEN DIE DATA ERGAENZT UND N IN"
110 PRINT "SATZ 150 VERAENDERT WERDEN."
120 PRINT :PRINT
140 PRINT "ZUR FORTSETZUNG BITTE CONT EINGEBEN !":STOP
150 N=10
155 DIM J$(5),S$(50),A$(1),K$(5)
160 PRINT CHR$(125)
170 REM TEST
180 PRINT "WIEVIELE ABFRAGEN WERDEN GEWUENSCHT ?"
190 PRINT :PRINT "BITTE ZAHL ANGEBEN : ";:INPUT Z
200 FOR I=1 TO Z
201 RESTORE
202 R=INT(RND(1)*N+1)
205 PRINT CHR$(125)
208 FOR J=1 TO R:READ J$,S$ :NEXT J
210 PRINT "WANN WAR: ":PRINT
230 PRINT S$:PRINT :PRINT
240 PRINT "IM JAHRE : ";: INPUT K$
245 PRINT :PRINT
250 IF K$=J$ THEN PRINT "          P R I M A !":GOTO 300
255 PRINT :PRINT
260 PRINT "       LEIDER FALSCH !"
265 PRINT :PRINT
270 PRINT "NOCH EIN VERSUCH (J/N) ";:INPUT A$
280 IF A$="J" THEN PRINT CHR$(125):GOTO 210
285 PRINT :PRINT
290 PRINT "DIE RICHTIGE JAHRESZAHL LAUTET: ";J$
300 FOR II=1 TO 1000:NEXT II:NEXT I
305 PRINT :PRINT :PRINT
310 PRINT "NOCH EIN TEST (J/N) ";:INPUT A$
320 IF A$="J" THEN PRINT CHR$(125):GOTO 180
330 PRINT :PRINT "ENDE":END
```

```
500 DATA 1939,BEGINN DES ZWEITEN WELTKRIEGS
510 DATA 800,KAISERKROENUNG KARLS DES GROSSEN
520 DATA 1949,GRUENDUNG DER BUNDESREPUBLIK
530 DATA 9,SCHLACHT IM TEUTOBURGER WALD
540 DATA 1832,HAMBACHER FEST
550 DATA 1871,GRUENDUNG DES DEUTSCHEN KAISERREICHS
560 DATA 1890,ERSTER EINSATZ DER LOCHKARTE
570 DATA 1812,NAPOLEON VOR MOSKAU
580 DATA 1763,FRIEDE VON HUBERTUSBURG
590 DATA 1495,REICHSTAG VON WORMS
```

5. SCHRITT : Variablenliste

A$ = Stringvariable für Antworten
I = Laufindex
II = Laufindex
J = Laufindex
J$ = Jahreszahlen
N = Anzahl der Datensätze
R = Zufallszahl
S$ = Historische Sachverhalte
Z = Anzahl der Tests

6. SCHRITT : Programmbeschreibung

Satz 10-120 : Überschrift und Erläuterungen.

Satz 140 : Warten.

Satz 150-170 : Vorgabe der Anzahl der Datensätze, Dimensionieren, Löschen des Bildschirms.

Satz 180-190 : Anforderung der Zahl der Tests.

Satz 200-300 : Tests:
Satz 201-245: Zufallsauswahl eines historischen Ereignisses und Anforderung der Jahreszahl als Antwort des Benutzers
Satz 250 : Falls korrekt, Meldung und weiter bei 300.

Satz 255-285: Falls nicht korrekt, Meldung und Frage, ob noch ein Versuch (wenn ja, zurück nach Satz 210).

Satz 290 : Wenn nein, korrekte Ausgabe.

Satz 300 : Warteschleife und dann zurück nach Satz 200.

Satz 305-330 : Frage, ob noch ein Testlauf gewünscht;

wenn ja, zurück zu Satz 180 nach Löschen des Bildschirms;

wenn nein, Ende des Programms.

Satz 500-590 : Daten.

7. SCHRITT : Ergebnisse

Die Ergebnisse brauchen hier nicht vorgestellt zu werden. Am sinnvollsten ist es, wenn der Leser unter Beachtung der Inhalte der DATA-Statements das Programm einfach ausprobiert und es dann gemäß seinen eigenen Wünschen erweitert.

7.3 Die Hauptstädte der Länder

1. SCHRITT : Vorstellung des Problems

Auch das Programm, das in diesem Abschnitt vorgestellt werden soll, ist ein sog. Trainingsprogramm. Der Benutzer soll auf die Fragen des Rechners Antwort geben. Gibt der Rechner zufällig ausgewählte Bundesländer vor, soll der Benutzer die Hauptstädte dieser Länder eingeben; gibt der Rechner hingegen Hauptstädte vor, soll der Benutzer die dazugehörigen Bundesländer nennen.

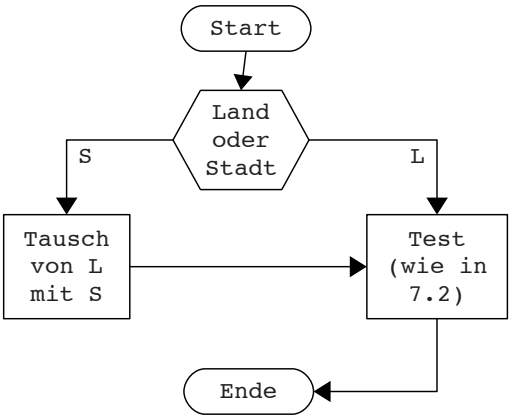
Insoweit bieten sich hier also keine neuen Probleme, so daß wir uns hier sehr kurz fassen können. Auch dieses Programm hat also eher exemplarischen Charakter:

Ohne Schwierigkeiten kann der Benutzer andere Sachverhalte in den DATA-Statements dieses Programms erfassen und gelangt auf diese Weise zu einem generell einsetzbaren Abfrageprogramm, das für die verschiedensten Schulfächer verwendet werden könnte.

2. SCHRITT : Problemanalyse

Es wurde schon erwähnt, daß hier keine neuen Probleme auftauchen , so daß dieser Schritt entsprechend kurz gehalten werden kann. Entsprechende Fragestellungen wurden in vorangegangenen Beispielen schon hinreichend diskutiert.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM E2-LAENDER
20 PRINT CHR$(125)
30 PRINT "      LERNKONTROLLE IN ERDKUNDE."
40 PRINT :PRINT :PRINT
50 PRINT "         PROF.DR.W.VOSS, 1984"
60 PRINT :PRINT :PRINT
70 PRINT "DIESES PROGRAMM FORDERT NACH ANGABE "
80 PRINT "DER DEUTSCHEN BUNDESLAENDER DEREN"
90 PRINT "HAUPTSTAEDTE AN UND UMGEKEHRT."
95 PRINT :PRINT
100 PRINT "BEI ANDEREN VORGABEN VON DATEN IN DEN"
110 PRINT "STATEMENTS 500 FF. UND AENDERUNG VON"
120 PRINT "N IN SATZ 160 KANN DAS PROGRAMM AUCH"
130 PRINT "FUER ANDERE ZWECKE VERWENDET WERDEN."
140 PRINT :PRINT
150 PRINT "ZUR FORTSETZUNG BITTE CONT EINGEBEN !":STOP
160 N=11
170 DIM L$(30),S$(30),H$(30),A$(1),C$(10),X$(30)
190 PRINT CHR$(125)
200 PRINT "SOLLEN LAENDER ODER HAUPTSTAEDTE VOR-"
210 PRINT "GEGEBEN WERDEN (L/S) ";:INPUT A$
215 C$="HAUPTSTADT"
220 IF A$="S" THEN C$="LAND"
230 REM TEST
240 PRINT CHR$(125)
250 PRINT "WIEVIELE ABFRAGEN ";:INPUT Z
255 PRINT :PRINT :PRINT
260 FOR I=1 TO Z
265 PRINT :PRINT
270 RESTORE
280 R=INT(RND(1)*N+1)
290 FOR J=1 TO R:READ L$,S$:NEXT J
300 IF A$="S" THEN GOSUB 1000:REM TAUSCH
310 PRINT L$
320 PRINT :PRINT :PRINT C$;" : ";:INPUT X$
330 IF X$=S$ THEN PRINT :PRINT "K O R R E K T !":GOTO 400
340 PRINT :PRINT "F E H L E R !":PRINT
350 PRINT :PRINT :PRINT
360 PRINT "DIE RICHTIGE ANTWORT LAUTET : "
370 PRINT :PRINT S$:PRINT :PRINT
400 NEXT I
405 PRINT :PRINT :PRINT
410 PRINT :PRINT "ENDE DES TESTS":END
```

```
500 DATA SCHLESWIG-HOLSTEIN,KIEL
510 DATA HAMBURG,HAMBURG,BREMEN,BREMEN
520 DATA BERLIN,BERLIN,NIEDERSACHSEN,HANNOVER
530 DATA NORDRHEIN-WESTFALEN,DUESSELDORF
540 DATA HESSEN,WIESBADEN,RHEINLAND-PFALZ,MAINZ
550 DATA SAARLAND,SAARBRUECKEN,BAYERN,MUENCHEN
560 DATA BADEN-WUERTEMBERG,STUTTGART
1000 REM UP TAUSCH
1010 H$=L$:L$=S$:S$=H$
1020 RETURN
```

5. SCHRITT : Variablenliste

A$ = Stringvariable

C$ = Hilfsvariable, die - je nach Auswahl - mit "Stadt" oder "Land" belegt wird

H$ = Hilfsvariable beim Tausch

I = Laufindex

J = Laufindex

L$ = Länder

N = Anzahl der Datensätze

S$ = Städte

X$ = Eingabe des Benutzers

6. SCHRITT : Programmbeschreibung

Satz 10-130 : Überschrift und Erläuterungen.

Satz 140-150 : Warten.

Satz 160-170 : Dimensionieren.

Satz 190-210 : Abfrage, ob Länder oder Hauptstädte vorgegeben werden sollen.

Satz 215-220 : Werden Städte gewählt, so wird die Hilfsvariable C$ mit "Land" belegt und es erfolgt ein Sprung nach 250.

Satz 23Ø-24Ø : Beginn des Abfragens.

Satz 25Ø : Anforderung der Zahl der Abfragen.

Satz 255 : Drei Leerzeilen.

Satz 26Ø-4ØØ : Durchführung der gewünschten Zahl von Abfragen.

265 :
29Ø : Restaurieren des Datenbestands und Zufallsauswahl eines Wertepaars.

3ØØ : Wurde bei der Menüvorgabe S gewählt (siehe 19Ø-21Ø), so wird S$ mit L$ im Unterprogramm 1ØØØ vertauscht.

31Ø : Ausgabe der Frage.

32Ø : Eingabe der Antwort des Benutzers.

34Ø :
37Ø : Ausgabe der korrekten Antwort.

Satz 4Ø5-41Ø : Beendigung des Programms.

Satz 5ØØ-56Ø : Daten.

Satz 1ØØØ-1Ø2Ø: Unterprogramm zum Tausch von S$ und L$ (siehe Satz 3ØØ).

7. SCHRITT : Ergebnisse

Auch bei diesem Programm ist es nicht erforderlich, Ergebnisausdrucke vorzustellen. Die Durchsicht des Programms oder besser noch das Ausprobieren zeigt sehr deutlich, was im einzelnen geschieht.

7.4 Bevölkerungsentwicklung in verschiedenen Nationen

1. SCHRITT : Vorstellung des Problems

Es soll in diesem Abschnitt ein Programm vorgestellt werden, welches für verschiedene Länder dieser Erde die Bevölkerungsentwicklung bis zum Jahre 2000 prognostiziert.

Als zu betrachtende Länder bzw. Ländergruppen haben wir ausgewählt:

Welt (als alle Länder umfassend)

- Industrieländer
- Entwicklungsländer

VR China
Indien
Japan
USA
UdSSR
Bundesrepublik Deutschland
DDR.

2. SCHRITT : Problemanalyse

Die geplanten Bevölkerungsprognosen machen es erforderlich, daß man sich für eine bestimmte Prognosemethode entscheidet.

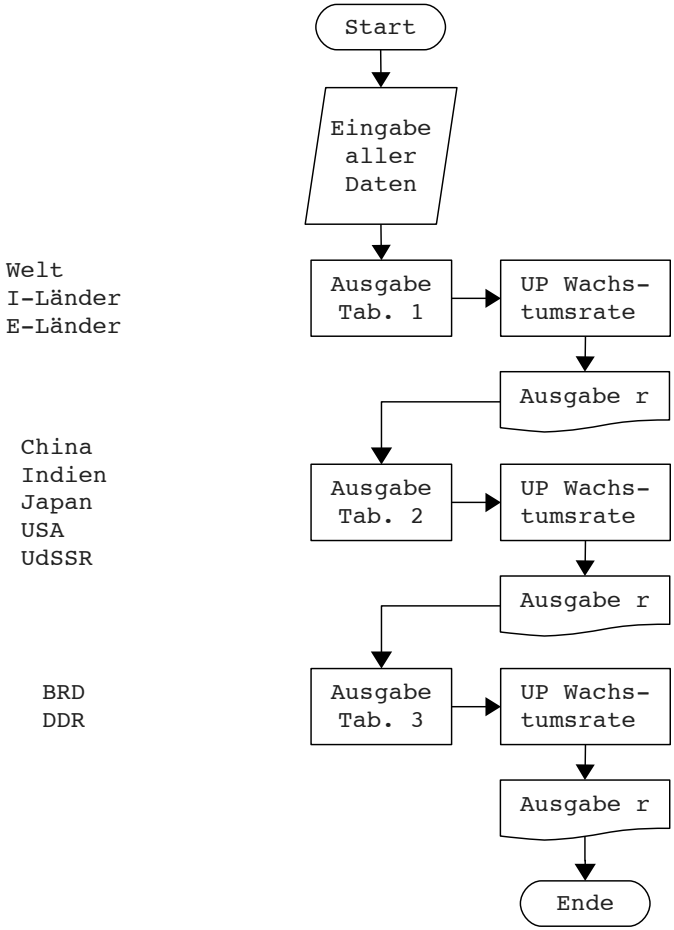
Es gibt eine große Zahl verschiedener Methoden, die im einzelnen nicht hier diskutiert werden können.

Wir haben für das folgende Programm die Werte übernommen, die sich bei den Modellprognosen ergeben haben, wie sie in dem Bericht GLOBAL 2000 (Washington 1980) veröffentlicht wurden.

Aus diesen Daten berechnen wir zusätzlich die jahresdurchschnittlichen Wachstumsraten, um damit durch das Programm eine zusätzliche Information bereitstellen zu lassen.

Unter programmlogischen Gesichtspunkten tauchen keine besonderen Schwierigkeiten auf, wenn man einmal davon absieht, daß wegen der Fülle der Ergebnisse an mehreren Stellen Programmunterbrechungen erforderlich sind.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM E3-BEVOELKERUNG
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR TABELLARISCHEN DARSTEL-"
40 PRINT "LUNG DER BEVOELKERUNGSENTWICKLUNG "
50 PRINT "        VERSCHIEDENER LAENDER."
55 PRINT :PRINT :PRINT
50 PRINT "          PROF.DR.W.VOSS, 1984"
70 DIM B(10,6),J(6),WR(10),L$(20)
71 DIM L1$(5),L2$(5),L3$(5),L4$(5),L5$(5)
72 DIM L6$(5),L7$(5),L8$(5),L9$(5),L10$(5)
80 FOR I=1 TO 6:READ X:J(I)=X:NEXT I
90 FOR L=1 TO 10:FOR I=1 TO 6:READ X
95 B(L,I)=X:NEXT I:NEXT L
100 L1$="WELT":L2$="IND."
101 L3$="ENT.":L4$="CHINA"
102 L5$="INDIEN":L6$="JAPAN"
103 L7$="USA":L8$="UDSSR":L9$="BRD"
104 L10$="DDR"
110 GOSUB 1000:REM WARTEN
140 PRINT "JAHR",L1$,L2$,L3$:PRINT
150 GOSUB 1200:REM STRICH
170 FOR I=1 TO 6
180 PRINT J(I),B(1,I),B(2,I),B(3,I)
190 NEXT I
195 GOSUB 1200:REM STRICH
200 FOR L=1 TO 3
210 X=B(L,1):Y=B(L,6)
220 GOSUB 1100:REM WACHSTUMSRATE
230 NEXT L
240 PRINT "RATE %",WR(1),WR(2),WR(3):PRINT
250 GOSUB 1000:REM WARTEN
260 PRINT CHR$(125)
270 PRINT "JAHR";"  ";L4$;"  ";L5$;"  ";
272 PRINT L6$;"  ";L7$;"  ";L8$
275 PRINT
280 GOSUB 1200:REM STRICH
290 FOR I=1 TO 6
300 PRINT J(I);"     ";B(4,I);"    ";B(5,I);
302 PRINT"   ";B(6,I);"    ";B(7,I);
304 PRINT"   ";B(8,I)
310 NEXT I
320 GOSUB 1200:REM STRICH
330 FOR L=4 TO 8
340 X=B(L,1):Y=B(L,6)
350 GOSUB 1100:REM WACHSTUMSRATE
360 NEXT L
370 PRINT "%";
```

```
380 PRINT "       ";WR(4);"   ";WR(5);
382 PRINT"   ";WR(6);"   ";WR(7);
384 PRINT"   ";WR(8)
385 PRINT :PRINT
390 GOSUB 1000:REM WARTEN
400 PRINT CHR$(125)
410 PRINT "JAHR",L9$,L10$
420 GOSUB 1200:REM STRICH
430 FOR I=1 TO 6
440 PRINT J(I),B(9,I),B(10,I):PRINT
450 NEXT I
460 GOSUB 1200:REM STRICH
470 FOR L=9 TO 10
480 X=B(L,1):Y=B(L,6)
490 GOSUB 1100:REM WACHSTUMSRATE
492 NEXT L
494 PRINT "%",WR(9),WR(10)
496 PRINT :PRINT "ENDE":END
500 DATA 1975,1980,1985,1990,1995,2000
505 REM WELT
510 DATA 4134,4549,5013,5545,6143,6798
515 REM INDUSTRIELAENDER
520 DATA 1131,1174,1224,1276,1327,1377
525 REM ENTWICKLUNGSLAENDER
530 DATA 3003,3375,3789,4269,4816,5420
535 REM VR CHINA
540 DATA 978,1071,1151,1241,1348,1468
545 REM INDIEN
550 DATA 618,694,786,894,1013,1142
555 REM JAPAN
560 DATA 112,117,122,127,131,135
565 REM USA
570 DATA 214,222,235,248,260,270
575 REM UDSSR
580 DATA 254,268,282,296,310,323
585 REM BRD
590 DATA 61.8,61.7,60,58.6,57.8,56.2
595 REM DDR
600 DATA 16.8,16.7,16.6,16.4,16.2,16.1
1000 REM UP WARTEN
1010 PRINT :PRINT :PRINT
1020 PRINT "BITTE CONT EINGEBEN !"
1030 STOP
1040 RETURN
1100 REM UP WACHSTUMSRATE
1110 R=EXP(LOG(Y/X)/25)-1
1120 R=R*100:R=INT(R*100+0.5)/100
1130 WR(L)=R
1140 RETURN
1200 REM UP STRICH
1210 FOR I=1 TO 38:PRINT "-";:NEXT I
1220 PRINT :RETURN
```

5. SCHRITT : Variablenliste

B = Bevölkerung

I = Laufindex

J = Jahr

L = Laufindex

L$ = Länder

WR = Wachstumsrate

X = Anfangswert (1975)

Y = Endwert (2000)

6. SCHRITT : Programmbeschreibung

Satz 10-60 : Überschrift.

Satz 70-104 : Einlesen aller Ausgangsdaten.

Satz 110 : Sprung ins UP 1000: Warten.

Satz 140 : Tabellenüberschrift.

Satz 150 : Sprung ins UP 1200: Unterstreichung.

Satz 170-190 : Ausgabe der Tabellenwerte.

Satz 195 : Wie Satz 150.

Satz 200-230 : Bestimmung jahresdurchschnittlicher Zuwachsraten für die Länder L= 1, L=2 und L= 3.

Satz 210 : Übernahme des Anfangs- und des Endwertes.

Satz 220 : Sprung ins UP 1000 : Berechnung der Wachstumsrate.

Satz 240 : Ausgabe der Wachstumsraten.

Satz 250 : Warten.

Satz 260-390 : Gleiche Vorgehensweise wie in Satz 140-250 beschrieben, nun aber für die zweite Tabelle.

Satz 400-494 : Entsprechend für die dritte Tabelle.

Satz 496 : Beendigung des Programms.

Satz 500-600 : Daten.

Satz 1000-1040: Unterprogramm Warten.

Satz 1100-1140: Unterprogramm jahresdurchschnittliche Wachstumsrate (sie wird mit Hilfe der Logarithmenrechnung bestimmt, siehe Satz 1110).

Satz 1200-1220: Unterprogramm Strich:

Durch Aneinanderfügung von Minuszeichen wird auf dem Bildschirm ein Strich gezeichnet.

7. SCHRITT : Ergebnisse

Dieses Programm erzeugt drei Tabellen nacheinander auf dem Bildschirm:

1. Tabelle (Angaben in Millionen)

JAHR	WELT	IND.	ENT.
1975	4134	1131	3ØØ3
198Ø	4549	1174	3375
1985	5Ø13	1224	3789
199Ø	5545	1276	4269
1995	6143	1327	4816
2ØØØ	6798	1377	542Ø
RATE %	2.Ø1	Ø.79	2.39

2. Tabelle (Angaben in Millionen)

JAHR	CHINA	INDIEN	JAPAN	USA	UDSSR
1975	978	618	112	214	254
198Ø	1Ø71	694	117	222	268
1985	1151	786	122	235	282
199Ø	1241	894	127	248	296
1995	1348	1Ø13	131	26Ø	31Ø
2ØØØ	1468	1142	135	27Ø	323
%	1.64	2.49	Ø.75	Ø.93	Ø.97

3. Tabelle (Angaben in Millionen)

JAHR	BRD	DDR
1975	61.8	16.8
198Ø	61.7	16.7
1985	6Ø	16.6
199Ø	58.6	16.4
1995	57.8	16.2
2ØØØ	56.2	16.1
%	-Ø.38	-Ø.17

ENDE

Kapitel 8: Die Wirtschaft

8.1 Vorbemerkung

Das wesentliche Einsatzgebiet der Rechner war lange Zeit der wirtschaftliche und speziell der betriebliche und der kaufmännische Bereich. Man kann sagen, daß auch heute noch in diesen Bereichen Computer mit am häufigsten eingesetzt werden.

Die Problemstellungen, um die es nämlich in diesen Bereichen geht, eignen sich in der Regel sehr gut dazu, durch Rechner, heute auch durch Mikrocomputer, gelöst zu werden.

Einige der grundlegenden wirtschaftlichen Probleme, zumindest so weit Rechenprobleme betroffen sind, werden in der Regel auch im schulischen Unterricht behandelt; beispielsweise in Handelsschulen natürlich in starkem Maße; aber auch in anderen Schultypen verschließt man nicht die Augen vor derartigen Fragestellungen, wie sie in diesem Kapitel behandelt werden sollen.

Auch hier gilt aber, daß die ausgewählten Probleme nur stellvertretend stehen für viele andere, die allein schon aus Platzgründen nicht aufgegriffen werden.

8.2 Die Zinsrechnung

1. SCHRITT : Vorstellung des Problems

Man stelle sich vor, jemand bringt einen bestimmten Kapitalbetrag zu einer Bank, die dafür eine jährliche Zinszahlung vorsieht, deren Höhe alternativ festsetzbar ist.

Nach jedem Jahr vermehrt sich also der Kapitalbestand um einen bestimmten Prozentsatz, wobei zu beachten ist, daß ab dem zweiten Jahr auch die schon angefallenen Zinsen wiederum verzinst werden (Zinseszinsrechnung).

Es soll nun ein Programm entwickelt werden, welches für einen beliebigen Ausgangsbetrag und für einen beliebigen prozentualen Zinssatz ausrechnet, wie hoch der Endbetrag nach einer beliebigen Anzahl von Jahren sein wird.

Ein derartiges Programm kann übrigens als Muster für die Simulation unterschiedlicher Wachstumsvorgänge benutzt werden.

2. SCHRITT : Problemanalyse

Nennen wir den Ausgangsbetrag KØ, den Zinssatz P (%) und die Dauer der Bankeinlage T (Jahre).

Dann gilt nach dem ersten Jahr, daß sich der Betrag X1 ergibt zu:

$$X1 = XØ + XØ * P/1ØØ = XØ * (1 + P/1ØØ)$$

Im zweiten Jahr ergibt sich entsprechend zum Jahresende:

$$X2 = X1 + X1 * P/1ØØ = X1 * (1 + P/1ØØ)$$

$$= XØ * (1 + P/1ØØ) * (1 + P/1ØØ)$$

$$= XØ * (1 + P/1ØØ)^2$$

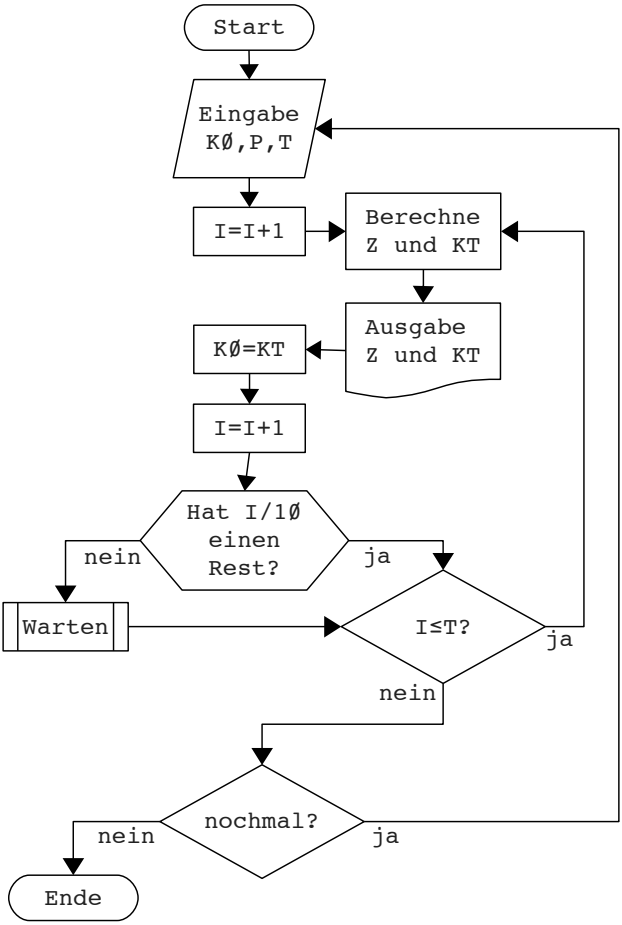
(Diesen Beziehungen sind wir übrigens schon in einem früheren Kapitel begegnet).

Nach T Jahren ergibt sich demnach allgemein:

$$XT = XØ * (1 + P/1ØØ)^T$$

Mit Hilfe dieser Beziehung kann nun ein geeignetes Programm aufgebaut werden.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM W1-ZINSRECHNUNG
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BERECHNUNG VON ZINSEN"
40 PRINT "BEI JAEHRLICHER ZINSZAHLUNG UND GEGE-"
50 PRINT "        BENEM AUSGANGSKAPITAL."
55 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
65 DIM A$(1)
70 PRINT :PRINT :PRINT :PRINT :PRINT
80 PRINT "AUSGANGSKAPITAL     : ";:INPUT K0
90 PRINT "JAEHRL.ZINSSATZ IN %: ";:INPUT P
100 PRINT "ANZAHL DER JAHRE    : ";:INPUT T
110 PRINT CHR$(125)
120 PRINT "JAHR","ZINS","KAPITAL"
125 PRINT
130 FOR I=1 TO T
150 Z=K0*P/100
160 KT=K0+Z
170 Z=INT(Z*100+0.5)/100:KT=INT(KT*100+0.5)/100
190 PRINT"  ";I,Z,KT
200 K0=KT
210 IF I/10=INT(I/10) THEN GOSUB 500:REM WRRTEN
220 NEXT I
225 PRINT :PRINT :PRINT
230 PRINT "NOCH EINE BERECHNUNG (J/N) ";
235 INPUT A$
240 IF A$="J" THEN PRINT CHR$(125):GOTO 80
250 PRINT :PRINT :PRINT "ENDE":END
500 REM UP WARTEN
510 POSITION 8,21
520 PRINT "BITTE CONT EINGEBEN !"
530 STOP
540 PRINT CHR$(125)
545 PRINT "JAHR","ZINS","KAPITAL"
546 PRINT
550 RETURN
```

5. SCHRITT : Variablenliste

A$ = Stringvariable für Antworten
I = Laufindex
KØ = Kapitalbestand am Anfang
KT = Kapitalbestand in den späteren Perioden
P = Jährlicher Zinssatz in Prozent
T = Dauer des Sparprozesses in Jahren
Z = Zinsbetrag pro Jahr

6. SCHRITT : Programmbeschreibung

Satz 1Ø-7Ø : Überschrift.

Satz 8Ø-1ØØ : Anforderung der Ausgangsdaten.

Satz 11Ø-12Ø : Räumen des Bildschirms und Ausgabe einer Tabellenüberschrift.

Satz 13Ø-22Ø : Berechnungen:

15Ø : Zinsbetrag.

16Ø : Kapitalbetrag in der nächsten Periode.

17Ø : Runden.

19Ø : Ausgabe.

2ØØ : Neubelegung von KØ mit KT, um in die nächste Periodenberechnung "einsteigen" zu können.

21Ø : Nach je 1Ø Perioden Programmunterbrechung durch Sprung ins UP 5ØØ.

Satz 23Ø-25Ø : Beendigung des Programms, es sei denn, es wird noch eine Berechnung gewünscht

(dann zurück zu Satz 8Ø).

Satz 5ØØ-55Ø : Unterprogramm zum Warten, wenn der Bildschirm voll ist:

51Ø-52Ø : Hinweis.

53Ø : Warten.

54Ø-546 : Löschen des Bildschirms und erneute Ausgabe der Tabellenüberschrift.

55Ø : Rücksprung.

7. SCHRITT : Ergebnisse

Geben wir als Anfangskapitalbestand DM 1ØØ ein, als Zinssatz 5.5 % und als Zeitdauer 6 Jahre, dann ergibt sich:

JAHR	ZINS	KAPITAL
1	5.5	1Ø5.5
2	5.8	111.3
3	6.12	117.42
4	6.46	123.88
5	6.81	13Ø.69
6	7.19	137.88

NOCH EINE BERECHNUNG (J/N) ?

8.3 Hypothekentilgung

1. SCHRITT : Vorstellung des Problems

Unter den wirtschaftlichen Fragestellungen spielen diejenigen eine wichtige Rolle, die mit der Rückzahlung von Krediten oder Hypotheken zu tun haben. Geeignete Computerprogramme können die Rückzahlungsdauer unter unterschiedlichen Ausgangsbedingungen, wie z.B. Höhe der Verbindlichkeit, Zinshöhe und Rückzahlungsmodalitäten bestimmen.

Es soll im folgenden ein Programm vorgestellt werden, welches für alternativ eingebbare Beträge die Tilgungsdauer berechnet, wenn alternative Zinssätze vorgegeben werden, bzw. wenn unterschiedlich hohe Rückzahlungsraten vereinbart werden.

Es ist leicht einzusehen, daß auch dieses Programm dann für generellere Fragestellungen dieser Art eingesetzt werden kann.

2. SCHRITT : Problemanalyse

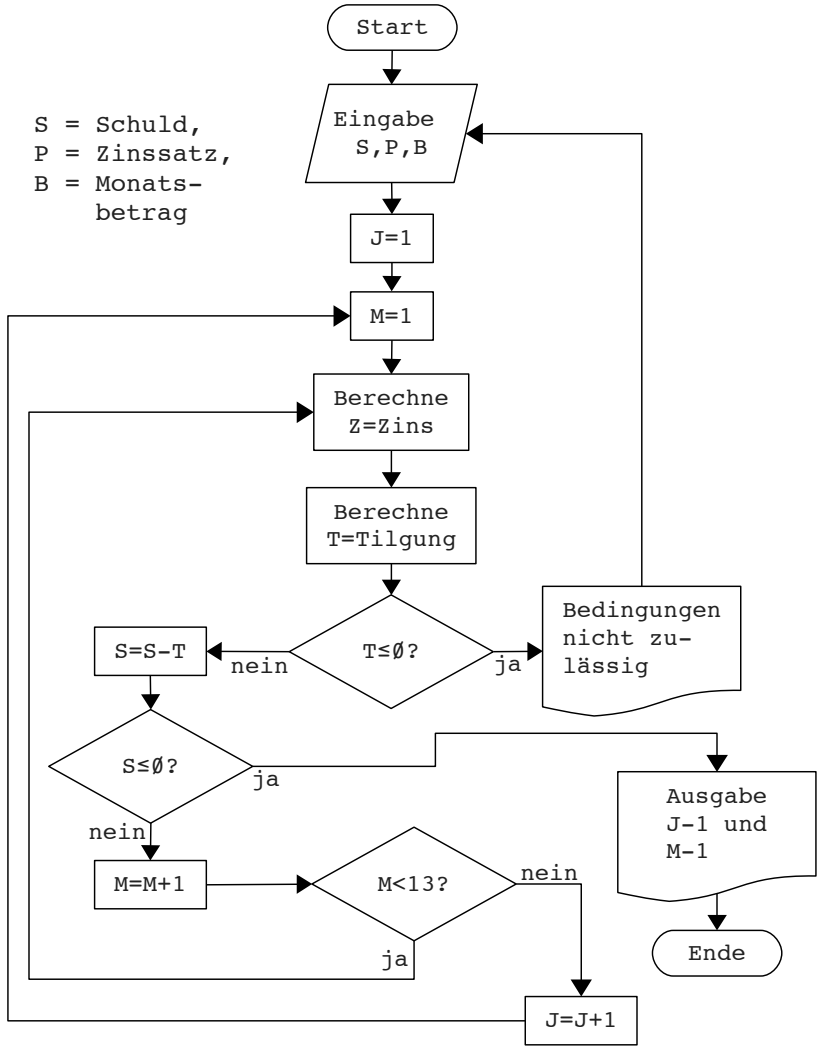
Zur Erläuterung der Problemlösung gehen wir von folgendem Beispiel aus:

Jemand nimmt eine Hypothek in der Höhe von S DM auf und vereinbart mit der hypothekgewährenden Bank eine monatliche Rückzahlung von B DM. Von diesem Monatsbetrag B müssen zunächst die Schuldzinsen beglichen werden. Der Betrag, der nach Abzug der Zinsen übrig bleibt, kann zur Tilgung der Schuld, d.h. zu ihrer Verringerung verwendet werden - vorausgesetzt, es bleibt überhaupt ein positiver Rest übrig.

Das zu entwickelnde Programm muß also Monat für Monat die anfallenden Zinsen berechnen und den Restbetrag, mit dem getilgt werden kann. Sollte dieser Restbetrag null oder negativ werden, so muß das Programm eine Meldung ausgeben, aus der erkenntlich wird, daß unter den gegebenen Bedingungen eine Verringerung der Schuld nicht möglich ist.

Jahr für Jahr soll der aktuelle Schuldenstand ausgegeben werden unä wenn der Schuldenstand null oder negativ geworden sein sollte, ist auszugeben, wie lange dieser Schuldentilgungsprozeß gedauert hat.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM W2-HYPOTHEK
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BERECHNUNG DER DAUER"
40 PRINT "       EINER HYPOTHEKENTILGUNG."
45 PRINT :PRINT :PRINT
50 PRINT "          PROF.DR.W.VOSS, 1984"
55 DIM A$(1)
60 PRINT :PRINT :PRINT :PRINT
70 PRINT "MONATLICHE RATE     : ";:INPUT B
80 PRINT "ZINSSATZ IN %       : ";:INPUT P
90 PRINT "AKTUELLE HYP.SCHULD : ";:INPUT S
100 PRINT CHR$(125)
110 PRINT "JAHR","SCHULD":PRINT
120 FOR I=1 TO 25:PRINT "-";:NEXT I:PRINT
130 J=1
140 M=1
150 Z=((P/100)*S)/12:T=B-Z
160 IF T<=0 THEN 250
170 S=INT((S-T)*100+0.5)/100
180 IF S<=0 THEN 300
190 M=M+1:IF M<13 THEN 150
200 PRINT " ";J,S
210 J=J+1:GOTO 140
250 PRINT :PRINT :PRINT
255 PRINT "DIE KONDITIONEN SIND DERART, DASS"
260 PRINT "NICHTS GETILGT WERDEN KANN."
270 PRINT :PRINT "BITTE NEUEINGABE"
280 PRINT :PRINT "BITTE CONT EINGEBEN":STOP
290 PRINT CHR$(125):GOTO 70
300 PRINT :PRINT :PRINT
305 PRINT "TILGUNGSDAUER : ":PRINT
310 PRINT J-1;" JAHRE UND ";M-1;" MONATE."
320 PRINT :PRINT :PRINT
325 PRINT "NOCH EINE BERECHNUNG (J/N) ";
330 INPUT A$
335 IF A$="J" THEN PRINT CHR$(125):GOTO 70
340 PRINT :PRINT "ENDE DER BERECHNUNGEN":END
```

5. SCHRITT : Variablenliste

A$ = Stringvariable (Ja, Nein)
B = Monatlicher Rückzahlungsbetrag
I = Laufindex
J = Jahreszähler
M = Monatszähler
P = Jährlicher Schuldzinssatz
T = Monatlicher Tilgungsbetrag
Z = Monatlicher Zinsbetrag

6. SCHRITT : Programmbeschreibung

Satz 10-60 : Überschrift.

Satz 70-90 : Vorgabe der Ausgangsinformationen.

Satz 100-120 : Ausgabe einer Tabellenüberschrift.

Satz 130-140 : Vorgabe eines Jahreszählers (J) und eines Monatszählers M.

Satz 150-170 : Berechnung von monatlichem Zins, monatlicher Tilgung und neuer Schuld.

Satz 160 : Wenn nicht getilgt werden kann, dann weiter bei 250.

Satz 180 : Ist die Schuld getilgt , dann weiter bei 300.

Satz 19Ø : Nächster Monat:
Falls M kleiner als 13 ist, weiter bei Satz 15Ø, andernfalls bei Satz 2ØØ.

Satz 2ØØ : Ausgabe der aktuellen Schuld.

Satz 21Ø : Nächstes Jahr und weiter bei Satz 14Ø.

Satz 25Ø-29Ø : Meldung bei unzulässigen Tilgungskonditionen und zurück zu Satz 7Ø.

Satz 3ØØ-31Ø : Ausgabe der Tilgungsdauer in Jahren und Monaten (der Leser überlege, warum dabei von J und von M jeweils der Wert 1 abgezogen werden muß).

Satz 32Ø-4ØØ : Beendigung des Programms, es sei denn es wird eine erneute Berechnung gewünscht (dann nach Löschen des Bildschirms zurück zum Satz 7Ø).

7. SCHRITT : Ergebnisse

Geben wir bei den Anforderungen des Rechners beispielsweise ein:

```
MONATLICHE RATE     : ? 5ØØ
ZINSSATZ IN %       : ? 6.5
AKTUELLE HYP.SCHULD : ? 4ØØØØ
```

so erhalten wir während des Programmablaufs:

```
JAHR      SCHULD
-------------------------
1        36496.86
2        32759.Ø9
3        28771.Ø2
4        24515.85
5        19975.7
6        15131.48
7        9962.85
8        4448.Ø7

TILGUNGSDAUER :

8 JAHRE UND 9 MONATE.


NOCH EINE BERECHNUNG (J/N) ?
```

8.4 Arithmetisches Mittel

1. SCHRITT : Vorstellung des Problems

Bei vielen wirtschaftlichen Fragestellungen ist es erforderlich, aus einer gegebenen Anzahl von Daten einen Durchschnitt auszurechnen, wobei üblicherweise das arithmetische Mittel benutzt wird.

Das folgende Programm soll für eine beliebige Anzahl von Daten nach Aufnahme dieser Werte diese Rechenprozedur durchführen.

2. SCHRITT : Problemanalyse

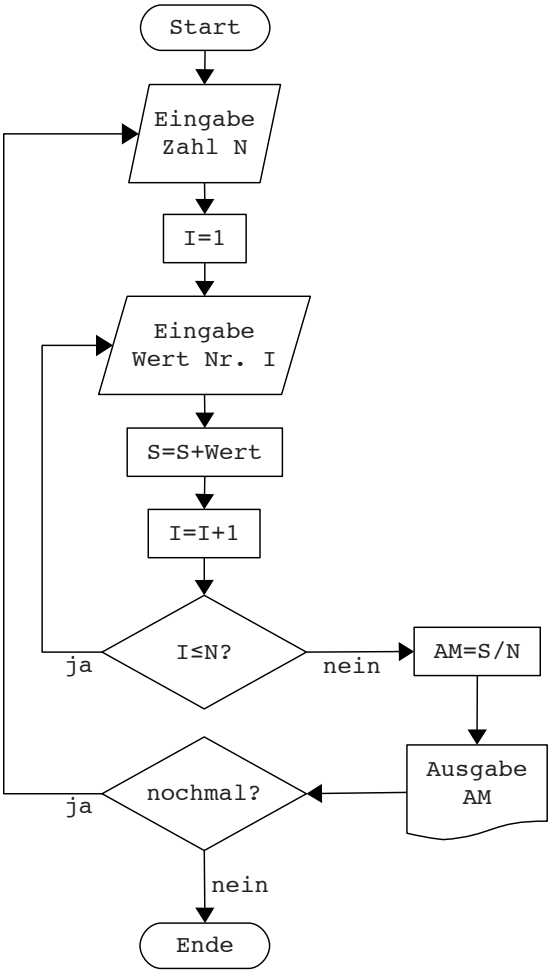
Bei der Problemanalyse treten keine besonderen Schwierigkeiten auf.

Es muß zunächst dafür gesorgt werden, daß eine offene Anzahl von Daten eingegeben werden kann, weshalb dem Rechner als erstes mitgeteilt werüen muß, wieviele Werte es überhaupt sein werden.

Bei der Eingabe der Werte können diese gleich aufaddiert werden (das arithmetische Mittel ist ja die Summe aller Werte geteilt durch ihre Anzahl). Die Division der entstehenden Summe durch die Anzahl der Werte kann dann als Ergebnis des Programms ausgegeben werden.

Soll eine weitere Berechnung durchgeführt werden, verzweigt das Programm wieder an den Anfang, andernfalls ist es beendet.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM W3-MITTELWERT
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BERECHNUNG EINES ARITH-"
40 PRINT "          METHISCHEN MITTELS."
42 DIM A$(1)
45 PRINT :PRINT :PRINT
50 PRINT "         PROF.DR.W.VOSS, 1984"
60 PRINT :PRINT :PRINT :PRINT :PRINT
70 PRINT "WIEVIELE WERTE : ";:INPUT N
80 PRINT CHR$(125):S=0
90 FOR I=1 TO N
100 PRINT I;".WERT : ";:INPUT X
110 S=S+X
120 NEXT I
130 AM=S/N
140 PRINT :PRINT :PRINT
150 PRINT "MITTELWERT = ";AM:PRINT :PRINT
160 PRINT "NOCHMAL (J/N) ";:INPUT A$
170 IF A$="J" THEN PRINT CHR$(125):GOTO 70
180 PRINT :PRINT :PRINT "ENDE":END
```

5. SCHRITT : Variablenliste

AM = Arithmetisches Mittel
A$ = Stringvariable (Ja, Nein)
I = Laufindex
N = Anzahl der zu mittelnden Werte
S = Summe der zu mittelnden Werte
X = Wert, der in die Mittelwertberechnung eingeht

6. SCHRITT : Programmbeschreibung

Satz 1Ø-9Ø : Überschrift.

Satz 11Ø : Warten.

Satz 12Ø-13Ø : Dimensionieren und Einlesen der Ausgangsdaten.

Satz 13Ø-22Ø : Berechnungen:

15Ø :
16Ø :

7. SCHRITT : Ergebnisse

Geben wir beispielsweise an, daß 3 Werte vorliegen, so fordert das Programm genau drei Werte an.

Sind dies z.B. die Werte 5, 7 und 9, so erhalten wir:

MITTELWERT = 7

NOCHMAL (J/N) ?

8.5 Häufigkeitsverteilung

1. SCHRITT : Vorstellung des Problems

Bei der Auswertung größerer Datenbestände, für die sich ja der Einsatz von Computern ganz besonders gut eignet, besteht ein erster statistischer Auswertungsschritt häufig darin, eine sog. Häufigkeitsverteilung zu erstellen.

Es geht dabei darum, den Merkmalswerten (oder Merkmalswertklassen) einer Untersuchungsvariablen die Häufigkeit ihres Auftretens graphisch oder tabellarisch zuzuordnen.

Das folgende Programm wird sowohl eine graphische wie auch eine tabellarische Häufigkeitsverteilung erstellen, wobei uns als Datenmaterial die Altersverteilung in der Bundesrepublik Deutschland im Jahre 1980 dient.

Diese Altersverteilung der westdeutschen Bevölkerung stellt sich folgendermaßen dar:

Altersverteilung in der Bundesrepublik Deutschland 198Ø:

Altersklasse	Anteil in %
unter 1Ø	1Ø.3
1Ø bis unter 2Ø	16.4
2Ø bis unter 3Ø	14.6
3Ø bis unter 4Ø	13.4
4Ø bis unter 5Ø	14.2
5Ø bis unter 6Ø	11.9
6Ø bis unter 7Ø	8.9
7Ø bis unter 8Ø	7.8
8Ø bis unter 9Ø	2.4
9Ø und mehr	Ø.2

2. SCHRITT : Problemanalyse

Bei der Problemanalyse gehen wir davon aus, daß uns die Daten wie in obiger Tabelle schon vorliegen, so daß diese Werte als Ausgangsmaterial dem zu entwickelnden Programm eingegeben werden können.

Sollte dies nicht der Fall sein, hätte man also beispielsweise individuelle Altersangaben in einer Fragebogenaktion o.ä. erhoben, so müßte in einem ersten Programmteil erst dafür gesorgt werden, daß die individuellen Angaben in die einzelnen, zunächst noch leeren, Altersklassen einsortiert werden (dies kann mit einer Folge von IF ..· THEN- Statements geleistet werden, bei der jeder einzelne Alterswert überprüft wird).

Ist dies geschehen, dann könnten die Klassenbesetzungszahlen, die sich dann ergeben haben, durch Beziehung auf die Gesamtzahl der Angaben relativiert werden, so daß man zu Prozentanteilen gelangt, wie sie in obiger Tabelle schon vorgegeben sind.

Dieser Programmteil könnte also schematisch folgendermaßen aussehen:

```
100 INPUT "Nächstes Alter : ";X
110 IF X < 10 THEN Z1 = Z1 + 1 : GOTO 100
120 IF X < 20 THEN Z2 = Z2 + 1 : GOTO 100
...
usw.

200 R1=(Z1/N)*100
210 R2=(Z2/N)*100
...
usw.
```

Hier in diesem Beispiel geht es aber lediglich darum, die schon vorgegebenen Werte zunächst tabellarisch darzustellen, was natürlich überhaupt keine programmtechnischen Schwierigkeiten verursacht.

Etwas komplizierter ist hingegen die graphische Darstellung dieser Häufigkeitsverteilung.

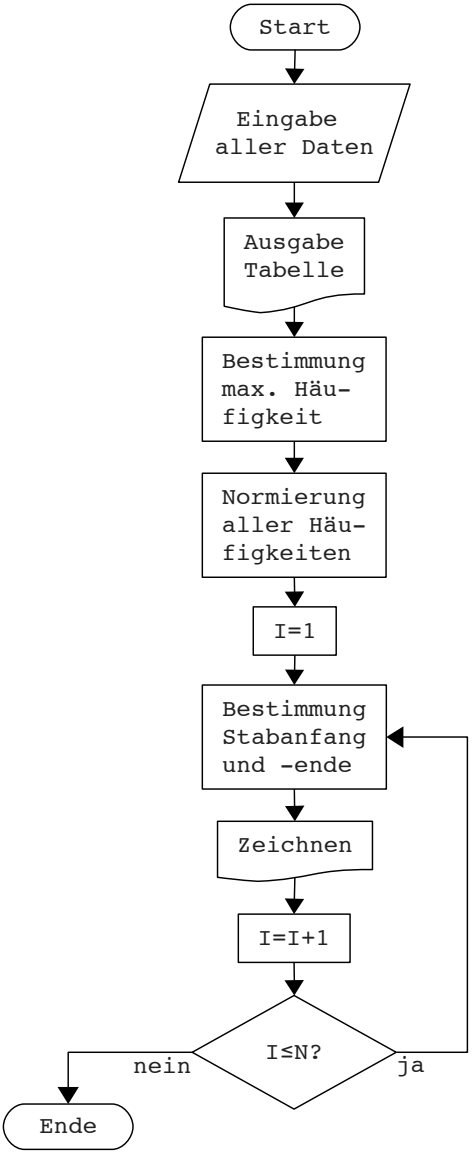
Um den Bildschirm für die graphische Darstellung optimal ausnutzen zu können, bestimmen wir zunächst aus dem vorliegenden Datenbestand die größte Häufigkeit und

rechnen alle anderen Häufigkeiten dann so um, daß die größte Häufigkeit in der graphischen Darstellung sich in der Vertikalen über 15 Bildschirmzeilen erstreckt.

Die Häufigkeiten selbst stellen wir als senkrechte Stäbe dar. Um diese Stäbe maßstabgerecht zeichnen zu lassen, muß ihr jeweiliger Endpunkt bestimmt werden.

Zusammen mit dem Anfangspunkt erhalten wir dann jeweils zwei Bildschirmpositionen und es geht dann nur noch darum, alle Positionen zwischen diesen beiden (unter Einschluß derselben) zu belegen.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM W4-VERTEILUNG
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR TABELLARISCHEN UND GRA-"
40 PRINT "PHISCHEN DARSTELLUNG EINER STATISTI-"
50 PRINT "    SCHEN HAEUFIGKEITSVERTEILUNG."
55 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT :PRINT :PRINT
80 PRINT "DARGESTELLT WIRD DIE ALTERSVERTEILUNG"
90 PRINT "IN DER BUNDESREPUBLIK DEUTSCHLAND 1980"
100 GOSUB 1000:REM WARTEN
120 DIM F(10),FT(10)
130 FOR I=1 TO 10:READ X:F(I)=X:NEXT I
140 PRINT "ALTER",,"    ";"ANTEIL":PRINT
150 FOR I=1 TO 35:PRINT "-";:NEXT I:PRINT
160 FOR I=1 TO 10
165 A=(I-1)*10:B=I*10
170 PRINT A;" BIS UNTER ";B;
176 PRINT ,F(I)
180 NEXT I
190 GOSUB 1000:REM WARTEN
220 REM NORMIERUNG
230 FM=0
240 FOR I=1 TO 10:IF F(I)>FM THEN FM=F(I)
250 NEXT I
260 FOR I=1 TO 10
270 FT(I)=F(I)*(15/FM)
280 NEXT I
290 REM ZEICHNEN
300 B=21
310 FOR I=1 TO 10
320 A=21-FT(I):A=INT(A+0.5)
330 FOR Z=A TO B
340 POSITION 3*I,Z
350 PRINT CHR$(160)
360 NEXT Z
370 NEXT I
380 POSITION 2,22
390 FOR I=1 TO 30
400 PRINT CHR$(18);
410 NEXT I
415 PRINT
420 FOR I=1 TO 10:PRINT " ";I*10;:NEXT I
430 PRINT :PRINT "ENDE":END
500 DATA 10.3,16.4,14.6,13.4,14.2
510 DATA 11.9,8.9,7.8,2.4,.2
1000 REM UP WARTEN
1010 PRINT :PRINT :PRINT
1020 PRINT "BITTE CONT EINGEBEN !"
1030 STOP
1040 PRINT CHR$(125):RETURN
```

5. SCHRITT : Variablenliste

A = Anfang der Altersklasse
später : Anfangsadresse des zu zeichnenden Stabes
B = Ende der Altersklasse
F = Häufigkeit
FM = maximale Häufigkeit
FT = transformierte Häufigkeit
I = Laufindex
Z = Laufindex (Bildschirmzeile)

6. SCHRITT : Programmbeschreibung

Satz 1Ø-9Ø : Überschrift.

Satz 11Ø : Warten.

Satz 12Ø-13Ø : Dimensionieren und Einlesen der Ausgangsdaten.

Satz 14Ø-15Ø : Ausgabe der Tabellenüberschrift.

Satz 16Ø-18Ø : Tabellenausgabe:

165 : Bestimmung von Klassenanfangs- und Klassenendpunkt.

17Ø-176 : Ausgabe.

Satz 19Ø : Warten.

Satz 22Ø-25Ø : Aufsuchen der größten Häufigkeit durch Vergleich aller Häufigkeiten mit der jeweils schon erreichten größten Häufigkeit (beim Wert null beginnend).

Satz 26Ø-28Ø : Bestimmung transformierter Häufigkeit durch Ausrichten der Häufigkeiten an der Maximalhäufigkeit.

Satz 29Ø-37Ø : Zeichnen von 1Ø Stäben:

3ØØ : Stabende.

32Ø : Bestimmung des Stabanfangs gemessen an der Bildschirmzeileneinteilung.

33Ø-36Ø : Zeichnen des Stabes durch Zusammenfügen von kleinen Quadraten in Spalte 3*I.

Satz 38Ø-41Ø : Waagrechte Achse.

Satz 415-42Ø : Beschriftung der waagrechten Achse.

Satz 43Ø : Ende des Programms.

Satz 5ØØ-51Ø : Daten.

Satz 1ØØØ-1Ø4Ø: Unterprogramm Warten.

7. SCHRITT : Ergebnisse

Die Ergebnisse dieses Programms sollen hier nicht vorgestellt werden. Viel sinnvoller ist es, wenn der Leser das Programm in seinen Rechner eingibt, um dann im Vergleich der Bildschirmausgabe mit dem vorgestellten Programm nachzuvollziehen, was im einzelnen geschieht.

Allerdings ist in diesem Zusammenhang die folgende Anmerkung sinnvoll:

Die Zeichnung, die in diesem Programm erstellt wird, bedient sich nicht derjenigen BASIC-Statements, die für hochauflösende Graphiken zur Verfügung stehen, sondern sie werden mit Hilfe von PRINT-Statements unter Benutzung der CHR$-Funktion erzeugt.

Dies hat zur Folge, daß die Zeichnung recht "grob" wird. Die Stäbe im Stabdiagramm entsprechen nicht exakt den darzustellenden Häufigkeiten, weil der kleinste "Stab-Baustein" ja ein Quadrat von der Größe des Cursors, also mit rund 7 mm Kantenlänge (auf einem normalen Computermonitor) ist.

Will man exaktere Graphiken, so gelingt dies nur mit der hochauflösenden Graphik, wie sie im folgenden Kapitel benutzt wird.

8.6 Die Wirtschaftskraft der Bundesländer

1. SCHRITT : Vorstellung des Problems

Von ganz anderer Art wie die vorangegangenen Programmbeispiele ist das folgende Programm. Hier geht es nämlich wieder, wie schon in manchen Kapiteln zuvor, darum, ein "Informationsprogramm" zu erstellen.

Es soll ein Programm entwickelt werden, welches für die Länder der Bundesrepublik Deutschland wichtige Wirtschaftsdaten auf Anforderung durch den Benutzer ausgibt.

Die Daten, die für diese Informationsbereitstellung vorgegeben werden, sind die folgenden:

1. Bevölkerungszahl (in Millionen)
2. Fläche (in 1000 Quadratkilometer)
3. Bruttosozialprodukt (in Mrd. DM)
4. Anzahl der Erwerbstätigen (in Millionen)
5. Anteil der ausländischen Erwerbstätigen (in %)
6. Steueraufkommen (in Mrd. DM)
7. Schulden (in Mrd. DM)
8. Anteil der industriellen Produktion (in %)

Es versteht sich von selbst, daß bei Bedarf oder Interesse diese Liste verlängert werden könnte, oder daß an-

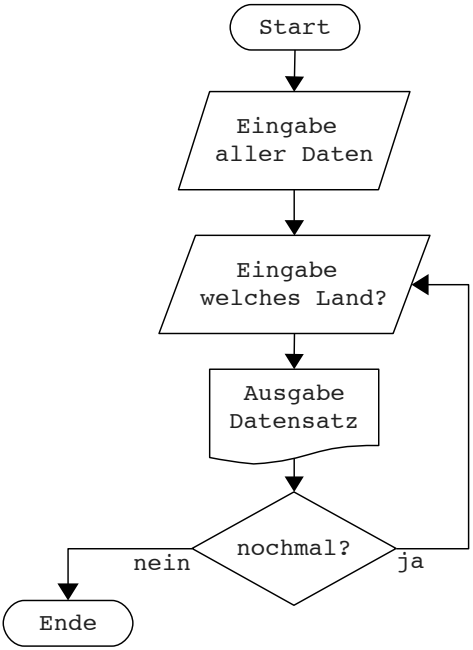
stelle der Bundesländer auch andere Sachverhalte (z.B. Nationen) betrachtet werden könnten. Natürlich müßte dann aber der Datenbestand entsprechend geändert werden.

2. SCHRITT : Problemanalyse

Der Leser wird schon festgestellt haben, daß sich über die verschiedenen Kapitel hinweg bestimmte Problemtypen wiederholen oder in ähnlicher Form wiederkehren. So ist dies auch bei dieser Problemstellung : Sie bietet im Prinzip keine neuen Probleme, so daß die Analyse recht einfach ist:

Es müssen zunächst alle relevanten Informationen eingegeben werden. Der Benutzer wird dann aufgefordert, anzugeben, über welches Land er Informationen wünscht. Nach der daraufhin erforderlichen Eingabe, erfolgt die Ausgabe des Datensatzes für das gewünschte Bundesland.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM W5-LAENDER
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR AUSGABE VON INFORMATIO-"
40 PRINT "NEN UEBER DIE WIRTSCHAFTSKRAFT DER"
50 PRINT "            BUNDESLAENDER."
55 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT :PRINT :PRINT
110 DIM L$(26),A$(1),V(8,12)
115 DIM B1$(26),B2$(26),B3$(26),B4$(26)
116 DIM B5$(26),B6$(26),B7$(26),B8$(26)
120 GOSUB 1000:REM WARTEN
125 PRINT "FUER WELCHES LAND WERDEN INFORMATIO-"
130 PRINT "NEN GEWUENSCHT ?":PRINT
140 FOR I=1 TO 12:READ L$
150 PRINT I,L$:NEXT I
160 PRINT :PRINT :PRINT
170 PRINT "BITTE NUMMER ANGEBEN: ";:INPUT Z
180 PRINT CHR$(125)
190 PRINT "          MOMENT BITTE":PRINT
200 FOR J=1 TO 8:FOR I=1 TO 12
210 READ X:V(J,I)=X:NEXT I:NEXT J
220 RESTORE
230 FOR I=1 TO Z:READ L$:NEXT I
250 B1$="BEVOELKERUNG (MILL.)       "
251 B2$="FLAECHE (1000 QKM)         "
252 B3$="SOZIALPRODUKT (MRD.DM)     "
253 B4$="ERWER8STAETIGE (MILL.)     "
254 B5$="AUSLAENDERANTEIL (%)       "
255 B6$="STEUERAUFKOMMEN (MRD.DM)   "
255 B7$="SCHULDEN (MRD.DM)          "
257 B8$="ANTEIL IND.PRODUKTION (%)  "
280 PRINT L$:PRINT :PRINT
290 PRINT B1$,V(1,Z)
291 PRINT B2$,V(2,Z)
292 PRINT B3$,V(3,Z)
293 PRINT B4$,V(4,Z)
294 PRINT B5$,V(5,Z)
295 PRINT B6$,V(6,Z)
295 PRINT B7$,V(7,Z)
297 PRINT B8$,V(8,Z)
320 GOSUB 1000:REM WARTEN
330 PRINT "NOCH EIN ANDERES LAND (J/N) ";
340 INPUT A$
350 IF A$="J" THEN PRINT CHR$(125) :RESTORE :GOTO 125
360 PRINT :PRINT "ENDE":END
```

```
500 DATA SCHLESWIG-HOLSTEIN,NIEDERSACHSEN
502 DATA NORDRHEIN-WESTFALEN,HESSEN
504 DATA RHEINLAND-PFALZ,SAARLAND
506 DATA BADEN-WUERTTEMBERG,BAYERN
508 DATA HAMBURG,BREMEN,BERLIN
510 DATA BUNDESREPUBLIK DEUTSCHLAND
600 DATA 0,0,0,0,0,0,0,0,0,0
602 DATA 0,0,0,0,0,0,0,0,0,0
604 DATA 0,0,0,0,0,0,0,0,0,0
606 DATA 0,0,0,0,0,0,0,0,0,0
608 DATA 0,0,0,0,0,0,0,0,0,0
610 DATA 0,0,0,0,0,0,0,0,0,0
612 DATA 0,0,0,0,0,0,0,0,0,0
614 DATA 0,0,0,0,0,0,0,0,0,0
616 DATA 0,0,0,0,0,0,0,0,0,0
618 DATA 0,0,0,0,0,0
1000 REM UP WARTEN
1010 PRINT :PRINT :PRINT
1020 PRINT "BITTE CONT EINGEBEN !"
1030 STOP
1040 PRINT CHR$(125):RETURN
```

5. SCHRITT : Variablenliste

A$ = Stringvariable
B$ = Variablen-Bezeichnungen
I = Laufindex (Länder)
J = Laufindex (Variablen)
L$ = Ländernamen
V = Variablen
X = Datenfeld
Z = Nummer des gewünschten Landes

6. SCHRITT : Programmbeschreibung

Satz 10-70 : Überschrift.

Satz 110-116 : Dimensionierungen.

Satz 120 : Warten.

Satz 125-170 : Anforderung der Nummer desjenigen Lan-

des, über das Informationen gewünscht werden, nachdem eine Länderliste auf dem Bildschirm vorgegeben wurde.

Satz 18Ø-21Ø : Löschen des Bildschirms und Einlesen der Daten.

Satz 22Ø : Restaurieren der Daten, damit der jeweils gewünschte Ländernamen wieder eingelesen und dann ausgegeben werden kann.

Satz 23Ø : Aufsuchen des gewünschten Landes.

Satz 25Ø-257 : Eingabe der Variablenbezeichnungen.

Satz 28Ø : Ausgabe des Ländernamens.

Satz 29Ø-297 : Ausgabe des Datensatzes.

Satz 32Ø : Warten.

Satz 33Ø-35Ø : Abfrage, ob noch eine weitere Ausgabe gewünscht wird (wenn ja, Löschen des Bildschirms und zurück zu Satz 125; wenn nein, Beendigung des Programms).

Satz 5ØØ-51Ø : Ländernamen in DATA-Statements.

Satz 6ØØ-618 : Variablenwerte:
Die Werte sind so anzugeben, daß für die erste Variable alle Länderwerte, dann für die zweite Variable alle Länderwerte usw. folgen.

ACHTUNG: IM PROGRAMM SIND NUR NULLEN (vergl. nächsten Arbeitschritt)!

Satz 1ØØØ-1Ø4Ø: Unterprogramm zum Warten.

7. SCHRITT : Ergebnisse

Im Datenbestand des vorgestellten Programms finden sich nur Nullen, so daß bei einem Probelauf tatsächlich auch nur Nullwerte auf dem Bildschirm erscheinen können.

Wenn ein "echter" Probelauf durchgeführt werden soll, dann müssen erst konkrete Daten eingegeben werden, die der Leser der folgenden Übersicht entnehmen kann:

Wirtschaftsdaten der Bundesländer (die Daten beziehen sich auf 1980-1982):

Land	Bevölk. (Mill.)	Fläche (1000 qkm)	Sozial-prod. (Mrd.)	Erwerbs-tätige (Mill.)
Schleswig-Holstein	2.6	16	53	1.1
Niedersachsen	7.3	48	152	3.2
Nordrhein-Westfalen	17.0	34	406	6.9
Hessen	5.6	21	140	2.4
Rheinland-Pfalz	3.6	20	81	1.6
Baden-Württemberg	9.3	36	234	4.3
Bayern	11.0	71	253	5.1
Saarland	1.1	3	24	.4
Hamburg	1.6	.7	75	.7
Bremen	0.7	.4	22	.3
Berlin	1.9	.5	53	.8
BUND	61.6	248	1484	26.8

Land	Ausländer-anteil	Steuer-aufkom-men (Mrd)	Schul-den (Mrd.)	Anteil In-dustrie-produktion
Schleswig-Holstein	2.6	10	8	34
Niedersachsen	3.3	31	18	41
Nordrhein-Westfalen	7.1	108	32	48
Hessen	7.6	34	13	45
Rheinland-Pfalz	3.6	16	9	44
Baden-Württem-berg	10.1	57	18	52
Bayern	5.8	54	14	45
Saarland	5.7	4	3	51
Hamburg	8.4	33	8	32
Bremen	6.4	6	6	34
Berlin	10.6	12	7	35
BUND	6.7	248	1484	26.8

Kapitel 9: Mathematik II

9.1 Vorbemerkung

Abschließend soll ein Kapitel angefügt werden, in dem weitere mathematische Probleme behandelt werden, und zwar solche, die einen etwas höheren Schwierigkeitsgrad aufweisen, als die bisher besprochenen Probleme. Damit soll der bereits fortgeschrittene Mikrocomputer-Benutzer erkennen, daß mit relativ einfachen Programmen auch schon komplexere Problemstellungen in Angriff genommen werden können.

Insoweit könnte dieses Schlußkapitel für den Anfänger eine leichte Uberforderung darstellen (aber das gerade übt ja auch manchmal einen beträchtlichen Reiz aus).

Diese etwas komplizierteren Probleme sollen so programmiert werden, daß keine zusätzlichen BASIC-Anweisungen benötigt werden. Das bedeutet, daß die entstehenden Programme (wie auch schon die vorhergehenden) nicht den Eleganzansprüchen geübter Programmierer entsprechen können. Darauf kommt es aber hier auch nicht an.

Im Gegensatz zu den früheren Kapiteln werden wir uns hier aber kürzer fassen, weil mehr als ein Dutzend zusätzlicher Probleme behandelt werden sollen.

Nach einer kurzen Beschreibung der Problemstellung und einigen Anmerkungen zur Problemanalyse stellen wir sofort das Programm vor und schließen eine knappe Programmbeschreibung an. Insbesondere auf die Arbeitsschritte "Flußdiagramm" und "Ergebnisse" wird also im folgenden verzichtet; nicht nur aus Gründen der Platzersparnis, sondern auch, weil sich unter rein programmlogischen Gesichtspunkten eigentlich keine neuen Detailprobleme mehr stellen.

9.2 Gitter

Problemstellung

Das folgende Programm dient dazu, auf dem Bildschirm ein Liniengitter mit beliebigen Zeilen und Spaltenabständen auszugeben.

Die Überlegungen, die diesem einfachen Programm zugrunde liegen, können so oder in ähnlicher Weise immer verwendet werden, wenn beispielsweise umfangreichere Tabellenausgaben durch Längs- und Querstriche gegliedert werden sollen.

Problemanalyse

Stichworte zur Problemanalyse sind hier fast entbehrlich. Vorzugeben sind vom Benutzer Angaben zum Zeilen und zum Spaltenabstand. Entsprechend dieser Angaben wird der Graphik-Cursor per PLOT-Anweisung positioniert, um dann per DRAWTO eine gerade Linie zu ziehen.

Programm

```
10 REM Z1-GITTER
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM ZEICHNEN EINES GITTERS."
35 PRINT :PRINT :PRINT
40 PRINT "         PROF.DR.W.VOSS, 1984"
45 FOR I=1 TO 8:PRINT :NEXT I
50 PRINT "SPALTENABSTAND (MAX.320): ";:INPUT S
60 PRINT "ZEILENABSTAND (MAX.160) : ";:INPUT Z
70 PRINT CHR$(125)
75 GRAPHICS 8
76 COLOR 1
80 FOR I=1 TO 320 STEP S
90 PLOT I,160:DRAWTO I,1
100 NEXT I
110 FOR I=0 TO 159 STEP Z
120 PLOT 1,I:DRAWTO 319,I
130 NEXT I
135 FOR I=1 TO 2000:NEXT I:GRAPHICS 0
140 POSITION 10,23:PRINT "ENDE":END
```

Variablenliste

I = Laufindex
S = Spaltenabstand
Z = Zeilenabstand

Programmbeschreibung

Satz 1Ø-45 : Überschrift.

Satz 5Ø-6Ø : Eingabe von Spalten- und Zeilenabstand.

Satz 7Ø : Löschen des Bildschirms.

Satz 75-76 : Umschalten auf Graphikmodus.

Satz 8Ø-1ØØ : Zeichnen der senkrechten Striche.

Satz 11Ø-13Ø : Zeichnen der waagrechten Striche.

Satz 135 : Warteschleife und Umschalten auf Textmodus.

Satz 14Ø : Beendigung des Programms.

9.3 Die Gerade

Problemstellung

Das folgende Programm dient dazu, beliebige Geraden in ein Achsenkreuz einzuzeichnen, dessen Ursprung in der Bildschirmmitte (16Ø,8Ø) liegt.

Problemanalyse

Die Lage einer Geraden wird von zwei Parametern bestimmt, dem Ordinatenabschnitt A und dem Tangens des Steigungswinkels B gemäß der folgenden Funktionalgleichung:

$$Y_i = A + B * X_i$$

Da der Koordinatenursprung im Punkt (16Ø,8Ø) liegen soll, ergibt sich:

$$Y_i = 8Ø + A + B * (X_i - 16Ø)$$

Allerdings müssen wir nun beachten, daß - im Gegensatz zum gewohnten Achsenkreuz - die niedrigen Y-Werte oben auf dem Bildschirm sind (Zeile Nr. Ø ist die oberste Bildschirmzeile).

Deshalb muß die Funktion folgendermaßen verändert werden:

$$Y_i = 16Ø - (8Ø + A + B * (x_i - 16Ø))$$

$$= 8Ø - A - B * (x_i - 16Ø)$$

Programm

```
10 REM Z2-GERADE
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM ZEICHNEN EINER GERADEN."
35 PRINT :PRINT :PRINT
40 PRINT "         PROF.DR.W.VOSS, 1984"
45 FOR I=1 TO 8:PRINT :NEXT I
50 PRINT "ORDINATENABSCHNITT A : ";:INPUT A
60 PRINT "STEIGUNG            B : ";:INPUT B
70 GRAPHICS 8
75 COLOR 1
80 PLOT 0,80:DRAWTO 319,80
90 PLOT 160,0:DRAWTO 160,159
100 FOR X=0 TO 319
110 Y=80-A-B*(X-160)
120 IF Y<=0 OR Y>=159 THEN 140
130 PLOT X,Y
140 NEXT X
150 FOR I=1 TO 2000:NEXT I
160 GRAPHICS 0
170 PRINT "ENDE":END
```

Variablenliste

A = Ordinatenabschnitt
B = Tangens des Steigungswinkels
I = Laufindex
X = Werte auf der X-Achse
Y = Funktionswerte

Programmbeschreibung

Satz 1Ø-45 : Überschrift.

Satz 5Ø-6Ø : Eingabe der Geradenparameter.

Satz 7Ø-75 : Graphikmodus.

Satz 8Ø-9Ø : Zeichnen des Achsenkreuzes.

Satz 1ØØ-14Ø : Zeichnen der Gerade über die gesamte Breite des Bildschirms (für unzulässige Y-Werte wird die Zeichenanweisung 13Ø übersprungen).

Satz 15Ø : Warteschleife.

Satz 16Ø : Textmodus.

Satz 17Ø : Beendigung des Programms.

9.4 Der Kreis

Problemstellung

Nach dem gleichen Muster wie im vorangegangenen Abschnitt ein Programm zum Zeichnen einer Geraden entwickelt wurde, soll nun ein Programm zum Zeichnen beliebiger Kreise vorgestellt werden. Dabei wiederholt sich nun vieles, so daß wir uns entsprechend kurz fassen können.

Problemanalyse

Wir stehen hier vor dem gleichen Problem wie im Abschnitt zuvor, mit dem Unterschied, daß wir nun von der Kreisgleichung ausgehen müssen.

Diese Kreisgleichung lautet allgemein:

$$(x - a)^2 + (y - b)^2 = r^2$$

Dabei ist r der Radius des Kreises, x und y sind die Koordinaten der Kreislinie und a und b sind die Koordinaten des Kreismittelpunktes.

Löst man diese Beziehung nach y auf (Lösen einer quadratischen Gleichung), so erhält man die Gleichungen für den oberen und den unteren Halbkreis.

Es gilt nämlich:

$$(y - b)^2 = r^2 - (x - a)^2 = D$$

$$y - b = \pm SQR(D)$$

$$Y_{1/2} = b \pm SQR(D)$$

Damit sind die Grundlagen für das folgende Programm klar:

Es benötigt als Input-Informationen die Koordinaten des Kreismittelpunktes a und b (wir nennen sie im folgenden Programm T und Z) und den Radius r (in BASIC: R). Dann können für alternative Werte von X die dazugehörigen Werte Y1 und Y2 gemäß der obigen Beziehung bestimmt werden und die Punkte X,Y1 und X,Y2 können gezeichnet werden.

Es ist dabei zu beachten, daß sich X nur sinnvollerweise im Bereich von T-R bis T+R bewegen kann, weil außerhalb dieses Bereichs keine Y-Werte im reellen Bereich definiert sind (die Hilfsgröße D würde dort negativ und der Versuch, aus ihr die Wurzel zu ziehen, würde zu einer Fehlermeldung und zum Programmabbruch führen).

Programm

```
10 REM Z3-KREIS
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM ZEICHNEN EINES KREISES."
35 PRINT "         PROF.DR.W.VOSS, 1984"
40 PRINT :PRINT :PRINT :PRINT
50 PRINT "DIESES PROGRAMM BENOETIGT ALS INPUT-"
60 PRINT "INFORMATIONEN SPALTE UND ZEILE DES"
70 PRINT "KREISMITTELPUNKTES UND DEN RADIUS."
80 PRINT :PRINT :PRINT
90 PRINT "MITTELPUNKT-SPALTE : ";:INPUT T
100 PRINT "MITTELPUNKT-ZEILE  : ";:INPUT Z
105 PRINT :PRINT
110 PRINT "RADIUS             : ";:INPUT R
120 GRAPHICS 8:COLOR 1
130 PLOT 0,0:DRAWTO 0,159
140 PLOT 0,159:DRAWTO 319,159
150 PLOT T,0:DRAWTO T,159
160 PLOT 0,Z:DRAWTO 319,Z
170 A=T-R:B=T+R
180 IF A<0 THEN A=0
190 IF B>319 THEN B=319
200 FOR X=A TO B
210 D=R*R-(X-T)^2
220 IF D<=0 THEN Y=Z:GOTO 250
230 Y=Z+SQR(D)
240 IF Y<0 OR Y>159 THEN 260
250 PLOT X,Y
260 NEXT X
270 FOR X=B TO A STEP -1
280 D=R*R-(X-T)^2
290 IF D<=0 THEN Y=Z:GOTO 320
300 Y=Z-SQR(D)
310 IF Y<0 OR Y>159 THEN 330
320 PLOT X,Y
330 NEXT X
340 FOR I=1 TO 2000:NEXT I
350 GRAPHICS 0
360 PRINT "ENDE":END
```

Variablenliste

A = linker Extrempunkt des Kreises
B = rechter Extrempunkt des Kreises
D = Hilfsgröße $D = R^2-(X-T)^2$
I = Laufindex
R = Radius
T = Spalte des Kreismittelpunkts
X = X-Koordinatenwerte
Y = Y-Koordinatenwerte
Z = Zeile des Kreismittelpunkts

Programmbeschreibung

Satz 10-80 : Überschrift und Erläuterungen.

Satz 90-120 : Input-Informationen und Graphikmodus.

Satz 130-140 : Achsenkreuz.

Satz 150-160 : Hilfs-Achsenkreuz.

Satz 170-190 : Festlegung des Wertebereichs für X.

Satz 200-260 : Zeichnen eines Halbkreises.

Satz 270-330 : Zeichnen des anderen Halbkreises.

Satz 340-360 : Beendigung des Programms.

9.5 Die Sinuslinie

Problemstellung

In einem letzten Beispiel hochauflösender Graphik wollen wir eine Sinusschwingung darstellen, der wir im Kapitel "Physik" schon einmal begegnet waren. Die Vorgehensweise ist die gleiche wie in den vorangegangenen beiden Abschnitten.

Problemanalyse

Die Problemanalyse ist hier einfach, weil keine neuen Probleme zu bewältigen sind. Wir haben lediglich darauf zu achten, daß die Periodizität der Sinusschwingung (Frequenz) und ihre Amplitude (Ausschlag) so eingerichtet werden, daß die auf dem Bildschirm auszugebende Schwingung optisch gut aussieht, d.h. den zur Verfügung stehenden Raum in geschickter Weise ausnutzt.

Programm

```
10 REM Z4-SINUS
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM ZEICHNEN EINER SINUSLINIE"
35 PRINT :PRINT :PRINT
40 PRINT "        PROF.DR.W.VOSS, 1984"
50 PRINT :PRINT :PRINT
60 PRINT "DAS PROGRAMM BENOETIGT DIE FOLGENDEN"
70 PRINT "INPUT-INFORMATIONEN :":PRINT :PRINT
80 PRINT "FREQUENZ (IN EINHEITEN VON PI) ";
85 INPUT P
90 PRINT "AMPLITUDE (0 BIS 79)           ";
95 INPUT A
100 GRAPHICS 8:COLOR 1
110 PLOT 0,80:DRAWTO 319,80
120 FOR X=0 TO 319
130 J=X/50:J=J*(1/P)
140 Y=SIN(J):Y=Y*A
150 Y=80-Y
160 PLOT X,Y
170 NEXT X
180 FOR I=1 TO 2000:NEXT I
190 GRAPHICS 0
200 PRINT "ENDE DER AUSGABE":END
```

Variablenliste

A = Amplitude
I = Laufindex
J = transformierter X-Wert
P = Frequenz
X = X-Koordinatenwert
Y = Y-Koordinatenwert

Programmbeschreibung

Satz 10-50 : Überschrift.

Satz 60-95 : Input-Informationen.

Satz 100-110 : Zeichnen einer waagrechten Mittellinie.

Satz 120-170 : Bestimmung der Sinuswerte und Zeichnen nach Transformation gemäß P und A.

Satz 180-200 : Textausgabe und Beendigung des Programms.

9.6 Mathematik Tafel 1

Problemstellung

Im folgenden Programm werden für alle natürlichen Zahlen von 1 bis 100 die Quadratzahlen und die Quadratwurzeln ausgegeben.

Insoweit ist dieses Programm quasi als "Nachschlageprogramm" zu verstehen.

Problemanalyse

Eine Problemanalyse dürfte bei dieser sehr einfachen Problemstellung entbehrlich sein.

Programm

```
10 REM Z5-TAFEL1
20 PRINT CHR$(125)
30 PRINT "PROGRAM ZUR ERZEUGUNG DER QUADRAT-"
40 PRINT "ZAHLEN UND DER QUADRATWURZELN FUER"
45 PRINT "ALLE NATUERLICHEN ZAHLEN (1-100)."
47 PRINT :PRINT :PRINT
50 PRINT "         PROF.DR.W.VOSS, 1984"
60 GOSUB 2000:REM WARTEN
70 FOR I=1 TO 100
80 Q=I*I:W=SQR(I):W=INT(W*1000+0.5)/1000
90 PRINT I,Q,W
100 IF I/15=INT(I/15) THEN GOSUB 2000
110 NEXT I
120 PRINT :PRINT "ENDE DER AUSGABE":END
1000 REM UP KOPFZEILE
1010 PRINT "ZAHL","QUADRAT","WURZEL"
1015 PRINT
1017 FOR J=1 TO 32:PRINT "-";:NEXT J
1020 PRINT :RETURN
2000 REM UP WARTEN
2010 PRINT :PRINT
2020 PRINT "BITTE CONT EINGEBEN !"
2030 STOP
2040 PRINT CHR$(125):GOSUB 1000
2050 RETURN
```

Variablenliste

I = Laufindex
J = Laufindex
Q = Quadratzahl
W = Quadratwurzel

Programmbeschreibung

Satz 10-50 : Überschrift.

Satz 60 : Sprung ins Unterprogramm 2000 zum Abwarten.

Satz 70-110 : Berechnung und Ausgabe der interessierenden Werte.

Satz 120 : Ende des Hauptprogramms.

Satz 1000-1020: Unterprogramm zur Ausgabe der Tabellen-Kopfzeile.

Satz 2000-2050: Unterprogramm zum Abwarten.

Satz 2010 : Ausgabe einer Meldung

Satz 2020-2030: Abwarten.

Satz 2040-2050: Löschen des Bildschirms, Ausgabe einer neuen Kopfzeile durch Sprung ins UP 1000 und danach Rückkehr ins Hauptprogramm.

9.7 Mathematik Tafel 2

Problemstellung

Dieses Programm dient zur tabellarischen Ausgabe der Winkelfunktionen

- Sinus (SIN)
- Cosinus (COS)

Diese Winkelfunktionen werden in Abhängigkeit von Einheiten des Kreisparameters π (im Programm Pi genannt) ausgegeben. Deshalb werden in der sich ergebenden Tabelle auch Winkelgrade angegeben.

Problemanalyse

Auch hier ist eine Problemanalyse nicht erforderlich, weil keine besonderen Schwierigkeiten auftauchen.

Programm

```
10 REM Z6-TAFEL2
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BESTIMMUNG VON SINUS-"
40 PRINT "        UND COSINUS-WERTEN."
50 PRINT :PRINT :PRINT
60 PRINT "           PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT
90 GOSUB 2000:REM WARTEN
100 FOR J=0 TO 40
102 I=J/10:I=INT(I*100+0.5)/100
105 A=(360*I)/(2*(22/7)):A=INT(A+0.5)
110 B=SIN(I):B=INT(B*1000+0.5)/1000
120 C=COS(I):C=INT(C*1000+0.5)/1000
135 PRINT I,A,B,C
137 IF J=0 THEN 150
140 IF J/10=INT(J/10) THEN GOSUB 2000
150 NEXT J
160 PRINT :PRINT "ENDE DER AUSGABE":END
1000 REM UP KOPFZEILE
1010 PRINT "PI","GRAD","SIN","COS":PRINT
1020 RETURN
2000 REM UP WARTEN
2010 PRINT :PRINT :PRINT
2020 PRINT "BITTE CONT EINGEBEN !"
2030 STOP
2040 PRINT CHR$(125):GOSUB 1000
2050 RETURN
```

Variablenliste

A = Winkel (in Grad)
B = Sinus
C = Cosinus
I = J/1Ø (=Abszissenwert der Winkelfunktionen)
J = Laufindex (hier Ø-4Ø)

Programmbeschreibung

Satz 1Ø-7Ø : Überschrift.

Satz 9Ø : Sprung ins Unterprogramm 2ØØØ zum Abwarten.

Satz 1ØØ-15Ø : Berechnung und Ausgabe der Ergebnisse. Es werden 41 Tabellenwerte bestimmt (J=Ø TO 4Ø), wobei die Winkelfunktionen auf I=J/1Ø angewandt werden (Satz 1Ø2).

Satz 1Ø5 : Umwandlung in Winkelgrad.
Satz 11Ø-12Ø: Bestimmung von sin und cos.
Satz 135-137: Ausgabe.
Satz 14Ø : Sprung ins Unterprogramm 2ØØØ nach jeder 1Ø. Runde. (Um bei J=Ø diesen Sprung zu vermeiden, der dann noch nicht sinnvoll ist, wird in Satz 137 für J=Ø zum Satz 15Ø gesprungen).

Satz 160 : Ende des Hauptprogramms.

Satz 1000-1020: Unterprogramm zur Ausgabe der Tabellen-Kopfzeile.

Satz 2000-2050: Unterprogramm zum Abwarten (Beschreibung siehe Programm Z6-TAFEL1).

9.8 Mathematik Tafel 3

Problemstellung

Das folgende Programm dient zur Ausgabe der natürlichen Logarithmen ln(X) im Wertebereich für X zwischen 1 und 1ØØ.

Problemanalyse

Auch hier ist eine Problemanalyse wegen der Schlichtheit der Fragestellung entbehrlich.

Programm

```
10 REM Z7-TAFEL3
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BESTIMMUNG DER WERTE DER"
40 PRINT "      NATUERLICHEN LOGARITHMEN."
50 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 GOSUB 2000
80 FOR I=1 TO 100
85 L=LOG(I)
90 PRINT I,L
100 IF I/15=INT(I/15) THEN GOSUB 2000
110 NEXT I
120 PRINT :PRINT "ENDE DER AUSGABE":END
1000 REM UP KOPFZEILE
1010 PRINT "ZAHL","LOGARITHMUS":PRINT
1020 RETURN
2000 REM UP WARTEN
2010 PRINT :PRINT :PRINT
2020 PRINT "BITTE CONT EINGEBEN !"
2030 STOP
2040 PRINT CHR$(125):GOSUB 1000
2050 RETURN
```

Variablenliste

A = Winkel (in Grad)
B = Sinus
C = Cosinus
I = Laufindex (hier 1-100)
J = Natürlicher Logarithmus von I

Programmbeschreibung

Satz 10-60 : Überschrift.

Satz 70 : Sprung ins Unterprogramm 2000 zum Abwarten.

Satz 80-110 : Berechnungen der Logarithmen und Ausgabe; nach jeder 15. Tabellenzeile Sprung ins Unterprogramm 2000 zum Warten.

Satz 120 : Beendigung des Hauptprogramms.

Satz 1000-1020: Unterprogramm zur Ausgabe der Tabellen-Kopfzeile.

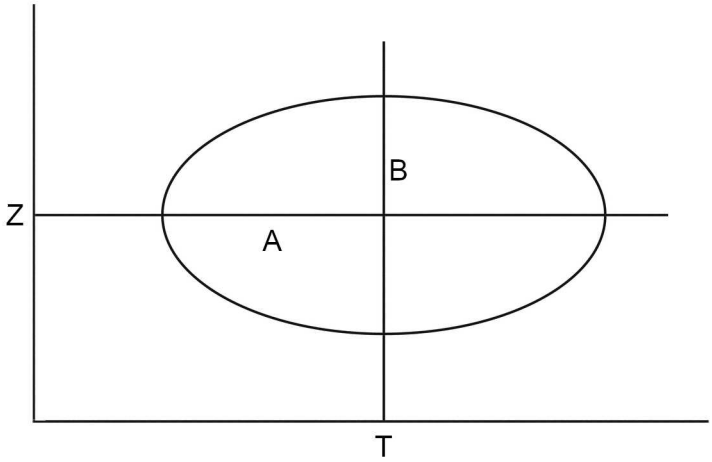
Satz 2000-2050: Unterprogramm zum Abwarten (Beschreibung siehe Programm Z6-TAFEL1).

9.9 Ellipse

Problemstellung

Das Programm zeichnet eine beliebige Ellipse.
Dabei sind folgende Werte einzugeben:

T: Spalte des Ellipsenmittelpunkts (Ø-319)
Z: Zeile des Ellipsenmittelpunkts (Ø-159)
A: Erste Halbachse der Ellipse
B: Zweite Halbachse der Ellipse



Problemanalyse

Die Ellipsengleichung lautet in der sog. Mittelpunkt-Schreibweise (Ellipsenmittelpunkt im Koordinatenursprung):

$$\frac{x^2}{a} + \frac{y^2}{b} = 1$$

Um eine derartige Funktion per BASIC-Programm zeichnen zu können, müssen wir sie nach y auflösen, wobei sich ergibt:

$$y_{1/2} = \pm\frac{b}{a}\sqrt{a^2-x^2}$$

Wenn wir nun berücksichtigen, daß eine beliebige Ellipse die Koordinaten (T,Z) und nicht immer nur (Ø,Ø) als Mittelpunktkoordinaten aufweist, so sind Verschiebungskorrekturen erforderlich (vergl. die folgende Programmbeschreibung).

Programm

```
10 REM Z8-ELLIPSE
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM ZEICHNEN EINER BELIEBI-"
40 PRINT "              GEN ELLIPSE."
45 PRINT :PRINT :PRINT
50 PRINT "          PROF.DR.W.VOSS, 1984"
60 PRINT :PRINT :PRINT :PRINT
70 PRINT "DIESES PROGRAMM BENOETIGT ALS INPUT-"
80 PRINT "INFORMATIONEN :":PRINT
90 PRINT "KOORDINATEN DES MITTELPUNKTS :":PRINT
100 PRINT "      - SPALTE :  ";:INPUT T
110 PRINT "      - ZEILE  :  ";:INPUT Z
115 PRINT :PRINT
120 PRINT "1. HALBACHSE  :  ";:INPUT A
125 PRINT "2. HALBACHSE  :  ";:INPUT B
130 GRAPHICS 8:COLOR 1
140 PLOT 0,159:DRAWTO 319,159
150 PLOT 0,159:DRAWTO 0,0
160 PLOT 0,Z:DRAWTO 319,Z
170 PLOT T,159:DRAWTO T,0
390 REM ELLIPSE
400 C=T-A:D=T+A
410 IF C<0 THEN C=0
420 IF D>319 THEN D=319
430 FOR X=C+1 TO D-1
440 DS=(B/A)*SQR(A*A-(X-T)^2)
450 Y=Z+DS
460 IF Y<0 OR Y>159 THEN 480
470 PLOT X,Y
480 NEXT X
490 FOR X=D-1 TO C+1 STEP -1
500 DS=(B/A)*SQR(A*A-(X-T)^2)
510 Y=Z-DS
520 IF Y<0 OR Y>159 THEN 540
530 PLOT X,Y
540 NEXT X
550 FOR I=1 TO 2000:NEXT I
560 GRAPHICS 0
570 PRINT "ENDE":END
```

Variablenliste

A = Erste Halbachse der Ellipse
B = Zweite Halbachse der Ellipse
C = Linker Extremwert der Ellipse
D = Rechter Extremwert der Ellipse
DS = Hilfsgröße aus der Ellipsengleichung
I = Laufindex
T = Spalte des Ellipsenmittelpunkts
X = Abszissenwert
Y = Ordinatenwert
Z =Zeile des Ellipsenmittelpunkts

Programmbeschreibung

Satz 1Ø-6Ø : Überschrift.

Satz 7Ø-125 : Vorgabe der Mittelpunktkoordinaten und der Halbachsen.

Satz 13Ø : Graphikmodus.

Satz 14Ø-15Ø : Zeichnen des Achsenkreuzes.

Satz 16Ø-17Ø : Zeichnen eines zweiten Achsenkreuzes, dessen Achsen durch den Ellipsenmittelpunkt laufen.

Satz 39Ø-48Ø : Zeichnen der unteren Ellipsenhälfte.

Satz 49Ø-54Ø : Zeichnen der oberen Ellipsenhälfte.

Satz 55Ø-57Ø : Beendigung des Programms.

9.10 Gauß'sche Normalverteilung

Problemstellung

Dieses Programm stellt die Gauß'sche Normalverteilung graphisch dar.
Sie hat die Dichtefunktion

$$f(x) = \frac{1}{SS*\sqrt{2\pi}}*e^{-\frac{1}{2}\left(\frac{x-MM}{SS}\right)^2}$$

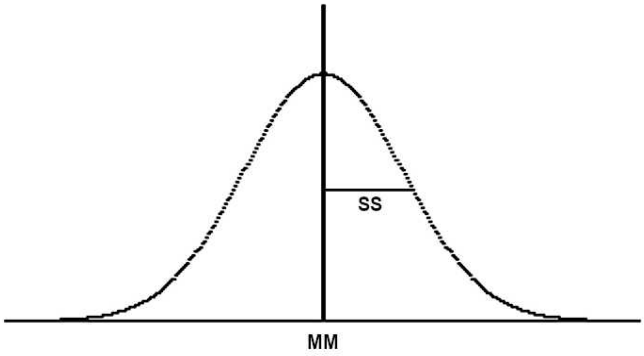
Dabei sind MM und SS die charakterisierenden Lageparameter (siehe folgende Skizze):

MM bezeichnet den Mittelwert der Verteilung

SS bezeichnet die Streuung der Verteilung (Standardabweichung = Abstand zwischen dem Lot im Mittelwert und einem der beiden Wendepunkte der Verteilung).

In diesem Programm haben wir vorgegeben:

MM = 16Ø
SS = 4Ø



Problemanalyse

Hier wiederum entbehrlich.

Programm

```
10 REM Z9-GAUSS
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUM ZEICHNEN DER NORMALVER-"
40 PRINT "         TEILUNG NACH GAUSS."
45 PRINT :PRINT :PRINT
50 PRINT "         PROF.DR.W.VOSS, 1984"
60 PRINT :PRINT :PRINT :PRINT
70 PRINT "DIESES PROGRAMM ZEICHNET DIE NORMAL-"
80 PRINT "VERTEILUNG MIT DEM MITTELWERT 160 UND"
90 PRINT "DER STANDARDABWEICHUNG 40."
100 PRINT :PRINT
105 PRINT "DIESE PARAMETER KOENNEN GGF. GEAEN-"
110 PRINT "DERT WERDEN (SATZ 120)."
115 GOSUB 1000:REM WARTEN
120 MM=160:SS=40
130 GRAPHICS 8:COLOR 1
140 PLOT 0,159:DRAWTO 319,159
150 PLOT 160,0:DRAWTO 160,159
390 REM NORMALVERTEILUNG
400 A=1/(SS*SQR(2*(22/7)))
410 FOR X=0 TO 319
420 B=((X-MM)/SS)^2
430 B=0.5*B*(-1)
440 Y=A*EXP(B)
450 Y=(500000/SS)*Y
455 Y=159-Y
460 IF Y<0 OR Y>159 THEN 480
470 PLOT X,Y
480 NEXT X
490 FOR I=1 TO 2000:NEXT I
500 GRAPHICS 0
510 PRINT "ENDE":END
1000 REM UP WARTEN
1010 PRINT :PRINT :PRINT
1020 PRINT "BITTE CONT EINGEBEN !"
1025 STOP
1030 RETURN
```

Variablenliste

A = Erster Term der Funktionalgleichung
B = Exponent in der Funktionalgleichung
I = Laufindex
MM = Mittelwert der Normalverteilung
SS = Standardabweichung der Normalverteilung
X = Abszissenwert
Y = Ordinatenwert

Programmbeschreibung

Satz 10-110 : Überschrift und Erläuterungen.

Satz 115 : Abwarten.

Satz 120-130 : Vorgabe der Funktionalparameter der Verteilung und Graphikmodus.

Satz 140-150 : Zeichnen des Achsenkreuzes.

Satz 390-480 : Errechnen der Ordinatenwerte der Normalverteilung, Normierung (Satz 450) und graphische Ausgabe.

Satz 490-510 : Ende des Programms.

Satz 1000-1030: Unterprogramm zum Abwarten.

9.11 Regression

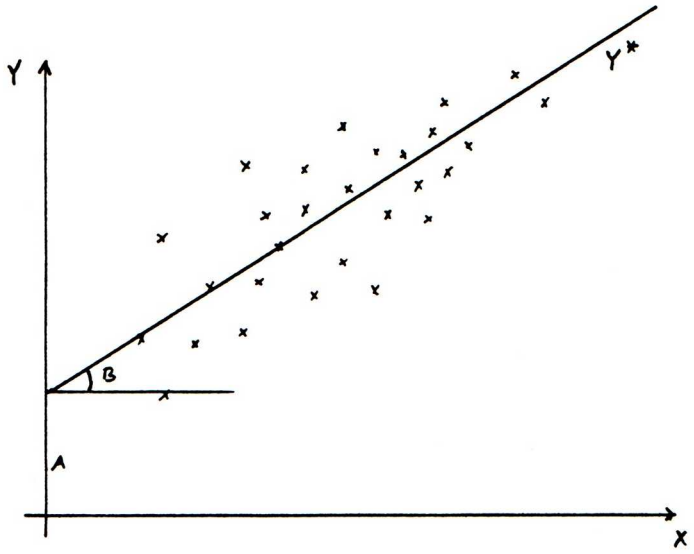
Problemstellung

Die Regressionsrechnung ist eine wichtige Aufgabenstellung aus dem Bereich der mathematischen Statistik. Es geht dabei darum, eine bivariate statistische Verteilung durch eine zusammenfassende mathematische Funktion, die im einfachsten Fall eine Gerade ist, zu beschreiben.

Eine bivariate statistische Verteilung entsteht beispielsweise dann, wenn wir eine gegebene Population nach zwei Merkmalen gleichzeitig untersuchen, wenn wir also zum Beispiel eine Reihe von Untersuchungspersonen nach Körpergröße und Körpergewicht gleichzeitig befragen.

Ziel einer solchen Untersuchung ist es, festzustellen, ob zwischen den beiden Variablen ein Zusammenhang besteht.

Eine solche bivariate Verteilung läßt sich anschaulich als "Punktwolke" in einem zweidimensionalen Koordinatensystem darstellen, wie es die folgende Skizze verdeutlicht.



Um nun die Art des Zusammenhangs zwischen beiden Variablen zu beschreiben, bzw. um bei einer Zeitreihe den linearen Trend zu bestimmen, was mathematisch die gleiche Aufgabe ist, versucht man, in die Punkte im Achsenkreuz möglichst gut eine lineare mathematische Funktion hineinzulegen.

Entsprechend gibt es auch nichtlineare Berechnungsansätze, auf die aber hier nicht eingegangen werden soll.

Problemanalyse

Die einzuzeichnende lineare oder nichtlineare Funktion bestimmt man in der Regel mit der Methode der kleinsten Quadrate.

Diese Methode gibt eine Rechenvorschrift an zur Bestimmung der Parameter der anzupassenden Funktion. Bei einer Geraden handelt es sich dabei um den Ordinatenabschnitt A und die Steigung B.

Die Methode der kleinsten Quadrate verlangt, daß diese beiden Parameter so zu bestimmen sind, daß die Summe der quadrierten Abweichungen zwischen den beobachteten Y-Werten y_i (senkrechte Koordinaten) und den auf der Gerade liegenden Y-Werten y_i^* zu minimieren ist, also:

$$\sum(y_i - y_i^*)^2 = \text{Min.}$$

Da die Werte y_i^* sich gemäß der Geradengleichung wie folgt berechnen:

$$y_i^* = a + b*x_i$$

ergibt sich aus der obigen Minimierungsregel:

$$\sum(y_i - a - b*x_i)^2 = \text{Min.}$$

Das Minimum dieser Beziehung findet man, indem man sie partiell nach den Größen a und b differenziert und diese beiden ersten partiellen Ableitungen gleich null setzt.

Auf diese Weise erhält man zwei Gleichungen mit zwei Unbekannten (a und b), die man nach eben diesen Unbekannten auflösen kann. Damit erhält man die folgenden beiden Bestimmungsgleichungen für a und b:

$$b = \frac{n*\sum x_i y_i - \sum x_i \sum y_i}{n*\sum x_i^2 - \sum x_i^2}$$

$$a = \frac{\sum y_i}{n} - b*\frac{\sum x_i}{n}$$

Das zu entwickelnde BASIC-Programm dient dazu, für beliebig einzugebende Wertepaare x_i, y_i diese beiden Parameter nach der Methode der kleinsten Quadrate zu errechnen und als Ergebnisse auszugeben.

Programm

```
10 REM Z10-REGRESSION
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BESTIMMUNG DER PARAMETER"
40 PRINT "EINER LINEAREN REGRESSIONSFUNKTION"
50 PRINT" BZW. EINER LINEAREN TRENDLINIE."
55 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT :PRINT :PRINT
80 PRINT "DIESES PROGRAMM BENOETIGT WERTEPAARE"
90 PRINT "ALS INPUT-INFORMATIONEN (Z.B. X- UND"
100 PRINT "Y-WERTE ODER ZEIT- UND MERKMALSWER-"
105 PRINT "              TE ETC.)"
110 GOSUB 1000:REM WARTEN
120 PRINT "WIEVIELE WERTE PAARE : ";:INPUT N
130 DIM X(N),Y(N)
140 PRINT :PRINT :PRINT "       EINGABE :"
145 PRINT :PRINT
150 FOR I=1 TO N
160 PRINT I;".PAAR  :   ";:INPUT Z1,Z2
165 X(I)=Z1:Y(I)=Z2
170 NEXT I
172 PRINT CHR$(125):PRINT "KONTROLLE"
173 PRINT :PRINT
174 PRINT "NR.","X-WERT","Y-WERT":PRINT
176 FOR I=1 TO N:PRINT I,X(I),Y(I)
178 IF I/10=INT(I/10) THEN GOSUB 1000
179 NEXT I
180 REM REGRESSION
185 GOSUB 1000:REM WARTEN
190 FOR I=1 TO N
200 S1=S1+X(I)*Y(I)
210 S2=S2+X(I):S3=S3+Y(I)
220 S4=S4+X(I)*X(I)
230 NEXT I
240 B=(N*S1-S2*S3)/(N*S4-S2*S2)
250 A=S3/N-B*S2/N
```

```
265 PRINT CHR$(125)
270 PRINT "ERGEBNIS :  ":PRINT :PRINT
280 PRINT "ORDINATENABSCHNITT A = ";A
285 PRINT
290 PRINT "STEIGUNG            B = ";B
300 PRINT :PRINT :PRINT
310 PRINT "MIT DIESEN PARAMETERN KANN NUN DAS"
320 PRINT "PROGRAMM ZUM ZEICHNEN EINER GERADEN"
330 PRINT "AUFGERUFEN WERDEN, WENN MAN DAS ER- "
340 PRINT "GEBNIS AUCH GRAPHISCH VERANSCHAULI-"
345 PRINT "CHEN MOECHTE."
347 PRINT :PRINT :PRINT
350 PRINT "ENDE DER AUSGABE":END
1000 REM UP WARTEN
1010 PRINT :PRINT :PRINT
1020 PRINT "BITTE CONT EINGEBEN !":STOP
1030 PRINT CHR$(125)
1040 RETURN
```

Variablenliste

A = Ordinatenabschnitt der Geraden
B = Steigung der Geraden
I = Laufindex
N = Zahl der Wertepaare
S1 =
S2 =
S3 =
S4 = } Hilfssummen zur Berechnung von B und A
X = Abszissenwert
Y = Ordinatenwert
Z1 =
Z2 = } Hilfsfelder zur Dateneingabe

Programmbeschreibung

Satz 10-110 : Überschrift, Erläuterungen und Warten.

Satz 120-130 : Angabe der Zahl der Paare von Merkmalswerten und Dimensionierung.

Satz 140-170 : Eingabe der Merkmalswerte über INPUT.

Satz 172-179 : Kontrollausgabe der Paare von Merkmalswerten mit Pause nach jedem zehnten Paar.

Satz 180-185 : Warten.

Satz 190-230 : Bildung der vier Summen, die in den Berechnungsformeln für A und B benötigt werden.

Satz 240-250 : Berechnung der Geradenparameter A und B.

Satz 265-300 : Ergebnisausgabe.

Satz 310-350 : Verzweigungshinweis und Beendigung des Programms.

Satz 1000-1040: Unterprogramm Warten.

9.12 Permutationen

Problemstellung

Die Berechnung von Permutationen beantwortet die Frage, auf wieviele unterschiedliche Arten verschiedene Elemente angeordnet werden können.

Problemanalyse

Haben wir beispielsweise drei verschiedene Elemente a, b und c, so sind die folgenden Anordnungsmöglichkeiten voneinander unterscheidbar:

abc, acb, bac, bca, cab, cba

Bei drei Elementen gibt es also sechs verschiedene Anordnungsmöglichkeiten.

Allgemein gilt die folgende Überlegung:

Für das erste von drei Elementen gibt es drei Möglichkeiten: Es kann an erster, an zweiter oder an dritter Stelle stehen;

für das zweite Element gibt es dann nur noch zwei Möglichkeiten, wenn entschieden worden ist, wo das erste Element stehen soll.

Dies bedeutet aber, daß es für die ersten beiden Elemente zusammen 3*2 = 6 Möglichkeiten gibt.

Für das dritte Element gibt es dann nur noch eine Möglichkeit und für alle drei Elemente also 3*2*1 = 6 verschiedene Anordnungsmöglichkeiten.

Hat man nun n verschiedene Elemente, so ergibt sich entsprechend:

Zahl der Anordnungsmöglichkeiten =

n*(n-1)*(n-2)*(n-3)* ... * 3*2*1

Diese Produktkette kürzt man ab durch die Schreibweise n! (sprich n-Fakultätl und sagt, daß n verschiedene Elemente auf n-Fakultät verschiedene Arten angeordnet werden können, oder:

Permutation von n verschiedenen Elementen ist n-Fakultät (n!).

Es ist zu beachten, daß bei großen Werten für n der Ausdruck n! rasch sehr groß wird, und zwar so rasch ansteigend, daß schon ab n = 34 der Rechner das Ergebnis 34! nicht mehr darstellen kann.

Programm

```
10 REM Z11-PERMUTATIONEN
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BESTIMMUNG DER ANZAHL"
40 PRINT "DER PERMUTATIONEN VON N VERSCHIEDE-"
50 PRINT" NEN ELEMENTEN (N-FAKULTAET = N!)"
55 PRINT :PRINT :PRINT
58 DIM A$(1)
60 PRINT "         PROF. DR.W. VOSS, 1984"
70 PRINT :PRINT :PRINT :PRINT
80 PRINT "ZAHL DER ELEMENTE : ";:INPUT N
90 P=N
100 IF N>33 OR N<2 THEN PRINT "N UNZULAESSIG !":GOTO 150
110 FOR I=N-1 TO 1 STEP -1
120 P=P*I
130 NEXT I
135 PRINT :PRINT :PRINT
140 PRINT "FAKULTAET VON ";N;" IST ";P
145 PRINT :PRINT :PRINT
150 PRINT "NOCH EINE RECHNUNG (J/N) ";:INPUT A$
160 IF A$="J" THEN PRINT CHR$(125):GOTO 80
170 PRINT :PRINT "ENDE":END
```

Variablenliste

A$ = Stringvariable für Antworten
I = Laufindex
N = Zahl der Elemente
P = Zwischen- und Endprodukt

Programmbeschreibung

Satz 1Ø-7Ø : Überschrift und Erläuterungen.

Satz 8Ø-9Ø : Anforderung der Input-Information (Zahl der zu permutierenden Elemente N) und Belegung des Produktfeldes P mit dem Wert im Feld N als erstem Faktor der zu bildenden Produktkette.

Satz 1ØØ : Meldung,wenn N gleich oder größer 34 ist, weil dazu die Rechenkapazitäten nicht ausreichen und Sprung nach 15Ø.

Satz 11Ø-13Ø : Bildung der Produktkette im Feld P.

Satz 14Ø : Ergebnisausgabe.

Satz 15Ø-17Ø : Beendigung des Programms, es sei denn, es wird eine erneute Berechnung gewünscht (dann weiter bei Satz 8Ø nach Löschen des Bildschirms).

9.13 Kombinationen

Problemstellung

Eine ähnliche Fragestellung wie im vorangegangenen Programm liegt vor, wenn berechnet werden soll, auf wieviele verschiedene Arten k Elemente aus n verschiedenen Elementen ausgewählt werden können.

Als Berechnungsformel ergibt sich:

$$\text{Anzahl} = \frac{n!}{k!(n-k)!}$$

Man nennt diese Größe Binomialkoeffizient und kürzt ihn ab mit folgender Schreibweise:

$$\binom{n}{k}$$

Er entspricht übrigens den Faktoren, die sich bei der Auflösung der sog. Binome ergeben:

$$\begin{aligned}(a+b)^2 &= a^2 + 2ab + b^2 \\ &= 1a^2b^0 + 2a^1b^1 + 1a^0b^2 \\ &= \binom{3}{0}a^2b^0 + \binom{2}{1}a^1b^1 + \binom{2}{2}a^0b^2\end{aligned}$$

oder:

$$\begin{aligned}(a+b)^3 &= a^3+3a^2b+3ab^2+b^3\\ &= 1a^3b^{\emptyset}+3a^2b^1+3a^1b^2+1a^{\emptyset}b^3\\ &= \binom{3}{\emptyset}a^3b^{\emptyset}+\binom{3}{1}a^2b^1+\binom{3}{2}a^1b^2+\binom{3}{3}a^{\emptyset}b^3\end{aligned}$$

Man sieht, daß jedes Binom sich in Summanden zerlegen läßt, wobei es immer ein Summand mehr ist als die Hochzahl des Binoms angibt (Hochzahl im letzten Beispiel war 3; Zahl der Summanden war 4).

Jeder Summand besteht aus 3 Faktoren:

- eine Konstante (der hier interessierende Binomialkoeffizient);
- Faktor " a hoch ... " mit einer Hochzahl, die von der Binomhochzahl (hier 3) auf Ø sinkt;
- Faktor "b hoch ... " mit einer Hochzahl, die von Ø zur Binomhochzahl (hier 3) ansteigt.

Im hier interessierenden Binomialkoeffizienten wiederum steht als oberer Wert die Binomhochzahl (hier 3); als unterer eine ganze Zahl, die von links nach rechts von Ø bis zur Binomhochzahl ansteigt.

Das Programm, um das es hier geht, bestimmt beliebige Binomialkoeffizienten, so daß wir also mit diesem Programm leicht auch kompliziertere Binome bestimmen können (z. B. $(a+b)^{27}$).

Aber auch für Berechnungen kann dieses Programm genutzt werden, wie die folgenden beiden Programme zeigen.

Schließlich kann mit einem derartigen Programm z.B. folgende konkrete Frage beantwortet werden:

"8 Läufer starten zu einem 100-m-Lauf. Auf wieviele verschiedene Arten können die ersten drei Plätze besetzt werden?"

Antwort:

$$\binom{8}{3} = \frac{8!}{3!}(8-3)! = \frac{8!}{3!5!}$$

$$= \frac{8*7*6}{3*2*1} = \underline{\underline{56}}$$

Programm

```
10 REM Z12-KOMBINATIONEN
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BESTIMMUNG DER ANZAHL"
40 PRINT "DER KOMBINATIONEN VON K AUS N ELEMEN-"
45 PRINT "                    TEN.": PRINT
50 PRINT "MAN SPRICHT IN DIESEM ZUSAMMENHANG"
55 PRINT "VON DEN SOG. BINOMIALKOEFFIZIENTEN."
57 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT :PRINT
80 PRINT "ZAHL DER ELEMENTE    N : ";:INPUT N
90 PRINT "AUSWAHLANZAHL        K : ";:INPUT K
100 Y=N-K
110 IF K=0 OR K=N THEN BK=1:GOTO 210
120 IF K>N/2 THEN H=Y:Y=K:K=H
130 GOSUB 500:REM FAK
140 PX=A:A=N
150 IF PX=1 THEN PN=N:GOTO 200
160 FOR I=N-1 TO N-K+1 STEP -1
170 A=A*I
180 NEXT I
190 PN=A
200 BK=PN/PX
205 PRINT :PRINT :PRINT
210 PRINT "ANZAHL DER KOMBINATIONEN VON ";K
220 PRINT "AUS ";N;" ELEMENTEN IST ";BK
225 PRINT :PRINT :PRINT
230 PRINT "ENDE DER BERECHUNGEN":END
500 REM UP FAK
510 A=K
520 IF K=0 OR K=1 THEN A=1:GOTO 560
530 FOR I=K-1 TO 1 STEP -1
540 A=A*I
550 NEXT I
560 RETURN
```

Variablenliste

A = Fakultät (Zwischen- und Endprodukt)
BK = Binomialkoeffizient
H = Hilfsgröße beim Tausch von K und Y
I = Laufindex (Faktor bei der Fakultätsberechnung)
K = Zahl der auszuwählenden Elemente
N = Zahl der Elemente
PN = Erste Fakultät im Binomialkoeffizient
PX = Zweite Fakultät im Binomialkoeffizient
Y = N-K

Programmbeschreibung

Satz 1Ø-7Ø : Überschrift und Erläuterungen.

Satz 8Ø-9Ø : Eingaben (Zahl der Elemente N und Zahl der Auszuwählenden K).

Satz 1ØØ : Belegung der Hilfsgröße Y.

Satz 11Ø : Vereinfachte Berechnung für K=Ø oder K=N, weil definitionsgemäß gilt:

$$\binom{N}{Ø} = \binom{N}{N} = 1$$

Satz 12Ø : Vertauschen von K und (N-K) (=Y), falls K>N/2. Dies ist möglich, weil gilt:

$$\binom{N}{K} = \frac{N!}{K!(N-K)!} = \binom{N}{N-K}$$

Auf diese Weise werden die folgenden Berechnungen vereinfacht.

Satz 13Ø : Berechnung von K! im Unterprogramm 5ØØ.

Satz 14Ø-19Ø : Berechnung der anderen Fakultät, die nach geeignetem Kürzen (siehe z. B. die Rennläufer-Aufgabe oben) noch übrig bleibt.

Satz 2ØØ-23Ø : Berechnung und Ausgabe des Binomialkoeffizienten und Beendigung des Hauptprogramms.

Satz 5ØØ-56Ø : Berechnung von K! in einem Unterprogramm.

9.14 Lotto

Problemstellung

Es soll ein Programm erstellt werden, welches die Chancen ausrechnet, beim Lottospiel (6 aus 49) zu gewinnen.

Problemanalyse

Das Programm berechnet die Wahrscheinlichkeit für 3, 4, 5 oder 6 Richtige im Zahlenlotto (6 aus 49).

Die entsprechenden Wahrscheinlichkeiten berechnen sich gemäß der folgenden Formel:

$$P(X) = \binom{6}{X} * \binom{43}{6-X} = \frac{B1*B2}{B3}$$

Dabei ist X die vom Benutzer anzugebende "Zahl der Richtigen".

Es muß hier also mit drei Binomialkoeffizienten (B1, B2 und B3) gearbeitet werden, denen wir ja schon im vorangegangenen Kapitel begegnet waren.

Deshalb sind detailliertere Erläuterungen hier entbehrlich.

Programm

```
10 REM Z13-LOTTO
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR BESTIMMUNG DER WAHR-"
40 PRINT "SCHEINLICHKEIT FUER 3,4,5 ODER 6"
50 PRINT "RICHTIGE IM ZAHLENLOTTO (6 AUS 49)."
55 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT :PRINT
80 PRINT "DIESES PROGRAMM BERUECKSICHTIGT NICHT"
90 PRINT "DIE MOEGLICHKEIT 5 RICHTIGE MIT ZU-"
100 PRINT "            SATZZAHL !"
110 GOSUB 1000:REM WARTEN
119 DIM A$(1)
120 PRINT "WIEVIELE RICHTIGE : ";:INPUT X
125 IF X=6 THEN B1=1:B2=1:GOTO 300
130 REM B1
135 IF X=5 THEN B1=6:B2=43:GOTO 300
140 Y=6-X
150 P=X:FOR I=X-1 TO 1 STEP -1
160 P=P*I:NEXT I:P1=P
170 P=Y:FOR I=Y-1 TO 1 STEP -1
180 P=P*I:NEXT I
190 P2=P
200 B1=720/(P1*P2)
210 REM B2
220 X=Y
230 P=43
240 FOR I=P-1 TO 43-X+1 STEP -1
250 P=P*I:NEXT I
260 P1=P
265 P=X
270 FOR I=X-1 TO 1 STEP -1
280 P=P*I:NEXT I
290 P2=P:B2=P1/P2
300 REM B3
310 P1=49:P2=6
320 FOR I=1 TO 5
330 P1=P1*(49-I)
340 P2=P2*(6-I)
350 NEXT I
360 B3=P1/P2:W=(B1*B2)/B3
```

```
370 PRINT C HR$(125)
375 IF X=2 THEN X=4
380 PRINT "     ERGEBNIS :":PRINT :PRINT
390 PRINT "DIE WAHRSCHEINLICHKEIT FUER ";X
400 PRINT "RICHTIGE IM LOTTO BETRAEGT :"
410 PRINT :PRINT :PRINT "      ";W
415 PRINT :PRINT :PRINT
420 PRINT "NOCH EINE RECHNUNG (J/N) ";:INPUT A$
430 IF A$="J" THEN PRINT CHR$(125):GOTO 120
440 PRINT :PRINT :PRINT "ENDE":END
1000 REM UP WARTEN
1010 PRINT :PRINT :PRINT
1020 PRINT "BITTE CONT EINGEBEN !":STOP
1030 PRINT CHR$(125)
1040 RETURN
```

Variablenliste

A$ = Stringvariable für Antworteingaben
B1 = Erster Term in der Berechnungsformel
B2 = Zweiter Term in der Berechnungsformel
B3 = Dritter Term in der Berechnungsformel
P = Produkt in den Fakultätenberechnungen
P1 = Erste Fakultät im Binomialkoeffizienten
P2 = Zweite Fakultät im Binomialkoeffizienten
W = Wahrscheinlichkeit
X = Anzahl der "Richtigen"
Y = 6-X

Programmbeschreibung

Satz 10-100 : Überschrift und Erläuterungen.

Satz 110 : Sprung ins Unterprogramm 1000 zum Warten.

Satz 120 : Eingabe der Zahl der Richtigen.

Satz 125 : Für X=6 gilt, daß B1=B2=1 (siehe Programm Z10); weiter bei Satz 300.

Satz 130-200 : Bestimmung von B1 gemäß der Überlegungen wie wir sie aus Programm Z10 schon kennen.

Satz 210-290 : Entsprechende Bestimmung des zweiten Binomialkoeffizienten B2.

Satz 300-360 : Entsprechend B3.

Satz 370-410 : Ausgabe des Ergebnisses.

Satz 420-440 : Beendigung des Hauptprogramms, es sei denn, es wird eine Alternativberechnung gewünscht.

Satz 1000-1040: Unterprogramm zum Abwarten.

9.15 Binomialverteilung

Problemstellung

Die Binomialverteilung ist eine Wahrscheinlichkeitsverteilung. Wahrscheinlichkeitsverteilungen wiederum dienen dazu, auf möglichst einfache Weise auch für kompliziertere Fragestellungen Wahrscheinlichkeiten zu berechnen.

Die binomische Verteilung ist auch deshalb besonders wichtig, weil sie als "Grundbaustein" für eine ganze Reihe darauf aufbauender Wahrscheinlichkeitsberechnungsverfahren dient.

Die binomische Verteilung antwortet auf die folgende Frage:

Wie groß ist die Wahrscheinlichkeit, daß bei n voneinander unabhängigen Versuchen x_i-mal das günstige Ereignis eintritt, wenn bei jedem Versuch nur zwei Ergebnisse möglich sind, und wenn die Eintrittswahrscheinlichkeit des günstigen Ereignisses bei einem Versuch P % beträgt ?

Ein einfaches Beispiel mag diese Frage etwas veranschaulichen:

Wie groß ist die Wahrscheinlichkeit, bei 10 Münzwürfen z.B. 5 Mal "Kopf" zu werfen?

Dies ist ein Anwendungsfall der binomischen Verteilung, weil bei jedem Versuch zwei Ereignisse möglich sind ("Kopf" und Zahl") und weil die Versuche (die Würfe) voneinander unabhängig sind.

Eine entsprechende Frage wäre beispielsweise:

Wie groß ist die Wahrscheinlichkeit, daß bei 5 Geburten genau 2 Mädchen zur Welt kommen ?

Problemanalyse

Zur Analyse des gestellten Problems beschäftigen wir uns einmal mit dem zuletzt erwähnten Beispiel:

Wenn bei fünf Geburten zwei Mädchen zur Welt kommen sollen, so wäre beispielsweise folgende "Kinder-Kombination" günstig im Sinne der Fragestellung:

M M K K K (M = Mädchen, K = Knabe)

Günstig wäre aber auch:

K K K M M

und viele andere Kombinationen.

Die Anzahl der insgesamt "günstigen" Kombinationen kann mit Hilfe der Permutationsberechnung gefunden werden, über die schon gesprochen wurde.

Ohne dies hier beweisen zu wollen, stellen wir fest, daß die Anzahl der Kombinationsmöglichkeiten von 2 Mädchen und 3 Knaben sich ergibt zu:

$$\text{Anzahl} = \frac{5!}{2!*3!} = \frac{5*4*3*2*1}{2*1*3*2*1}$$

$$= \frac{5*4}{2*1} = 10$$

Allgemein gilt, daß die Anzahl der verschiedenen Anordnungsmöglichkeiten von n Elementen, von denen x_i Stück gleich sind (im Sinne von "günstig") und weitere $n-x_i$ ihrerseits gleich sind (im Sinne von "ungünstig") sich ergibt zu:

$$\text{Anzahl} = \frac{n!}{x_i!*(n-x_i)!}$$

Wir gehen bei der weiteren Problemanalyse davon aus, daß die Wahrscheinlichkeit einer Mädchengeburt P betrage (P = Ø.5) - dies ist die Eintrittswahrscheinlichkeit des günstigen Ereignisses bei einem Versuch.

Betrachten wir nun noch einmal das Ergebnis:

M M K K K

Wie groß ist die Wahrscheinlichkeit eines solchen Ergebnisses bei fünf Versuchen ?

Betrachten wir zunächst die erste Mädchengeburt: Ihre Wahrscheinlichkeit ist offenbar gleich Ø.5.

Auch die Wahrscheinlichkeit der zweiten Mädchengeburt ist Ø.5.

Die Wahrscheinlichkeit nun dafür, daß die ersten beiden Geburten zusammen beides Mädchengeburten sind, beträgt dann Ø.5 * Ø.5 (allgemein P * P).

Die Wahrscheinlichkeit für eine Knabengeburt beträgt 1 - Ø.5 (allgemein 1-P). Deshalb ergibt sich nach der gleichen Überlegung wie oben, daß die Wahrscheinlichkeit dafür, bei der dritten und bei der vierten und bei der fünften Geburt Knaben zu erhalten, sich ergibt zu:

$$(1 - 0.5)*(1 - 0.5)*(1 - 0.5)$$

oder allgemein:

$$(1 - P)*(1 - P)*(1 - P)$$

Für die Kombination M M K K K erhalten wir deshalb insgesamt die folgende Wahrscheinlichkeit in allgemeiner Schreibweise:

$$P * P * (1 - P) * (1 - P) * (1 - P)$$

$$= \quad p^2 \; * \; (1 - P)^3$$

$$= \quad p^{x_1} \; * \; (1 - P)^{n - x_1}$$

In unserem Beispiel würde sich deshalb ergeben:

Wahrscheinlichkeit für M M K K K =

$$0.5^2 * (1 - 0.5)^{5-2} \;=\; 0.5^5 \;=\; 1/32$$

Nun sind aber in diesem Beispiel 10 verschiedene M,K-Kombinationen günstig im Sinne der Fragestellung.

Deshalb ergibt sich die endgültige Wahrscheinlichkeit, weil für jede der 10 Kombinationen sich die Einzelwahrscheinlichkeit 1/32 ergibt, zu:

Wahrscheinlichkeit für 2 Mädchen bei fünf Geburten = 10 * 1/32 = 10/32 = 31.25 %

Aus diesem Beispiel können wir nun die allgemeine Berechnungsformel herleiten:

Die Wahrscheinlichkeit, bei n Versuchen xi-mal das günstige Ereignis zu haben, wenn dessen Einzeleintrittswahrscheinlichkeit P beträgt, ergibt sich zu:

$$W = \frac{n!}{x_i! * (n - x_i)!} * p^{x_i} * (1 - p)^{n - x_i}$$

Diese Formel sieht recht kompliziert aus, läßt sich aber vergleichsweise leicht programmieren.

Programm

```
10 REM Z14-BINOMIAL
20 PRINT CHR$(125)
30 PRINT " PROGRAMM ZUR BERECHNUNG DER WAHR-"
40 PRINT "SCHEINLICHKEITEN MIT DER BINOMISCHEN"
50 PRINT "            VERTEILUNG."
55 PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT
80 PRINT "MIT DIESEM PROGRAMM BZW. MIT DER BI-"
90 PRINT "NOMISCHEN VERTEILUNG KANN DIE FOLGEN-"
100 PRINT "DE FRAGE BEANTWORTET WERDEN :"
105 PRINT
110 PRINT "WIE GROSS IST DIE WAHRSCHEINLICHKEIT,"
120 PRINT "DASS BEI N VERSUCHEN X-MAL DAS GUEN-"
130 PRINT "STIGE EREIGNIS EINTRITT, WENN BEI "
140 PRINT "EINEM VERSUCH DESSEN WAHRSCHEINLICH-"
150 PRINT "KEIT P BETRAEGT ?"
160 GOSUB 1000:REM WARTEN
170 PRINT "DIESES PROGRAMM BENOETIGT ALS INPUT-"
180 PRINT "INFORMATIONEN : ":PRINT
189 DIM A$(1)
190 PRINT "ZAHL DER VERSUCHE  : ";:INPUT N
195 PRINT
200 PRINT "ZAHL DER GUENSTIGEN"
210 PRINT "EREIGNISSE         : ";: INPUT X
215 PRINT :Z=X
220 PRINT "EINTRITTSWAHR-"
225 PRINT "SCHEINLICHKEIT"
228 PRINT "BEI EINEM VERSUCH"
230 PRINT "(ALS DEZIMALZAHL)  : ";:INPUT P
240 REM BINOMIALKOEFFIZIENT
245 Y=N-X:Q=1-P
250 IF X=0 OR X=N THEN BK=1:GOTO 360
260 IF X>N/2 THEN H=Y:Y=X:X=H:G=Q:Q=P:P=G
270 F=X:GOSUB 500:REM FAK
290 PX=A:A=N
295 IF PX=1 THEN PN=N:GOTO 350
300 FOR I=N-1 TO N-X+1 STEP -1
310 A=A*I
320 NEXT I
```

```
330 PN=A
350 BK=PN/PX
360 W=BK*P^X*Q^Y
365 PRINT CHR$(125)
370 PRINT "      ERGEBNIS":PRINT :PRINT
380 PRINT "DIE WAHRSCHEINLICHKEIT, DASS BEI ";N
390 PRINT "VERSUCHEN ";Z;" MAL DAS GUENSTIGE"
400 PRINT "EREIGNIS EINTRITT, BETRAEGT :"
410 PRINT : PRINT : PRINT "    ";W
415 PRINT :PRINT :PRINT
420 PRINT "NOCH EINE RECHNUNG (J/N) ";:INPUT A$
430 IF A$="J" THEN PRINT CHR$(125):GOTO 190
440 PRINT :PRINT :PRINT "ENDE":END
500 REM UP FAK
510 A=X
520 IF X=0 OR X=1 THEN A=1:GOTO 560
530 FOR I=X-1 TO 1 STEP -1
540 A=A*I
550 NEXT I
560 RETURN
1000 REM UP WARTEN
1010 PRINT :PRINT :PRINT
1020 PRINT "BITTE CONT EINGEBEN !":STOP
1030 PRINT CHR$(125)
1040 RETURN
```

Variablenliste

A = Produkt bei der Fakultätenberechnung
A$ = Stringvariable für Antworteingabe
BK = Binomialkoeffizient
F = Hilfsgröße (F=X)
G = Hilfsgröße beim Tausch von P und Q
H = Hilfsgröße beim Tausch von X und Y
I = Laufindex
N = Zahl der Versuche
P = Eintrittswahrscheinlichkeit des günstigen Ereignisses bei einem Versuch
PN = Erster Term des Binomialkoeffizienten
PX = Zweiter Term des Binomialkoeffizienten
Q = 1-P
W = Wahrscheinlichkeit
X = Zahl der günstigen Ereignisse
Y = N-X
Z = X

Programmbeschreibung

Satz 1Ø-15Ø : Überschrift und Erläuterungen.

Satz 16Ø-23Ø : Warten und danach Eingabe der Input-Informationen (Zahl der Versuche, Zahl der günstigen Ereignisse, Eintrittswahrscheinlichkeit).

Satz 24Ø-35Ø : Bestimmung des Binomialkoeffizienten:

245: Hilfsfelder;

25Ø: Ist X=Ø oder X=1, dann ist der BK definiert als 1; weiter bei 36Ø;

26Ø: Ist X> N/2, so ist, um die Kürzungsmöglichkeiten auszunutzen, X mit N-X und P mit 1-P zu tauschen.

27Ø: Bestimmung von x!;

29Ø: Belegung des Nenners von BK (=PX) und Neubelegung von A, falls Var. A erneut benutzt wird (3ØØ);

295: Ist der Nenner des BK = 1, so ist der Zähler des BK definiert als N; weiter bei Satz 35Ø.

3ØØ-32Ø: Bestimmung des Zählers des Binomialkoeffizienten BK;

35Ø: Berechnung von BK.

Satz 36Ø : Berechnung der gesuchten Wahrscheinlichkeit.

Satz 365-410 : Ausgabe des Berechnungsergebnisses.

Satz 420-499 : Beendigung des Programms, es sei denn es wird eine erneute Berechnung gewünscht (dann nach Löschen des Bildschirms zurück zu Satz 190.

Satz 500-560 : Unterprogramm zur Berechnung einer Fakultät.

Satz 1000-1040: Unterprogramm zum Abwarten.

9.16 Zentrales Grenzwerttheorem (ZGWT)

Problemstellung

Gesetzt den Fall, man zieht aus einer vorliegenden Grundgesamtheit eine Zufallsstichprobe vorn Umfang 5Ø (beispielsweise werden 5Ø zufällig ausgewählte Schüler einer Schule nach ihrer Zufriedenheit mit den Lehrern befragt).

Man kann dann in einer solchen Stichprobe unterschiedliche statistische Maßzahlen berechnen, wie beispielsweise ein arithmetisches Mittel (Durchschnittsalter der Schüler oder dergl.).

Wenn man nun alle verschiedenen Zufallsstichproben vom Umfang n=5Ø ziehen würde, die überhaupt möglich sind (kein Mensch wird dies in der Praxis allerdings tun), dann erhält man ggf. sehr viele dieser Durchschnittswerte. Zeichnet man die Häufigkeitsverteilung all dieser Mittelwerte auf, so erhält man eine Verteilung, die sehr präzise dem Modell der Normalverteilung entspricht, unabhängig davon, wie die interessierende Variable (z.B. das Alter der Schüler) in der Grundgesamtheit verteilt ist, aus der die Stichproben gezogen wurden.

Auf diese Weise ist es möglich, wahrscheinlichkeitsbehaftete Aussagen über den Durchschnittswert in nur einer einzigen Zufallsstichprobe (und das ist die Praxis) zu machen.

Wir können dann errechnen, in welchen Grenzen der unbekannte Durchschnitt der Grundgesamtheit mit angebbarer Wahrscheinlichkeit (die unter Umständen recht hoch werden kann; z.B. 9Ø, 95 oder 99 %) liegen muß, wenn ein Stichprobenmittelwert in bestimmter Größe festgestellt worden ist.

Diese Wahrscheinlichkeitsberechnungen sind nur deshalb möglich, weil wir die Gültigkeit der Normalverteilung unterstellen können, wie es das Zentrale Grenzwerttheorem von LaPlace und Liapunoff bewiesen hat:

Parameter von Zufallsstichproben folgen approximativ der Gauß'schen Normalverteilung, unabhängig davon, wie die jeweilige Verteilung in der Grundgesamtheit aussieht.

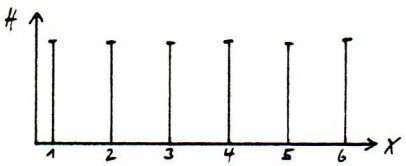
An einem einfachen Beispiel erweist das folgende Programm die Gültigkeit dieses Theorems. Es zeigt, daß Mittelwerte aus Zufallsstichproben approximativ normal verteilt sind, obwohl die Grundgesamtheit, aus der diese Stichproben gezogen werden alles andere als "normal" (im Sinne von Gauß) ist.

Zur Darstellung des Grenzwerttheorems lassen wir den Rechner 80 Zufallsstichproben, jeweils vom Umfang n = 50 ziehen. Die Untersuchungsvariable seien Würfelwürfe.

Das heißt also, wir lassen den Rechner 50 Mal würfeln und berechnen die durchschnittliche Augenzahl in dieser Stichprobe; danach würfelt er erneut 50 Mal und berechnet den Durchschnitt, würfelt erneut 50 Mal und berechnet den Durchschnitt, usw., usw., und das alles insgesamt 80 Mal.

Nach dieser Prozedur liegen 80 Durchschnitte vor, die ihrerseits in Form einer Häufigkeitsverteilung durch den Rechner dargestellt werden können.

Es wird sich bei der Ausgabe dieser Häufigkeitsverteilung zeigen, daß sie keineswegs mehr dem Modell einer Würfel-Augenzahlverteilung entspricht, das ja schematisch folgendermaßen aussehen müßte:



Im Gegenteil : Es wird sich eine Verteilung ergeben, die der Normalverteilung recht nahe kommt (übrigens ist die Annäherung an das Modell der Gauß'schen Normalverteilung gemäß des Zentralen Grenzwerttheorems um so besser, je größer die Zufallsstichproben sind).

Zusätzlich zur graphischen Veranschaulichung der entstehenden Häufigkeitsverteilung, berechnen wir auch die kennzeichnenden Parameter, nämlich das arithmetische Mittel und die Standardabweichung. Dabei gilt:

Arithmetisches Mittel $= \dfrac{\sum x_i}{n}$

(entsprechend kann auch ein Mittelwert aller Stichprobenmittelwerte definiert werden);

Standardabweichung $= \sqrt{\dfrac{1}{n}\sum (x_i - \bar{x})^2}$

(dabei ist x das jeweilige arithmetische Mittel); entsprechend kann auch eine Standardabweichung aller Stichprobenmittelwerte definiert werden).

Im einzelnen ist also folgendes zu tun:

1. In dem zu entwickelnden Programm sollen 8Ø Mal je 5Ø Würfelwürfe simuliert werden und aus jeder 5Øer Serie ist das arithmetische Mittel zu berechnen.

2. Zusätzlich wird der Gesamtmittelwert berechnet und darüber hinaus als Streuungsmaß für alle insgesamt 4ØØØ Würfelwürfe die Standardabweichung.

3. In einem dritten Schritt erstellen wir dann die Häufigkeitsverteilung der 8Ø erzeugten Stichprobenmittelwerte, indem die Mittelwerte in ein Klassifizierungsraster einsortiert werden, wobei die Klassen jeweils Ø.1 Einheiten breit sind (dies wird sich als zweckmäßige Klassenbreite erweisen).

4. Im Anschluß daran wird der Mittelwert und die Standardabweichung dieser Häufigkeitsverteilung bestimmt und diese beiden Maße werden mit den entsprechenden Werten, wie sie in Punkt 2. berechnet wurden, verglichen.

5. Schließlich wird die Häufigkeitsverteilung der Stichprobenmittelwerte auch der Anschaulichkeit halber graphisch ausgegeben.

Programm

```
10 REM Z15-ZGWT
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR ILLUSTRATION DES ZENTRA-"
40 PRINT "      LEN GRENZWERTTHEOREMS."
50 PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT :PRINT :PRINT
80 PRINT "DIESES PROGRAMM FUEHRT 80 SERIEN ZU"
90 PRINT "JE 50 WUERFELWUERFEN DURCH UND BE-"
100 PRINT "STIMMT DEN MITTELWERT IN JEDER SERIE."
110 PRINT :PRINT "UEBER DIESE MITTELWERTE WIRD DANN"
120 PRINT "EINE HAEUFIGKEITSVERTEILUNG ERSTELLT."
122 GOSUB 1100:REM WARTEN
125 N=80
130 DIM AM(N),KA(11),KE(11),H(11),W(N,50)
140 FOR I=1 TO N
150 S=0
160 FOR J=1 TO 50
170 W(I,J)=INT(RND(1)*6+1)
180 S=S+W(I,J):SS=SS+W(I,J)
190 NEXT J
200 AM(I)=S/50
210 NEXT I
```

```
215 MW=SS/(N*50):PRINT :MW=INT(MW*1000+0.5)/1000
216 PRINT "GESAMTMITTELWERT = ";MW
217 PRINT :GOSUB 1100
218 FOR I=1 TO N:FOR J=1 TO 50
219 S1=S1+(W(I,J)-MW)^2:NEXT J:NEXT I
223 S9=SQR(S1/(N*50))
224 S9=INT(S9*1000+0.5)/1000
225 PRINT :PRINT "STANDARDABWEICHUNG = ";S9
226 PRINT :PRINT :PRINT :PRINT :PRINT
227 REM HAEUFIGKEITSVERTEILUNG
230 GOSUB 1000:REM STOP
240 PRINT "MITTELWERT IN 50-ER","ANZAHL"
250 PRINT "       SERIEN"
260 FOR I=1 TO 36:PRINT "-";:NEXT I:PRINT
270 FOR K=1 TO 10
280 KA(K)=3+(K-1)*0.1
290 KE(K)=KA(K)+0.1
300 NEXT K
305 FOR K=1 TO 11:H(K)=0:NEXT K
310 FOR I=1 TO N
320 FOR K=1 TO 10
330 IF AM(I)<KE(K) THEN H(K)=H(K)+1:GOTO 350
340 NEXT K
345 H(11)=H(11)+1
350 NEXT I
360 FOR K=1 TO 10
370 PRINT KA(K);" BIS UNTER ";KE(K),,H(K)
380 NEXT K
```

```
385 PRINT "      UEBER   ";KE(10),,H(11)
390 REM MITTELWERT DER SPV
400 S=0
410 FOR J=1 TO N
420 S=S+AM(J)
430 NEXT J
440 A=S/N
445 PRINT :PRINT
450 PRINT "MITTELWERT DER VERTEILUNG : ";A
455 REM STREUUNG DER SPV
460 S=0
470 FOR J=1 TO N
480 S=S+(AM(J)-A)^2
490 NEXT J
500 SS=SQR(S/N)
502 SS=INT(SS*100+0.5)/100
505 PRINT :PRINT
510 PRINT "STREUUNG DER VERTEILUNG   : ";SS
515 GOSUB 1000:REM ABWARTEN
520 PRINT "GEGENUEBERSTELLUNG :":PRINT
530 PRINT ,,"WUERFEL","GAUSS":PRINT
540 MW=INT(MW*100+0.5)/100:A=INT(A*100+0.5)/100
550 S9=INT(S9*100+0.5)/100
560 PRINT "MITTELWERT",MW,A
570 PRINT :PRINT "STREUUNG",,S9,SS
580 GOSUB 1000:REM WARTEN
```

```
600 REM GRAPHIK
630 FOR K=1 TO 11
640 E=20:A=E-H(K):IF A<3 THEN A=3
645 IF A=E THEN 680
650 FOR J=A-1 TO E
660 POSITION 3*K,J:PRINT CHR$(160)
670 NEXT J
680 NEXT K
690 FOR I=1 TO 35:PRINT CHR$(18);:NEXT I
740 PRINT :PRINT "ENDE DER AUSGABE":END
1000 REM UP ABWARTEN
1020 PRINT "BITTE CONT EINGEBEN !":STOP
1030 PRINT CHR$(125)
1040 RETURN
1100 REM UP WARTEN
1110 PRINT :PRINT :PRINT
1115 PRINT "BITTE WARTEN, ICH RECHNE"
1120 PRINT :PRINT :RETURN
```

Variablenliste

A = Mittelwert aller Mittelwerte,
später Anfangsadresse zum Zeichnen
AM = Stichprobenmittelwert
E = Endadresse beim Zeichnen
H = Häufigkeit
I = Laufindex (Serie Nr.)
J = Laufindex (Wurf Nr.)
K = Laufindex (Klasse)
KA = Klassenanfangspunkt
KE = Klassenendpunkt
MW = Gesamtmittelwert
N = Zahl der Würfe
S = Seriensumme
SS = Gesamtsumme (Später Standardabweichung)
S1 = Quadratsumme
S9 = Standardabweichung
W = Würfelwurf

Programmbeschreibung

Satz 1Ø-12Ø : Überschrift und Erläuterungen.

Satz 122 : Warten (per Unterprogramm 11ØØ).

Satz 125-13Ø : Angabe der Zahl der zu ziehenden Stichproben und Dimensionierung der benötigten Variablenfelder (es wird dabei ein Stichprobenumfang von 5Ø unterstellt).

Satz 14Ø-21Ø : Erzeugung von N Stichproben zu je 5Ø Würfelwürfen:

15Ø: Belegung eines Summenfeldes S mit dem Wert Ø;

16Ø-19Ø: Es wird 5Ø Mal gewürfelt.

17Ø: Würfelwurf;

18Ø: Bildung der stichprobeninternen Augensumme S und der Gesamtsumme SS;

2ØØ: Bestimmung des Stichprobenmittelwertes AM.

Satz 215-216 : Errechnung des Gesamtmittelwertes MW und Ausgabe.

Satz 217 : Abwarten.

Satz 218-225 : Bildung der Gesamt-Standardabweichung als Maß der Streuung der Augenzahlen und Ausgabe dieses Maßes.

219: Addition aller quadrierten Abweichungen der Augenzahlen vom Gesamtmittelwert im Feld S1;

223: Berechnung der Standardabweichung;

225: Ausgabe des Ergebnisses.

Satz 227-230 : Warten.

Satz 240-260 : Ausgabe einer Tabellenüberschrift für die Häufigkeitsverteilung der Stichprobenmittelwerte.

Satz 270-300 : Bestimmung der Klassenuntergrenzen (KA) und der Klassenobergrenzen (KE) für die zu erzeugende Häufigkeitsverteilung (Klassenbreite = 0.1).

Satz 305 : Belegung der Häufigkeitsfelder mit 0.

Satz 31Ø-35Ø : Einordnen der Stichprobenmittelwerte in die Häufigkeitsklassen;

31Ø: Vorgabe eines Mittelwerts (von den insgesamt N Mittelwerten);

32Ø: Vorgabe einer Häufigkeitsklasse (von den insgesamt 1Ø Klassen);

33Ø: Ist der Mittelwert kleiner als die Klassenobergrenze KE, so ist er in diese Klasse einzuordnen (H=H+1); dann weiter bei 35Ø (nächster Mittelwert); wenn nicht, weiter bei 34Ø (nächste Klasse);

34Ø: nächste Klasse;

345: Besetzung einer Restklasse;

35Ø: nächster Mittelwert.

Satz 36Ø-385 : Ausgabe der Häufigkeitsverteilung.

Satz 39Ø-45Ø : Berechnung des Mittelwerts aller Stichprobenmittelwerte (= Mittelwert der sog. Stichprobenverteilung):

4ØØ: Summenfeld S = Ø;

41Ø-43Ø: Aufsummierung aller Stichprobenmittelwerte AM;

44Ø: Berechnung des Gesamtmittels A;

45Ø: Ausgabe von A.

Satz 455-51Ø : Berechnung der Streuung aller Stichprobenmittelwerte (= sog. Stichprobenfehler):

46Ø: Summenfeld S = Ø;

47Ø-49Ø: Bildung der Summe aller quadrierten Abweichungen zwischen Stichprobenmittelwerten und Gesamtmittelwert;

5ØØ: Berechnung der Standardabweichung (= Streuung der Stichprobenverteilung);

51Ø: Ausgabe.

Satz 515 : Warten.

Satz 52Ø-57Ø : Gegenüberstellung der einzelnen errechneten Maßzahlen.

Satz 58Ø : Warten.

Satz 6ØØ-7ØØ : Erzeugung des graphischen Abbilds der Häufigkeitsverteilung:

63Ø: Vorgabe einer Häufigkeitsklasse;

640: Bestimmung des Endpunktes und des Anfangspunktes des zu zeichnenden Balkens, der die jeweilige Häufigkeit repräsentieren soll;

645: Unterdrückung, falls H=0.

650-670: Zeichnen des Balkens.

680: Nächste Klasse.

Satz 740 : Beendigung des Programms.

Satz 1000-1040: Unterprogramm zum Abwarten.

Satz 1100-1120: Unterprogramm Wartemeldung.

Wenn dieses Programm gestartet wird, dann benötigt es zunächst ca. 3 Minuten, um das erste Zufallsergebnis zu produzieren:

GESAMTMITTELWERT = 3.455 (zufallsabhängig)

Nach weiteren 15 Minuten meldet der Rechner dann:

STANDARDABWEICHUNG = 1.676 (zufallsabhängig)

Bei einem völlig korrekten Würfel liegt übrigens der zu erwartende Mittelwert bei 3.5 und die zu erwartende Standardabweichung bei 1.7Ø8.

Danach ergibt sich die folgende Häufigkeitsverteilung, die wohlgemerkt ein Zufallsergebnis darstellt:

MITTELWERT IN 5Ø-ER SERIEN	ANZAHL
3 BIS UNTER 3.1	5
3.1 BIS UNTER 3.2	8
3.2 BIS UNTER 3.3	7
3.3 BIS UNTER 3.4	11
3.4 BIS UNTER 3.5	12
3.5 BIS UNTER 3.6	1Ø
3.6 BIS UNTER 3.7	13
3.7 BIS UNTER 3.8	9
3.8 BIS UNTER 3.9	2
3.9 BIS UNTER 4	2
UEBER 4	1

MITTELWERT DER VERTEILUNG : 3.45

STREUUNG DER VERTEILUNG : Ø.25

Weiterhin gibt der Rechner aus:

GEGENUEBERSTELLUNG :

	WUERFEL	GAUSS
MITTELWERT	3.46	3.45
STREUUNG	1.68	Ø.25

Wir erkennen, daß die Streuung der Stichprobenverteilung, die - wie schon erwähnt wurde - annäherungsweise der Gauß'schen Normalverteilung entspricht, sehr viel kleiner ist als die der ursprünglichen Verteilung. Darauf beruht letzten Endes die Präzision, die bei Hochrechnungen, ausgehend von Zufallsstichproben, erreicht werden kann.

Das Zentrale Grenzwerttheorem beweist u.a. die folgenden Beziehungen:

Mittelwert der Stichprobenverteilung	~	Mittelwert der Grundgesamtheit
3.45	~	3.46

Streuung der Stichprobenverteilung	~	Streuung der Grundgesamtheit dividiert durch die Wurzel aus dem Stichprobenumfang
Ø.25	~	$1.68/\sqrt{5Ø}$
		$= 1.68/7.Ø7...$
		$= Ø.248$

Schließlich produziert das Programm noch eine graphische Darstellung der oben vorgeführten Häufigkeitsverteilung für Stichprobenmittelwerte, wobei die einzelnen Häufigkeiten als Balken unterschiedlicher Länge dargestellt werden.

9.17 Wahlen

Problemstellung

Das Programm veranschaulicht graphisch in Form eines Balkendiagramms Wahlergebnisse.
Dazu muß der Benutzer die prozentualen Stimmenanteile für CDU, SPD, F.D.P., Grüne und Sonstige eingeben.

Problemanalyse

Eine Problemanalyse ist bei dieser recht einfachen Aufgabe wieder entbehrlich.

Programm

```
10 REM Z16-WAHLEN
20 PRINT CHR$(125)
30 PRINT "PROGRAMM ZUR GRAPHISCHEN DARSTELLUNG"
40 PRINT "          VON WAHLERGEBNISSEN."
50 PRINT :PRINT :PRINT :PRINT
60 PRINT "         PROF.DR.W.VOSS, 1984"
70 PRINT :PRINT :PRINT
75 DIM F(5),G(5):PRINT :PRINT
77 PRINT :PRINT :PRINT
80 PRINT "EINZUGEBEN SIND DIE WAHLERGEBNISSE (%)"
90 PRINT :PRINT
100 PRINT "CDU/CSU    :  ";:INPUT X:F(1)=X
110 PRINT "SPD        :  ";:INPUT X:F(2)=X
120 PRINT "F.D.P.     :  ";:INPUT X:F(3)=X
130 PRINT "GRUENE     :  ";:INPUT X:F(4)=X
140 PRINT "SONSTIGE   :  ";:INPUT X:F(5)=X
141 S=0
142 FOR I=1 TO 5:S=S+F(I):NEXT I
143 PRINT :PRINT :PRINT
144 IF S<>100 THEN PRINT "EINGABEFEHLER":GOTO 77
150 PRINT CHR$(125)
160 FM=0
180 FOR I=1 TO 5:IF F(I)>FM THEN FM=F(I)
190 NEXT I
200 FOR I=1 TO 5:G(I)=(13/FM)*F(I)
210 G(I)=INT(G(I)+0.5):NEXT I
220 POSITION 30,2:PRINT "CDU   ";F(1)
222 POSITION 30,3:PRINT "SPD   ";F(2)
224 POSITION 30,4:PRINT "FDP    ";F(3)
226 POSITION 30,5:PRINT "GRUENE ";F(4)
228 POSITION 30,6:PRINT "SONST. ";F(5)
230 GOSUB 2000:REM GRAPHIK
```

```
240 POSITION 1,17
245 FOR I=1 TO 30:PRINT CHR$(18);:NEXT I
246 PRINT
250 POSITION 1,20
260 PRINT "CDU   SPD   FDP   GRU   SONST"
280 PRINT :PRINT "ENDE":END
2000 REM UP GRAPHIK
2010 FOR J=1 TO 5
2020 B=17:A=17-G(J)
2030 FOR I=A TO B
2040 POSITION (J-1)*6+2,I:PRINT CHR$(160)
2050 POSITION (J-1)*6+3,I:PRINT CHR$(160)
2060 NEXT I
2070 NEXT J
2080 RETURN
```

Variablenliste

A = Startzeile der Zeichnung
B = Endzeile der Zeichnung
F = Prozentanteile
FM = Maximalwert
G = Transformierte Prozentanteile
I = Laufindex
J = Laufindex
S = Summenfeld
X = Feld zur Dateneingabe

Programmbeschreibung

Satz 10-70 : Überschrift und Erläuterungen.

Satz 75 : Dimensionierungen.

Satz 77-140 : Eingabe der Inputinformationen.

Satz 141-144 : Probe.

Satz 150 : Löschen des Bildschirms.

Satz 160-190 : Aufsuchen der maximalen Prozentangabe.

Satz 200-210 : Umrechnen der Prozentangaben derart, daß der Maximalwert den auf dem Bildschirm zur Verfügung stehenden Platz ausnutzt.

Satz 220-228 : Ausgabe der Informationen.

Satz 230 : Sprung ins UP 2000 zum Zeichnen.

Satz 240-260 : Ausgabe der Texte CDU, SPD, F.D.P., GRU, SONST und eines waagrechten Strichs.

Satz 280 : Ende des Hauptprogramms.

Satz 2000-2080: Unterprogramm zum Zeichnen der Balken.

2020: Bestimmung von Anfang- und Endadresse des Balkens.

2030-2060: Zeichnen von inversen Blanks an die per POSITION bestimmten Stellen.

2080: Rücksprung.

10

Kapitel 1Ø: Ausblick

In den vorangegangenen Kapiteln haben wir versucht,zu zeigen, wie mit Hilfe vergleichsweise einfacher BASIC-Programme typische Probleme, wie sie im Schulalltag auftreten, gelöst werden können.

Mit Nachdruck muß darauf hingewiesen werden, daß die ausgewählten Beispiele nur exemplarischen Charakter haben, und daß wir keinen Wert darauf gelegt haben, optimale Programme zu erstellen.

Der Leser sollte vielmehr in die Lage versetzt werden, die Funktionsweise der Programme rasch nachvollziehen zu können. Damit wird er dann in der Lage sein, auch für andere Probleme, die ja häufig den hier vorgestellten Beispielen ähneln werden, die entsprechenden Lösungsansätze zu finden.

Entscheidend beim Computereinsatz ist nämlich nicht der Rechner selbst, sondern die Fähigkeit des Benutzers, bestimmte Probleme so aufzubereiten, daß er dem Rechner den adäquaten Lösungsweg in Form eines Programms geben kann.

Wichtig ist nämlich nicht in erster Linie, daß ein Programm schon beim ersten Versuch fehlerfrei läuft (auch wenn dies natürlich sehr zu begrüßen und erfreulich ist), sondern daß der Computerbenutzer es lernt, wie ein gestelltes Problem gedanklich vorstrukturiert werden muß, damit es durch einen Rechner dann bearbeitet werden kann:

Ein Computer nützt uns nämlich nichts, wenn wir, die Benutzer, die jeweiligen Problemlösungswege nicht schon im Kopf haben.

Und wenn einmal ein Programm nicht so funktionieren sollte, wie der Benutzer das erwartet und zum Beispiel mit einer Fehlermeldung der Programmlauf abgebrochen wird, dann hilft immer noch ein Blick in das Benutzerhandbuch.

Für diejenigen Leser, denen die vorgestellten Beispiele zu einfach waren, sei darauf hingewiesen, daß dieses Buch in erster Linie für BASIC-Einsteiger gedacht war, die wir nicht schon durch die eventuelle Komplexität der Aufgabenstellungen überfordern wollten. Zu ihrem Trost aber verweisen wir auf einen in Bälde erscheinenden Fortsetzungsband zu diesem Buch, in dem dann insbesondere komplexere und vor allem mathematische Fragestellungen aufgegriffen werden sollen.

A Abbildung, optische 163
Abfrage 28
ABS 25
Addition 23
Anfangswert 32
Arbeitsschritt 41
Archimedes 151
Argument 24
Array 11
ASCII-Code 96
Ausgabe 12
Ausprägung 1Ø

B BASIC 7, 17
BASIC-Anweisung 95, 179
BASIC-Graphikanweisung 136
Betriebssystem 12
Bevölkerungsentwicklung 245
Bildschirm 13, 31
Bildschirmzone 2Ø
Binomialverteilung 347
Biologie 211
Bit 9
Blockgraphik 132, 158
BREAK-Taste 36
Bruttobetrag 8Ø
Buchstabe 9
Bundesländer 282
Byte 9

C Chemie 95
CHR$ 25, 36, 97
COLOR 134
Computerkunst 138, 143
CONT 36
COS 25, 3Ø9
Cosinus 3Ø9

D	DATA	99
	Datei	11
	Daten	8, 12
	Dateneingabe	99
	Datensatz	11
	Datenverarbeitung	8
	DIM	30
	Dimensionierung	100
	Diskette	13, 34
	Division	23
	DOS	35
	DRAWTO	135
	Dreieck, rechtwinkliges	42
	Dreisatz	81
	Dreisatzrechnung	80
	Drucker	13
E	E	74
	Eingabe	12
	Eingabefehler	40
	Element, chemisches	122
	Ellipse	316
	END	18
	Endlosschleife	36
	Endwert	32
	Englisch-Vokabeln	196
	Englisch-Vokabeltest	196
	Erdkunde	231
	Ergebnis	41
	Ergebnisausgabe	17
	Eva-Prinzip	42, 152
	EXP	25
	Exponentialdarstellung	92
F	Farbe	134
	Federwaage	144
	Feld	10
	File	11
	FILES	35
	Flußdiagramm	14, 41
	FOR ... TO	32
	Französisch-Vokabeln	189

F	Fremdsprache	179
	Funktion	24, 25
	Funktionsname	24
	Funktionsweise	12
G	Gauß	32Ø
	Geheimschrift	2Ø3
	Gerade	296
	Geschichte	231
	Gesetz, Ohm'sches	173
	GGT	5Ø
	Gitter	293
	Gleichung, quadratische	68
	GOSUB	18Ø
	GOTO	3Ø
	GRAPHICS	132
	Graphik	132, 293
	Graphik, hochauflösende	132
	Graphik-Programmierung	132
	Graphikzeichen	96
	Grenzwerttheorem	358
H	Häufigkeitsverteilung	273
	Hauptstädte	238
	Hypothek	262
	Hypothekentilgung	262
I	IF ... THEN	28
	Indizierung, doppelte	1Ø1
	Informationen	8
	Informationseingabe	26
	Inhaltsverzeichnis	35
	INPUT	26, 98
	INT	25
J	Jahreszahlen	232
	Jahreszahlen, historische	232
K	Kassette	13
	KGV	5Ø
	Kombinationen	336
	Kommando	12, 2Ø

K Korrektur 24
Kräfteparallelogramm 168
Kreis 299

L Laden 34
Länder 238, 282
Laufvariable 32
Leerzeile 18
LEN 2Ø4
LET 23, 98
Linse 163
LIST 2Ø
LOAD 34
LOG 25, 313
Logarithmus 313
Loop 32
Lotto 342

M Mathematik 39, 291
Menü 174
Menütechnik 174
Mittel, arithmetisches 269
Mittelwert 27, 269
Multiplikation 23

N Name 1Ø
Nation 245
Nettobetrag 8Ø
NEW 21
NEXT 32
Normalgraphik 132, 158
Normalverteilung 32Ø, 358

O Ökologie 211
OPEN 35

P Pendel 157
Pendelbewegung 157
Periodensystem 122
Permutationen 332
Physik 131
POSITION 21

P	Potenzierung	23
	Primzahl	59
	Primzahlprüfung	59
	PRINT	18
	Problemanalyse	14, 41
	Problemvorstellung	41
	Programm	41
	Programmanweisung	11, 12
	Programmbeschreibung	41
	Programmieren	16
	Programmschleife	32
	Programmverzweigung	28
	Prozentrechnung	80
	Punkt	297, 298
	Pythagoras	41
Q	Quadratzahl	306
R	READ	99
	Reaktion, chemische	109
	Reaktionsgleichung	109
	Rechengenauigkeit	92
	Rechenoperation	23
	Rechnen, stöchiometrisches	116
	Rechnung	18
	Regression	324
	REM	37
	RESET-Taste	35
	RESTORE	102
	RETURN	181
	RETURN-Taste	24
	RND	25, 85
	RUN	19
S	Satz	17
	Satznummer	28
	SAVE	34
	Schleife	32
	Schrittweite	32
	SHIFT	98
	SIN	25
	Sinus	303, 309

S	Sinuslinie	303
	Sonderzeichen	9
	Sortierprogramm	207
	Spaltenindex	101
	Speicher, externer	13, 34
	Speichern	34
	Speicherplatz	100
	Speicherstelle	10
	Sprache	179
	Sprung	28
	Sprung, bedingter	28
	Sprung, unbedingter	28
	SQR	25
	Statement	17
	STEP	33
	Steuerbetrag	80
	Stöchiometrie	116
	STOP	36
	Strich	136
	String	11, 18, 30
	Stringvariable	22, 30, 101
T	Tangens	309
	Tastatur	13, 96
	Textmodus	132
	Trend	325
U	Umweltverschmutzung	223
V	Variable	10, 22
	Variable, indizierte	100
	Variable, numerische	22
	Variablenliste	41
	Variablenname	18, 22
	Verarbeitung	12
	Verben	182
	Verben, englische	182
	Verben, unregelmäßige	182
	Verzweigung	28
	Vokabeln	189
	Vokabeltest	196

W Wachstum 212
Wachstum, gebremstes 217
Wachstum, ungebremstes 212
Wahlen 376
Wassermolekül 1Ø3
Wert 1Ø
Wertzuweisung 22
Wirtschaft 255
Wirtschaftskraft 282
Würfel 85
Wurzel 3Ø6

Z Zahl 18
Zahl, Euler'sche 74
Zeilenindex 1Ø1
Zentraleinheit 13
ZGWT 358
Ziffer 9
Zins 256
Zinsrechnung 256
Zufallszahl 85

DAS STEHT DRIN:

Besonders für Schüler der Mittel- und Oberstufe geschrieben, enthält das Buch viele Interessante Problemlösungs- und Lernprogramme, die besonders ausführlich und leicht verständlich beschrieben sind. Sie ermöglichen ein intensives Lernen am ATARI 600XL/ 800XL mit viel Spaß.

Aus dem Inhalt:

- Satz des Pythagoras
- Geometrische Reihen
- Geheimschrift
- Exponentielles Wachstum
- Unregelmäßige Verben
- Quadratische Gleichungen
- Pendelbewegungen
- Molekülbildung
- Vokabeln lernen
- Zinseszinsrechnung

Ein kurzer Überblick über die Grundlagen der EDV, eine knappe Wiederholung der wichtigsten BASIC-Elemente und eine Einführung in die Grundzüge der Problemanalyse vervollständigen das Ganze.

UND GESCHRIEBEN HAT DIESES BUCH:

Werner Voß ist Professor für Statistik an der Universität Bochum. Zahlreiche Veröffentlichungen im Bereich Statistik und Datenverarbeitung.

ISBN 3-89011-045-2

ATARI

Voß / Das Schulbuch zu ATARI ® 600XL / 800XL