

Michael Geipel

# NEUE LERNPROGRAMME

Für Commodore 64/128





Michael Geipel (Hrsg.)  
Neue Lernprogramme  
Für Commodore 64/128



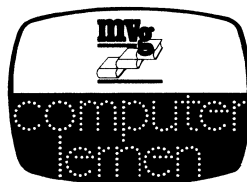
**Michael Geipel (Hrsg.)**

# **Neue Lernprogramme Für Commodore 64/128**

**7 dokumentierte Lernprogramme**

**Autoren:**

**Christoph Bronold · Gertrud und Michael Geipel ·  
Helmut Klatt · Manfred Raß · Claus Richter**



CIP-Kurztitelaufnahme der Deutschen Bibliothek

---

**Geipel, Michael:**

Neue Lernprogramme – Für Commodore 64/128 / Michael

Geipel (Hrsg.) – Landsberg am Lech: mvg verlag

(mvg-computer lernen)

[Textbd.] (1985).

ISBN 3-478-09270-1

---

© Alle Rechte bei mvg-verlag

8910 Landsberg am Lech

Umschlaggestaltung: Hendrik van Gemert

Satz: H. Buck, 8300 Kumhausen

Druck- und Bindearbeiten: Mühlthaler, München

Printed in Germany 090 270/985302

ISBN 3-478-09270-1 (Buch)

ISBN 3-478-99270-2 (Diskette)

# Inhalt

Vorwort .....	7
Einführung .....	9
Mathematik	
1 Mathematik-Trainer .....	12
2 Sinus und Kosinus .....	32
Sprachen	
3 Deutsche Literatur (Grundwissen) .....	47
4 Die Steigerung im Englischen .....	77
Sonstige Fächer	
5 Physik-Trainer .....	101
6 Grundlagen der Buchführung .....	113
7 Lernfiler .....	126
Anhang	
Hinweise zur Anwendung der Diskette .....	150





## Vorwort

»Neue Lernprogramme – Für Commodore 64/128« ist das dritte Buch aus der Reihe »Leichter lernen«. Wie die beiden Titel »Lernprogramme für den VC 20« und »Lernprogramme für den Commodore 64« enthält auch dieses Buch grundlegende Lernprogramme für Schüler der Sekundarstufe I.

Die neu entwickelten Programme wurden nicht nur optisch anspruchsvoller, sondern bieten auch mehr *Features* und ein größeres Anwendungsspektrum.

Damit wurden auch die *Listings* länger. Vom Leser, der die Programme selber in den Computer tippen möchte, werden Ausdauer, Konzentration und nicht zuletzt elementare BASIC-Kenntnisse verlangt.

Die Autoren unterstützen den *Do-it-yourself-Fan* durch zahlreiche Abbildungen und detaillierte Programm-Dokumentationen. Die *Listings* sind aus Gründen der Übersicht in Originalgröße gedruckt.

Ziel dieses Buches ist es,

- dem interessierten Schüler und Besitzer eines C 64/128 grundlegende Lernprogramme für den Schulalltag an die Hand zu geben,
- durch das Studium der Dokumentationen und die Eingabe der *Listings* die Programmierkenntnisse des Anwenders zu vertiefen,
- dem Jugendlichen bewußt zu machen, daß man mit Computern nicht nur Spiele konsumieren, sondern auch seine geistigen Fähigkeiten erweitern kann.

(Michael Geipel)



# Einführung

## Keine Angst vor dem Computer

Professor Konrad Zuse beantwortete einmal in einer Fernsehsendung die Frage, ob er sich vorstellen könne, daß viele Menschen Angst vor dem Computer haben, mit einem schlichten »Ja, natürlich«. Eine zunächst verblüffende Antwort des Mannes, der in Deutschland den programmgesteuerten Computer erfand. Von guter Menschenkenntnis zeugt die Begründung seiner Antwort. Sinngemäß sagte er: Der Mensch hat immer Angst vor dem Unbekannten. Wenn er aufgrund fehlender Kenntnisse die Folgen einer Sache nicht einschätzen kann, fühlt er sich unsicher. Unsicherheit aber erzeugt Angst.

Mit diesem neuen Unterrichtsmittel sollte man sich vertraut machen. Viele betrachten einen Computer auch heute noch als »Rechner«. Ein moderner Computer kann aber viel mehr als nur rechnen.

## Können Computer denken?

Die volkstümliche Bezeichnung »Elektronengehirn« verführt zum Vergleich mit dem menschlichen Gehirn. Doch die technikgläubigen 60er Jahre, in denen ein Fachmann für Computerübersetzungen noch selbstbewußt verkündete: »If a human being can do it, a suitably programmed computer can do it, too«\*, sind lange vorbei. Die Ursachen für die Ernüchterung, zumindest unter denen, die sich mit computerunterstütztem Lernen befaßten, waren die geringe Speicherkapazität und die gewaltigen Kosten der damaligen EDV-Anlagen.

Während sich die Sprachforschung des 19. Jahrhunderts noch vornehmlich auf den Sprachinhalt konzentrierte, vollzog sich unter dem Einfluß des Computers im 20. Jahrhundert fast unmerklich der Wandel vom inhaltlichen zum formalen Denken. Dies übrigens nicht nur im Bereich der Sprachwissenschaften, sondern nahezu in allen Bereichen des modernen Lebens.

Der wohl bekannteste Sprachforscher auf diesem Gebiet, Noam Chomsky, setzte die strukturelle Sprachwissenschaft, als deren Schöpfer Ferdinand de Saussure gilt, erfolgreich fort und erzielte eindrucksvolle Ergebnisse. So ist es heute möglich, rein formal durch Austausch von Elementen, gemäß der von ihm gefundenen Strukturregeln, mit dem Computer Sätze zu erzeugen. Diese sind grammatikalisch richtig, aber leider inhaltlich nicht immer sinnvoll. Dies ist die Barriere, die bis heute nicht überwunden werden konnte. Inhaltliches Denken ist dem Menschen vorbehalten. Im formalen »Denken« jedoch ist uns der Computer weit überlegen. Diese Überlegenheit sollten wir nutzen.

---

\* »Wenn es der Mensch kann, dann kann es auch ein entsprechend programmierter Computer.«

## In den »Niederungen« des Schulalltags

Begeben wir uns also bescheiden in die »Niederungen« des Schulalltags. Ein nicht geringer Teil des Unterrichts, insbesondere in den unteren Klassen, bewegt sich auf der formalen Ebene:

- Aneignung von Faktenwissen
- Erweiterung des Wortschatzes
- Verbesserung der Orthographiekenntnisse
- Einüben von Grammatikregeln und ihre Anwendung
- Erweiterung des Wortschatzes

Auch viele Tests enthalten formalistische Bestandteile:

- Falsch-Richtig-Entscheidungen
- Mehrfachauswahl-Antworten
- Einsetzübungen und Lückentexte

Reproduktiv »denken«, d.h. Informationen speichern, ordnen oder zufallsgesteuert und zeitverschoben, eventuell auch anders strukturiert wiedergeben, kann der Computer sicherlich besser als der Schüler, und so kann er ihn bei seiner Lernarbeit unterstützen. Wohlgermerkt, der Computer soll dem Schüler helfen, schneller zu lernen. Er soll nicht das Lernen ersetzen!

## Das Listing auf dem Weg in den C 64 oder PC 128

Nach zahlreichen Programmtests wurden die *Listings* auf einem Epson FX-80 ausgedruckt und auf fotomechanischem Wege in diesem Buch in Originalgröße reproduziert. Dies bietet optimale Gewähr, daß die Programme, vorausgesetzt, sie werden vom Anwender fehlerfrei abgetippt, auch laufen.

Achten Sie beim Eintippen bitte unbedingt darauf, daß die Programmzeilen auch wirklich in allen Einzelheiten mit der Vorlage übereinstimmen. Um diese Arbeit etwas zu erleichtern, wurden die *Listings* mit 40 Zeichen pro Zeile wiedergegeben und stimmen somit mit der Darstellung auf dem Bildschirm überein.

Eine Fehlerfalle sind erfahrungsgemäß die commodore-typischen Steuer- und Grafikzeichen. Aus diesem Grunde wurden sie, soweit sinnvoll, durch *Characterstrings* ersetzt.

Finden Sie eines der folgenden Zeichen in einem Listing, so geben Sie bitte die in Klammern gesetzte Tastenkombination ein.

<b>☐</b>	= Im Kleinschriftmodus RVS einschalten (CTRL + 9)
<b>☐</b>	= Im Kleinschriftmodus RVS ausschalten (CTRL + 0)
<b>☐</b>	= Grafikzeichen CHR\$(221) (SHIFT+ -)
<b>\$</b>	= Grafikzeichen CHR\$(192) (SHIFT+ *)
<b>β</b> oder <b>^</b>	= Exponentialpfeil (↑)

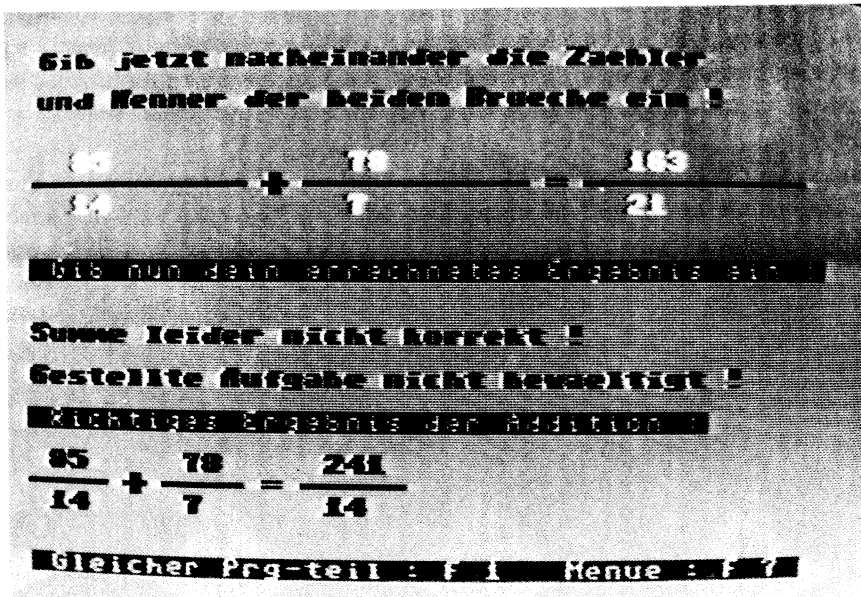
Testen Sie bereits einzelne Programmabschnitte und speichern Sie diese auf Diskette ab. Bevor die Tipp-Lust zum Tipp-Frust wird, sollte man unbedingt die lauffertige Diskette zu diesem Buch erwerben.

Einige Programme arbeiten mit Sonderzeichen. Um unerwünschte Nebenwirkungen zu vermeiden, sollten Sie den Computer in der Grundversion laufen lassen. Entfernen Sie also gegebenenfalls Steckmodule, Programmierhilfen und sonstiges. Weiter hat es sich als zweckmäßig erwiesen, das Gerät kurz aus- und wieder einzuschalten, wenn man nach einem Sonderzeichenprogramm mit anderen Programmen weiterarbeiten möchte.

Besitzen Sie einen Commodore PC 128, so fügen Sie den Programmen in diesem Buch lediglich die Zeile **1 GO 64** hinzu. Der Befehl **GO 64** kann natürlich auch im Direktmodus eingegeben werden.

(Michael Geipel)

# 1 Mathematik-Trainer



## Programm-Telegramm

- Drei Sachgebiete in einem Programm.
- Bruchrechnung in allen vier Grundrechenarten.
- ggT und kgV zur Findung der optimalen Kürzungszahl bzw. des Hauptnenners.
- Primfaktorenzerlegung zur Berechnung des ggT und kgV in schwierigen Fällen, zur Primzahlprüfung und Teilerbestimmung.
- Zufallsgesteuerte Aufgabenstellung mit Lösungsanzeige nach dem 3. Versuch.

## Sinn und Zweck des Programms

### Einsatzmöglichkeiten

Der »Mathematik-Trainer« ist ein Programm, das für Schüler ab Klasse 5 geeignet ist. Es vereinigt in sich drei thematisch eng zusammengehörende mathematische Sachgebiete: Bruchrechnung, ggT und kgV, Primfaktorenzerlegung.

Die Bruchrechnung – und das trifft besonders für ihre mannigfaltigen Anwendungen zu – gilt unter vielen Schülern als unbeliebter, weil wenig überschaubarer mathe-

matischer Unterrichtsinhalt. Ein wichtiger Grund ist sicherlich darin zu sehen, daß der Übergang von den natürlichen Zahlen zu den Bruchzahlen für die Lernenden in der Regel die erste (und vielleicht schwierigste) Zahlbereichserweiterung darstellt. Der Mathematikunterricht kann mit seinen durchschnittlich vier Wochenstunden in bezug auf die Vermittlung einer sicheren Beherrschung der Bruchrechnung nur Begrenztes leisten. Das hat zur Folge, daß einer Reihe von Jugendlichen fundierte Regelkenntnisse ebenso fehlen wie zufriedenstellende Rechenfertigkeit. Mangelnde Motivation führt dann dazu, daß aufgetretene Wissenslücken trotz schulischer Bemühungen nicht immer geschlossen werden können. Zwar wird man in der (beruflichen) Praxis in aller Regel nicht mit Brüchen, sondern mit Dezimalzahlen konfrontiert; ein sicherer Umgang mit den letzteren verlangt jedoch Grundkenntnisse über den Umgang mit den ersteren. Genau hier nun setzen Sinn und Zweck des vorliegenden Programms ein. Dieses geht davon aus, daß dem Üben das Regelwerk der Verknüpfungen von Bruchzahlen im großen und ganzen bekannt ist, es ihm aber an der notwendigen Sicherheit und Gewandtheit im Umgang mit Bruchzahlen mangelt.

Bei der Konzeption des Programms stellte sich nun die Frage, ob es zur Verwirklichung des angestrebten Ziels vorteilhafter sei, vom Computer (per RND-Funktion) entsprechende Aufgaben erzeugen oder die zur Bearbeitung vorliegenden Aufgaben vom Programmbenutzer selbst eingeben zu lassen. Vornehmlich aus zwei Gründen wurde mit diesem Programm die zweite Möglichkeit realisiert. Zum einen gibt es schon eine Reihe von Programmen, bei denen der Computer die Aufgaben »stellt« und der Anwender (ohne Programmänderung) meist nur geringfügigen Einfluß auf die Art der zu lösenden Aufgaben hat, zum anderen bietet die hier gewählte Vorgehensweise dem Lernenden wahrscheinlich die effizientere Lernhilfe an, weil sie eine auf die konkret vorliegende Aufgabe bezogene Hilfestellung leistet. Setzt man beim Üben ein gewisses Maß an Selbstdisziplin voraus, kann ein solches Programm etwa bei der Erledigung von Hausaufgaben oder sonstiger Übungen eine Aufsichtsperson überflüssig machen.

Unter schulischem Blickwinkel betrachtet, muß wohl als wichtigster Programmteil der erste angesehen werden; die beiden anderen sind diesem dienend zugeordnet, obwohl sie ebenfalls in sich selbständige Themen darstellen und mit ihnen auch ohne ständigen Blick auf die Bruchrechnung vielfältige Übungen möglich sind.

### **Programmteil »Bruchrechnung«**

Im Programmteil »Bruchrechnung« stehen dem Üben sämtliche vier Grundrechenarten zur Verfügung. Auf eine weitere, häufig vorkommende Verknüpfung, das Potenzieren von Bruchzahlen, wurde verzichtet, weil eine Potenz – sofern die Hochzahl eine natürliche Zahl darstellt – als Produkt mit gleichen Faktoren geschrieben werden kann. Dabei ist die Basis (Grundzahl) der mehrfach auftretende Faktor; der Exponent gibt dann an, wie oft der gleiche Faktor auftritt. Aus Gründen der Übersichtlichkeit erlaubt das Programm immer nur die Verknüpfung zweier Bruchzahlen. Dieses stellt jedoch keine Einschränkung dar, weil sich auch komplexe Bruchterme als Folge von

einfachen Rechenschritten darstellen lassen, bei denen unter Beachtung entsprechender Termumformungsregeln stets genau zwei Bruchzahlen durch eine Bruchzahl (Teilergebnis) ersetzt werden können.

### Programmteil »ggT und kgV«

Mit Hilfe von Programmteil »ggT und kgV« kann der Lernende überprüfen, ob er aus einer beliebigen Anzahl natürlicher Zahlen deren größten gemeinsamen Teiler (ggT) und deren kleinstes gemeinsames Vielfaches (kgV) herausfinden konnte. Die Bestimmung dieser Zahlen ist kein Selbstzweck, sie findet vielmehr ihre Anwendung in zwei speziellen Fragestellungen der Bruchrechnung.

a) Die Frage nach dem ggT zweier natürlicher Zahlen ist die Frage nach der optimalen Kürzungszahl, der größten Zahl also, durch die sich eine Bruchzahl kürzen läßt. Sollte die Antwort »1« lauten, so ist der entsprechende Bruch eben nicht mehr »echt« kürzbar.

b) Die Frage nach dem kgV zweier oder mehrerer natürlicher Zahlen ist die Frage nach dem sogenannten »Hauptnenner«, der kleinsten Zahl also, die ein Vielfaches sämtlicher auftretender Nenner darstellt. Sie stellt sich beim Größenvergleich sowie bei der Addition und Subtraktion von Bruchzahlen.

### Programmteil »Primfaktorenzerlegung«

Der Programmteil »Primfaktorenzerlegung« schließlich kann dreifach genutzt werden.

#### a) ggT und kgV in schwierigeren Fällen berechnen

Mit Hilfe der Primfaktorenzerlegung kann man in den Fällen, in denen ggT und kgV mehrerer Zahlen nicht auf Anhieb erkennbar sind, (relativ) bequem die beiden gesuchten Werte finden. Das Verfahren dazu findet sich in entsprechenden Büchern (Schulbuch, Mathematik-Lexikon u.a.).

#### b) prüfen, ob eine natürliche Zahl Primzahl ist

Der Übende kann leicht feststellen, ob eine natürliche Zahl überhaupt in ein Produkt mit Primzahlen als Faktoren zerlegbar ist; läßt sich nämlich eine solche Zerlegung nicht finden, dann handelt es sich um eine Primzahl.

*Anmerkung:*

Jede natürliche Zahl mit genau zwei Teilern heißt Primzahl.

#### c) alle Teiler einer natürlichen Zahl bestimmen

Besteht die Primfaktorenzerlegung einer natürlichen Zahl aus höchstens vier verschiedenen Faktoren, so lassen sich durch Kombination aller möglichen (Teil-)Produkte und durch Hinzufügen des trivialen Teilers »1« sämtliche Teiler der zerlegten Zahl finden.

Beispiel:

$$210 = 2 * 3 * 5 * 7$$

Kombinationen:

$$1, 2, 3, 5, 7, 2*3, 2*5, 2*7, 3*5, 3*7, 5*7, \\ 2*3*5, 2*3*7, 2*5*7, 3*5*7, 2*3*5*7$$

Teiler von 210:

$$1, 2, 3, 5, 7, 6, 10, 14, 15, 21, 35, 30, 42, 70, 105, 210$$



Sind mehrere Faktoren gleich, so verringert sich die Anzahl der möglichen Kombinationen. Bei mehr als vier unterschiedlichen Faktoren wird durch die Vielzahl der zu bildenden (Teil-)Produkte das Verfahren unpraktisch, weil unübersichtlich.

## Programmablauf

### Blick in das Menü

Nach dem Starten des Programms wird dem Benutzer auf dem Bildschirm angezeigt, welche mathematischen Sachgebiete er üben kann. Ein blinkender *Cursor* fordert ihn auf, sich durch Eingabe einer Ziffer für »Bruchrechnung« (=1), »ggT und kgV« (=2) oder »Primfaktorenzerlegung« (=3) zu entscheiden.

In allen drei Programmteilen stehen dem Übenden drei Versuche zur Verfügung, um die gesuchte(n) Lösung(en) zu bestimmen. Erst nach dem dritten vergeblichen Versuch erhält er die korrekten Zahlenwerte auf dem Bildschirm angezeigt. War der Anwender im ersten oder zweiten Versuch erfolgreich, so erfährt er Lob vom Computer und kann dann mittels *f1* oder *f7* entscheiden, ob er im gleichen Programmteil weiterarbeiten oder lieber zum Menü zurückkehren möchte. Alle Eingaben des Benutzers erscheinen in weißer Schrift, damit sie sich vom übrigen Text (hellblau auf dunkelblauem Grund) farblich bzw. vom Kontrast her deutlich unterscheiden. Die Aufmerksamkeit des Übenden soll dadurch besonders auf die eigenen Eingaben gelenkt werden. Diverse Eingabeprüfungen sorgen in diesem Zusammenhang dafür, daß keine sinnlosen Werte zur Berechnung kommen.

### 1 Bruchrechnung

Ist der Lernende durch »1« in den Programmteil »Bruchrechnung« gelangt, kann er durch Angabe eines Zeichens (»+« für Addition, »-« für Subtraktion, »\*« für Multiplikation, »/« für Division) eine der vier Grundrechenarten auswählen. Er muß sodann die Zähler und Nenner der beiden (durch die gewünschte Rechenoperation zu verknüpfenden) Brüche sowie die des Ergebnisses eingeben. Die *Cursor*position zeigt ihm dabei stets unmißverständlich, welche Eingabe gerade gefordert ist. Das Programm akzeptiert hier nur maximal neunstellige ganzzahlige Werte mit oder ohne Vorzeichen. Sollte die als Lösung eingegebene Bruchzahl vom Wert her richtig, jedoch noch nicht gekürzt sein, so wird die vollständig gekürzte Darstellung des Bruches angezeigt und die notwendige Kürzungszahl genannt. Bei ganzzahligen Lösungen wird das Ergebnis in Bruchform (Nenner=1) und in herkömmlicher Form (nur die ganze Zahl) auf den Bildschirm gebracht. Überhaupt wird zur Darstellung der korrekten Lösung immer die Form der Gleichung gewählt, wie sie auch in der Schule im Mathematikunterricht in der Regel Verwendung findet.

## 2 ggT und kgV

Hat sich der Lernende durch »2« für »ggT und kgV« entschieden, so wird er aufgefordert, einzeln nacheinander alle natürlichen Zahlen einzugeben, von denen er den größten gemeinsamen Teiler (ggT) sowie das kleinste gemeinsame Vielfache (kgV) bestimmen möchte. Mit Ausnahme von »←« (Pfeil nach links), mit dem die Eingabe der Zahlen abgeschlossen wird, nimmt das Programm nur maximal zwölfstellige natürliche Zahlen an. Der Anwender muß nun seine errechneten Werte für ggT und kgV eintippen. Er erfährt dann, ob und welche Werte korrekt bestimmt wurden und kann dann möglicherweise weitere Versuche unternehmen. Nur der falsche Wert wird jeweils gelöscht, und es muß ein neuer eingegeben werden; der richtig bestimmte wird durch Bestätigung mittels **RETURN** übernommen.

## 3 Primfaktorenzerlegung

Mit »3« schließlich begibt sich der Lernende in den Programmteil »Primfaktorenzerlegung«. Nach Eingabe der zu zerlegenden natürlichen Zahl (maximal zwölfstellig) stellt sich die Aufgabe, die selbst vorgegebene Zahl als Produkt zu formulieren, bei dem sämtliche Faktoren Primzahlen sind. Außer Ziffern läßt die Eingabeprüfung nur »\*« und »Space« (= Leertaste) passieren. Sollte der Benutzer eine Zerlegung finden, die zwar die vorgegebene Zahl ergibt, jedoch den Mangel hat, daß nicht alle Faktoren Primzahlen darstellen, so wird er durch eine entsprechende Meldung darauf aufmerksam gemacht und gegebenenfalls zu einem weiteren Versuch aufgefordert. Ergibt die eingegebene Zerlegung eine andere Zahl als die zu zerlegende, so wird die entsprechende Zahl zusammen mit der eingetippten Zerlegung als Gleichung auf dem Bildschirm angezeigt. War die anfangs vorgegebene Zahl eine Primzahl, so erfolgt eine entsprechende Meldung; eine Primfaktorenzerlegung ist dann selbstverständlich nicht möglich.

## Programmbeschreibung

### Die vier Programmteile

Da das Programmlisting sehr viele REM-Anweisungen enthält, sich dadurch gewissermaßen selbst dokumentiert, kann sich die Erläuterung auf einige wenige Punkte beschränken.

a) Das Programm besteht aus vier Teilen:

Zeilen 10 – 770: Initialisierung

Zeilen 780 – 1800: Bruchrechnung

Zeilen 1810 – 2530: ggT und kgV

Zeilen 2540 – 3430: Primfaktorenzerlegung

Sollten Sie nicht alle drei Programmteile benötigen, so können Sie folgendermaßen verfahren:

Sie geben die Zeilen 10 – 770 ein und zwar unabhängig davon, welchen Programmteil Sie zunächst als ersten benötigen.

Sie geben die entsprechenden Zeilen des ausgewählten Programmteils ein.

Beachten Sie bitte, daß die Zeilen 880, 1900, 2760 auf alle Fälle im Programm enthalten sein müssen, da sie durch die ON . . . GOTO . . . -Anweisung in Zeile 540 möglicherweise angesprungen werden. Fehlt Ihnen beispielsweise noch der letzte Programmteil, so leistet »2760 RUN« Abhilfe.

### »Wahr« oder »falsch«?

b) An mehreren Stellen im Programm tauchen Befehlsfolgen mit folgender Struktur auf:

”ON –(logischer ausdruck) GOTO zeilennummer x : GOTO zeilennummer y”  
Diese Anweisung gleicht einen Mangel des Commodore 64 Basic V2 aus, nämlich keine IF . . . THEN . . . ELSE . . . -Anweisung zu besitzen. Die aufgeführte Anweisungsfolge basiert auf folgender Überlegung:

Ein logischer Ausdruck ist entweder »wahr« oder »falsch«. Der Computer ordnet nun dem Wahrheitswert des Ausdrucks zwei Integer-Zahlen zu: ”0” für »falsch« und ”-1” für »wahr«. Ist in unserem Fall der logische Ausdruck »wahr«, so ergibt ” –(logischer ausdruck)” den Wert ” –(-1)” also ”1”. Das Programm wird dann in der Zeile x fortgesetzt. Ansonsten jedoch wird die gesamte ON . . . GOTO . . . -Anweisung ignoriert und die dahinterliegende Anweisung ausgeführt, das Programm also in der Zeile y fortgesetzt.

Beispiel: Statt zwei Zeilen zu verwenden

” 100 if fl=1 then 500 ”

” 110 goto 600 ”

kann man in einer Zeile schreiben

” 100 on –(fl=1) goto 500 : goto 600 ”

### Experimente?

c) Wollen Sie mit dem Programm experimentieren, so können Sie unbeschadet seiner grundlegenden Funktionen an folgenden Stellen Änderungen vornehmen:

Zeilen 110/120:

Farbgebung

Zeile 160:

Dimensionierung der Felder

Zeilen 820/1850/2580/2640:

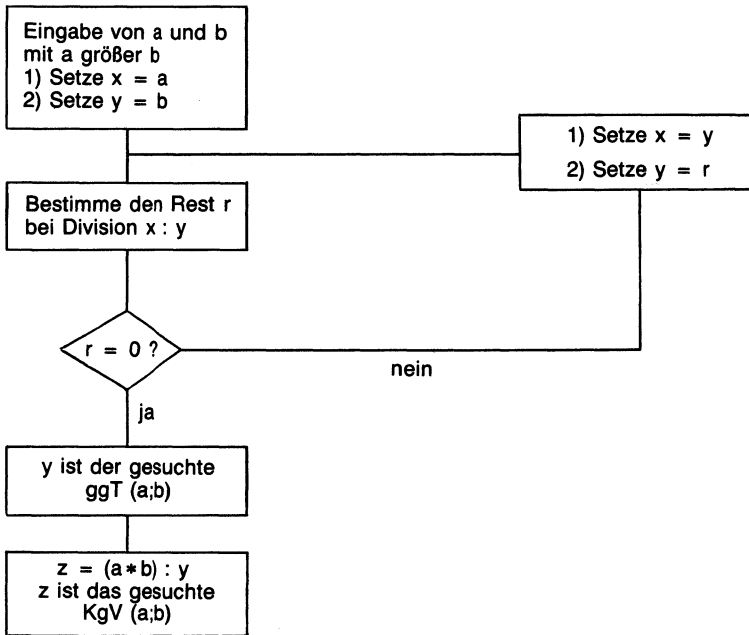
erlaubte Stringlängen bei Eingabe

### Der Euklidische Divisionsalgorithmus

d) In den Zeilen 1410/1420 und 2140/2150 sowie 2190/2200 tritt der sogenannte »Euklidische Divisionsalgorithmus« auf. Es handelt sich dabei um ein Verfahren, mit dem man auf einfache Weise den ggT zweier natürlicher Zahlen  $a$  und  $b$  bestimmen kann. Hat man diesen gefunden, so ergibt sich das kgV als Quotient aus dem Produkt der beiden natürlichen Zahlen und dem errechneten ggT. Der Vorteil dieses Verfahrens zeigt sich besonders bei sehr großen Zahlen, bei denen die Primfaktorenzerlegung schwierig herzuleiten ist.

Den mathematischen Hintergrund des Verfahrens, das zuweilen auch »Wechselwegnahme« genannt wird, können Sie aus nachstehendem Flußdiagramm ersehen.

### Flußdiagramm zum euklidischen Divisionsalgorithmus



Beispiel: gesucht ggT/kgV von 128 und 48

$$128 = 48 * 2 + 32$$

$$48 = 32 * 1 + 16$$

$$32 = 16 * 2 + 0$$

Ergebnis : 16 = ggT(128;48)

$$128 * 48 = 6144$$

$$6144 = 16 * 384$$

Ergebnis : 384 = kgV(128;48)

## Liste der wichtigsten Variablen

i	Laufvariable
a\$(i)	situationsbezogene Antworten
z1\$,z3\$	obere/untere Begrenzung des Menüs
z2\$,s\$	linke/rechte Begrenzung des Menüs
z4\$	Leerstring zum Löschen von Teilen des Bildschirms
v\$	Bruchstrich (Aufgabenstellung)
b1\$	Bruchstrich (erster Bruch)
b2\$	Bruchstrich (zweiter Bruch)
b3\$	Bruchstrich (Ergebnis)
z\$	Eingabe
c1	Cursorzeile
c2	Cursorspalte
c3	Länge des zu löschenden Strings auf dem Bildschirm
o	Anzahl der zu löschenden Zeilen
fl	Flagge (gelöscht = 0, gesetzt = 1)
p\$,d\$	Strings zur Überprüfung der Eingabe
f	Anzahl der Fehler
m\$	Rechenzeichen
n\$	Art des Verknüpfungsergebnisses
o\$	Name der Verknüpfung
z1,n1	Zähler/Nenner des ersten Bruches
z2,n2	Zähler/Nenner des zweiten Bruches
a1,a2	Zähler/Nenner der vom Benutzer ermittelten Ergebnisbruchzahl
z,n	Zähler/Nenner der ungekürzten Ergebnisbruchzahl
k,l	Zähler/Nenner der gekürzten Ergebnisbruchzahl
a	Kürzungszahl
p(i)	natürliche Zahlen
g,k	ggT, kgV der Zahlen
gp,kp	vom Benutzer ermittelte Werte für ggT und kgV
x	zu zerlegende natürliche Zahl
a\$	vom Benutzer ermittelte Primfaktorenzerlegung
r(i)	Primfaktoren der korrekten Zerlegung
t\$(i)	ingegebene Faktoren
pr	Produkt der Faktoren

(Manfred Raiß)

```

10 rem "*****"
15 rem "*"
20 rem "* Mathematik - Trainer  *"
25 rem "*"
30 rem "* 1. Bruchrechnung  *"
35 rem "* 2. 'ggT' und 'kgV'  *"
40 rem "* 3. Primfaktorenzerlegung  *"
45 rem "*"
50 rem "*****"
60 :
65 rem "Copyright 1985 by          "
70 rem "moderne verlags gesellschaft  "
75 :
80 rem "Autor: Manfred Raiss          "
85 :
100 rem ***** "Initialisierung" *****
*
110 poke53280,6:poke53281,6: rem "Rahmen
farbe , Bildschirmfarbe"
120 ef=1:cf=13:bf=14 :rem "Cursorfarbe
bei Eingabe/Text/Programmfortsetzung"
130 poke53272,23:poke646,cf: rem "Gross/
klein-Modus , Cursorfarbe"
140 cz=214:cs=211: rem "Zeropageadressen
fuer Cursor (Zeile/Spalte)"
150 mp=58732: rem "Cursor an die in 214/
211 hinterlegte Bildschirmposition"
160 dima$(18),p$(40),t$(30),p(30),q(30),
r(50)
170 rem chr$(145): "Cursor nach oben"
180 rem chr$(18) : "Revers-Modus ein"
190 rem chr$(146): "Revers-Modus aus"
200 rem chr$(147): "Bildschirm loeschen"
210 for i=0 to 18:read a$(i):next i
220 data" Programmfortsetzung : beliebig
e Taste "
230 data"Ergebnis stimmt ! Mach weiter s
o !"
240 data" leider nicht korrekt !","Versu
ch es ein zweitesmal !"
250 data"Unternimm noch einen dritten Ve
rsuch !"
260 data"Ergebnis zutreffend, aber noch
kuerzen !"
270 data"Notwendige Kuerzungszahl ist"

```

```

280 data"Gestellte Aufgabe nicht bewael
igt !"
290 data"gut ! ggT und kgV korrekt ermit
telt !"
300 data"ggT und kgV nicht zutreffend be
stimmt !"
310 data"ggT gefunden, kgV leider falsch
!"
320 data"kgV gefunden, ggT leider falsch
!"
330 data"ist eine Primzahl, daher","kein
e Primfaktorenzerlegung moeglich !"
340 data"Primfaktorenzerlegung gelungen
!"
350 data"Faktorenzerlegung ergibt andere
Zahl !"
360 data"Zerlegung ergibt zwar die vorge
gebene"
370 data"Zahl, aber nicht alle Faktoren
stellen"
380 data"wie gefordert Primzahlen dar !"
390 rem ** "benoetigte Strings erzeugen"
400 fori=1to38:z0$=z0$+chr$(192):nexti:s
$=chr$(221):v$=left$(z0$,11)
410 z1$=chr$(176)+z0$+chr$(174):z3$=chr$
(173)+z0$+chr$(189)
420 fori=1to40:z4$=z4$+chr$(32):nexti:z2
$=s$+left$(z4$,38)+s$
430 rem ***** "Menue anzeigen" *****
440 printchr$(147);z1$;z2$;s$;" Mit di
esem Programm kannst Du 3      ";s$;z2$;
450 prints$;" thematisch eng zusammeng
ehoerende ";s$;z2$;
460 prints$;" mathematische Sachgebiet
e ueben. ";s$;z2$;z2$;z2$;
470 prints$;" Bruchrechnung
"chr$(18)" 1 "chr$(146);" ";s$;
480 printz2$;s$;" 'ggT' und 'kgV'
"chr$(18)" 2 "chr$(146);" ";
490 prints$;z2$;s$;" Primfaktorenzerle
gung "chr$(18)" 3 "chr$(146);
500 print" ";s$;z2$;z3$;z1$;z2$;s$;"
Was moechtest Du ueben ? ";
510 prints$;z2$;s$;" Gib die entsprech
ende Zahl ein ! ";s$;z2$;z3$;

```

```
520 c1=19:c2=34:gosub600:a=val(left$(right$(z$,6),1))
530 ifa<>1anda<>2anda<>3thenc3=5:gosub640:goto520
540 onagoto880,1900,2760
550 :
560 rem *** "Subroutinen, die" ***
570 rem *** "von allen Programm-" ***
580 rem *** "teilen benutzt werden" ***
590 :
600 rem **** "Cursorpositionierung" ***
610 rem **** "und Stringeingabe" ****
620 z$="":pokec,c1:pokec,c2:sysmp:poke646,ef:open1,0:input#1,z$:close1
630 poke646,cf:return
640 rem ***** "String loeschen" *****
650 pokec,c1:pokec,c2:sysmp:printleft$(z4$,c3):return
660 rem ***** "Zeilen loeschen" *****
670 pokec,c1:pokec,c2:sysmp:forx=1to0:printz4$:nextx:return
680 rem **** "Programmfortsetzung" ****
690 pokec,24:pokec,0:sysmp:poke646,bf:printchr$(18)a$(0)chr$(146);
700 geta$:on-(a$="")goto700:poke646,cf:printchr$(145)z4$:return
710 rem ** "Auswahl mittels F-Tasten" **
720 pokec,24:pokec,0:sysmp:poke646,bf
730 printchr$(18)" Gleicher Prg-teil : F
  1  Menue : F 7 "chr$(146):poke646,cf
740 geta$:ifa$=""or(a$<>chr$(133)anda$<>chr$(136))then740
750 ifa$=chr$(136)thenf1=1:return
760 return
770 :
780 rem ***** "Subroutine fuer" *****
790 rem ***** "Programmteil 1" *****
800 :
810 rem ***** "Eingabepuefung" *****
820 iflen(p$)>9thenrun
830 iflen(p$)=0thenf1=1:return
840 w=len(p$):forj=1tow:d$=mid$(p$,j,1)
850 ifasc(d$)<48orasc(d$)>57thenifasc(d$)<>43andasc(d$)<>45then870
860 nextj:return
```



```

870 j=w+1:nextj:f1=1:return
880 rem ***** "Hauptprogramm" *****
890 rem ***** "Bruchrechnung" *****
900 printchr$(147)"Mit diesem Programmte
il kannst Du":print
910 print"  Addition          ("chr$(
(18)" + "chr$(146)"):print
920 print"  Subtraktion      ("chr$(
(18)" - "chr$(146)"):print
930 print"  Multiplikation   ("chr$(
(18)" * "chr$(146)"):print
940 print"  Division         ("chr$(
(18)" / "chr$(146)"):print
950 print"von Bruchzahlen ueben. Solltes
t Du die":print
960 print"Aufgabe nach drei Versuchen ni
cht ge-":print
970 print"loest haben, so erhaeltst Du d
as Ergeb-":print
980 print"nis vollstaendig gekuerzt ausg
edruckt.":print
990 print"Bei ganzzahligen Loesungen mus
t Du im":print
1000 print"im Nenner des Bruches eine '1
' eingeben."
1010 pokecz,22:pokecs,0:sysmp:print"Ents
prechendes Rechenzeichen waehlen !"
1020 c1=24:c2=0:gosub600:f=0
1030 ifz$("<"+"andz$("<"-"andz$("<"*"andz$("<
>"/"thenc3=39:gosub640:goto1020
1040 ifz$="+"thenm$="+"n$="Summe":o$="A
ddition : "
1050 ifz$="-"thenm$="-":n$="Differenz":o
$="Subtraktion : "
1060 ifz$="*"thenm$="*":n$="Produkt":o$=
"Multiplikation : "
1070 ifz$="/"thenm$="":n$="Quotient":o$
="Division : "
1080 rem ***** "Aufgabe eingeben" *****
1090 u$=z$:printchr$(147)"Gib jetzt nach
einander die Zaehler":print
1100 print"und Nenner der beiden Brueche
ein !":print:print:print
1110 printv$;" ";m$;" ";v$;" = ";v$;
1120 f1=0:c1=5:c2=2:gosub600:p$=z$:gosub

```

```

810: if f1=1 then c3=10: gosub 640: goto 1120
1130 z1=val(z$)
1140 f1=0: c1=7: c2=2: gosub 600: p$=z$: gosub
810: if f1=1 then c3=10: gosub 640: goto 1140
1150 n1=val(z$): if n1=0 then c3=10: gosub 640
: goto 1140
1160 f1=0: c1=5: c2=16: gosub 600: p$=z$: gosub
810: if f1=1 then c3=10: gosub 640: goto 1160
1170 z2=val(z$): if z2=0 and u$="/" then c3=10
: gosub 640: goto 1160
1180 f1=0: c1=7: c2=16: gosub 600: p$=z$: gosub
810: if f1=1 then c3=10: gosub 640: goto 1180
1190 n2=val(z$): if n2=0 then c3=10: gosub 640
: goto 1180
1200 if u$="+" then 1240
1210 if u$="-" then 1290
1220 if u$="*" then 1340
1230 if u$="/" then 1360
1240 rem ***** "Addition" *****
1250 if (z1=0 and z2<>0) then z=z2: n=n2: goto 1
380
1260 if (z1<>0 and z2=0) then z=z1: n=n1: goto 1
380
1270 if (z1=0 and z2=0) then 1460
1280 z=z1*n2+n1*z2: n=n1*n2: on-(z=0) goto 1
460: goto 1380
1290 rem ***** "Subtraktion" *****
1300 if (z1=0 and z2<>0) then z=-z2: n=n2: goto
1380
1310 if (z1<>0 and z2=0) then z=z1: n=n1: goto 1
380
1320 if (z1=0 and z2=0) then 1460
1330 z=z1*n2-n1*z2: n=n1*n2: on-(z=0) goto 1
460: goto 1380
1340 rem ***** "Multiplikation" *****
1350 on-(z1=0 or z2=0) goto 1460: z=z1*z2: n=n
1*n2: goto 1380
1360 rem ***** "Division" *****
1370 on-(z1=0) goto 1460: z=z1/n2: n=n1*z2
1380 rem * "Euklidischer Algorithmus" **
1390 x=abs(z): y=abs(n): if x>=y then a=x: b=y
: goto 1410
1400 a=y: b=x
1410 if b=0 then 1430
1420 r=a-b*int(a/b): a=b: b=r: goto 1410

```

```

1430 k=z/a:l=n/a:ifk<0andl<0thenk=abs(k)
:l=abs(l):goto1470
1440 ifk>0andl<0thenk=-k:l=-1
1450 goto1470
1460 k=0:l=1
1470 rem ***** "Ergebnis eingeben" *****
1480 pokecz,10:pokecs,0:sysmp
1490 printchr$(18)" Gib nun dein errechn
etes Ergebnis ein !"
1500 f1=0:c1=5:c2=30:gosub600:p$=z$:gosu
b810:iff1=1thenc3=10:gosub640:goto1500
1510 a1=val(z$)
1520 f1=0:c1=7:c2=30:gosub600:p$=z$:gosu
b810:iff1=1thenc3=10:gosub640:goto1520
1530 a2=val(z$):ifa2=0thenc3=10:gosub640
:goto1520
1540 on-(a1=abs(k)anda2=abs(l)anda1/a2=k
/1)goto1590:ifa1/a2=k/1then1630
1550 rem **** "Ergebnis war falsch" ****
1560 f=f+1:pokecz,13:pokecs,0:sysmp:prin
tn$;a$(2):iff>2then1610
1570 pokecz,15:pokecs,0:sysmp:printa$(2+
f):gosub680
1580 c1=5:c2=27:c3=10:gosub640:c1=7:gosu
b640:c1=13:c2=0:o=3:gosub660:goto1470
1590 rem **** "Ergebnis war richtig" ***
1600 pokecz,13:pokecs,0:sysmp:printa$(1)
:goto1790
1610 rem ***** "drei Fehlversuche" *****
1620 pokecz,15:pokecs,0:sysmp:printa$(7)
:goto1660
1630 rem ** "Ergebnis nicht gekuerzt" **
1640 pokecz,12:pokecs,0:sysmp:printa$(5)
1650 pokecz,14:pokecs,0:sysmp:printa$(6)
;abs(a2/1)
1660 rem "Bruchstrichlaengen bestimmen"
1670 bz=len(str$(z1)):bn=len(str$(n1)):b
1=bn:ifbz>bnthenb1=bz
1680 b1$=mid$(z1$,2,b1+1)
1690 bz=len(str$(z2)):bn=len(str$(n2)):b
2=bn:ifbz>bnthenb2=bz
1700 b2$=mid$(z1$,2,b2+1)
1710 bz=len(str$(k)):bn=len(str$(1)):b3=
bn:ifbz>bnthenb3=bz
1720 b3$=mid$(z1$,2,b3+1)

```

```

1730 rem *"Richtiges Ergebnis drucken" *
1740 pokecz,17:pokecs,0:sysmp:printchr$(
18)" Richtiges Ergebnis der ";o$
1750 pokecz,19:pokecs,0:sysmp:printz1;ta
b(b1+4)z2;tab(b1+b2+8)k
1760 pokecz,20:pokecs,0:sysmp:printb1$;"
";m$;" ";b2$;" ";=";" ";b3$;
1770 ifl=1thenprint" = ";k
1780 pokecz,21:pokecs,0:sysmp:printn1;ta
b(b1+4)n2;tab(b1+b2+8)l
1790 fl=0:gosub710:on-(fl=1)goto430:goto
880
1800 :
1810 rem ***** "Subroutine fuer" *****
1820 rem ***** "Programmteil 2" *****
1830 :
1840 rem ***** "Eingabepuefung" *****
1850 iflen(p$)>12thenrun
1860 iflen(p$)=0thenfl=1:return
1870 w=len(p$):forj=1tow:d$=mid$(p$,j,1)
1880 ifasc(d$)<48orasc(d$)>57thenifasc(d
$)<>95thenj=w+1:nextj:fl=1:return
1890 nextj:return
1900 rem ***** "Hauptprogramm" *****
1910 rem ***** "ggT und kgV" *****
1920 printchr$(147)"Mit diesem Programmt
eil kannst Du in":print
1930 print"3 Versuchen feststellen, ob D
u den":print
1940 print"groessten gemeinsamen Teiler
";chr$(18)"'ggT'"chr$(146);" und":print
1950 print"das kleinste gemeinsame Vielf
ache ";chr$(18)"'kgV'"chr$(146):print
1960 print"einer beliebigen Anzahl natue
rlicher":print
1970 print"Zahlen korrekt bestimmt hast.
":print
1980 print"Gib jetzt die Zahlen einzeln
nachein-":print
1990 print"ander ein. Nach der letzten E
ingabe":print
2000 print"muss mit '";chr$(95);"' abges
chlossen werden."
2010 rem ** "Eingabe der Zahlenwerte" **
2020 n=1:m=1:g=1:k=1:f=0:c3=20

```

```

2030 f1=0:c1=17+m:c2=16:gosub600:p$=z$:g
osub1840:iff1=1thengosub640:goto2030
2040 ifz$=chr$(95)andn>2thenn=n-1:goto20
80
2050 p(n)=val(z$):ifp(n)=0thengosub640:g
oto2030
2060 n=n+1:m=n:ifn>7thenc1=24:c2=15:c3=1
5:gosub640:m=7
2070 goto2030
2080 printchr$(147)"Folgende Zahlen hast
Du eingegeben : "
2090 pokecz,2:pokecs,0:sysmp
2100 fori=1ton:printp(i);:nexti
2110 fori=1ton:q(i)=p(i):nexti
2120 fori=1to(n-1):a=p(i):b=p(i+1)
2130 rem **** "Bestimmung des ggT" *****
2140 r=a-b*int(a/b):ifr=0theng=b:goto216
0
2150 a=b:b=r:goto2140
2160 p(i+1)=g:c=q(i):d=q(i+1)
2170 rem **** "Bestimmung des kgV" *****
2180 z=c*d
2190 r=c-d*int(c/d):ifr=0thenk=z/d:goto2
210
2200 c=d:d=r:goto2190
2210 q(i+1)=k
2220 nexti
2230 rem ** "Eingabe von ggT und kgV" **
2240 pokecz,4:pokecs,0:sysmp:printchr$(1
8)"Gib nun ein, was Du als 'ggT' und";
2250 print"      ":printchr$(18)" 'kgV' di
eser Zahlen errechnet hast ! "
2260 f1=0:pokecz,8:pokecs,10:sysmp:print
"ggT = ";:c1=8:c2=16:gosub600
2270 p$=z$:gosub1840:iff1=1thenc3=20:gos
ub640:goto2260
2280 gp=val(z$)
2290 f1=0:pokecz,10:pokecs,10:sysmp:prin
t"kgV = ";:c1=10:c2=16:gosub600
2300 p$=z$:gosub1840:iff1=1thenc3=20:gos
ub640:goto2290
2310 kp=val(z$)
2320 ifgp=gandkp=kthenpokecz,13:pokecs,0
:sysmp:printa$(8):goto2520
2330 ifgp=gandkp<>kthenpokecz,13:pokecs,

```

```
0:sysmp:printa$(10):goto2400
2340 ifgp<>gandkp=kthenpokecz,13:pokecs,
0:sysmp:printa$(11):goto2440
2350 rem **** "ggT und kgV falsch" *****
2360 pokecz,13:pokecs,0:sysmp:printa$(9)
2370 pokecz,15:pokecs,0:sysmp:f=f+1:iff>
2then2480
2380 printa$(2+f):gosub680:c1=8:c2=15:c3
=15:gosub640
2390 c1=10:c3=15:gosub640:c1=13:c2=0:o=3
:gosub660:goto2240
2400 rem ***** "nur kgV falsch" *****
2410 pokecz,15:pokecs,0:sysmp:f=f+1:iff>
2then2480
2420 printa$(2+f):gosub680:c1=10:c2=15:c
3=15:gosub640
2430 c1=13:c2=0:o=3:gosub660:goto2240
2440 rem ***** "nur ggT falsch" *****
2450 pokecz,15:pokecs,0:sysmp:f=f+1:iff>
2then2480
2460 printa$(2+f):gosub680:c1=8:c2=15:c3
=15:gosub640
2470 c1=13:c2=0:o=3:gosub660:goto2240
2480 rem ***** "drei Fehlversuche" *****
2490 pokecz,15:pokecs,0:sysmp:printa$(7)
:pokecz,17:pokecs,0:sysmp
2500 printchr$(18)" Richtige Loesung fue
r ggT und kgV : "
2510 print:printtab(10)"ggT =";g:print:p
rinttab(10)"kgV =";k
2520 fl=0:gosub710:on-(fl=1)goto430:goto
1900
2530 :
2540 rem ***** "Subroutinen fuer" *****
2550 rem ***** "Programmteil 3" *****
2560 :
2570 rem *** "Eingabepuefung Zahl" ****
2580 iflen(p$)>12thenrun
2590 iflen(p$)=0thenfl=1:return
2600 w=len(p$):forj=1tow:d$=mid$(p$,j,1)
2610 ifasc(d$)<48orasc(d$)>57thenj=w+1:n
extj:fl=1:return
2620 nextj:return
2630 rem * "Eingabepuefung Zerlegung" *
2640 iflen(p$)>40thenrun
```

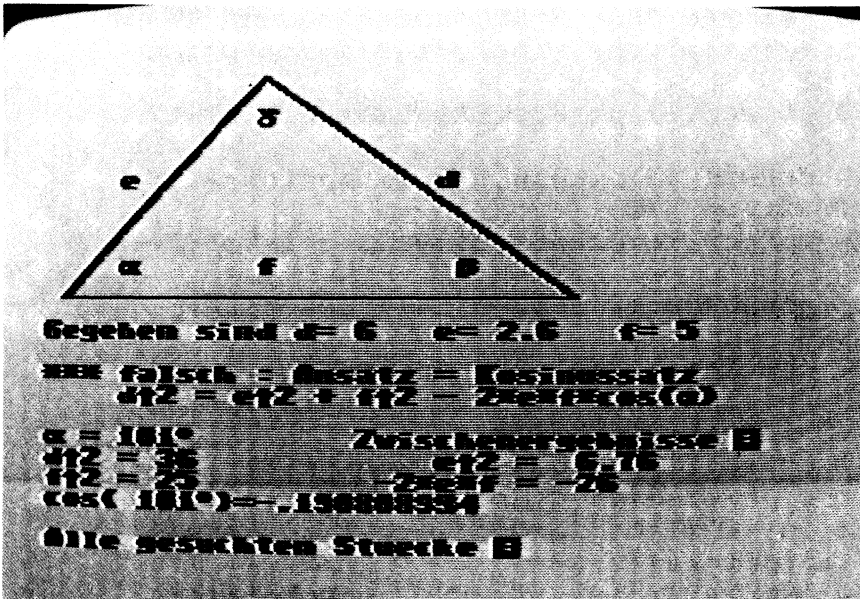
```
2650 if len(p$)=0 then f1=1: return
2660 w=len(p$): for j=1 to w: d$=mid$(p$,j,1)
2670 if asc(d$)<48 or asc(d$)>57 then if asc(d
$)<>32 and asc(d$)<>42 then j=w+1: goto 2690
2680 next j: return
2690 next j: poke cz,6: poke cs,(len(x$)+2): s
ysmp: print left$(z4$,40-(len(x$)+2));
2700 f1=1: return
2710 rem **** "Leerzeichenpruefung" ****
2720 w=len(a$): for i=1 to w: p$(i)=mid$(a$,i
,1)
2730 if p$(i)=" " then 2750
2740 next i: return
2750 t1$=left$(a$,i-1): t2$=right$(a$,w-i
): a$=t1$+t2$: i=w+1: next i: goto 2720
2760 rem ***** "Hauptprogramm" *****
2770 rem ***** "Pr.-fakt.-zerl." *****
2780 print chr$(147)"Mit diesem Programmt
eil kannst Du die": print
2790 print chr$(18)" Primfaktorenzerlegun
g "chr$(146)" einer natuer-": print
2800 print "lichen Zahl einueben. Nach Ei
ngabe der": print
2810 print "betreffenden Zahl sollst Du s
ie als Pro-"
2820 print "dukt schreiben, bei dem alle
Faktoren": print
2830 print "Primzahlen sind. Es kommt dab
ei jedoch": print
2840 print "nicht auf die Reihenfolge der
einzelnen": print
2850 print "Faktoren an. So ist 600 = 2*2
*2*3*5*5": print
2860 print "ebenso korrekt wie 600 = 2*3*
5*2*5*2.": print
2870 print "600 = 2*2*10*15 hingegen stel
lt nur": print
2880 print "eine Faktorenzerlegung dar, w
eil": print
2890 print "beispielsweise 15 keine Primz
ahl ist.": gosub 680
2900 f=0: print chr$(147)"Welche Zahl moec
hstest Du zerlegen ?"
2910 f1=0: c1=2: c2=10: gosub 600: p$=z$: gosu
b 2570: if f1=1 then c3=20: gosub 640: goto 2910
```

```
2920 x=val(z$):ifx<2thenc3=20:gosub640:g
oto2910
2930 xd=x:x$=right$(str$(x),len(str$(x)
-1)+" "
2940 rem * "pruefen auf Zerlegbarkeit" *
2950 on-(x=2orx=3)goto2990:ifx/2=int(x/2
)then3010
2960 k=3
2970 on-(x/k=int(x/k))goto3010:ifk2>xth
en2990
2980 k=k+2:goto2970
2990 rem * "eingegebene Zahl war prim" *
3000 pokecz,4:pokecs,0:sysmp:printx$;a$(
12):print:printa$(13):goto3430
3010 rem *** "Computer bestimmt PFZ" ***
3020 h=1:y=x:a=2
3030 on-(a>y/2)goto3050:m=x/a:n=int(x/a)
:ifm<>nthena=a+1:goto3030
3040 r(h)=a:x=m:h=h+1:on-(x=1)goto3050:g
oto3030
3050 rem *** "Anwender bestimmt PFZ" ***
3060 fl=0:pokecz,4:pokecs,0:sysmp
3070 printchr$(18)" Gib nun die Primfakt
orenzerlegung ein !"
3080 pokecz,6:pokecs,0:sysmp:printx$;"=
";c1=6:c2=len(x$)+2:gosub600
3090 p$=z$:gosub2630:on-(f1=1)goto3060:a
$=z$:gosub2710
3100 rem ***** "Stringauswertung" *****
3110 a$=a$+"*":z=0
3120 w=len(a$):fori=1toz:p$(i)=mid$(a$,i
,1)
3130 ifp$(i)="*"thenz=z+1:t$(z)=left$(a$
,i-1):a$=right$(a$,w-i):goto3150
3140 nexti:goto3160
3150 i=w+1:nexti:goto3120
3160 pr=1:foru=1toz:pr=pr*val(t$(u)):nex
tu:ifpr<>xdthen3320
3170 rem *** "alle Faktoren prim ?" ***
3180 foru=1toz:fr=val(t$(u))
3190 on-(fr=2orfr=3)goto3230:iffr/2=int(
fr/2)then3260
3200 k=3
3210 on-(fr/k=int(fr/k))goto3260:ifk2>f
rthen3230
```



```
3220 k=k+1:goto3210
3230 nextu
3240 rem ***** "PFZ ist richtig" *****
3250 pokecz,9:pokecs,0:sysmp:printa$(14)
:goto3430
3260 rem *** "nur Faktorenzerlegung" ***
3270 f=f+1:pokecz,9:pokecs,0:sysmp:print
a$(16)
3280 pokecz,11:pokecs,0:sysmp:printa$(17
)
3290 pokecz,13:pokecs,0:sysmp:printa$(18
):iff>2then3370
3300 pokecz,16:pokecs,0:sysmp:printa$(2+
f)
3310 gosub680:c1=6:c2=0:o=11:gosub660:go
to3080
3320 rem ***** "falsche Zerlegung" *****
3330 f=f+1:pokecz,9:pokecs,0:sysmp:print
a$(15):a$=z$:gosub2710
3340 pokecz,11:pokecs,0:sysmp:printright
$(str$(pr),len(str$(pr))-1);" = ";a$
3350 on-(f>2)goto3370:pokecz,13:pokecs,0
:sysmp:printa$(2+f)
3360 gosub680:c1=6:c2=0:o=8:gosub660:got
o3080
3370 rem ***** "drei Fehlversuche" *****
3380 pokecz,16:pokecs,0:sysmp:printa$(7)
:pokecz,18:pokecs,0:sysmp
3390 printchr$(18)" Richtige Primfaktore
nzerlegung : "chr$(146)
3400 pokecz,20:pokecs,0:sysmp:printx$;"=
";
3410 s=h-1:forh=1to(s-1):w=len(str$(r(h)
)):printright$(str$(r(h)),w-1);"*";
3420 nexth:w=len(str$(r(s))):printright$
(str$(r(s)),w-1)
3430 f1=0:gosub710:on-(f1=1)goto430:goto
2900
ready.
```

## 2 Sinus und Kosinus



### Programm-Telegramm

- Winkelbestimmung und Dreiecksberechnung durch zufallsgesteuerte Aufgabenstellung.
- Wirklichkeitsgetreue Darstellung durch Planfigur und Sonderzeichen.
- Mit schrittweiser Hinführung zur Lösung.

### Sinus & Co

Wie war das noch?

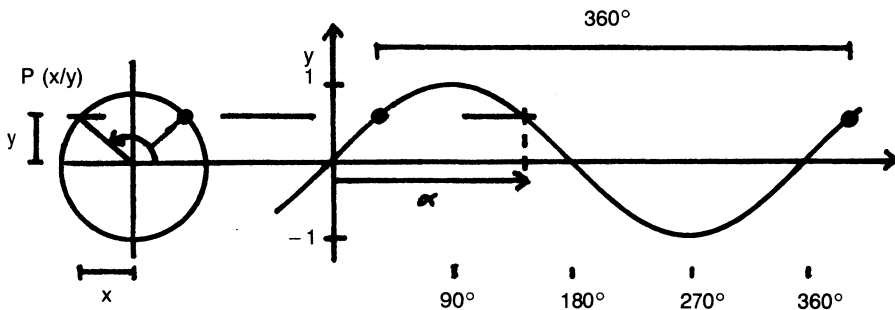
Aus dem Latein-Wörterbuch: sinus = Rundung, Krümmung, Bogen, Falte, Meerbusen, Bucht, Schlucht, Vorsprung, Landspitze, Busen, Brust, Schoß, Tasche, Geldbeutel, Liebe, Fürsorge, Schutz, Verborgenheit, Innerstes, Herz, Gewalt, Macht.

Solche Vieldeutigkeit ist Mathematikern ein Greuel. Sie definieren den »Sinus« lieber ganz neu: Der Sinus ist eine (Winkel-)Funktion: Jedem Winkel  $\alpha$  wird ein Zahlenwert  $\sin(\alpha)$  zugeordnet. In einem rechtwinkligen Dreieck ist dieser Wert gleich dem Längen-

verhältnis von Gegenkathete/Hypotenuse. Die Hypotenuse ist die dem rechten Winkel gegenüberliegende Seite, die Gegenkathete liegt dem Winkel  $\alpha$  gegenüber. Das Verhältnis aus Ankathete/Hypotenuse definiert den »Kosinus« von  $\alpha$  (die Ankathete liegt am Winkel  $\alpha$ ).

Damit aber nicht genug: Im rechtwinkligen Dreieck kann  $\alpha$  nur Werte zwischen  $0^\circ$  und  $90^\circ$  annehmen. Man erweitert die Definition schließlich so, daß jeder Winkel seinen »Sinus« bekommt. Dazu benutzt man den Einheitskreis, einen Kreis im Koordinatensystem mit dem Mittelpunkt  $(0/0)$  und dem Radius 1:

Zu jedem Wert von  $\alpha$  gibt es einen Punkt P auf dem Einheitskreis. Seine Koordinaten  $(x,y)$  definieren Sinus und Kosinus:  $\cos(\alpha) := x$ ,  $\sin(\alpha) := y$ . Die Vorzeichen von  $x$  und  $y$  ergeben sich aus der Lage im Koordinatensystem.

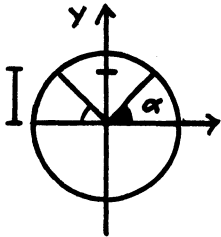


Zeichnet man nun noch den Graphen dieser Funktion daneben, dann sieht man, womit sich der Sinus seinen Namen verdient hat.

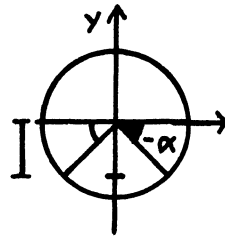
### Reichlich verwinkelte Beziehungen

Wie man sieht, läßt diese Definition auch Winkel  $> 360^\circ$  oder  $< 0^\circ$  zu. Der Sinus von  $40^\circ$  ist dabei »genau soviel wert« wie der Sinus von  $40 + 360^\circ$ ,  $40 + 720^\circ$ ,  $40 - 360^\circ$ , usw., wenn man sich die Winkel als Drehwinkel vorstellt. Negative Winkel entstehen bei umgekehrter Drehung (im Uhrzeigersinn). Die Sinus- und Kosinuswerte solcher Winkel rechnet jeder etwas bessere Taschenrechner brav aus. Denn wie bei jeder Funktion herrschen auch bei der Sinusfunktion eindeutige Verhältnisse. Wehe aber, wenn man einen Sinuswert oder Kosinuswert hat und man sucht »den« richtigen Winkel. Da gibt es nämlich unendlich viele! Der Taschenrechner nennt nur einen. Gibt man z.B. ein 0.643 INV SIN (statt INV für »Umkehrung«, verwenden manche Taschenrechner auch ARC, SECOND, SIN hoch  $-1$ , ...), so bekommt man z.B. den Wert  $\alpha = 40^\circ$ . Weitere Lösungen sind aber auch  $\alpha = 40 + n \times 360^\circ$ ,  $\alpha = 40 - n \times 360^\circ$  mit  $n=1, 2, 3 \dots$ . Wie man oben sieht, gibt es jedoch auch im Bereich von  $0 - 360^\circ$  noch einen zweiten Wert (mit ebenso vielen »Verwandten«). Im Mathematikbuch findet man dazu eine Reihe von Umrechnungsformeln, die man natürlich auswendig lernen kann, etwa

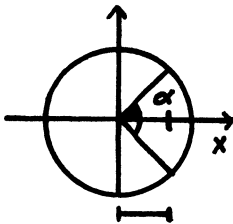
$\sin(\alpha) = \sin(180^\circ - \alpha)$ ,  $\sin(-\alpha) = \sin(180^\circ + \alpha)$ ,  $\sin(\alpha) = \sin(360^\circ - \alpha)$ ,  $\cos(\alpha) = \cos(360^\circ - \alpha)$ . Auf Dauer hilft bei solchen zum Verwechselln ähnlichen Formeln nur eines: Man merkt sich die Symmetrien am Kreis:



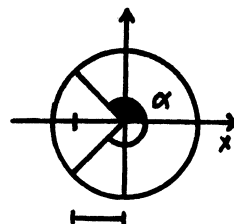
$$\sin(\alpha) = \sin(180^\circ - \alpha)$$



$$\sin(-\alpha) = \sin(180^\circ + \alpha)$$



$$\cos(\alpha) = \cos(360^\circ - \alpha)$$



$$\cos(\alpha) = \cos(360^\circ - \alpha)$$

**Faustregel: Sinus = y-Wert**

(also symmetrisch zur y-Achse)

**Kosinus = x-Wert**

(also symmetrisch zur x-Achse)

Zur Not zeichnet man sich kurz einen Kreis auf. Der erste Teil unseres Programms trainiert diese Überlegungen zum Auffinden des zweiten Winkelwertes.

## Dreiecksverhältnisse!

### Dreiecksberechnungen mit Hilfe von Sinus- und Kosinussatz

In der Regel kann man aus drei gegebenen Stücken die übrigen drei Stücke eines Dreiecks berechnen. »Stücke« sind dabei Winkel oder Seiten. In einigen Fällen gibt es jedoch keine eindeutige Lösung (Kongruenzsätze!): (WWW)=drei Winkel und (SSW)=ein Winkel und zwei Seiten, wobei dem Winkel die kürzere der beiden Seiten gegenüberliegt. Außerdem darf die Winkelsumme z.B. nicht  $180^\circ$  überschreiten, oder

eine Seite darf nicht länger sein als die Summe der beiden anderen. Diese Bedingungen werden von unserem Programm automatisch kontrolliert.

Sinussatz:  $a/\sin(\alpha) = b/\sin(\beta) = c/\sin(\gamma)$

*Merke:* Seite und gegenüberliegender Winkel bilden ein Paar.

Kosinussatz: 
$$\begin{aligned} c^2 &= a^2 + b^2 - 2ab \cdot \cos(\gamma) \\ a^2 &= b^2 + c^2 - 2bc \cdot \cos(\alpha) \\ b^2 &= a^2 + c^2 - 2ac \cdot \cos(\beta) \end{aligned}$$

Die Anwendung des Sinussatzes ist rechnerisch nicht so aufwendig wie die des Kosinussatzes. Man sieht schnell, daß man den Kosinussatz grundsätzlich nur bei der Berechnung des ersten gesuchten Stückes benötigt, und dies auch nur dann, wenn alle drei Seiten (SSS) oder zwei Seiten und der davon eingeschlossene Winkel (SWS) gegeben sind.

## Programmablauf

In einem Vorprogramm wird der Zeichensatz im RAM abgelegt (ab 2048) und geändert. Neben einigen Sonderzeichen (griech. Buchstaben) wird vor allem die Planfigur (Dreiecksberechnungen) mit Hilfe des Zeichensatzes dargestellt. Um Zeit zu sparen, verwendet man hierfür am besten ein kleines Maschinenprogramm. Gleichzeitig wird das Hauptprogramm nach (4096) verlegt. Will man nur den ersten Programmteil (Bestimmung des zweiten Winkels) benutzen, kommt man auch ohne das Vorprogramm aus.

Das Vorprogramm wird normal geladen und gestartet. Nach der READY-Meldung kann auch das Hauptprogramm normal geladen und gestartet werden. Im Programm wird dann zunächst danach gefragt, ob man die Winkelbestimmung (Teil I) oder die Dreiecksberechnung (Teil II) trainieren möchte.

### Teil I (Winkelbestimmung)

Zu einem mit der RANDOM-Funktion ermittelten Sinus- bzw. Kosinuswert sind zunächst die Bereiche der beiden zugehörigen Winkel anzugeben. Macht man einen Fehler, bekommt man eine grafische Hilfestellung. Danach wird einer der beiden Winkel angegeben (dieser entspricht dem Wert, den auch ein Taschenrechner angeben würde). Ist z.B. der Sinus negativ, so wird von den meisten Taschenrechnern ein negativer Winkel angegeben. Für den zweiten Winkelwert (zwischen  $0^\circ$  und  $360^\circ$ ) werden drei Lösungen (mit entsprechender Umrechnungsformel) angeboten.

Am Schluß einer Übung kann man mit einer beliebigen Taste die nächste Übung anfordern, oder mit »↑« über das Hauptmenü in den Teil II gelangen.

## Teil II (Dreiecksberechnung)

Das Programm stellt Aufgaben zur Dreiecksberechnung. Dazu werden die üblichen Hilfsmittel (Taschenrechner mit Sinus und Kosinus, Papier, Kugelschreiber) benötigt. Zu Beginn jeder Aufgabe erscheint ein Dreieck auf dem Bildschirm, bei dem Winkel und Seiten bezeichnet sind. Neben der Auswahl der gegebenen Stücke ändern sich mit jeder Aufgabe auch die Bezeichnungen! Es wird nun danach gefragt, ob der Sinus- oder der Kosinussatz verwendet werden muß.

Dann wird danach gefragt, welches Stück berechnet werden kann. *Die Winkel  $\alpha$ ,  $\beta$  und  $\gamma$  werden dabei mit den Tasten Commodore  $a$ ,  $b$  und  $c$  eingegeben.* Nach richtiger Eingabe wird die Formulierung des entsprechenden Ansatzes verlangt. Er wird nicht eingetippt, sondern auf einem Zettel formuliert. Mit »=« (oder einer anderen Taste) kann man den eigenen Ansatz mit dem des Computers vergleichen. Hier sind einige Besonderheiten zu beachten:

- 1) Im Falle (SSS) sind drei Ansätze mit dem Kosinussatz möglich. Der Computer nennt deshalb vorher den gesuchten Winkel.
- 2) Im Falle (WSW) kann man den Sinussatz erst nach Berechnung des dritten Winkels anwenden. In diesen Fällen gibt der Computer nur den Hinweis auf den Winkelsummensatz und geht zur nächsten Aufgabe über.
- 3) Die vom Computer zum Fall (SSW) gestellten Aufgaben sind zwar immer lösbar, jedoch kann es auch noch eine zweite Lösung geben (SSW). Das Programm gibt bei der Formulierung des Ansatzes einen entsprechenden Hinweis.

Ist der Ansatz kontrolliert, kann das gesuchte Stück auf dem Zettel ausgerechnet und anschließend mit »=« überprüft werden. Falls der eigene Wert falsch ist, kann man sich schließlich noch alle dazugehörigen Zwischenergebnisse ausgeben lassen. Will man auch die beiden letzten Stücke noch berechnen, so gibt das Programm auch hierzu noch die richtigen Ergebnisse an.

Will man eine Aufgabe nicht fortsetzen, so kann man jederzeit mit »↑« zur nächsten Aufgabe übergehen.

Ein vorzeitiger Übergang zur nächsten Aufgabe ist z.B. sinnvoll, wenn man den im Ansatz verlangten Kosinussatz noch gar nicht kennt. Für diesen Fall kann man zu Beginn der Berechnungen auch den Anteil an Kosinussatzaufgaben auf 0 setzen. Normalerweise wird man hier wohl den Wert 50 (= 50 Prozent) eingeben.

## Die wichtigsten Programmzeilen

- 100 – 780 : ALLGEMEINE VORBEREITUNGEN
- 310 – 560 : Sonder- und Steuerzeichen  
570 – 610 : Aufgabenliste (2 mal 9+3 Ansätze)  
620 – 690 : Funktion-Unterprogramme  
700 : Bildschirmfarbe  
710 – 780 : Teile-Menü (I oder II)
- 790 – 1340 : I WINKELBESTIMMUNG
- 860 : Sinus-/Kosinuswert (Zufallswahl)  
870 – 940 : Bereich für 1. Winkel  
950 – 1010 : Bereich für 2. Winkel  
1080 : Winkel-Alternativen für 2. Winkel  
1290 – 1340 : Graphische Hilfe
- 1350 – 2820 : II DREIECKSBERECHNUNGEN
- 1350 – 1380 : Warteschleife  
1430 – 1470 : Anteil Kosinussatz  
1500 : 6 Reihenfolgen für 3 Seiten  
1540 – 1570 : Zufallswahl der Seitenbezeichnungen  
1580 – 1770 : Planfigur drucken  
1820 : Zufallswahl Sinus-/Kosinussatz  
1840 – 1920 : Sinussatz (15 Ansätze)  
1930 – 1980 : Kosinussatz (6 Ansätze)  
1990 – 2080 : Alphabetische Ordnung der Stücke  
2090 – 2160 : Zufallswahl sämtlicher Ergebnisse  
2170 – 2220 : Gegebene Stücke  
2230 – 2310 : Sinus- oder Kosinussatz?  
2320 – 2410 : Gesuchtes Stück?  
2420 – 2560 : Formulierter Ansatz  
2570 – 2820 : Ergebnisse

(Helmut Klatt)

```
1 REM *****
2 REM *      VORPROGRAMM  ZU      *
3 REM *      SINUS UND KOSINUS    *
4 REM *****
5 :
6 REM  COPYRIGHT BY
7 REM  MODERNE VERLAGS GESELLSCHAFT
8 :
9 REM  AUTOR:  HELMUT KLATT
10 :
11 REM -- ZEICHENSATZ NACH 2048
12 A=49151:FOR J=1TO16:S=0
13 FORI=1TO18:READX:A=A+1:POKEA,X:S=S+X
14 NEXTI:READX:IFX<>STHENPRINT"FEHLER IN
";18+2*J:STOP
15 NEXTJ:S=0:FORI=1TO9:READX:A=A+1:POKEA
,X:S=S+X:NEXTI
16 READX:IFX<>STHEN PRINT"FEHLER IN 52";
:STOP
17 SYS 49294
18 REM -----
20 DATA 8,11,1,3,6,12,24,48,96,192,16,11
,1,3,6,12,24,48,522
22 DATA 96,255,24,11,0,0,128,96,48,24,6,
255,40,11,0,0,0,994
24 DATA 16,56,102,195,48,11,192,96,16,12
,2,1,0,0,56,11,0,0,814
26 DATA 0,0,0,128,96,48,64,11,12,6,1,0,0
,0,0,0,72,11,449
28 DATA 0,0,128,192,48,24,6,3,80,11,12,6
,1,0,0,0,0,255,766
30 DATA 240,11,48,72,72,48,0,0,0,0,248,1
1,0,60,102,124,102,102,1240
32 DATA 124,96,128,11,0,0,59,106,68,106,
59,0,48,9,0,0,59,106,979
34 DATA 68,106,59,0,224,11,0,102,55,24,5
2,108,108,56,0,0,173,14,1160
36 DATA 220,41,254,141,14,220,165,1,41,2
51,133,1,169,8,133,254,169,216,2431
38 DATA 133,252,169,0,133,251,133,253,16
0,0,177,251,145,253,200,192,0,208,2910
40 DATA 247,230,252,230,254,165,252,201,
224,208,235,165,1,9,4,133,1,173,2984
42 DATA 14,220,9,1,141,14,220,169,192,13
3,254,169,0,133,253,160,0,177,2259
```



```

44 DATA 253,133,251,230,253,177,253,240,4
5,133,252,230,253,160,0,177,253,145,3438
46 DATA 251,200,192,8,208,247,24,165,253
,105,8,133,253,76,213,192,145,253,2926
48 DATA 192,8,208,231,24,165,251,105,9,1
44,2,230,252,133,251,76,213,192,2686
50 DATA 169,19,141,24,208,169,0,141,0,16
,169,1,133,43,169,16,133,44,1595
52 DATA 133,46,169,3,133,45,76,116,164,8
85

```

```

100 rem *****
110 rem * *
120 rem" * Sinus und Kosinus * "
130 rem * *
140 rem *****
150 rem
160 rem" Copyright 1985 by "
170 rem moderne verlags gesellschaft
180 rem
190 rem" Autor: Helmut Klatt "
200 rem
210 rem teil i winkelbestimmungen
220 rem
230 rem teil ii dreiecksberechnungen
240 rem
250 rem neuer Zeichensatz ab 2048 wird
260 rem im Vorprogramm eingerichtet
270 rem (umschalten mit poke 53272,19)
280 rem Hauptprogramm beginnt bei 4096
290 rem -----
295 poke 53280,1:poke 53281,1
300 dim au$(11),da(16)
310 rem ---sonderzeichen i+ii
320 w$=chr$(38):rem alpha i
330 g$=chr$(190):rem grad
340 d$=chr$(17) :rem down=cursor runter
350 u$=chr$(145) :rem up= cursor rauf
360 l$=chr$(157) :rem left= cursor links
370 r$=chr$(29) :rem cursor rechts
380 hm$=chr$(19) :rem cursor home
390 cl$=chr$(147):rem clear screen
400 ri$=chr$(18) :rem revers an

```

```
410 ro$=chr$(146):rem revers aus
420 le$="":rem19*leer
430 dc$=chr$(164):rem dreiecksseite c
440 db$=chr$(161):rem dreiecksseite b
450 a$=chr$(176):rem alpha ii
460 b$=chr$(191):rem beta
470 c$=chr$(188):rem gamma
480 i$=chr$(221):o$=chr$(42)
490 s$=o$+i$+o$+" "+i$+" "+i$+" "+o$+i$
+o$+" "+i$+o$+" "+i$
500 s$=s$+o$+o$+i$+" "+o$+i$+" "
510 rem --- dreieck ii ecken
520 rem ecken
530 ea$=chr$(162):eb$=chr$(163):ec$=chr$(165)
540 rem --- dreieck ii seite a
550 dd$=chr$(166)+chr$(167):ee$=chr$(168)
+chr$(169)
560 rem
570 rem --- aufgabenliste zu ii
580 data 3450,4351,5342
590 data 3041,3052,4130,4152,5230,5241
600 data 3120,4021,5012
610 fori=0to11:readau$(i):next
620 rem --- funktionen -----
630 rem --- zufallszahlen 0,1,2,...,x-1
640 x=rnd(-ti)
650 def fnr(x)=int(rnd(1)*x*.9999)
660 rem --- arcsin(x)
670 def fns(x)=.01*int(.5+18000/π*atn(x/
sqr(-x*x+1)))
680 rem --- arccos(x)
690 def fnc(x)=.01*int(9000.5-18000/π*at
n(x/sqr(-x*x+1)))
700 poke53281,peek(53280):poke646,0
710 rem --- menue i oder ii
720 printcl$"Sinus und Kosinus * Trainin
gsprogramm":printd$"von Helmut Klatt"
730 printd$;d$;d$;d$"1 Teil I - Winkel
bestimmung"
740 printd$;d$"2 Teil II - Dreiecksbere
chnungen"
750 getz$:ifz$=""then750
760 ifz$="1"then790
770 ifz$="2"then1390
```

```

780 goto750
790 rem -----
800 rem i 2 winkel zwischen 0 und 360
810 rem
820 rem --- sinus bzw cosinus vorgeben
830 f$(1)="sin":f$(2)="cos"
840 a1(0)=6:a1(1)=24:a1(2)=18:a1(3)=12
850 wi$="      180"+g$+"-180"+g$+"+360"+g
   $+"-":zi$="12"
860 m=-1:u=fnr(2000)-1000:u=u/1000:v=fnr
   (2)+1:m=(m+v)*2
870 ifu<0thenm=m+1
880 printcl$;f$(v)"("w$")=";u;" =>"d$
890 fork=1to4:printtab(14);k;:a1$=str$(k
   *90-90):a2$=str$(k*90)
900 a1$=right$(le$+a1$,3):a2$=right$(le$
   +a2$,3):print"  "a1$;g$<"w$;
910 print" <"a2$;g$
920 next:print
930 input"Welcher Bereich";h:h=int(h)
940 ifh<1orh>4then printu$;u$:goto 930
950 rem --- richtig ?
960 h1=20h;if(h1anda1(m))=0thenra=1:goto
   1290
970 a2=a1(m)-h1:print"OK! ";
980 input"und";h2:h2=int(h2)
990 ifh2<0orh2>4then980
1000 h3=20h2;if(h3anda2)=0thenra=2:goto1
   290
1010 e1=0:ifv=2then1270
1020 rem --- sin
1030 w0=fns(u):e0=2:ifm<1then1240
1040 rem --- sin <0
1050 w1=-w0:zi=1:e0=3:e1=4
1060 printcl$"OK!  "f$(v)"("w$")=";u:pri
   nt
1070 printw$;"0 =";w0;g$ =>"d$;k1=1
1080 w(1)=w1:w(2)=180-w1:w(3)=180+w1:w(4
   )=360-w1
1090 fork=k1to4:h$=mid$(wi$,k*5-4,5):pri
   nt
1100 printtab(1);k;"  "w$;mid$(zi$,zi,
   1);" =";h$;w1;g$=";w(k);g$
1110 nextk:print
1120 ifm<>1then1170

```

```
1130 inpute:ife=e1thenprintd$"OK! ";:got
o1160
1140 ife=e0thene0=e1:printd$"OK! ";:goto
1160
1150 e=e1:printd$"leider falsch:"
1160 printw$"1 ="mid$(wi$,e*5-4,5);w1;g$
"=";w(e);g$:print"und";
1170 inpute:ife=e0then1190
1180 printd$"leider falsch":goto1170
1190 printd$"OK! ";w$;"2 ="mid$(wi$,e
*5-4,5);w1;g$"=";w(e);g$
1200 printd$;d$"↑ Menue"
1210 getz$:ifz$=""then1210
1220 ifz$="↑"then710
1230 goto 860
1240 w1=w0:zi=2
1250 printcl$"OK! ";f$(v);"(";w$;")=";u
:print
1260 printw$"1 ="w0;g$" =>"d$:k1=2:goto
1080
1270 rem --- cos
1280 w0=fnc(u):e0=4:goto1240
1290 rem --- falsch = hilfe
1300 print"leider falsch:";
1310 printtab(30);mid$(s$,m*6+1,3)
1320 printtab(30);chr$(192);chr$(219);ch
r$(192)
1330 printtab(30);mid$(s$,m*6+4,3)
1340 onragoto930,980
1350 rem --- up eingabe -----
1360 getz$:ifz$=""then1360
1370 ifz$="↑"then1540
1380 return
1390 rem -----
1400 rem ii dreiecksberechnungen mit
1410 rem     sinus- und kosinussatz
1420 rem
1430 rem --- anteil kosinussatz
1440 printcl$;"Wieviel Prozent der Aufga
ben sollen"
1450 printd$"zum Kosinussatz gestellt we
rden ?"
1460 printd$"(Sinussatz + Kosinussatz =
100 Prozent)":print:input qk
1470 qk=.01*qk
```

```

1480 an$(1)="S":an$(2)="Kos"
1490 rem --- bezeichnungen
1500 se$="345354435453534543"
1510 s$(0)=a$:s$(1)=b$:s$(2)=c$
1520 rem
1530 rem --- neue aufgabe
1540 k=fnr(6)*3+1:bz$=mid$(se$,k,3):k=fn
r(3)+68
1550 s$(val(left$(bz$,1)))=chr$(k)
1560 s$(val(mid$(bz$,2,1)))=chr$(k+1)
1570 s$(val(right$(bz$,1)))=chr$(k+2)
1580 rem --- planfigur
1590 printcl$:tab(11);ec$
1600 rem --- seite b
1610 fori=1to9:printtab(11-i)db$:next
1620 rem --- seite c
1630 print" ";ea$;:fori=1to24:printdc$;:
next:printeb$
1640 printhm$:fori=1to11:printr$;:next
1650 printd$;c$;d$;d$;d$;:fori=1to8:prin
tl$;:next:prints$(4);
1660 fori=1to15:printr$;:next:prints$(3)
1670 printd$;d$;d$;r$;r$;r$;r$;a$;
1680 fori=1to6:printr$;:next:prints$(5);
1690 fori=1to9:printr$;:next:printb$;d$;
d$
1700 printhm$:fori=1to12:printr$;:next
1710 rem --- seite a
1720 fori=1to4
1730 printdd$;d$;l$;ee$;d$;
1740 next
1750 printdd$;d$;l$;chr$(170);
1760 print:print
1770 rem
1780 rem --- wahl der aufgabe - ansatz
1790 rem al$ = ansatzparameter
1800 rem au = nummer des ansatzes
1810 rem gg$ gegeben - xx$ gesucht
1820 qr=rnd(1):sz=1:ifqr<qkthensz=2
1830 ifsz>1then1940
1840 rem --- sinussatz
1850 au=fnr(9)+3:al$=au$(au):ifau>8then1
920
1860 u=fnr(2)+1:ifu>1then1890
1870 gg$=right$(al$,3):xx$=left$(al$,1)

```

```
1880 s1=0:goto2000
1890 gg#=left$(a1$,1)+right$(a1$,2):xx#=
mid$(a1$,2,1):s1=1
1900 goto2000
1910 rem --- winkelsumme
1920 s1=2:goto1980
1930 rem --- kosinussatz
1940 s1=-1:au=fnr(3):a1#=au$(au)
1950 u=fnr(2):ifu<1then1980
1960 gg#=right$(a1$,3):xx#=left$(a1$,1)
1970 goto2000
1980 gg#=left$(a1$,3):xx#=right$(a1$,1)
1990 rem --- ordnen gg# und hh#
2000 x0=val(xx#):hh#="":ff#=""
2010 fori=1to3:x#=mid$(bz$,i,1)
2020 forj=1to3:ifx#=mid$(gg$,j,1)thenff#
=ff#+x#:goto2040
2030 nextj:hh#=hh#+x#
2040 nexti
2050 fori=0to2:x#=right$(str$(i),1)
2060 forj=1to3:ifx#=mid$(gg$,j,1)thenff#
=ff#+x#:goto2080
2070 nextj:hh#=hh#+x#
2080 nexti
2090 rem --- zahlenwerte fuer parameter
2100 s(0)=20+fnr(116):s0=0:s(3)=0:s(4)=0
:s(5)=0:ug=π/180
2110 s(1)=20+fnr(141-s(0)):ifs(1)<s(s0)t
hens0=1
2120 s(2)=180-s(0)-s(1):ifs(2)<s(s0)then
s0=2
2130 s(s0+3)=1.5+fnr(21)/10:qs=10*s(s0+3
)/sin(ug*s(s0))
2140 ifs(3)=0thens(3)=.1*int(qs*sin(ug*s
(0))+.5)
2150 ifs(4)=0thens(4)=.1*int(qs*sin(ug*s
(1))+.5)
2160 ifs(5)=0thens(5)=.1*int(qs*sin(ug*s
(2))+.5)
2170 rem --- sinus- oder kosinussatz
2180 print"Gegeben sind ";
2190 fori=1to3:k=val(mid$(ff$,i,1))
2200 prints$(k);"=";s(k);:ifk<3thenprint
l$;g$;
2210 print"  ";
```

```

2220 nexti
2230 printd$:print"Ansatz ?";
2240 print"  1 Sinussatz    2 Kosinussatz
"
2250 get z$:ifz$=""then2250
2260 if z$="1"then 2290
2270 if z$="2"then 2290
2280 goto2250
2290 ifval(z$)<>szthenprintu$ "*** falsch
:";
2300 ifval(z$)=szthenprintu$ "*** richtig
:";
2310 print" Ansatz = ";an$(sz);"inussatz
"

2320 rem --- gesuchtes stueck
2330 print"Welches Stueck kann man berec
hnen?":print"(↑ Ende)    ";
2340 fori=1to3:prints$(val(mid$(hh$,i,1)
));" ";:nexti
2350 print"  ?"
2360 gosub1360
2370 rem --- kosinussatz -- 3 winkel
2380 ifsz<2then2400
2390 ifx0<3then2420
2400 ifz$=s$(x0)then2420
2410 printu$; "*** falsch! ";:goto2340
2420 printu$;u$"OK! Formuliere den Ansatz
z fuer    ";s$(x0)
2430 print"auf einem Zettel. Dann vergle
iche "ri$"="ro$
2440 gosub1360
2450 ifsl<2then2490
2460 print"Winkelsumme = 180";g$" ! Neue
Aufgabe":gosub1360:goto1540
2470 rem --- ansatz und ergebnisse
2480 rem    vorbereiten
2490 s1=val(left$(a1$,1)):s2=val(mid$(a1
$,2,1))
2500 s3=val(mid$(a1$,3,1)):s4=val(right$
(a1$,1))
2510 s1$=s$(s1):s2$=s$(s2):s3$=s$(s3):s4
$=s$(s4)
2520 ifsz>1then2560
2530 si$=left$(le$,13):ifsl=0then2550
2540 si$=" "+s3$+">"+s1$+" "+ri$+" Ssw

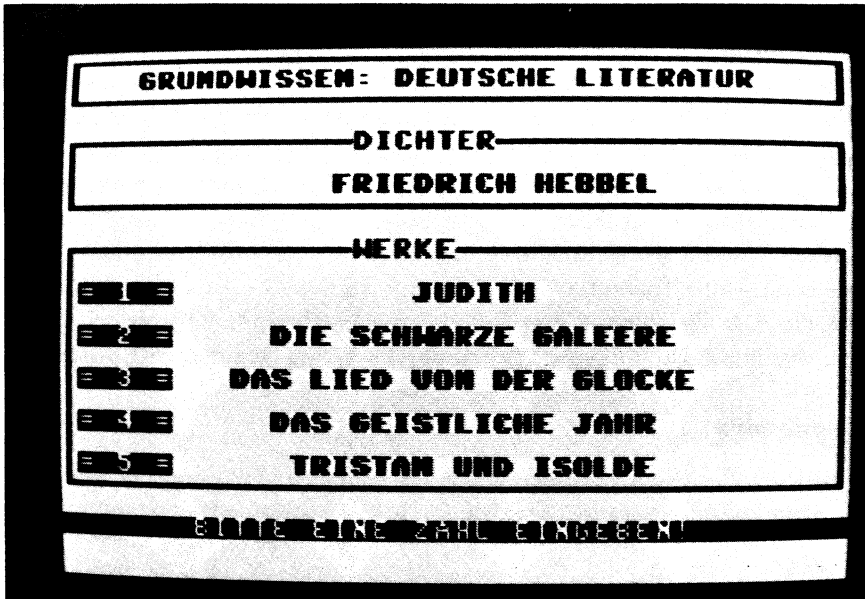
```

```
"+ro$+" ?"
2550 printu$;u$"      "s1$"/sin("s2$") = "
s3$"/sin("s4$")"si$:goto2570
2560 printu$;u$"      "s1$^2 = "s2$^2 +
"s3$^2 - 2*"s2$"*"s3$"*cos("s4$")      "
2570 printle$+le$:print"Ergebnis fuer "s
$(x0)" "ri$"="ro$;le$
2580 gosub1360
2590 printu$;s$(x0)" =";s(x0);l$;
2600 ifx0<3thenprintg$;
2610 printleft$(le$,8);
2620 print"Zwischenergebnisse "ri$"="ro$
2630 gosub1360
2640 ifsz>1then2700
2650 rem --- zwischenergebnisse sinus
2660 print"sin(";s(s2);l$;g$;)"=";sin(ug
*s(s2))
2670 s5=sin(ug*s(s4)):print"sin(";s(s4);
l$;g$)"=";s5
2680 prints3$"/sin(";s(s4);l$;g$)"=";s(s
3)/s5:goto2760
2690 rem --- zwischenergebnisse kosinus
2700 prints1$^2 =";s(s1)*s(s1);
2710 printtab(20)s2$^2 = ";s(s2)*s(s2)
2720 prints3$^2 =";s(s3)*s(s3);
2730 printtab(17)"-2*"s2$"*"s3$" = ";-2*
s(s2)*s(s3)
2740 print"cos(";s(s4);l$;g$;)"=";cos(ug
*s(s4))
2750 rem --- alle endergebnisse
2760 print:print"Alle gesuchten Stuecke
"ri$"="ro$
2770 gosub1360
2780 fori=1to3:k=val(mid$(hh$,i,1)):prin
ts$(k);" =";s(k);
2790 ifk<3thenprintl$;g$;
2800 print" ";:nexti:print" "ri$^"ro$
2810 getz$;ifz$=""then2810
2820 goto1540
```



### 3

## Deutsche Literatur (Grundwissen)



### Programm-Telegramm

- Ein »farbiger« Test in 4 Schwierigkeitsstufen für 1 bis 5 Teilnehmer.
- Zuordnung von Dichter, Werk, Epoche und Gattungsform im Wettlauf gegen die Zeit.
- Mit aktueller Sieger- und Punkteliste nach jeder Runde!

### Ein Pfad durch die deutsche Literatur

#### Abgrenzungen

In diesem Programm werden Fakten aus der Literaturgeschichte überprüft; ein Werturteil wird nicht gefällt. Dem Leser soll eine Grundlage vermittelt werden, auf der er weiter aufbauen kann. Dieser knappe Überblick mag als Orientierungshilfe dienen.

Die Dichtung der frühgermanischen Zeit wurde nicht in dieses Programm aufgenommen. Sie besteht häufig nur aus mündlicher Überlieferung, wobei der Verfasser unbekannt bleibt. Zudem wird sie im Schulunterricht nur verhältnismäßig knapp behan-

delt. Die relativ unbedeutende, meist bürgerliche Dichtung gegen Ende des Mittelalters (14. Jh.) wurde hier ebenfalls nicht berücksichtigt.

Aufgrund der äußerst umfangreichen Dichtung vom Ende des 19. Jahrhunderts bis zur Gegenwart mit ihren zahlreichen unterschiedlichen Strömungen, erscheint es ratsam, daß der Anwender mit diesen Daten ein eigenes Programm füllt.

Der Bogen der Einträge spannt sich von der ersten Blütezeit der deutschen Dichtung, der Höfischen Klassik, bis zum Ausklang des Realismus.

Innerhalb der Epochen mußte bisweilen, insbesondere bei der Lyrik, eine exemplarische Auswahl getroffen werden. Ein wesentliches Kriterium war hierbei auch die Bedeutung, die den verschiedenen Werken in den Lehrplänen zugemessen wird. In Ausnahmefällen ist die Epochenzugehörigkeit nicht allgemein verbindlich festgelegt. Einige Vertreter des literarischen Biedermeier (z.B. *Stifter*, *Droste-Hülshoff* u.a.) werden in manchen Literaturgeschichten auch dem Frühen Realismus zugeordnet.

Daten bedeutender Theoretiker, wie *Opitz* für das Barock oder *Gottsched* für die Aufklärung, die zwar die Literatur ihrer Zeit entscheidend beeinflussten, bleiben jedoch der eingehenden Behandlung in einem Werk der Literaturgeschichte vorbehalten.

## Gattungsformen

Die einzelnen Werke der Dichter müssen im Computer-Programm diesen drei Gattungsformen richtig zugeordnet werden:

EPIK  
LYRIK  
DRAMATIK

Die EPIK umfaßt die erzählende Dichtung, die nochmals unterteilt wird:

**Epos:** Ursprüngliche Form erzählender Dichtung, die das Leben eines Volkes oder Helden in Versform wiedergibt (Höfische Klassik).

**Roman:** Umfangreiche Prosa-Darstellung auf verschiedenen Handlungsebenen mit breit ausgeführten Passagen. Eine besonders wichtige Rolle im behandelten Zeitraum spielt der Entwicklungsroman (Erziehungsroman, Bildungsroman), der den Reifeprozess eines Menschen im Laufe seines Lebens verfolgt.

Beispiele:

*Goethe*, »*Wilhelm Meister*«

*Keller*, »*Der grüne Heinrich*«.

**Novelle:** Sie berichtet in äußerst knapper, straffer Form, auf einer einzigen Handlungsebene, von einem ungewöhnlichen Ereignis (Wendepunkt) im Leben des Menschen.

Beispiele:

*Kleist*, »*Michael Kohlhaas*«

*Droste-Hülshoff*, »*Die Judenbuche*«.

**Erzählung:** Ein Abschnitt des Lebens wird kurz, in schlichter, aber anschaulicher Sprache beschrieben.

Beispiel:

*Stifter, »Bunte Steine«.*

Kurzgeschichte, Märchen, Legende, Sage und Anekdote gehören ebenfalls zur EPIK. Sie seien der Vollständigkeit halber hier erwähnt, obwohl sie in diesem Programm nicht erscheinen.

Die LYRIK vermittelt Gefühle, Empfindungen und Stimmungen in gebundener Form. Als Ausdrucksmittel dienen vor allem Vers, Reim und Rhythmus.

Zur LYRIK zählen Ballade, Lied, Ode, Elegie, Hymne und Sonett.

**Ballade:** Ursprünglich ein Tanzlied, heute ein Gedicht, das düsteres, unheimliches Geschehen widerspiegelt. Die Ballade vereinigt lyrische (Rhythmus), epische (Erzählweise) und dramatische (Dialoge) Elemente.

Beispiele:

*Goethe, »Erkönig«*

*Schiller, »Die Bürgschaft«.*

Sie spielt im aufgeführten Zeitraum eine bedeutende Rolle.

**Elegie:** Ein Gedicht mit wehmütig-klagendem Inhalt.

»Elegie« von *Walther von der Vogelweide* (Höfische Klassik) ist hier nicht nur Name des Werkes, sondern auch gleichzeitig Gattungsform.

Auf dem Gebiet der DRAMATIK entstehen, insbesondere in der Klassik, bedeutende Dichtungen.

**Drama:** Es stellt eine spannungsreiche Handlung in Dialog und Monolog dar, die auf der Bühne durch die schauspielerische Leistung unterstützt wird. Das klassische Drama gliedert sich in

- Einleitung
- Mittelteil (Verwicklung, Höhepunkt, Krisis)
- Entscheidung.

Von den 3 Einheiten

- der Zeit (Ablauf des Geschehens in 24 Stunden)
- des Ortes (gleichbleibender Schauplatz)
- der Handlung (Abstimmung der Nebenhandlungen auf eine Haupt-handlung)

ist seit *Lessing* allein die Einheit der Handlung noch gültig.

Zum Drama gehören Tragödie und Komödie:

**Tragödie:** Das Trauerspiel beinhaltet den Kampf des Helden gegen seine Leidenschaften, die Umwelt oder das Schicksal und endet mit seinem Untergang.

Beispiel:

*Schiller, »Maria Stuart«.*

**Komödie:** Das Lustspiel entlarvt mit Humor und Ironie menschliche Schwächen und löst so den Konflikt.

Beispiel:

*Lessing, »Minna von Barnhelm«.*

## Literaturverständnis

»Literatur« leitet sich vom lateinischen Wort für »Buchstabe« (*littera*) ab. Aus den nur 26 Buchstaben unseres Alphabets, die in zahllosen Kombinationen aneinandergereiht werden, entstehen Wörter, Sätze, Texte; eine kaum vorstellbare Fülle von Schrifttum. Wer sich die Mühe gemacht hat, sich einen kleinen Überblick zu verschaffen, will meist mehr erfahren. Vielleicht wünscht man sich, angeregt durch dieses Computer-Programm, weitere Informationen über eine Epoche oder das Leben eines Dichters. Eventuell wird man auch inspiriert, sich mit einem der angegebenen Werke näher zu befassen. Ziel dieses Programmes ist es, neben der Vermittlung von rein formalem Grundwissen Verständnis und Interesse für die deutsche Literatur zu wecken.

## Programmablauf

### Wer macht mit?

Bei diesem Literatur-Test sind maximal 5 Personen zugelassen. Der Computer bittet um die Namen. Geben Sie aus Gründen des Datenschutzes nichts ein, also drücken Sie lediglich die **RETURN**-Taste, so stört das den Computer nicht. Während des Tests werden Sie dann allerdings auch nicht »persönlich angesprochen«! Dies gilt nur für Solo-Spieler. Drängen sich mehrere um das Gerät, sollten Sie Ihren Computer schon mit irgend etwas füttern; wenigstens mit Ihrem Künstlernamen! Lassen Sie ihn hungern, indem Sie lediglich die **RETURN**-Taste drücken, so springt er nämlich direkt ins Menü. Sicher haben Sie schon »gehört«, daß alle Ihre Eingaben mit einem »Piep« quittiert werden.

### Wer die Wahl hat, hat die Qual

Soll ein Dichter vorgegeben werden, dem Sie das richtige Werk zuordnen, oder wollen Sie es lieber umgekehrt?

<b>GRUNDWISSEN: DEUTSCHE LITERATUR</b>	
<b>GEGEBEN:</b>	<b>GESUCHT:</b>
<b>=1= DICHTER</b>	<b>WERKE</b>
<b>=2= WERK</b>	<b>DICHTER</b>
<b>ANZAHL DER VORGEGEBENEN DICHTER? 5</b>	
<b>WIEVIEL SEKUNDEN PRO AUFGABE WÜNSCHEN SIE?</b>	
? <input type="text" value="15"/>	

Wünschen Sie sich einen Dichter, so drücken Sie einfach Taste 1.

Wollen Sie herausfinden, wer was geschrieben hat, so tippen Sie auf die Taste 2.

Sie müssen sich entscheiden, die anderen Tasten sind blockiert.

Sie können sich vorstellen, daß es einfacher ist, wenn nur 2 Dichter oder 2 Werke zur Auswahl stehen statt deren fünf. Geben Sie also nun den gewünschten Schwierigkeitsgrad ein. Je schwieriger Sie sich den Test gestalten, um so mehr Punkte können Sie erzielen; vorausgesetzt, Sie überfordern sich nicht. Schätzen Sie Ihre Fähigkeiten richtig ein, und verbessern Sie notfalls Ihr Grundwissen über die deutsche Literatur mit Hilfe der Tabelle »Ausgewählte Dichter und ihre Werke«.

Wählen Sie abschließend noch die Ratezeit pro Frage oder übernehmen sie einfach die Vorgabe von 15 Sekunden.

Hatten Sie Ihren Namen und/oder den von Mitspielern eingegeben, so werden Sie jetzt mit der in wechselnder Farbe blinkenden Aufforderung »LOS GEHT'S, (Name)!« in die Startlöcher komplimentiert.

In jedem Fall werden 5 Werke und die dazugehörigen Dichter ausgewählt. Auf dem Bildschirm sehen Sie aber nur so viele, und in der Anordnung, wie Sie selber festlegen.

## Die Lösung

Sie müssen nicht mehr eintippen als die den Werken oder Dichtern zugeordnete Ziffer.

Je schneller Sie die richtige Zahl eingeben, um so mehr Punkte erhalten Sie. Erscheint der Balken mit der Inschrift »EINGABEZIT ABGELAUFEN«, so können Sie zwar noch Ihre Lösung eingeben, aber Punkte gibt es dafür nicht mehr! Die richtige Lösung wird in jedem Fall angezeigt. Sie können dennoch ruhig weiterspielen, denn dem Werk sind noch die Gattungsform und die richtige Epoche zuzuordnen.

<b>GRUNDWISSEN: DEUTSCHE LITERATUR</b>
<b>ERLKOENIG V O N JOHANN WOLFGANG VON GOETHE</b>
<b>WELCHE GATTUNG?</b>
<input type="checkbox"/> 1 EPIK
<input type="checkbox"/> 2 LYRIK
<input type="checkbox"/> 3 DRAMATIK
<b>BITTE EINE ZAHL EINGEBEN!</b>

Pluspunkte

<b>WELCHER EPOCHE IST DER ERLKOENIG ZUZUORDNEN, MARKUS?</b>
<b>0. HOEFISCHE KLASSIK 1. HUMANISMUS 2. BAROCK 3. AUFKLAERUNG 4. STURM UND DRANG 5. KLASSIK 6. ZWISCHEN KLASSIK UND ROMANTIK 7. ROMANTIK 8. BIEDERMEIER 9. REALISMUS</b>
<b>BITTE EINE ZAHL EINGEBEN!</b>

Hier läuft keine Stoppuhr mehr im Hintergrund, und selbst wenn Sie im ersten Teil Pech gehabt haben sollten; hier sind noch ein paar Punkte für Sie übrig! Präzise gesagt, 5 für die richtige Gattung und 15 für die Eingabe der passenden Epoche.

Sicher wollen Sie nun Ihren Punktestand erfahren. Rufen sie ihn durch einen Tastendruck ab. Er könnte folgendermaßen aussehen:



## **Epochenüberblick**

Die Epochen sind zeitlich meist nicht exakt abgrenzbar; oft herrschen unterschiedliche Auffassungen. Die Übergänge sind fließend; manche Epochen überschneiden sich auch zeitlich oder laufen parallel (z.B. Klassik und Romantik; Biedermeier und Realismus). Daher werden in der folgenden Aufstellung die Epochenangaben nicht in Jahreszahlen, sondern auf das Jahrhundert gerundet wiedergegeben.

### **Höfische Klassik (12./13. Jh.)**

Die Ritterdichtung entwickelte sich an Fürstenhöfen. Leitmotiv ist die Sage um den geheimnisvollen König Artus und seine Tafelrunde. Auf der Grundlage christlichen Lebensgefühls und der Ideale des Ritterstandes (Ehre, Treue, Minne, Beständigkeit, Selbstzucht) entstehen vollendete Werke (meist Kunstepen). Diese Epoche ist die erste Blütezeit der deutschen Dichtung.

### **Humanismus (15./16. Jh.)**

Kunst- und Lebensauffassung der Antike gelten als Vorbild. Anstelle des Autoritätsglaubens tritt kritische Forschung.

### **Barock (17. Jh.)**

Das Barock ist durch starke Spannungen und den Gegensatz zwischen Lebensfreude und Vergänglichkeitsbewußtsein gekennzeichnet. Eine Fülle von Bildern und gesteigertes Pathos prägen die Sprache.

### **Aufklärung (18. Jh.)**

Rationalismus (Bsp.: Lessing), Pietismus (Bsp.: Klopstock) und Rokoko (Bsp.: Wieland) sind die drei großen Strömungen der Aufklärung. Die Dichtung wendet sich an Verstand und kritisches Bewußtsein; sie setzt sich Belehrung und Besserung des Menschen zum Ziel und zeigt Harmonie und Lebensfreude.

### **Sturm und Drang (Ende des 18. Jhs.)**

Grundthema der Dichtung ist der ungebändigte Freiheitsdrang der jungen Generation und ihre Auflehnung gegen die Herrschaft der Vernunft, gegen Konventionen und Standesvorurteile. Die sprachliche Gestaltung zeigt Gefühl und Phantasie, wirkt ursprünglich, lebendig und überschwenglich.

### **Klassik (Wende vom 18. zum 19. Jh.)**

»Edle Einfalt und stille Größe« sind Kennzeichen antiker Kunst, dem Vorbild der Klassik. Es entstehen Werke, die in ihrer Ausgewogenheit, Gesetzmäßigkeit der Form und aufgrund der differenzierten Hochsprache als vollkommen gelten. Hauptthema ist die Entwicklung des Menschen zur edlen Persönlichkeit. Harmonie von Körper und Geist,



Freiheit und Menschlichkeit sind Ideale der Klassik. Diese Epoche ist die zweite Blütezeit der deutschen Dichtung.

**Romantik** (1. Hälfte des 19. Jhs.)

Sehnsucht nach der Vergangenheit, der Ferne und Unendlichkeit führt zur Auseinandersetzung mit anderen Epochen und Kulturen. Phantasie und Traum schaffen geheimnisvolle und märchenhafte Welten. Grenzenloses Gefühl sprengt alle formalen und sprachlichen Gesetzmäßigkeiten der Dichtung.

**Biedermeier** (1. Hälfte des 19. Jhs.)

Maßvolle Bescheidenheit, Beschränkung auf den engen, schlichten Wirkungskreis, Rückzug in die bürgerliche Idylle, Flucht in die Erinnerung sind Wesensmerkmale der Epoche, deren Lebensgefühl die Resignation ist.

**Realismus** (19. Jh.)

Neue wirtschaftliche und weltanschauliche Tendenzen führen zu wirklichkeitsbewußter, diesseitsorientierter Lebenseinstellung und zu objektiver Wiedergabe der Realität. Charakteristisch sind der Sinn für Natur, Geschichte und Volkstum und die Einordnung des Menschen in Staat und Gesellschaft.

**Ausgewählte Dichter und ihre Werke**

DICHTER	EPOCHE/WERK	GATTUNG* Kennzahl
	<b>Höfische Klassik</b>	
Hartmann von der Aue	Iwein	E 0101
	Der arme Heinrich	E 0101
Wolfram von Eschenbach	Parzival	E 0102
Gottfried von Straßburg	Tristan und Isolde	E 0103
Walther von der Vogelweide	Elegie	L 0204
	<b>Humanismus</b>	
Johann von Tepl	Der Ackermann aus Böhmen	E 1105
	<b>Barock</b>	
Christoph von Grimmelshausen	Simplicissimus	E 2106
	<b>Aufklärung</b>	
Gotthold Ephraim Lessing	Minna von Barnhelm	D 3307
	Emilia Galotti	D 3307
	Nathan der Weise	D 3307
Friedrich Gottlieb Klopstock	Der Messias	E 3108
Christoph Martin Wieland	Agathon	E 3109
	<b>Sturm und Drang</b>	
Johann Wolfgang von Goethe	Die Leiden des jungen Werther	E 4110
	Urfaust	D 4310
	Götz von Berlichingen	D 4310
	Prometheus	L 4210
Friedrich Schiller	Die Räuber	D 4311
	Kabale und Liebe	D 4311
	Don Carlos	D 4311
	<b>Klassik</b>	
Johann Wolfgang von Goethe	Iphigenie	D 5310
	Egmont	D 5310
	Torquato Tasso	D 5310
	Faust	D 5310
	Erlkönig	L 5210
	Hermann und Dorothea	E 5110
	Wilhelm Meister	E 5110
	Die Wahlverwandtschaften	E 5110
	West-östlicher Divan	L 5210
	Dichtung und Wahrheit	E 5110

\*E=Epik L=Lyrik D=Dramatik

DICHTER	EPOCHE/WERK	GATTUNG* Kennzahl
Friedrich Schiller	Der Taucher	L 5211
	Die Kraniche des Ibykus	L 5211
	Die Bürgschaft	L 5211
	Das Lied von der Glocke	L 5211
	Wallenstein	D 5311
	Maria Stuart	D 5311
	Die Jungfrau von Orleans	D 5311
	Die Braut von Messina	D 5311
	Wilhelm Tell	D 5311
<b>Zwischen Klassik und Romantik</b>		
Friedrich Hölderlin	Hyperion	E 6112
Heinrich von Kleist	Michael Kohlhaas	E 6113
	Der zerbrochene Krug	D 6313
	Das Käthchen von Heilbronn	D 6313
	Prinz Friedrich von Homburg	D 6313
Jean Paul	Titan	E 6114
<b>Romantik</b>		
Novalis	Heinrich von Ofterdingen	E 7115
	Hymnen an die Nacht	L 7215
Joseph Freiherr von Eichendorff	Aus dem Leben eines Taugenichts	E 7116
	Ahnung und Gegenwart	E 7116
E. T. A. Hoffmann	Das Fräulein von Scudery	E 7117
<b>Biedermeier</b>		
Eduard Mörike	Maler Nolten	E 8118
	Mozart auf der Reise nach Prag	E 8118
	Der Feuerreiter	L 8218
Adalbert Stifter	Der Hochwald	E 8119
	Bunte Steine	E 8119
	Nachsommer	E 8119
	Witiko	E 8119
Grillparzer	Das goldene Vlies	D 8320
	Ein Bruderzwist im Hause Habsburg	D 8320
	Weh dem der lügt	D 8320
Annette von Droste-Hülshoff	Das Geistliche Jahr	L 8221
	Heidebilder	L 8221
	Der Knabe im Moor	L 8221
	Die Judenbuche	E 8121

DICHTER	EPOCHE/WERK	GATTUNG* Kennzahl
	<b>Realismus</b>	
Heinrich Heine	Deutschland. Ein Wintermärchen	E 9122
Georg Büchner	Dantons Tod	D 9323
	Woyzeck	D 9323
Gottfried Keller	Der grüne Heinrich	E 9124
	Die Leute von Seldwyla	E 9124
Conrad Ferdinand Meyer	Das Amulett	E 9125
	Gustav Adolfs Page	E 9125
	Der Schuß von der Kanzel	E 9125
Gustav Freytag	Soll und Haben	E 9126
Jeremias Gotthelf	Die schwarze Spinne	E 9127
	Uli der Knecht	E 9127
	Uli der Pächter	E 9127
Friedrich Hebbel	Judith	D 9328
	Maria Magdalena	D 9328
	Agnes Bernauer	D 9328
	Die Nibelungen	D 9328
Theodor Storm	Immensee	E 9129
	Pole Poppenspüler	E 9129
	Der Schimmelreiter	E 9129
Wilhelm Raabe	Die Chronik der Sperlingsgasse	E 9130
	Die schwarze Galeere	E 9130
	Der Hungerpastor	E 9130
Theodor Fontane	Die Brücke am Tay	L 9231
	Irrungen, Wirrungen	E 9131
	Effi Briest	E 9131
	Der Stechlin	E 9131

## Programmbeschreibung

Zeile 115:

Sie sorgt dafür, daß die erste Zufallszahl nach dem Programmstart wirklich immer verschieden ist.

Zeilen 116 – 120:

Da DW = 100 ist, werden die Werke auf WE\$ (100) DIMensioniert, das heißt begrenzt; ebenso die Sperre H (100), die in diesem Programm Doppelnennungen verhindert. We-

gen der geringeren Anzahl der Dichter DI\$ (50) genügt 50. Die Indizes gelten für die Daten dieses Programmes. Werden zusätzliche Werke oder Dichter eingegeben und dabei die Grenzzahlen überschritten, so ist die DIMensionierung entsprechend anzupassen. Da der Stern auch mitzählt, muß der Index immer um 1 höher liegen als die betreffende Datenmenge.

Zeile 130:

POKE 53281,1 ist der »Weißmacher« für den Bildschirm-Hintergrund.

Zeile 140:

CHR\$(9) entriegelt die **SHIFT**-Taste, CHR\$(142) schaltet in den Großschrift-Modus, und CHR\$(8) verriegelt die **SHIFT**-Taste wieder.

Zeile 145:

In R = 53280 ist die Zahl für die Zelle des Bildschirm-Rahmens abgelegt. Da im Laufe des Programms die Rahmenfarbe mehrfach wechselt, erspart man sich so etwas Schreiarbeit. Aus eben diesem Grunde ist hier auch SZ = 58732 abgelegt, die Zahl für SYS. Die Position der nächsten PRINT-Ausgabe wird öfter gebraucht. Mit S = 211 wird die Spalte und mit Z = 214 die Zeile gePOKEt.

Beispiel:

Mit POKE S,3:POKE Z,5:SYS SZ wird das folgende PRINT in Spalte 3 – dies entspricht einem PRINTTAB(3) – und Zeile 5 ausgegeben. Letzteres entspricht 5mal »CURSOR nach unten«, falls dieser vorher in HOME-Stellung gebracht wurde.

Zeilen 180 – 240:

Um Ihnen die Arbeit des Abtippens etwas zu erleichtern, arbeite ich mit *Character-Strings*. Im Commodore 64 Handbuch, Anhang F (Seiten 135 – 137) finden Sie die Tabelle, die Ihnen die CHR\$Codes näher bringt.

BL\$ = CHR\$(147) = Bildschirm löschen  
CL\$ = CHR\$(157) = CURSOR nach links  
CR\$ = CHR\$(29) = CURSOR nach rechts  
CO\$ = CHR\$(145) = CURSOR nach oben  
FB\$ = CHR\$(31) = Zeichenfarbe blau  
FS\$ = CHR\$(144) = Zeichenfarbe schwarz  
FR\$ = CHR\$(28) = Zeichenfarbe rot  
FG\$ = CHR\$(30) = Zeichenfarbe grün  
RA\$ = CHR\$(18) = RVS ON (negative Zeichenwiedergabe einschalten)  
RE\$ = CHR\$(146) = RVS OFF (negative Zeichenwiedergabe ausschalten)

Die Steuerzeichen werden, wie wir in Zeile 240 sehen, mit Hilfe der PRINT-Anweisung gesendet.

Zeilen 275 – 360:

CHR\$ (113) beinhaltet einen Ball. In einer Schleife 40mal aneinandergereiht ergibt sich eine Zeile mit Bällen, die wir später noch brauchen werden.

Auch die folgenden Zeichenketten sind sehr gefragt und werden daher schon hier festgelegt.

Z2\$ beispielsweise gibt den Text von Z1\$ in Negativdarstellung wieder.

TD\$ beinhaltet in einem grünen Balken die Aufforderung »BITTE EINE BELIEBIGE TASTE DRÜCKEN«.

RL\$ ist die linke und rechte Begrenzung für ein Kästchen.

LL\$ erscheint später auf dem Bildschirm im unteren Viertel einer Zeile als Linie.

T1\$ ist zunächst eine Mittellinie, wird aber in der folgenden Zeile zur oberen Begrenzung eines Kästchens erweitert.

T2\$ ist die Titelzeile und T3\$ die untere Begrenzung eines Kästchens.

TT\$ faßt T1\$ – T3\$ zum Titelkopf zusammen.

Zeilen 410 – 500:

Unterhalb des Titelkopfes wird mittels eines roten Balkens mit Inschrift um die Namen der Test-Teilnehmer gebeten.

Die Schleife von Zeile 420 – 500 gestattet die Eingabe von 5 Teilnehmern. Jeder eingetippte Name wird in der indizierten Variablen N\$ (N) gesondert festgehalten, um später weiter verwendet werden zu können. Man könnte sagen, das Programm bekommt dadurch seine »persönliche Note«.

Drücken Sie aber lediglich die **RETURN**-Taste, geben Sie also "" ein, dann springt das Programm sofort aus der Schleife und begibt sich zum nächsten Menüpunkt.

Zeilen 520 – 610:

Dieser Abschnitt beginnt mit einigen PRINT-Anweisungen, gefolgt von einem GET, damit wir 1 oder 2 eingeben können. Interessanter sind die Zeilen 580 und 590: Erinnern Sie sich noch an Z1\$ und Z2\$? Hier blinkt nun die Aufforderung, eine Zahl einzugeben!

Die Warteschleife bremst die Blinkgeschwindigkeit, und je 2mal CO\$ (CURSOR nach oben) sorgt dafür, daß sich die beiden Zeilen überlagern und somit eine Zeile die andere wieder löscht (Blinkeffekt).

Die Umwandlung eines Strings in eine Zahl erfolgt mehrfach in diesem Programm, da ich bei versehentlich falscher Eingabe keine REDO FROM START-Meldung erhalten möchte. Zeile 610 verhindert schließlich, daß unbrauchbare Eingaben akzeptiert werden.

Zeile 625:

Hier treffen wir erstmals in diesem Programm auf eine Positionierungs-Anweisung. Erinnern Sie sich noch an die Zeile 145? Das nächste PRINT erscheint demnach in Bildschirmzeile 13. Vorher gehen wir aber noch in das Unterprogramm, welches in Zeile 2303 beginnt. GOSUB = GO to SUBroutine = GEHE ins UNTERprogramm.

Diese Subroutine erzeugt den Piepston, der jede Eingabe bestätigt.

Zeilen 630 – 680:

POKE R,6 läßt den Bildschirmrahmen blau werden, falls Sie sich entschieden, Dichter auszuwählen.

POKE 198,0:WAIT 198,1 löscht den Tastaturpuffer und wartet dann, bis ein Zeichen eingeschrieben wird.

Etwa gleiche Wirkung hätte auch folgende Zeile: 650 GET ZW\$:IF ZW\$= "" THEN 650.

Zeilen 710 – 720:

Diese Zeilen entsprechen einem INPUT mit Vorgabe.

Wie Sie wissen, kann man diese Vorgabe auch folgendermaßen realisieren: INPUT"Textvorgabe";SE

Um aber die Negativdarstellung und die Farbgebung zu implementieren, müßten commodore-eigene Steuerzeichen verwendet werden. Auch für die Vorgabe der Zahl 15 würde man diese Zeichen benötigen.

Das sähe so aus:

```

                INPUT"#####15#####";SE
                |
3x CURSOR nach rechts ─┘ ┌ 4x CURSOR nach links
                |
                2 Zeichen
    
```

Allgemein gilt: Anzahl der CURSOR-Bewegungen nach rechts = Anzahl der Zeichen der Vorgabe + 1.  
 Anzahl der CURSOR-Bewegungen nach links = Anzahl der Zeichen der Vorgabe + 2.

Sehen Sie sich nun die hier leicht veränderte Zeile 715 an:

```
PRINT;RE$;F$;CR$;CR$;CR$;"15";CL$;CL$;CL$;CL$;:INPUT SE
```

Zeile 725:

Wünschen Sie nach einer Spielrunde eine Rückkehr in das Menü, so bewirkt diese Zeile, daß die Daten nicht erneut eingelesen werden.

Zeile 760 – 1030:

760 – 780: Festlegung der Gattungsbezeichnungen.

820 – 910: Festlegung der Epochenbezeichnungen.

950 – 970: Einlesen der Dichter mit Hilfe der READ-Anweisung.

Dieses Einlesen wird erst beendet, wenn ein vorher festgelegtes Zeichen gelesen wird; in unserem Fall der Stern. Natürlich könnte dieses »Kenn«-Zeichen auch durch ein »Kenn«-Wort ersetzt werden.

1010 – 1030: Einlesen der Werke.

Zeilen 1070 – 1190:

WX ist eine Zufallszahl, die durch die Anzahl der in den DATA-Zeilen aufgeführten Werke begrenzt wird.

H ist die zweistellige Dichterkennzahl aus der vierstelligen Kennzahl, die jedem Werk vorangestellt ist.

Sehen Sie sich die DATAs ab Zeile 2680 an!

Dieser Wert wird als Index für H(H) verwendet.

Auf diese Weise wird ausgeschlossen, daß von einem bestimmten Dichter mehr als ein Werk ausgewählt wird.

In den folgenden Zeilen werden nun die Hilfsvariablen erstellt. Sie enthalten die Kennzahlen des jeweiligen zufallsgewählten Werkes:

$E(V)$  = Epochenkennzahl dieses Werkes.

$G(V)$  = Gattungskennzahl dieses Werkes.

$D(V)$  = Dichterkennzahl dieses Werkes.

$A\$(V)$  = Werk ohne Kennziffer.

Mit Hilfe der Dichterkennzahl kann der zugehörige Dichtername ermittelt werden:  $DI\$(D(V))$ .

Da, wie wir hörten, jeder Dichter nur mit einem Werk vertreten ist, erhielten wir mit den 5 Werken auch 5 Dichter. Die Zeile 1190 sorgt übrigens dafür, daß es bei 5 Dichtern und Werken bleibt.

Zeilen 1230 – 1235:

ZW ist eine Zahl zwischen 2 und 5. Ihr aktueller Wert wurde in Zeile 650 mit GET festgelegt.

X ist eine Zufallszahl zwischen 1 und ZW. Dieses X fischt uns als Index des Dichter- bzw. Werke-Strings einen Dichter bzw. ein Werk heraus.

Zeilen 1270 – 1360:

Wir befinden uns nun im bereits angesprochenen »persönlichen« Teil unseres Programmes.



Die erste Programmzeile sorgt dafür, daß nach Aufruf aller eingegebenen Namen wieder von vorne begonnen wird.

1290: Bildschirm löschen und Druck des Programmkopfs.

POKE 646,I+2 färbt die blinkende Aufforderung "LOS GEHT'S, (Name)" verschieden ein (Schleife!).

Den namentlichen Aufruf erledigt N\$(NN).

1340: Jeder Farbbalken bleibt 300 Zählerinheiten lang auf dem Schirm.

Zeilen 1420 – 1660:

Nach all diesen Vorarbeiten beginnt nun der eigentliche Literatur-Test.

Die Zeilen 1420 – 1540 befassen sich mit der Anzeige eines Werkes und mehrerer Dichter.

Die Programmzeilen 1550 – 1640 sind ähnlich aufgebaut. Sie zeigen aber einen Dichter und mehrere Werke an. Auf Grund der zahlreichen anderen Maßnahmen, die das Display freundlicher gestalten, bringt hier ein Dreiecksaustausch keinen echten Programmervorteil.

1435 und 1555: Werk/e und Dichter werden eingerahmt.

1445 und 1570: AX und BX liefern einen Wert, der zusammen mit PRINTTAB() den/die Dichternamen bzw. das Werk oder die Werke zentriert auf den Bildschirm PRINTet.

Zeilen 1700 – 1740:

Der Faktor »Zeit« soll in unserem Test auch eine Rolle spielen: TI\$ wird auf Null gesetzt. RZ% = Restzeit in Form einer Ganzzahl(%=INTEger). Ist unsere restliche Ratezeit abgelaufen, so zeigt uns dies Zeile 1730 mit dem inversen Schriftzug " \* \* \* EINGABEZIT ABGELAUFEN \* \* \* " an.

Zeilen 1800 – 1860:

Hier müssen Dichter und Werk richtig zugeordnet werden. Die der GET-Abfrage folgenden 3 Zeilen sollen unsinnige Eingaben verhindern. Bei richtiger Eingabe wird in Zeile 1850 die errungene Punktezahl berechnet. Sie ist das Produkt aus einem empirisch ermittelten Faktor, der Zahl der angezeigten Werke und der Rest-Ratezeit zum Zeitpunkt unserer Eingabe, dividiert durch das Quadrat der gewählten Maximal-Ratezeit in Sekunden. Die Verwendung möglichst vieler Variablen ermöglicht eine individuelle Punktebewertung.

1860: Natürlich werden wir, wer hätte es anders erwartet, mit

"\*\*\*\*\* R I C H T I G \*\*\*\*\*" gelobt. Dieses RICHTIG flackert, wie den Zeilen 1860 und 1865 zu entnehmen ist, in den Farben des Regenbogens. Zeile 1863 bewirkt die Überlagerung von Positiv- und Negativschrift und sorgt nach 8maligem Wechsel für den Fortgang des Programms..

Zeilen 1880 – 1935:

*Nobody is perfect.* Deshalb brauchen wir auch diese Zeilen:

”DAS WAR LEIDER FALSCH!  
ES MUSS HEISSEN:”

In einem Kästchen wird die richtige Lösung eingeblendet.

Damit wir Zeit haben, uns die richtige Lösung einzuprägen, geht es erst nach einem Tastendruck – Zeile 1935 – weiter.

Zeilen 1955 – 2026:

Erst einmal wird der Bildschirmrahmen gelb: POKE R,7.

1960: Ein Rahmen wird »gezimmert«.

1965: Werk und Dichter werden in diesen Rahmen geschrieben.

1970: Es wird nach der Gattung gefragt.

1980: Die 3 Gattungsbegriffe werden angeboten.

1990: Aufforderung, eine Zahl (Z1\$) einzugeben).

1991: Sprung in das Unterprogramm: Bestätigung der Eingabe durch Piepston.

2000: Wurde die Gattung richtig eingegeben, gibt es fünf Punkte.

2005: Positionierung auf Bildschirmzeile 13.

2010 – 2013: Mehrfarbig blinkende Bestätigung der richtigen Eingabe.

2016 – 2021: Anzeige der richtigen Lösung bei Fehleingabe.

Zeilen 2055 – 2115:

Der Programm-Abschnitt Epoche ähnelt dem Abschnitt Gattung. Unterschiede finden Sie in der Farbgebung, im Text bei richtiger und falscher Eingabe sowie in der Anordnung der Textelemente.

Zeilen 2160 – 2290:

2160: Warten, bis ein Zeichen im Tastaturpuffer ist.

2161: Sprung ins Unterprogramm. Rahmenfarbe türkis (*cyan*).

2165: Bildschirm löschen. Drei Zeilen Bälle (BA\$) mit der Inschrift ”PUNKTESTAND”.

2170: Rundenzähler RD.

2212: Berechnung der Gesamtpunktzahl durch Aufaddieren der Rundenpunkte PK(NN). Der Index NN bewirkt, daß die Punkte den einzelnen Spielern zugeordnet werden können.

2213 – 2214: Ausdruck der Namen, der Rundenpunkte und Gesamtpunkte.

2215 – 2218: Ermitteln der Rekordpunktzahl (Hi Score) und Ausdruck.

2219 – 2220: Name und Punktstand des jeweiligen Spielers werden in Negativschrift an der richtigen Stelle der Spielertabelle durch Überlagerung eingeblendet.

2222: Information über Möglichkeit der Rückkehr ins Menü, des Spielabbruchs und Weiterspielens.

2242 – 2290: Vorbereitungen für die nächste Spielrunde. Sprünge gemäß der vom Anwender in Zeile 2040 gemachten Eingabe.

Zeilen 2303 – 2306:

In diesem Unterprogramm wird ein Piepston erzeugt.

S1 = 54272 = Basisadresse des SID (*Sound Interface Device*).

POKE S1+24,15: Lautstärke einschalten. 24 = 1. bis 3. Stimme.  
15 = laut (0-15; leise → laut).

POKE S1+11,17: Ton einschalten. 11 = 2. Stimme.  
17 = Wellenform: Dreieck.

POKE S1+7,207: 7 = 2. Stimme. 207 = Frequenz, Low-Byte.

POKE S1 +8,34: 8 = 2. Stimme. 34 = Frequenz, High-Byte.

POKE S1+13,240: Ton halten. 13 = 2. Stimme. 240 = 15 x 16.  
15 = laut.

PO = 1 TO 50: Warteschleife für Tondauer.

POKE S1+11,0: Ton ausschalten. 11 = 2. Stimme. 0 = aus.

} Note C  
5. Oktave

Wollen Sie diesen Piepston verändern, so sehen Sie sich im Commodore 64 Handbuch, Anhang O+P (Seiten 155 – 159) die Tabellen an.

Zeilen 2340 – 2630:

Hier sind die Namen der Dichter eingegeben. Die Dateneingabe muß mit einem Stern in Anführungszeichen abschließen. Sie können natürlich auch ein anderes Daten-schluß-Zeichen verwenden. Ändern Sie dann aber unbedingt auch die Programmzeilen 970 und 1030.

Weitere Dichternamen können Sie nach Zeile 2625 einfügen.

Zeilen 2670 – 5000:

Die erste Zeile erklärt Ihnen nochmal die 4-stellige Kennzahl:

Unter E steht die Epochen-Kennzahl (0 – 9).

Unter G finden Sie die Gattungs-Kennzahl (1 – 3).

Unter DI schreiben Sie die 2-stellige Dichter-Kennzahl (01 – 31).

Die passende Dichter-Kennzahl erhält man, wenn man auszählt, an welcher Stelle in den DATAs der Dichtername steht. Maximal können 99 Dichter und eine beliebige Anzahl von Werken eingegeben werden.

Ein Werk-Name soll, einschließlich Leerzeichen, weniger als 35 Zeichen umfassen.

Kommt darin ein Komma oder Doppelpunkt vor, wie in Zeile 3490, so ist die Eingabe zwischen Anführungszeichen zu setzen.

Die letzte Zeile lautet auch hier xxxx DATA ”\*”.

(Gertrud und Michael Geipel)

```
10 REM *****
20 REM *
30 REM * DEUTSCHE LITERATUR *
40 REM * - GRUNDWISSEN - *
60 REM *
70 REM *****
80 :
85 REM COPYRIGHT 1985 BY
90 REM MODERNE VERLAGS GESELLSCHAFT
95 REM AUTOR: MICHAEL GEIPEL
110 :
115 X=INT(-TI)
116 DW=100
120 DIM WE$(DW),DI$(DW/2),H(DW)
130 POKE 53281,1
140 PRINT CHR$(9);CHR$(142);CHR$(8)
145 R = 53280: SZ = 58732: S = 211: Z =
214
150 :
160 REM ** STEUERZEICHENDEFINITIONEN ***
170 :
180 BL$= CHR$(147)
190 CL$= CHR$(157) : CR$= CHR$(29)
200 CO$= CHR$(145) : CU$= CHR$(17)
210 FB$= CHR$(31) : FS$= CHR$(144)
220 FR$= CHR$(28) : FG$= CHR$(30)
230 RA$= CHR$(18) : RE$= CHR$(146)
240 PRINT FS$;BL$
250 :
260 REM ** WEITERE STRINGDEFINITIONEN **
264 :
265 FOR B=1 TO 9: BA$=BA$+CHR$(113):NEXT
270 R1$=CHR$(32)+BA$+" R I C H T I G
"+BA$+CHR$(32)
275 BA$="":FOR B=1 TO 40: BA$=BA$+RA$+CH
R$(113)+RE$:NEXT
280 Z1$=" BITTE EINE ZAHL EINGEBEN
!
"
290 Z2$=RA$+Z1$+FS$+RE$
300 TD$=RA$+FG$+ " BITTE EINE BELIEBIGE
TASTE DRUECKEN! "+FS$+RE$
315 RL$=CHR$(98)+"
"+CHR$(98)
320 FOR LL=1TO 40:LL$=LL$+CHR$(175):NEXT
321 FOR T1=1TO 38:T0$=T0$+CHR$(99):NEXT
```

```

330 T1$=CHR$(176)+T0$+CHR$(174)
340 T2$=CHR$(98)+" GRUNDWISSEN: DEUTSC
HE LITERATUR "+CHR$(98)
350 T3$=CHR$(173)+T0$+CHR$(189)
360 TT$=T1$+T2$+T3$
370 :
380 REM ***** MENUE *****
390 :
400 PRINT TT$
410 PRINT " ";RA$;FR$;" DIE NAMEN DER TES
T-TEILNEHMER BITTE! ";RE$;FS$:PRINT
420 FOR N=1 TO 5
440 INPUT " ";N$(N):GOSUB 2303
460 IF N$(N)=""THEN 505
470 N1$(N)=" LOS GEHT'S, "+N$(N)+"!"
480 N2$(N)=N$(N)+" HAT "
490 PRINTCO$;N;CL$;". "
500 NEXT
505 PRINT BL$;TT$
520 PRINTTAB(46)"GEGEBEN: GESUCHT: ";P
RINT LL$
530 PRINTCHR$(32);RA$" =1=";RE$;FB$;" DI
CHTER ";FS$;" ";FG$;"WERKE";FS$
540 PRINT
550 PRINTCHR$(32);RA$" =2=";RE$;FG$;" WE
RK ";FS$;" ";FB$;"DICHTER";FS$
560 PRINT LL$
570 GET AU$
580 FOR B =1 TO 100:NEXT:PRINT Z1$;CO$;C
O$
590 FOR B =1 TO 100:NEXT:PRINT Z2$;CO$;C
O$
600 AU=VAL(AU$)
610 IF AU=1 OR AU=2 THEN 625
620 GOTO 570
625 POKE Z,13:SYS SZ :GOSUB 2303
630 IF AU=1 THEN PRINTRAS$;FG$;" ANZAHL D
ER VORGEGEBENEN WERKE? ";RE$;GOTO 650
640 POKE R,6:PRINTRAS$;FB$;" ANZAHL DER V
ORGEGEBENEN DICHTER? ";RE$;FS$;
650 PRINT:POKE 198,0:WAIT 198,1:GET ZW$:
PRINT TAB(35);CO$;CO$;ZW$
651 GOSUB 2303
660 ZW=VAL(ZW$)
680 IF ZW<=1 OR ZW=>6 THEN 625

```

```
710 PRINT:PRINT RA$;FR$;" WIEVIEL SEKUND
EN PRO AUFGABE WUENSCHEN ";
711 PRINTRA$;"
      ";RE$;CO$;
712 PRINT RA$;FR$;" SIE?";RE$;FS$
715 PRINT:PRINT;RE$;FS$;CR$;CR$;CR$;"15"
;CL$;CL$;CL$;CL$;
720 INPUT SE:GOSUB 2303:IF SE=0 THEN PRI
NT CO$;;GOTO 720:PRINT L$
725 IF WT$="M" THEN 1070
730 :
740 REM *** VARIABLEN : GATTUNG *****
750 :
760 G$(1)="EPIK"
770 G$(2)="LYRIK"
780 G$(3)="DRAMATIK"
790 :
800 REM *** VARIABLEN : EPOCHE *****
810 :
820 E$(0)="HOEFISCHE KLASSIK"
830 E$(1)="HUMANISMUS"
840 E$(2)="BAROCK"
850 E$(3)="AUFKLAERUNG"
860 E$(4)="STURM UND DRANG"
870 E$(5)="KLASSIK"
880 E$(6)="ZWISCHEN KLASSIK UND ROMANTIK
"
890 E$(7)="ROMANTIK"
900 E$(8)="BIEDERMEIER"
910 E$(9)="REALISMUS"
920 :
930 REM ***** DICHTER EINLESEN *****
940 :
950 DI=DI+1
960 READ DI$(DI)
970 IF DI$(DI)<>"*" THEN 950
980 :
990 REM ***** WERKE EINLESEN *****
1000 :
1010 WE=WE+1
1020 READ WE$(WE)
1030 IF WE$(WE)<>"*" THEN 1010
1040 :
1050 REM ** ZUFALLSAUSWAHL:
      5 WERKE UND 5 DICHTER *****
```

```

1060 :
1070 WX=INT(RND(1)*(WE-1))+1
1080 H=VAL(MID$(WE$(WX),3,2))
1090 IF H(H)=1 THEN 1070
1100 H(H)=1
1110 V=V+1
1130 A=LEN(WE$(WX))
1140 A$(V)=RIGHT$(WE$(WX),A-5)
1150 E(V)=VAL(LEFT$(WE$(WX),1))
1160 G(V)=VAL(MID$(WE$(WX),2,1))
1170 D(V)=VAL(MID$(WE$(WX),3,2))
1180 B$(V)=DI$(D(V))
1190 IF V <=6 THEN 1070
1200 :
1210 REM ** ZUFALLSAUSWAHL:
           1 WERK ODER 1 DICHTER *****
1220 :
1230 X=INT(RND(1)*ZW)+1
1235 NM=NM+1:IF NM=>N THEN MN=1
1240 :
1250 REM ***** LOS GEHT'S *****
1260 :
1270 IF NN=>N-1 THEN NN=0
1280 NN=NN+1
1290 PRINTBL$;TT$
1300 FOR I=1 TO 4
1310 POKE646,I+2
1320 PRINT:IF I=5 THEN PRINT FS$;
1330 PRINTRAS$;N1$(NN);RE$
1335 IF LEN(N$(NN))>0 THEN N2$="," "+N$(N
N) : N3$=N$(NN)+", "
1336 N1$=N$(NN)
1340 FOR U=1TO300:NEXT
1350 PRINTCO$;CO$;FS$;
1360 NEXT I
1370 V=0
1380 :
1390 REM ** BILDSCHIRMANZEIGE *****
1400 REM ** 1 WERK - MEHRERE DICHTER ***
1410 :
1420 IF AU=1 THEN 1550
1430 POKE R,6:PRINT FG$;T1$
1435 PRINT CO$;RL$;RL$;T3$;CO$;CO$
1440 POKE S,15:POKE Z,4:SYS SZ:PRINT"WER
K"

```

```
1445 AX=LEN(A$(X)):AX=(38-AX)/2
1450 PRINTTAB(AX)FS$;A$(X);FG$
1460 PRINT
1470 PRINTFB$;T1$
1480 PRINTTAB(15)CO$;CO$;"DICHTER"
1490 FOR U=1 TO ZW
1500 PRINT CO$;RL$;CO$:PRINTCHR$(98);RA$
"-";U;CL$;"-";RE$;CO$
1505 BU=LEN(B$(U)):BU=INT(6+((32-BU)/2))
1510 PRINTTAB(BU)FS$;B$(U);FB$;CO$
1515 PRINTTAB(39)CHR$(98)
1520 NEXT U
1530 PRINT FB$;CO$;T3$;FS$
1540 GOTO 1660
1544 :
1545 REM ** 1 DICHTER - MEHRERE WERKE **
1546 :
1550 PRINTFB$;T1$
1555 PRINT CO$;RL$;RL$;T3$;CO$;CO$
1560 POKE 9,15:POKE 2,4:SYS SZ:PRINT"DIC
HTER"
1570 BX=LEN(B$(X)):BX=INT(6+((32-BX)/2))
1580 PRINTTAB(BX)FS$;B$(X);FB$
1590 PRINT:PRINT FG$;T1$
1600 PRINTTAB(15)CO$;CO$;"WERKE"
1610 FOR U=1TO ZW
1620 PRINTCO$;RL$;CO$:PRINTCHR$(98);RA$
="";U;CL$;"=";RE$;CO$
1625 AX=LEN(A$(U)):AX=INT(4+((34-AX)/2))
1630 PRINTTAB(AX)FS$;A$(U);FG$;CO$
1635 PRINTTAB(39)CHR$(98)
1640 NEXT U
1650 PRINT FG$;CO$;T3$;FS$
1660 PRINT Z2$
1670 :
1680 REM ** ZEIT FUER LOESUNG *****
1690 :
1700 TI$="000000"
1710 AB=0
1720 RZ%=SE-TI/60: IF RZ%<=0 THEN RZ%=0
1730 IF RZ%<=0 THEN PRINT CO$;CO$;RA$;"
*** EINGABEZIT ABGELAUFEN! **** "
1740 IF RZ% = 0 THEN POKE 198,0:WAIT 198
,1:AB=1
1770 :
```



```

1780 REM ** ABFRAGE DICHTER <=> WERK ***
1790 :
1800 GET EG$:EG=VAL(EG$)
1810 IF AB=1 AND EG= 0 THEN 1800
1820 IF AB=1 AND EG=>U THEN 1800
1830 IF EG <=0 OR EG =>U THEN 1720
1831 GOSUB 2303
1850 IF D(EG)=D(X) THEN PRINT BL$:PK(NN)
=INT((251*ZW*RZ%)/SE↑2)
1856 R2#=RA#+R1#+RE#:R3=0
1857 POKE Z,2:SYS SZ
1860 IF D(EG)=D(X) THEN FOR B =1 TO 100:
NEXT:PRINT R1$:POKE 646,INT(R3)+2
1863 R3=R3+1:POKE Z,2:SYS SZ:POKE R,R3:I
F R3=>9 THEN 1890
1865 IF D(EG)=D(X) THEN FOR B =1 TO 100:
NEXT:PRINT R2$:PRINTTAB(13)N1$:GOTO 1857
1880 PRINT BL$:TT$:FR$:RA$
1881 POKE S,4:POKE Z,4:SYS SZ:PRINT N3$:
PRINT
1882 PRINTTAB(4)"DAS WAR LEIDER FALSCH!"
1883 PRINT:PRINTTAB(4)"ES MUSS HEISSEN:"
1890 AX=INT((38-LEN(A$(X)))/2)+2
1900 PRINT:PRINT FS$:T1$:RL$:RL$:T3$:CO$
;CO$:CO$:FG$:PRINTTAB(AX) A$(X)
1920 PRINT:PRINTTAB(17)FS$: "V O N"
1925 BX=INT((38-LEN(B$(X)))/2)+2
1930 PRINT T1$:RL$:RL$:T3$:CO$:CO$:CO$ :
PRINTTAB(BX)FB$:B$(X):PRINT
1935 PRINTTD$:POKE 198,0:WAIT 198,1:POKE
198,0:PRINTBL$:FS$:TT$:PRINT
1940 :
1950 REM ** ABFRAGE DER GATTUNG *****
1951 :
1955 POKE R,7
1960 PRINT T1$:RL$:RL$:RL$:RL$:RL$:T3$:P
OKE S,AX:POKE Z,6:SYS SZ
1965 PRINT FG$:A$(X):PRINTTAB(57)FR$: " V
O N":PRINTTAB(40+BX)FB$:B$(X)
1970 POKE S,4:POKE Z,13:SYS SZ:PRINT FS$
;"WELCHE GATTUNG ?":PRINT LL$
1980 FOR I=1 TO 3:PRINTTAB(4)RA$:I;RE$:"
";G$(I):PRINT:NEXT I
1990 PRINT RA$:FR$:Z1$:RE$:FS$:POKE 198,
0:WAIT 198,1:GET GT$:GT=VAL(GT$)

```

```
1991 GOSUB 2303
2000 R3=0 : IF GT=G(X) THEN PK(NN)=PK(NN)
+5
2005 POKE Z,13:SYS SZ
2010 IF GT=G(X) THEN FOR B =1 TO 100:NEX
T:PRINT R1$: POKE 646,INT(R3)+2
2012 R3=R3+1:POKE Z,13:SYS SZ:POKE R,R3:
IF R3=>9 THEN 2021
2013 IF GT=G(X) THEN FOR B=1TO100:NEXT:P
RINT R2$:GOTO 2005
2016 PRINT" NEIN! DIES IST DIE RICHTIGE
LOESUNG:"
2021 POKE S,8:POKE Z,14+(G(X)*2):SYS SZ:
PRINT RA$;G$(G(X));" ";RE$
2022 POKE Z,22: SYS SZ:PRINT TD$
2025 POKE 198,0:WAIT 198,1:PRINTBL$
2026 GOSUB 2303
2030 :
2040 REM *** ABFRAGE DER EPOCHE *****
2050 :
2055 POKE R, 4
2060 PRINTTAB(3)FS$;"WELCHER EPOCHE IST"
:PRINT
2065 PRINTTAB(3)FG$;A$(X);FS$ :PRINT
2066 PRINTTAB(3)"ZUZUORDNEN";N2$;" ?":PR
INTLL$
2070 FOR I=0TO 9:PRINT I;CL$;". ";E$(I):N
EXT I
2075 POKE Z,19: SYS SZ:PRINT LL$;FR$;RA$
;Z1$;RE$;FS$
2080 POKE 198,0:WAIT 198,1:GET EP$:EP=VA
L(EP$)
2081 GOSUB 2303
2100 IF EP=E(X) THEN PRINTTAB(3)"GUT GEM
ACHT!":PK(NN)=PK(NN)+15:GOTO 2110
2104 PRINTCHR$(19)
2105 PRINT" LEIDER FALSCH!
":PRINT
2106 PRINT" ICH ZEIGE IHNEN DIE R I C
H T I G E "
2107 PRINT" LOESUNG";N2$;". "
2110 POKE S,3:POKE Z,8+(E(X)): SYS SZ
2115 PRINT RA$;FR$;E$(E(X));" ";RE$;FS$
2130 POKE Z,19:SYS SZ:PRINT LL$;TD$
2140 :
```

```

2150 REM *** PUNKTESTAND *****
2151 :
2160 POKE 198,0:WAIT 198,1
2161 GOSUB 2303 :POKE R,3
2165 PRINT BL$;FR$;BA$;BA$;BA$;FS$;CO$;C
O$;SPC(8);" P U N K T E S T A N D "
2170 RD=RD+1
2180 PRINTTAB(83)"RUNDE: ";RD;:PRINTTAB(
32)" PUNKTE"
2190 PRINT LL$
2212 PT(NN)=PT(NN)+PK(NN)
2213 FOR H=1 TO N-1:PRINTTAB(3)N$(H);:PR
INTTAB(25);": ";INT(PK(H));
2214 PRINTTAB(32)FR$;INT(PT(H));FS$:PRIN
T
2215 IF PT(H)>PT(H-1)THEN RK=PT(H)
2217 NEXT
2218 POKE S,18:POKE Z,4:SYS SZ:PRINT"REK
ORD :";INT(RK)
2219 POKE S,3:POKE Z,(2*NN-1)+6:SYS SZ:P
RINT RA$;N$(NN);RE$
2220 POKES,33:POKEZ,(2*NN-1)+6:SYSSZ:PRI
NTRA$;FR$;INT(PT(NN));CL$" "RE$:PK(NN)=0
2221 POKE Z,16:SYS SZ:PRINT T1$;RL$;T3$
2222 POKE S,1:POKE Z,17 :SYS SZ:PRINT"ME
NUE = M * ENDE = E * WEITER = RETURN"
2230 PRINT:PRINT TD$
2240 POKE 198,0:WAIT 198,1:GET WT$
2241 GOSUB 2303
2242 RR=5:IF AU=2 THEN RR=6:POKE R,RR
2250 IF WT$="E" THEN END
2270 FORI=1TO DW:H(I)=0:NEXT:I=0
2280 IF WT$="M" THEN 505
2290 U=0:GOTO 1070
2300 :
2301 REM * UNTERPROGRAMM: EINGABE-TON *
2302 :
2303 S1=54272
2304 POKE S1+24,15:POKE S1+11,17:POKE S1
+7,207:POKE S1+8,34
2305 POKES1+13,240:FOR PO=1TO50:NEXT:POK
E S1+11,0:POKE S1 24,0
2306 RETURN
2307 :
2310 REM DATEN DICHTER

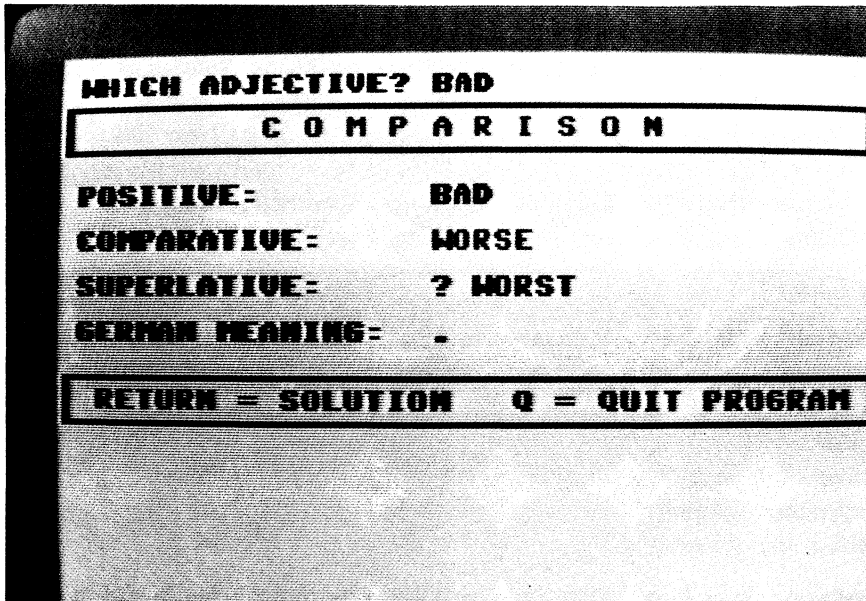
```

2320 :  
2340 DATA HARTMANN VON DER AUE  
2350 DATA WOLFRAM VON ESCHENBACH  
2360 DATA GOTTFRIED VON STRASSBURG  
2370 DATA WALTHER VON DER VOGELWEIDE  
2380 DATA "JOHANN VON TEPL (AUCH: SAAZ)"  
2390 DATA CHRISTOPH VON GRIMMELSHAUSEN  
2400 DATA GOTTHOLD EPHRAIM LESSING  
2410 DATA FRIEDRICH GOTTLIEB KLOPSTOCK  
2420 DATA CHRISTOPH MARTIN WIELAND  
2430 DATA JOHANN WOLFGANG VON GOETHE  
2440 DATA FRIEDRICH SCHILLER  
2450 DATA FRIEDRICH HOELDERLIN  
2460 DATA HEINRICH VON KLEIST  
2470 DATA JEAN PAUL  
2480 DATA NOVALIS  
2490 DATA JOSEPH VON EICHENDORFF  
2500 DATA E. T. A. HOFFMANN  
2510 DATA EDUARD MOERIKE  
2520 DATA ADALBERT STIFTER  
2530 DATA FRANZ GRILLPARZER  
2540 DATA ANNETTE VON DROSTE-HUELSHOFF  
2545 DATA HEINRICH HEINE  
2550 DATA GEORG BUECHNER  
2560 DATA GOTTFRIED KELLER  
2570 DATA CONRAD FERDINAND MEYER  
2580 DATA GUSTAV FREYTAG  
2590 DATA JEREMIAS GOTTHELF  
2600 DATA FRIEDRICH HEBBEL  
2610 DATA THEODOR STORM  
2620 DATA WILHELM RAABE  
2625 DATA THEODOR FONTANE  
2630 DATA "\*" "  
2640 :  
2650 REM \*\* DATEN UND WERKE DER DICHTER  
2660 :  
2670 REM EGD I  
2680 DATA 0101 IWEIN  
2690 DATA 0101 DER ARME HEINRICH  
2700 DATA 0102 PARZIVAL  
2710 DATA 0103 TRISTAN UND ISOLDE  
2720 DATA 0204 ELEGIE  
2730 DATA 1105 DER ACKERMANN AUS BOEHMEN  
2740 DATA 2106 SIMPLICISSIMUS  
2750 DATA 3307 MINNA VON BARNHELM

2760 DATA 3307 EMILIA GALOTTI  
2770 DATA 3307 NATHAN DER WEISE  
2780 DATA 3108 DER MESSIAS  
2790 DATA 3109 AGATHON  
2800 DATA 4110 DIE LEIDEN DES JUNGEN WERTHER  
2810 DATA 4310 URFAUST  
2820 DATA 4310 GOETZ VON BERLICHINGEN  
2830 DATA 4210 PROMETHEUS  
2840 DATA 4311 DIE RAEUBER  
2850 DATA 4311 KABALE UND LIEBE  
2860 DATA 4311 DON CARLOS  
2870 DATA 5310 IPHIGENIE  
2880 DATA 5310 EGMONT  
2890 DATA 5310 TORQUATO TASSO  
2900 DATA 5310 FAUST  
2910 DATA 5210 ERLKOENIG  
2920 DATA 5110 HERMANN UND DOROTHEA  
2930 DATA 5110 WILHELM MEISTER  
2940 DATA 5110 DIE WAHLVERWANDTSCHAFTEN  
2950 DATA 5210 WEST-OESTLICHER DIVAN  
2960 DATA 5110 DICHTUNG UND WAHRHEIT  
2965 DATA 5211 DER TAUCHER  
2970 DATA 5211 DIE KRANICHE DES IBYKUS  
2980 DATA 5211 DIE BUERGSCHAFT  
2985 DATA 5211 DAS LIED VON DER GLOCKE  
2990 DATA 5311 WALLENSTEIN  
3000 DATA 5311 MARIA STUART  
3010 DATA 5311 DIE JUNGFRAU VON ORLEANS  
3020 DATA 5311 DIE BRAUT VON MESSINA  
3030 DATA 5311 WILHELM TELL  
3040 DATA 6112 HYPERION  
3050 DATA 6113 MICHAEL KOHLHAAS  
3060 DATA 6313 DER ZERBROCHENE KRUG  
3070 DATA 6313 DAS KAETHCHEN VON HEILBRONN  
3080 DATA 6313 PRINZ FRIEDRICH VON HOMBURG  
3090 DATA 6114 TITAN  
3100 DATA 7115 HEINRICH VON OEFERDINGEN  
3110 DATA 7215 HYMNEN AN DIE NACHT  
3120 DATA 7116 AUS DEM LEBEN EINES TAUGENICHTS  
3130 DATA 7116 AHNUNG UND BEGENWART  
3140 DATA 7117 DAS FRAEULEIN VON SCUDERY  
3150 DATA 8118 MALER NOLTEN

3160 DATA 8118 MOZART AUF DER REISE NACH  
PRAG  
3170 DATA 8218 DER FEUERREITER  
3180 DATA 8119 DER HOCHWALD  
3190 DATA 8119 BUNTE STEINE  
3200 DATA 8119 NACHSOMMER  
3210 DATA 8119 WITIKO  
3220 DATA 8320 DAS GOLDENE VLIES  
3225 DATA 8320 EIN BRUDERZWIST IM HAUSE  
HABSBURG  
3230 DATA 8320 WEH DEM DER LUEGT  
3240 DATA 8221 DAS GEISTLICHE JAHR  
3250 DATA 8221 HEIDEBILDER  
3260 DATA 8221 DER KNABE IM MOOR  
3270 DATA 8121 DIE JUDENBUCH  
3280 DATA 9122 DEUTSCHLAND. EIN WINTERMA  
ERCHEN  
3290 DATA 9323 DANTONS TOD  
3300 DATA 9323 WOYZECK  
3310 DATA 9124 DER GRUENE HEINRICH  
3320 DATA 9124 DIE LEUTE VON SELDWYLA  
3330 DATA 9125 DAS AMULETT  
3340 DATA 9125 GUSTAV ADOLFS PAGE  
3350 DATA 9125 DER SCHUSS VON DER KANZEL  
3360 DATA 9126 SOLL UND HABEN  
3370 DATA 9127 DIE SCHWARZE SPINNE  
3380 DATA 9127 ULI DER KNECHT  
3390 DATA 9127 ULI DER PAECHTER  
3395 DATA 9328 JUDITH  
3400 DATA 9328 MARIA MAGDALENA  
3410 DATA 9328 AGNES BERNAUER  
3420 DATA 9328 DIE NIBELUNGEN  
3430 DATA 9129 IMMENSEE  
3435 DATA 9129 POLE POPPENSPAELER  
3440 DATA 9129 DER SCHIMMELREITER  
3450 DATA 9130 DIE CHRONIK DER SPERLINGS  
GASSE  
3460 DATA 9130 DIE SCHWARZE GALEERE  
3470 DATA 9130 DER HUNGERPASTOR  
3480 DATA 9231 DIE BRUECKE AM TAY  
3490 DATA "9131 IRRUNGEN, WIRRUNGEN"  
3500 DATA 9131 EFFI BRIEST  
3510 DATA 9131 DER STECHLIN  
5000 DATA "\*"

## 4 Die Steigerung im Englischen



### Programm-Telegramm

- Erlernen der Steigerung durch sequentielle Wahl oder Zufallsauswahl englischer Adjektive.
- Wahlweises Ausfüllen von ein bis drei Lücken.
- Überprüfung der richtigen Schreibweise bei selbst eingegebenen Adjektiven.
- Nach Übungsende Ausgabe einer individuellen Lernliste auf dem Bildschirm.

### Steigern Sie Ihre Kenntnisse!

Bevor Sie in das Computerprogramm einsteigen, können Sie hier Ihre Kenntnisse über die Steigerung der englischen Adjektive und Adverbien steigern oder zumindest etwas auffrischen.

### Germanisch oder romanisch?

Wie man weiß, gehört Englisch zu den germanischen Sprachen. Es kennt daher die synthetische oder germanische Art der Steigerung durch Anhängen von *-er* oder *-est* an die Grundform.

Die analytische oder romanische Steigerung durch Vorsetzen von *more* oder *most* erinnert daran, daß bereits die Römer in England waren. Sie hinterließen dort nicht nur den Hadrianswall, die nach Kaiser Hadrian benannte Mauer, sondern auch sprachliche Spuren.

Nachhaltiger beeinflussten allerdings die Normannen, die im Jahre 1066 mit Wilhelm dem Eroberer ins Land kamen, die englische Sprache. So finden wir heute neben einem alten germanischen Wort häufig ein Synonym französischen Ursprungs.

### Eine Silbe

Die einsilbigen Wörter sind oft germanischen Ursprungs. Es gilt daher die Faustregel:

Einsilbige Adjektive werden durch Anfügen von <b>-er</b> bzw. <b>-est</b> gesteigert.
---

Beispiel:

Positive: great

Comparative: greater

Superlative: greatest

Natürlich gibt es Ausnahmen. *Strict*, einsilbig, aber romanischer Herkunft, wird mit **more** und **most** gesteigert.

### Drei und mehr Silben

Adjektive romanischer Herkunft steigert man meist durch Vorstellen von **more** und **most**.

Beispiel:

im:portant (3 Silben : = Silbentrennung)

**more** important

**most** important

Da der sprachliche Ursprung vieler Wörter heute nur schwer erkennbar ist, gilt die Faustregel:

Drei- und mehrsilbige Adjektive werden romanisch gesteigert.
--



## Zwei Silben

Da gibt es noch die zweisilbigen Adjektive, die eine Zwischenstellung einnehmen. Auch hier zunächst die Faustregel:

Zweisilbige Adjektive, die auf **-y**, **-ow**, **-er** oder **-le** enden, steigern mit **-er** und **-est**.

Hierbei spielt es keine Rolle, welchen Ursprungs das Wort ist.

Beispiel:

noble (2 Silben)

nobler

noblest

## Rechtschreibregeln

Wie Sie am obigen Beispiel bereits erkennen, genügt es nicht, einfach **-er** oder **-est** anzuhängen.

Bei der Schreibung ist folgendes zu beachten:

1. Endkonsonanten werden nach **kurzem**, betontem Vokal **verdoppelt**.

Beispiel:

big – bigger – biggest

2. End **-e** entfällt:

Beispiel:

nice – nicer – nicest

3. **y** → **i**

Beispiel:

happy – happier – happiest

### Unregelmäßige Steigerung

Diese **unregelmäßigen** Formen sind in das Computerprogramm eingebaut:

<b>good</b>	<b>better</b>	<b>best</b>	+
<b>bad</b>	<b>worse</b>	<b>worst</b>	-
<b>much</b>	<b>more</b>	<b>most</b>	+
<b>little</b>	<b>less</b>	<b>least</b>	-
<b>far</b>	<b>farther</b>	<b>farthest</b>	+
near	nearer	nearest	-

Man kann sie sich leichter merken, wenn man sie, wie hier geschehen, nach positivem/negativem Sinninhalt geordnet lernt.

### Alternativformen

Beachten Sie auch diese Alternativformen (! = Erläuterung, nicht Übersetzung):

	good	/ well (Adverb!)
	bad	/ <b>badly</b> (Adverb!)/ <b>ill</b> (krank)
(viel)	much	/ <b>many</b> (viele)
(Entfernung!)	nearest	/ <b>next</b> (Reihenfolge!)
(räumlich!)	farther	/ <b>further</b> (örtlich + übertragen!)
	farthest	/ <b>furthest</b> (selten!)

### Eingabe

Adverbien werden wie Adjektive gesteigert. Die synthetische Steigerung (*more, most*) wird aber hier bereits bei zweisilbigen Wörtern nahezu ausschließlich angewandt.

Sollten Sie bei der Eingabe der Daten Schwierigkeiten haben, so sehen Sie in einem Grammatikbuch nach. Geben Sie dann notfalls von der tatsächlichen Silbenzahl abweichend

1 ein, wenn mit **-er/-est** gesteigert wird, und

3 wenn durch Voranstellen von **more** und **most** gesteigert wird.

### Programmablauf

Nach dem Laden und Starten des Programms wird der Bildschirmrahmen gelb und das Feld weiß. Das erste Bild sieht folgendermaßen aus:

C O M P A R I S O N	
<b>1</b>	1 GAP
<b>2</b>	2 GAPS
<b>3</b>	3 GAPS
GAP AT ...	
<b>4</b>	POSITIVE
<b>5</b>	COMPARATIVE
<b>6</b>	SUPERLATIVE
<b>7</b>	GERMAN MEANING
H I T N U M B E R K E Y	

Geben Sie eine Zahl zwischen 1 und 7 ein. Ihre Eingabe wird durch einen Piepston quittiert.

Nun werden Sie gefragt, welche Lektion Sie wünschen. 0–9 wird angeboten, wobei 0 bedeutet, daß der gesamte Wortschatz abgefragt wird. Hierbei kann es durchaus vorkommen, daß sich ein Adjektiv wiederholt, nämlich dann, wenn ein und dasselbe Wort in verschiedenen Lektionen eingegeben wurde.

Das 3. Bild sehen Sie als Bildschirmfoto zu Beginn dieses Kapitels. Wählen Sie Nummer 3 (EIGENE EINGABE), so müssen Sie zunächst das Adjektiv eintippen, welches Sie steigern wollen. Es hat keinen Sinn, etwas Unsinniges einzugeben, denn der Computer überprüft, ob sich das betreffende Wort in seinem Speicher befindet und akzeptiert die Eingabe nur dann, wenn dies der Fall ist.

Wählen Sie 1 oder 2, so sucht der Computer für Sie ein entsprechendes Adjektiv aus und bietet es Ihnen zum Steigern an. Seine Wahl wird bei Modus 1 vom Zufall bestimmt, während im Modus 2 die Daten in der gespeicherten Reihenfolge präsentiert werden.

Nun geben Sie die Lösung ein, die an die Stelle gehört, wo der *Cursor* gerade blinkt. Beachten Sie bei der Eingabe der deutschen Bedeutung, daß dieses Programm weder »ß« noch Umlaute kennt. Das Wort »früh« sähe also folgendermaßen aus: FRUEH.

Sie können Ihre Eingabe mit Hilfe der **INST/DEL**-Taste verbessern. INSTAnt DELete heißt »sofort löschen«.

Achten Sie aber darauf, daß Sie diese Taste nur ganz kurz drücken. Sie sollten auf diese Weise nicht die Zeile verlassen. Ebenso sollten Sie nicht die **CRSR**-Taste (*Cursor*) verwenden!

Nachdem Sie die Lösung blau eingetippt haben, drücken Sie kurz die **RETURN**-Taste. Jedes Antippen dieser Taste bewirkt die Ausgabe der richtigen Lösung und einen Sprung des *Cursors* zur nächsten Lücke.

Sind alle Lücken ausgefüllt, so können Sie Ihre Leistung betrachten. Rote Wörter sind Fehleingaben, grüne hingegen markieren richtige Lösungen. Die Vorgaben werden in schwarzer Schrift angezeigt. In einem Kästchen steht, daß Sie sich mit Hilfe der **RETURN**-Taste die Lösung zeigen lassen können und mit **Q** das Programm verlassen. Ein blinkender roter Balken fordert Sie auf, die Leertaste zu drücken. Bevor Sie dies tun und zum nächsten Adjektiv-Set kommen, sollten Sie sich noch kurz über Ihren Leistungsstand informieren. Unterhalb der Blinkanzeige sind die Anzahl Ihrer Versuche, die von Ihnen erreichte PunktezahI sowie die Note, welche Sie aufgrund Ihres gegenwärtigen Leistungsstandes erhalten würden, angezeigt.

Tippen Sie nun auf die Leertaste, und das nächste Adjektiv-Set erscheint. Genau genommen muß es kein Adjektiv-Set sein, denn auch Adverbien können gesteigert werden. Ist die Übung zu Ende, weil Sie **Q** eintippten oder der Datenvorrat verbraucht ist, so erscheint auf dem Monitor eine abschließende Übersicht. Sie könnte folgendermaßen aussehen:

<b>C O M P A R I S O N</b>	
	<b>5 SET/S OF ADJECTIVES</b>
	<b>15 FILLED GAP/S</b>
	<b>12 CORRECT INPUT/S</b>
	<b>3 WRONG INPUT/S</b>
<b>YOUR MARK:</b>	<b>2</b>
<b>LEARN THE 'RED' ADJECTIVE FORMS:</b>	
<b>1. KALT</b>	<b>COLD COLDER COLDEST</b>

Sollten Sie, wie in unserem Beispiel, mehr richtige Eingaben bestätigt bekommen als Steigerungsübungen angezeigt werden, so hatten Sie im Menü sicher zwei oder drei Lücken gewählt. Halt, Sie sehen natürlich nichts!

Ihre Fehlerauflistung erscheint erst auf Tastendruck. Drücken Sie also ganz kurz die Leertaste! Sie sehen nun jene Adjektiv-Sets, in denen Sie einen Fehler gemacht hatten und werden aufgefordert, die in Rot wiedergegebenen Formen zu lernen. Dieser letzte Teil ist wesentlicher Bestandteil des Lernprogramms, und wenn dieses Programm wirklich zur Verbesserung Ihrer Leistungen beitragen soll, dürfen Sie auf keinen Fall diese letzte Bitte Ihres Computers ignorieren.

Selbstverständlich können Sie sich diese Liste auch auf Ihrem Drucker ausgeben lassen. Leider ist sie dann nicht dreifarbig (Vorgabe = schwarz, richtige Eingabe = grün, falsche Eingabe = rot), sondern in der Tabelle wird vor die zu lernenden Formen ein schwarzer »Ball« gesetzt. Die Liste könnte beispielsweise so aussehen:

## LEARN THE MARKED (●) ADJECTIVE-FORMS OF UNIT 8

1	TERRIBLE	MORE TERRIBLE	●MOST TERRIBLE	●SCHRECKLICH
2	POPULAR	MORE POPULAR	MOST POPULAR	BELIEBT
3	ELEGANT	MORE ELEGANT	MOST ELEGANT	ELEGANT
4	MODERN	●MORE MODERN	MOST MODERN	MODERN
5	●TIDY	●TIDIER	TIDIEST	●ORDENTLICH
6	YOUNG	YOUNGER	YOUNGEST	JUNG
7	●FEW	FEWER	FEWEST	●WENIGE
8	LITTLE	LESS	●LEAST	WENIG
9	●CLEVER	●CLEVERER	CLEVEREST	KLUG
10	MUCH	●MORE	MOST	●VIEL
11	●HEALTHY	●HEALTHIER	HEALTHIEST	●GESUND
12	●POLITE	MORE POLITE	MOST POLITE	●HOEFLICH
13	●COSY	COSIER	●COSIEST	●GEMÜTLICH

Sollten Sie versehentlich die Leertaste gedrückt haben, ohne daß Ihr Drucker eingeschaltet war, so erhalten Sie die Meldung: DEVICE NOT PRESENT ERROR IN 1695, was ins Deutsche übertragen heißt, daß Ihr Gerät (= Drucker) nicht eingeschaltet war. Keine Panik, Sie können durch direkte Eingabe von GOTO 1500 ohne Datenverlust wieder in das Programm einsteigen.

Selbst wenn Sie nach diesem Einstieg aus einem unerfindlichen Grund Ihren Drucker wieder nicht eingeschaltet hatten, können Sie Ihre Daten retten, wenn Sie die Meldung: FILE OPEN ERROR IN 1695 mit der Eingabe von CLOSE 4 im Direktmodus beheben.

### Programmbeschreibung

- 10 – 90: Titel, Copyright, Verfasser.
- 110: W sollte etwas größer als die Zahl der in den DATA-Zeilen abgelegten Adjektiv-*Sets* sein.
- 120: Dimensionierung:  
 VE\$(W) ist für die Grundform des Adjektivs reserviert.  
 SI(W) wird die Silbenzahl aufnehmen.  
 GE\$(W) erhält die deutsche Bedeutung zugewiesen.  
 VM\$(W,4) wird alle auf dem Bildschirm ausgegebenen Adjektiv-Formen speichern.  
 VH(W) vermeidet Wiederholungen bei der Zufallsauswahl der Datensätze.
- 130: S = *Soundchip*-Startadresse.
- 140: Eine Zufallszahl wird erzeugt. Sie bewirkt, daß bei jedem Start die folgende Zufallszahl in Zeile 950 verschieden ist.

- 150: Der Bildschirmrahmen wird gelb, der Hintergrund wird weiß gePOKEt.
- 160: SHIFT entriegeln, Großschrift einschalten und SHIFT wieder blockieren.
- 200 – 240: Die verwendeten Steuerzeichen- und Farbvariablen wurden bereits im Programm »Deutsche Literatur« besprochen.
- 320: Das 40mal aneinandergereihte  $\blacksquare$  (CHR\$ (175)) ergibt einen Strich.
- 321: Hier entsteht ein feiner Mittelstrich  $\square$  (CHR\$ (99)) von 38 Zeichen Länge.  
Achten Sie darauf, daß Sie wirklich T0\$ und nicht TO\$ eingeben.
- 330: T1\$ sieht so aus:  $\square$ . Die Winkel sind CHR\$ (174) und CHR\$ (176).
- 340: Programmtitel COMPARISON mit linker und rechter Begrenzung (= CHR\$ (98)) wird zusammengesetzt.
- 350: Dies ist T3\$:  $\square$ . Die Winkel sind CHR\$ (173) und CHR\$ (189).
- 360: TT\$ ergibt folgendes Bild auf dem Schirm:

C O M P A R I S O N
---------------------

- 370 – 371: T5\$ ist der Balken, der auffordert (T4\$), eine Zahl einzutippen.
- 380: T5\$ produziert einen roten Balken mit dem eingeschriebenen Text: LEERTASTE DRUECKEN.
- 420 – 490: Die indizierte Variable G\$ ( ) enthält das Angebot des ersten Menüs.
- 500 – 510: Das komplette Menü, einschließlich Titel und Wartetaste, wird auf den Bildschirm gebracht.
- 520 – 530: Absicherungen gegen Fehleingaben und Sprung in das Unterprogramm PIEPSTON.
- 560 – 591: Die in SE\$ gespeicherte Wahl des Anwenders  
1 = Zufallsauswahl der Adjektive,  
2 = eingespicherte Reihenfolge der Adjektive,  
3 = eigene Eingabe eines Adjektives,  
wird später noch gebraucht.
- 592 – 593: Mit Hilfe dieser zwei Zeilen entsteht eine Blinkschrift.
- 610 – 625: Nur nach Eingabe der Ziffern 1, 2 oder 3 geht es weiter.  
Wenn SE = 3 ist, werden die folgenden beiden Zeilen übersprungen, und über Zeile 625 wird das Unterprogramm PIEPSTON abgearbeitet. Ist SE = 1 oder 2, so wird das Unterprogramm LEKTIONSWAHL angesprungen.
- 751 – 754: Einlesen der Daten. Der Asterix schließt das Einlesen ab. Wurden jedoch keine Daten eingelesen, d.h. steht B noch auf 1, dann erfolgt ein Sprung in den Programmteil MELDUNG: UNIT EXISTIERT NICHT.

- 785 – 812: Die Bildschirmmaske wird auf den Schirm gebracht. Hier werden auch die Inhalte von G\$ (4) bis G\$ (7) wieder verwendet.  
Dies ist ihr Inhalt:  
POSITIVE (Grundstufe)  
COMPARATIVE (1. Steigerungsstufe)  
SUPERLATIVE (Höchststufe)  
GERMAN MEANING (deutsche Bedeutung)  
Die Übersetzung der PRINT-Anweisung lautet:  
»Ein Druck auf die RETURN-Taste zeigt die Lösung an.« und  
»Mit Q kann man die Übung abbrechen.«
- 840: Der *Cursor* geht in HOME-Stellung, d.h. in das linke obere Eck des Bildschirms.
- 940: Wenn SE=2 (sequentielle Wahl), dann muß der Programmteil 945 – 1030 übersprungen werden.
- 945: Wenn SE=3, dann wird ab Zeile 2020 das Unterprogramm EIGENE EINGABE abgearbeitet.
- 950: Es bleibt nur noch SE=1 übrig. Hier wird die Zufallsauswahl der Adjektive getroffen.
- 970 – 975: VH (I) vermeidet Wiederholungen.
- 1040: Z zählt hoch, falls der Anwender SE auf 2 setzte. So wird in den Zeilen 1080 – 1110 ein Datensatz nach dem anderen umgespeichert.
- 1050: Diese Programmzeile sorgt für ein Ende der Übung, wenn alle Daten ausgegeben wurden.
- 1080 – 1100: Umspeichern eines Datensatzes (Adjektive) zur weiteren Verwendung.
- 1140 – 1190: Die Lücken – 1 bis maximal 3 – werden zufallsgesteuert ermittelt. G zählt die Runden. D (H) vermeidet eine Doppelbelegung.
- 1220 – 1260: Die vorgegebenen Adjektivformen und die Lücken (für letztere dient ein Punkt als Platzhalter) werden auf dem Bildschirm plziert.
- 1270: In dieser Zeile beginnt die Schleife, die dazu beiträgt, daß INPUT, das ist die Anwendereingabe, genau in die richtige/n Lücke/n gesetzt wird.
- 1275: POKE 211,18 entspricht einem PRINTTAB (18) oder einem PRINT SPC (18). In POKE 214,2 \* Y+4 wird die Zeile ermittelt: 4 Zeilen werden für die Titeleinblendung benötigt. Zwischen jedem Eintrag sorgt eine Leerzeile für bessere Übersichtlichkeit; daher 2 \* Y.
- 1280 – 1310: Wenn DH(Y)=0, so bedeutet dies, daß auch die zugehörige *Stringvariable* V\$ (Y) »leer« ist. Deshalb wird das Schleifenende (Zeile 1370) angesprungen. Andernfalls (Zeile 1310) wird eine Eingabe (INPUT) verlangt.
- 1320 – 1321: Die Länge der Eingabe und die des Lösungsadjektivs werden ermittelt. Falls die Zeichenzahl der Eingabe größer als die des fragten Adjektivs sein sollte – diese Lösung wäre natürlich falsch – dann wird ein *Leersstring* erzeugt, der die ursprüngliche Eingabe mit *Blanks* überschreibt

- und auf diese Weise dafür sorgt, daß neu eingegeben werden kann, ohne daß störende Wortreste auf dem Bildschirm sichtbar bleiben. Ein starres Überschreiben mit Leerzeichen könnte den *Cursor* – unerwünscht – in die nächste Zeile rutschen lassen.
- 1340+1350: Bei richtiger Lösungseingabe durch den Anwender wird diese grün überschrieben. RG zählt die richtigen Lösungen.
- 1344 – 1349: Alternativlösungen.
- 1360: Bei falscher Lösungseingabe wird die Eingabe des Anwenders rot überschrieben.
- 1400 – 1415: Nach jedem Durchlauf erscheint im unteren Teil des Bildschirms die Leistungsanzeige.
- Die Anzahl der Versuche ergibt sich aus der Summe falscher (F) und richtiger (RG) Eingaben.
  - Die Punktezahl entspricht der Menge der richtigen (RG) Eingaben.
  - FF ist die Note, die nach oben mit der Zahl 6 begrenzt werden muß. Die untere Begrenzung ist notwendig für den Fall, daß der Anwender null Fehler haben sollte. In Zeile 1409 wird F (I) auf 1 gesetzt. Das ist nötig, um bei Eingabe einer falschen Lösung das betreffende Adjektiv erneut anbieten und bei Abschluß der Übung auflisten zu können. Ein Blick auf Zeile 1615 erklärt die obige Maßnahme.
- 1420 – 1430: Blinkanzeige.
- 1440 – 1450: Variablen werden auf Null gesetzt.
- 1460: Bei Übungsende Sprung in die Leistungsanzeige.
- 1480: Sprung zum nächsten Durchlauf.
- 1510 – 1570: Die gePRINTete Anzeige gibt bei Übungsende oder vorzeitigem Abbruch die Zahl der
- Adjektiv-Sets
  - ausgefüllten Lücken
  - richtigen Eingaben
  - falschen Eingaben
  - und die Note an.
- 1580 – 1585: Nach einem Tastendruck (TS\$) werden wir aufgefordert, die rot markierten Adjektivformen zu lernen. Mit Hilfe von zweimal *Cursor* nach oben (CO\$) überlagert der letzte Schriftzug den vorangegangenen.
- 1610 – 1650: Anzeige aller Adjektiv-Sets, bei denen eine Fehleingabe gemacht wurde. Sie sind im zweidimensionalen Feld VM\$ (I,Y) enthalten.
- 1660 – 1690: GET YN\$ fragt, ob Sie die Liste ausgedruckt haben wollen. Wenn YN\$= N, dann ist das Programm beendet (END). Bei Eingabe eines beliebigen anderen Zeichens, einschließlich Leerzeichen, kommen Sie, wenn der Drucker bereit ist, zur nächsten Abfrage: TASTE DRUECKEN (TS\$).



- Mit OPEN 4,4:CMD4 werden die folgenden PRINTs auf den Drucker umgeleitet.  
 Die Ausgabe wurde auf einem EPSON FX-80 mit Görlitz-*Interface* getestet. Wegen des einfachen Aufbaus dürfte es aber auch mit anderen Druckern keine Schwierigkeiten geben.
- 1696 – 1716: Den »Ball« in Zeile 1696 + 1705 erhalten Sie mit SHIFT + Q (im Großschrift-Modus). In der Schleife von 1700 – 1716 werden die Adjektive des zweidimensionalen Feldes VM\$ (I, Y) gelesen. Damit eine Tabelle entsteht, wird zunächst die Stringlänge ermittelt (LF). Dieser »Adjektiv«-*String* kann eine Farbinformation (rot oder grün) enthalten. Diese Information verlängert den String um ein Zeichen. Auf dem Bildschirm ist dies nicht sichtbar, bringt aber im Ausdruck unsere Tabelle in Unordnung. Daher sind die Maßnahmen in den Zeilen 1705 – 1710 notwendig.
- In Zeile 1705 wird das erste Zeichen (=Farbinformation) entfernt, falls der ASCII-Code 28 (=Zeichenfarbe rot) ist. Es wird durch einen »Ball« ersetzt.
- In Zeile 1706 wird das erste Zeichen ersatzlos entfernt, falls es den ASCII-Code 30 (=Zeichenfarbe grün) hat.
- Die Anzahl der Leerzeichen (*Space* = SPC) und der Zeichen des Adjektivs soll addiert immer 18 betragen. Die Menge der SPCs wird daher in TB durch Subtraktion gewonnen.
- IF TB < 1 THEN TB = 1 brauchen wir, um kein negatives TB zu erhalten bzw. bei längeren Adjektiven einen Abstand zwischen den Wörtern sicherzustellen.
- TB = 1: IF I = > 9 then TB = 0 ist nötig, um eine Verschiebung der Tabellen zu vermeiden, wenn die Ordnungszahlen zweistellig werden.
- 1735 – 1740: Die romanische Steigerung braucht kaum Platz. Es muß lediglich die Grundform H\$ mit der Textkonstanten MORE bzw. MOST verkettet werden. Auf diese Weise entstehen die zwei noch fehlenden Formen Komparativ und Superlativ.
- 1790: Die Länge R der Grundform H\$ wird ermittelt.
- 1800: In R1\$ steht der letzte Buchstabe.
- 1810: Der vorletzte Buchstabe ist in M2\$ abgelegt.
- 1830: Die beiden letzten Buchstaben werden in R2\$ gespeichert.
- 1840: L1\$ ist die Grundform ohne den letzten Buchstaben.
- 1845: Adverbien, die auf LY enden, werden romanisch gesteigert. Die Silbenzahl SI wird daher auf 3 gesetzt.
- 1850: Endet die Grundform auf Y, so wird IER bzw. IEST an die Grundform gehängt.
- 1860 – 1880: Hier wird geprüft, ob das Adjektiv auf OW, ER oder LE endet; wenn ja, wird germanisch gesteigert.

- 1890: Diese Zeile befaßt sich mit der Regel: Auslautendes E entfällt.
- 1895 – 1896: Konsonantenverdoppelung bei bestimmten Vokalkombinationen wird ausgeschlossen, d.h. U=0.
- 1900 – 1920: Die Vokalkontrolle wird später eventuell für die Konsonantenverdoppelung gebraucht.
- 1940 – 1960: Die Prüfung auf Konsonanten sorgt im Verbund mit der positiven Vokalkontrolle, deren Ergebnis in U abgelegt wurde, für die Verdoppelung des Konsonanten. Zeile 1945 schließt jedoch die Verdoppelung bei bestimmten »zweistelligen« Buchstabenkombinationen aus. Die Sterne sind Platzhalter für 3 weitere Kombinationen für den Fall, daß sich bei bestimmten Adjektiven fehlerhafte Bildungen zeigen sollten.
- V\$ (2) = H\$ + R1\$ + "ER"  
Komparativ = Grundform + letzter Buchstabe + ER  
BIGGER = BIG + G + ER
- 1990: Bei negativem Ergebnis der Kontrollschleifendurchläufe (wenn U=0), wird die Steigerung ohne Konsonantenverdoppelung durchgeführt.
- 1995: U muß für den nächsten Durchlauf wieder auf Null gesetzt werden.
- 2002 – 2006: Falls eine Adverbform, die unregelmäßig gesteigert wird, aus der DATA-Zeile gelesen wurde, wird hier die richtige Form erzeugt.
- 2012 – 2014: Unterprogramm PIEPSTON: Zunächst wird die Lautstärke voll, d.h. auf 15 geschaltet. Mit POKE S+11,17 wird der Ton eingeschaltet. Dann kommt die Frequenz LO, anschließend HI. Diese Lautstärke muß in der nächsten Zeile gehalten werden. Eine Schleife bestimmt die Länge des Tons. Der Ton wird wieder ausgeschaltet; anschließend wird die Lautstärke auf Null gesetzt. Mit RETURN geht es zurück ins Hauptprogramm. Wer den Ton verändern möchte, schlägt am besten im Commodore 64 Handbuch, Anhang P, S. 158-159 nach.
- 2020 – 2130: Das Unterprogramm EIGENE EINGABE ermöglicht es dem Anwender, selbst ein Adjektiv auszuwählen, mit dem er üben möchte, statt dies dem Zufallsgenerator zu überlassen. Damit aber kein Unsinn eingegeben wird, muß hier geprüft werden, ob das betreffende Wort bereits im Speicher ist. Verläuft die Prüfung negativ, so wird MM\$ ausgegeben: DIESES WORT IST NICHT GESPEICHERT! Natürlich besteht bei der Eingabe jederzeit die Möglichkeit, das Programm zu verlassen, indem man Q (=QUIT) eingibt. MN\$ ist die Textkonstante, die uns Meldung erstattet, falls ein bestimmtes Adjektiv im Laufe unserer Übung bereits eingegeben wurde. So sollen Wiederholungen vermieden werden. Zeile 2090 sorgt dafür, daß die Textausgabe immer an einer bestimmten Stelle erfolgt. Die eigentliche Prüfung auf Übereinstimmung findet in den Zeilen 2050 – 2080 statt.

- 2530–2560: Das Unterprogramm LEKTIONSWAHL gestattet, eine bestimmte Lektion (UNIT) herauszugreifen und nur diese zu üben.
- 2610–2660: Hier wird gegebenenfalls die Meldung, ob eine im vorangegangenen Unterprogramm gewählte Lektion nicht existiert, ausgegeben. Die Prüfung dieses Tatbestandes findet in Zeile 753 im Programmteil ADJEKTIVE EINLESEN statt. Beachten Sie die regenbogenfarbige Blinkanzeige. Die Farben zaubert POKE 646,E+3 hinein!
- 3020–5000: Der Datensatz wird folgendermaßen eingegeben:  
xxxx DATA Lektionszahl, englisches Adjektiv, Silbenzahl, deutsches Adjektiv  
Beispiel:  
3410 DATA 5,SHORT,1,KURZ  
Die letzte Zeile muß immer so aussehen:  
xxxx DATA 1,"\*",1,"\*"

(Michael Geipel)

```
10 REM *****
20 REM *
30 REM *   DIE STEIGERUNG   *
40 REM *   IM ENGLISCHEN   *
50 REM *
60 REM *****
70 :
80 REM COPYRIGHT 1985 BY
85 REM MODERNE VERLAGS GESELLSCHAFT
90 REM AUTOR:  MICHAEL GEIPEL
100 :
110 W =100
120 DIM F(W),VE$(W),SI(W),GE$(W),VM$(W,4),VH(W)
130 S = 54272
140 X = INT(-TI)
150 POKE 53280,7:POKE 53281,1
160 PRINTCHR$(9);CHR$(142);CHR$(8)
170 :
180 REM *** STEUERZEICHEN DEFINITIONEN *
190 :
200 CO$=CHR$(145): CU$=CHR$(17)
210 CL$=CHR$(157): BL$=CHR$(147)
```

```
220 FS$=CHR$(144): FR$=CHR$(28)
230 FG$=CHR$(30)
240 RA$=CHR$(18) : RE$=CHR$(146)
250 :
260 REM *** WEITERE STRINGDEFINITIONEN *
270 :
320 FOR LL=1TO 40:LL$=LL$+CHR$(175):NEXT
321 FOR T1=1TO 38:T0$=T0$+CHR$(99):NEXT
330 T1$=CHR$(176)+T0$+CHR$(174)
340 T2$=CHR$(98)+FG$+"          C O M P A
    R I S O N          "+FS$+CHR$(98)
350 T3$=CHR$(173)+T0$+CHR$(189)
360 TT$=T1$+T2$+T3$
370 T4$=CHR$(31)+"          HIT NUMBER
    KEY          "+FS$
371 T5$=RA$+T4$+RE$
380 TS$=RA$+FR$+"          HIT SPACE
BAR          "+FS$+RE$
390 :
400 REM *** MENUE: LUECKE/N *****
405 :
410 PRINTBL$;FS$
420 G$(1)="1 GAP"
430 G$(2)="2 GAPS"
440 G$(3)="3 GAPS"
460 G$(4)="POSITIVE"
470 G$(5)="COMPARATIVE"
480 G$(6)="SUPERLATIVE"
490 G$(7)="GERMAN MEANING"
500 PRINT:PRINT TT$
501 FOR U=1 TO 7:PRINT" ";RA$;" ";CL$;U;
CL$;" ";RE$;" ";G$(U):PRINT
502 IF U=3 THEN PRINT FG$;" GAP AT ...";
FS$:PRINT
503 NEXT U
505 FOR C=1 TO 100:NEXT:PRINT T4$;CO$;CO
$
506 GET D$
507 FOR C=1 TO 100:NEXT:PRINT T5$;CO$;CO
$
510 D=VAL(D$)
520 IF D=0 OR D=>8 THEN 505
530 GOSUB 2012
540 :
550 REM *** MENUE: ZUFALL / SEQUENTIELL
```

```
555 :
560 PRINT BL$:PRINT TT$:PRINT
570 PRINT" 1 = RANDOM SELECTION":PRINT
580 PRINT" 2 = SEQUENTIAL ORDER":PRINT
585 PRINT" 3 = OWN INPUT"
590 PRINT:PRINT
591 GET SE$
592 FOR C=1 TO 100:NEXT:PRINT T4$;CO$;CO
$
593 FOR C=1 TO 100:NEXT:PRINT T5$;CO$;CO
$
600 SE=VAL(SE$)
610 IF SE=3 THEN 625
615 IF SE=1 OR SE =2 THEN GOSUB2012:GOSU
B 2530:GOTO 625
620 GOTO 591
625 GOSUB 2012
630 :
700 REM *** ADJEKTIVE EINLESEN *****
710 :
751 B=B+1
752 READ UT,VE$(B),SI(B),GE$(B)
753 IF VE$(B)="*" AND B=1 THEN 2620
754 IF VE$(B)="*" THEN B=B-1:GOTO 785
755 IF UN=0 THEN 751
756 IF SE<>3 AND UN<>UT THEN 752
757 GOTO 751
760 :
770 REM *** BILDSCHIRM-MASKE *****
780 :
785 PRINT BL$:PRINT:PRINT TT$
800 FOR U=4 TO 7:PRINT CHR$(32);G$(U);":
":PRINT
810 NEXT U
811 PRINTCHR$(31);
812 PRINT T1$;CHR$(98);" RETURN = SOLUTI
ON Q = QUIT PROGRAM ";CHR$(98);T3$
840 PRINT FS$;CHR$(19)
920 :
930 REM *** ZUFALLSAUSWAHL DER ADJEKTIVE
935 :
940 IF SE=2 THEN 1040
945 IF SE=3 THEN 2020
950 I=INT(RND(1)*B)+1
970 IF VH(I)=1 THEN 950
```

```
975 VH(I)=1
980 V=V+1
1000 GOTO 1080
1010 :
1020 REM ** SEQUENTIELLE AUSWAHL *****
1030 :
1040 Z=Z+1:I=Z
1050 IF Z=>B+1 THEN 1510
1060 :
1070 REM ** EIN ADJEKTIV-SET UMSPEICHERN
1075 :
1080 V$(1)=VE$(I)
1090 SI =SI(I)
1100 V$(4)=GE$(I)
1110 GOTO 1735
1120 :
1130 REM ** ZUFALLSAUSWAHL DER LUECKEN *
1135 :
1140 IF D=>4 THEN DH(D-3)=1:GOTO 1220
1150 FOR Y = 1 TO D
1160 A=INT(RND(1)*4)+1
1170 IF DH(A)=1 THEN 1160
1180 DH(A)=1
1190 NEXT Y
1200 :
1210 REM ** BILDSCHIRMANZEIGE *****
1215 :
1220 G=G+1
1225 FOR Y = 1 TO 4:VM$(G,Y)=V$(Y)
1230 POKE211,18:POKE 214,2*Y+4:SYS 58732
1240 IF DH(Y)=0 THEN PRINT V$(Y):GOTO 1260
1250 PRINTTAB(18)". "
1260 NEXT Y
1270 FOR Y = 1 TO 4
1275 POKE 211,18:POKE 214,2*Y+4:SYS 58732
1280 IF DH(Y)=0 THEN 1370
1310 PRINTCHR$(31);:INPUT W$:PRINTFS$;:G
OSUB 2010
1315 IF W$=CHR$(32) THEN 1310
1320 L1= LEN(W$):L2=LEN(V$(Y))
1321 IF L1-L2=>0 THEN FOR U=1TO(L1-L2+2)
:LZ$=LZ$+CHR$(32):NEXT U
1340 IF W$=V$(Y) THEN PRINT CO$;:VM$(G,Y
```

```
)=FG$+V$(Y):PRINTTAB(18)VM$(G,Y);FS$;
1341 :
1342 REM ** ALTERNATIVLOESUNGEN *****
1343 :
1344 IF W$="NEXT" THEN W$="NEAREST":GOTO
  1340
1345 IF W$="WELL" THEN W$="GOOD":GOTO 13
  40
1346 IF W$="BADLY" OR W$="ILL" THEN W$="
BAD":GOTO 1340
1347 IF W$="MANY" THEN W$="MUCH":GOTO 13
  40
1348 IF W$="FURTHER" THEN W$="FARTHER":G
OTO 1340
1349 IF W$="FURTHEST" THEN W$="FARTHEST"
:GOTO 1340
1350 IF W$=V$(Y) THEN PRINT LZ$:RG=RG+1:
PRINT:GOTO 1370
1360 PRINT CO$;:VM$(G,Y)=FR$+V$(Y):PRINT
TAB(18)FR$;VM$(G,Y);FS$;LZ$:F=F+1:PRINT
1370 NEXT Y
1380 PRINT:PRINT:LZ$=CHR$(32)
1400 POKE 214,20:SYS 58732
1405 PRINT"  TRIES:";F+RG;"      SCORE:";R
G;
1406 FF=INT(F*10/(F+RG)):IF FF>6 THEN FF
=6
1407 IF FF=0 THEN FF=1
1408 PRINT"      MARK:";FF
1409 IF F=F1 THEN F(I)=1
1410 POKE 214,18:SYS 58732
1415 F1=F
1420 FOR C=1 TO 100:NEXT:PRINT TS$;CO$;C
O$
1430 FOR C=1TO 100:NEXT:PRINT"
          ";CO$;CO$
1440 GET HH$:IF HH$="" THEN 1410
1450 FOR U=1 TO 4:DH(U)=0:NEXT U
1455 IF HH$="Q" THEN GOSUB 2012:V=B+1
1460 IF V=>B THEN 1510
1480 GOTO 785
1490 :
1500 REM ** LEISTUNGSANZEIGE *****
1505 :
1510 PRINT BL$:PRINT:PRINT TT$:PRINT
```

```
1535 PRINTTAB(13)G:PRINTTAB(16)CO$;" SET
/S OF ADJECTIVES"
1540 PRINTTAB(13)RG+F:PRINTTAB(16)CO$;"
FILLED GAP/S"
1550 PRINTTAB(13)RG:PRINTTAB(16)CO$;" CO
RRECT INPUT/S"
1560 PRINTTAB(13)F:PRINTTAB(16)CO$;" WRO
NG INPUT/S"
1570 PRINT:PRINT"    YOUR MARK:";FF
1571 IF FF=1 THEN PRINT LL$:END
1575 PRINT:PRINT
1580 PRINT TS$:POKE 198,0:WAIT 198,1:GOS
UB 2010
1585 PRINTTAB(3)CO$;CO$;FR$;" LEARN THE
'RED' ADJECTIVE FORMS: ";FS$
1586 PRINT
1600 POKE 198,0:WAIT 198,1
1610 FOR I=1 TO 6:PRINT I;CL$;". ";CO$;FS
$:PRINTTAB(6)VM$(I,4);FS$;
1620 FOR Y=1 TO 3:PRINTTAB(20)VM$(I,Y);F
S$:NEXT Y
1630 PRINT:POKE 198,0:WAIT 198,1
1650 NEXT I
1655 :
1656 REM ** LISTE DRUCKEN ? *****
1657 :
1660 PRINTCU$;CU$;" LIST ON PRINTER? (Y/
N)"
1670 POKE 198,0:WAIT 198,1:GET YN$:IF YN
$="N" THEN END
1680 GOSUB 2012:PRINTBL$;CU$;TT$;CU$;CU$
;TS$;CU$
1690 PRINT SPC(8)"IF YOUR PRINTER IS REA
DY.":POKE 198,0:WAIT 198,1
1695 OPEN 4,4:CMD4
1696 PRINT"    LEARN THE MARKED (●) ADJEC
TIVE-FORMS OF UNIT";UN:PRINTLL$;LL$
1700 TB=1:FOR I=1TO6:PRINTI;:FOR Y=1TO 4
1701 LF=LEN(VM$(I,Y))-1
1705 IF ASC(LEFT$(VM$(I,Y),1))=28 THEN V
M$(I,Y)="●"+RIGHT$(VM$(I,Y),LF)
1706 IF ASC(LEFT$(VM$(I,Y),1))=30 THEN V
M$(I,Y)=RIGHT$(VM$(I,Y),LF)
1710 PRINT SPC(TB)VM$(I,Y);:TB=18-LEN(VM
$(I,Y)):IF TB<1 THEN TB=1
```



```
1715 NEXT:PRINT:TB=1:IF I=>9 THEN TB=0
1716 NEXT:CLOSE 4:PRINTLL$:LL$:END
1720 :
1730 REM ** ROMANISCHE STEIGERUNG **
1732 :
1735 H$=V$(1):U=0
1740 IF SI=>3THEN V$(2)="MORE "+H$:V$(3)
="MOST "+H$:V$(5)=V$(2):V$(6)=V$(3)
1745 IF SI=>3 THEN 1140
1750 :
1760 REM ** GERMANISCHE STEIGERUNG **
1770 REM ** RECHTSCHREIBUNG **
1780 :
1790 R=LEN(H$)
1800 R1$=RIGHT$(H$,1)
1810 M2$=MID$(H$,R-1,1)
1830 R2$=RIGHT$(H$,2)
1840 L1$=LEFT$(H$,R-1)
1845 IF R2$="LY" AND SI=>2 THEN SI=3:GOT
O 1735
1850 IF R1$="Y" THEN V$(2)= L1$+"IER":V$
(3)=L1$+"IEST":SI=1:U=1:GOTO 1980
1860 IF R2$="OW" THEN SI=1:GOTO 1980
1870 IF R2$="ER" THEN SI=1
1880 IF R2$="LE" THEN SI=1
1890 IF R1$="E" THEN V$(2)=H$+"R":V$(3)
=H$+"ST":U=1:GOTO 1980
1895 IF MID$(H$,R-2,2)="OO" OR MID$(H$,R
-2,2)="EA" THEN U=0:GOTO1990
1896 IF R2$="UR" THEN U=0:GOTO 1990
1900 FOR I=1 TO 5
1910 IF M2$=MID$("AEIOU",I,1) THEN U=1:G
OTO 1940
1920 NEXT I
1930 GOTO 1980
1940 FOR I=1 TO 5
1945 IF R2$=MID$("EREWED****",I,2) THEN
U=0
1950 IF U=1 AND R1$<>MID$("AEIOU",I,1) T
HEN V$(2)=H$+R1$+"ER":V$(3)=H$+R1$+"EST"
1960 NEXT I
1980 IF SI=2 THEN V$(5)="MORE "+H$:V$(6)
="MOST "+H$
1990 IF U=0 THEN V$(2)=H$+"ER":V$(3)=H$+
"EST"
```

```
1992 V$(5)=V$(2):V$(6)=V$(3)
1995 U=0
1998 :
1999 REM ** AUSNAHMEN *****
2000 :
2001 IF V$(1)="QUIET"THEN V$(2)="QUIETER
":V$(3)="QUIETEST"
2002 IF V$(1)="GOOD" THEN V$(2)="BETTER"
:V$(3)="BEST"
2003 IF V$(1)="BAD" THEN V$(2)="WORSE"
:V$(3)="WORST"
2004 IF V$(1)="LITTLE" THEN V$(2)="LESS"
:V$(3)="LEAST"
2005 IF V$(1)="MUCH" THEN V$(2)="MORE":V
$(3)="MOST"
2006 IF V$(1)="FAR" THEN V$(2)="FARTHER"
:V$(3)="FARTHEST"
2008 GOTO 1140
2009 :
2010 REM ** UNTERPROGRAMM : PIEPSTON ***
2011 :
2012 POKE S+24,15:POKE S+11,17:POKE S+7,
207:POKE S+8,34
2013 POKE S+13,240:FOR P=1 TO 50:NEXT:PO
KE S+11,0:POKE S+24,0
2015 RETURN
2016 :
2017 REM * UNTERPROGRAMM: EIGENE EINGABE
2018 :
2020 MM$=" THIS WORD ISN'T STORED!"
2030 INPUT " WHICH ADJECTIVE";AD$:GOSUB 2
012:IF AD$="Q" THEN GOSUB 2012:GOTO 1510
2040 MN$= " "+RA$+" "+AD$+" "+RE$+" IS A
PREVIOUS INPUT!"
2050 FOR AD=1 TO B
2060 IF VE$(AD)=AD$ AND VH(AD)=1 ORGE$(A
D)=AD$ANDVH(AD)=1THEN MM$=MN$:GOTO 2080
2070 IF VE$(AD)=AD$ OR GE$(AD)=AD$ THEN
I=AD:Z=Z+1:VH(AD)=1:GOTO 1050
2080 NEXT AD
2090 POKE 214,18:SYS 58732:PRINT MM$;CO$
;CO$
2095 IFVE$(AD)<>"*" THEN 2050
2100 FOR C =1 TO 200:NEXT:PRINT"
```

..

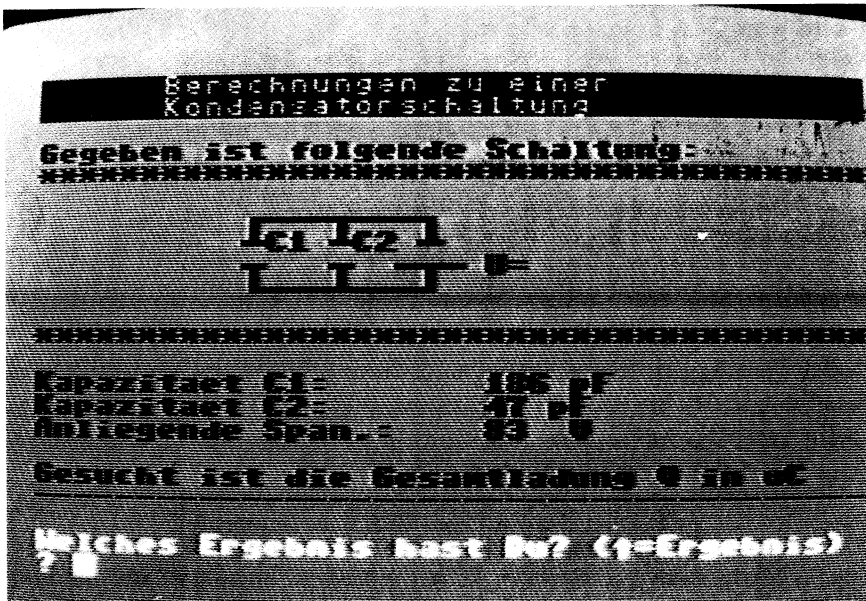
```
2120 E=E+1:IF E=> 10 THEN PRINTCHR$(19):
PRINTTAB(18)"
2125 IF E=>15 THEN E=0:PRINTCHR$(19):GOT
O 2020
2130 GOTO 2090
2500 :
2510 REM ** UNTERPROGRAMM: LEKTIONS-WAHL
2520 :
2530 PRINTBL$;CU$;TT$;CU$;CU$
2540 PRINT" WHICH UNIT, PLEASE? (0-9)";
2550 POKE 198,0:WAIT 198,1:GET UN$
2560 UN=VAL(UN$):RETURN
2600 :
2610 REM ** MELDUNG:UNIT EXISTIERT NICHT
2615 :
2620 PRINTBL$;CU$;CU$;TT$;CU$
2630 FOR C=1TO 50:NEXT:PRINTRE$;"
";CO$
2640 FOR C=1TO 200:NEXT:PRINT SPC(6)RA$;
" THIS UNIT ISN'T STORED! ";CO$
2645 E=E+1
2650 IF E=>10 THEN PRINT CU$;CU$;TS$:GOS
UB 2012:POKE 198,0:WAIT 198,1
2651 IF E=>10 THEN E=0:RESTORE:B=0:GOTO
615
2660 POKE 646,E+3:GOTO 2630
2990 :
3000 REM ** ADJEKTIVE UND SILBENZAHL ***
3005 :
3010 REM ***** UNIT 1 *****
3015 :
3020 DATA 1,NEW,1,NEU
3030 DATA 1,OLD,1,ALT
3040 DATA 1,DIFFICULT,3,SCHWER
3050 DATA 1,GOOD,1,GUT
3060 DATA 1,MUCH,1,VIEL
3070 DATA 1,EASY,1,LEICHT
3090 DATA 1,SMALL,1,KLEIN
3100 DATA 1,FINE,1,FEIN
3110 DATA 1,FULL,1,VOLL
3120 DATA 1,LATE,1,SPAET
3130 DATA 1,NEAR,1,NAHE
3140 DATA 1,LITTLE,1,WENIG
3150 DATA 1,NICE,1,NETT
3160 DATA 1,LONG,1,LANG
```

3170 DATA 1,NORMAL,3,NORMAL  
3175 :  
3180 REM \*\*\*\*\* UNIT 2 \*\*\*\*\*  
3185 :  
3190 DATA 2,NICE,1,NETT  
3200 DATA 2,HAPPY,2,GLUECKLICH  
3210 DATA 2,QUIET,1,RUHIG  
3220 DATA 2,FAR,1,WEIT  
3230 DATA 2,INTERESTING,3,INTERESSANT  
3240 DATA 2,BUSY,1,GESCHAEFTIG  
3250 DATA 2,GREAT,1,GROSS  
3260 DATA 2,EARLY,1,FRUEH  
3270 DATA 2,OFTEN,3,OFT  
3275 :  
3280 REM \*\*\*\*\* UNIT 3 \*\*\*\*\*  
3285 :  
3290 DATA 3,HUNGRY,1,HUNGRIG  
3300 DATA 3,BAD,1,SCHLECHT  
3310 DATA 3,LITTLE,1,WENIG  
3315 :  
3320 REM \*\*\*\*\* UNIT 4 \*\*\*\*\*  
3325 :  
3330 DATA 4,COLD,1,KALT  
3340 DATA 4,HOT,1,HEISS  
3350 DATA 4,GOOD,1,GUT  
3360 DATA 4,LOVELY,1,LIEBLICH  
3370 DATA 4,SOON,1,BALD  
3375 :  
3380 REM \*\*\*\*\* UNIT 5 \*\*\*\*\*  
3385 :  
3390 DATA 5,BEAUTIFUL,3,SCHOEN  
3400 DATA 5,EXCITING,3,SPANNEND  
3410 DATA 5,SHORT,1,KURZ  
3420 DATA 5,LONG,1,LANG  
3430 DATA 5,SILLY,1,ALBERN  
3440 DATA 5,STUPID,3,DUMM  
3450 DATA 5,FREE,1,FREI  
3460 DATA 5,SUNNY,1,SONNIG  
3470 DATA 5,PRETTY,1,HUEBSCH  
3480 DATA 5,POOR,1,ARM  
3485 :  
3490 REM \*\*\*\*\* UNIT 6 \*\*\*\*\*  
3495 :  
3500 DATA 6,BRIGHT,1,KLAR  
3510 DATA 6,COOL,1,KUEHL

3520 DATA 6,CLOUDY,1,WOLKIG  
3530 DATA 6,WET,1,NASS  
3540 DATA 6,STRONG,1,STARK  
3550 DATA 6,CLEAN,1,SAUBER  
3560 DATA 6,DIRTY,1,SCHMUTZIG  
3570 DATA 6,KIND,1,FREUNDLICH  
3580 DATA 6,CHEAP,1,BILLIG  
3590 DATA 6,PLEASANT,3,ANGENEHM  
3600 DATA 6,TERRIBLE,3,SCHRECKLICH  
3610 DATA 6,NECESSARY,3,NOTWENDIG  
3615 :  
3620 REM \*\*\*\*\* UNIT 7 \*\*\*\*\*  
3625 :  
3630 DATA 7,FAMOUS,3,BERUEHMT  
3640 DATA 7,HELPFUL,3,HILFREICH  
3650 DATA 7,WELL-KNOWN,3,BEKANNT  
3660 DATA 7,SUCCESSFUL,3,ERFOLGREICH  
3670 DATA 7,PLEASED,3,ERFREUT  
3680 DATA 7,MARVELLOUS,3,HERRLICH  
3690 DATA 7,GOOD,1,GUT  
3700 DATA 7,ANGRY,1,AERGERLICH  
3710 DATA 7,SAFE,1,SICHER  
3715 :  
3720 REM \*\*\*\*\* UNIT 8 \*\*\*\*\*  
3725 :  
3730 DATA 8,YOUNG,1,JUNG  
3740 DATA 8,MUCH,1,VIEL  
3750 DATA 8,LITTLE,1,WENIG  
3760 DATA 8,ELEGANT,3,ELEGANT  
3770 DATA 8,MODERN,3,MODERN  
3780 DATA 8,COSY,1,GEMUETLICH  
3790 DATA 8,TIDY,1,ORDENTLICH  
3800 DATA 8,HEALTHY,1,GESUND  
3810 DATA 8,TERRIBLE,3,SCHRECKLICH  
3820 DATA 8,CLEVER,1,KLUG  
3830 DATA 8,POLITE,3,HOEFLICH  
3840 DATA 8,FEW,1,WENIGE  
3850 DATA 8,POPULAR,3,BELIEBT  
3855 :  
3860 REM \*\*\*\*\* UNIT 9 \*\*\*\*\*  
3865 :  
3870 DATA 9,NECESSARY,3,NOTWENDIG  
3880 DATA 9,USEFUL,3,NUETZLICH  
3890 DATA 9,RICH,1,REICH  
3894 :

```
3895 REM **** ADDITIONAL ADJECTIVES ****
3896 :
3900 DATA 0,STUFFY,1,MUFFIG
3910 DATA 0,FUNNY,2,SPASSIG
3920 DATA 0,TIRED,1,MUEDE
3930 DATA 0,HARD,1,HART
3940 DATA 0,EXPENSIVE,3,TEUER
3950 DATA 0,SLOW,1,LANGSAM
3960 DATA 0,HIGH,1,HOCH
3970 DATA 0,LOW,1,NIEDRIG
3980 DATA 0,HEAVY,1,SCHWER
4000 :
8990 REM    DATENEINGABE  IMMER  MIT
          1,"*",1,"*" ABSCHLIESSEN!
9000 DATA 1,"*",1,""
```

## 5 Physik-Trainer



### Programm-Telegramm

- Jederzeit Lösungsabfrage durch Eingabe von ↑ möglich.
- Leistungsbewertung durch Kommentar, nicht durch Schulnote.  
»Physik-Trainer« stellt menügesteuerte Aufgaben zum Thema ...
- Widerstandsschaltungen mit 2, 3, 4 oder 5 Widerständen,
- Spulenberechnungen,
- Kondensatorschaltungen.

### Programmablauf

#### Wie man mit diesem Programm arbeitet

Nachdem das Programm eingetippt ist und keine Syntaxfehler mehr vorhanden sind, kann es gestartet werden. Als erstes erscheint eine Übersicht der zu trainierenden Gebiete. Der Schwerpunkt liegt dabei auf Widerstandsschaltungen.

Man kann zwischen Schaltungen mit zwei, drei, vier oder fünf und mehr Widerständen wählen. Die restlichen beiden Gebiete befassen sich mit Spulenberechnung und Kondensatorschaltung. Eine fehlende Größe soll bestimmt werden. Es besteht auch die Möglichkeit, alle Gebiete gemischt zu üben. Durch Druck auf die richtige Taste mit der zugehörigen Ziffer 8 verzweigt das Programm zu dem angesprochenen Gebiet. Folgend wird jedes Gebiet einzeln erklärt.

## Widerstandsschaltungen

Zuerst die dazu notwendigen Formeln:

$$\text{Reihenschaltung: } R_{\text{ges}} = R_1 + R_2 + \dots R_n$$

$$\text{Parallelschaltung: } \frac{1}{R_{\text{ges}}} = \frac{1}{R_1} + \frac{1}{R_2} + \dots \frac{1}{R_n}$$

Bei allen Teilaufgaben wird zuerst der Bildschirm gelöscht, und in der oberen Hälfte erscheint die zu berechnende Schaltung. Gesucht ist hier der Gesamtwiderstand. Darunter gibt das Programm zufällig ausgewählte Widerstandswerte aus. Nun sollst du die gegebene Schaltung berechnen. Nach der Frage

»Welches Ergebnis hast Du? (↑ = Ergebnis)«

ist man aufgefordert, den errechneten Wert einzugeben. Hierbei ist anzumerken, daß bei der Eingabe von Rechenresultaten eine Genauigkeit von einer Kommastelle genügt. Ist das Ergebnis richtig, was hoffentlich meist der Fall ist, erscheint

»Sehr gut! Du hast richtig gerechnet!«

Wenn »Noch einmal? (j/n)« mit **j** beantwortet wird, stellt der Computer eine neue Aufgabe. Bei der Eingabe von **n** kehrt das Programm zum Hauptmenü zurück.

Wurde ein falsches Ergebnis eingegeben, wird dies in roter Schrift mit dem Hinweis

»Leider falsch! Überprüfung notwendig?«

mitgeteilt. Jetzt solltest du deine Rechnung nach eventuellen Fehlern durchsuchen. Die erneute Eingabe geht wie oben beschrieben vor sich. Konntest du nach mehreren Versuchen noch kein richtiges Resultat ermitteln, kann die Abfrage mit der Eingabe von ↑ abgebrochen werden. Der Pfeil nach oben soll immer an diesen »Fluchtweg« erinnern. Das Programm gibt das richtige Ergebnis folgendermaßen aus:

»Mein Ergebnis zum Vergleich: xxxxxx.«

Es empfiehlt sich dann, den betreffenden Stoff erneut in einem Physikbuch oder -heft anzusehen.

Die hier beschriebene Eingabe von Rechenresultaten gilt ebenfalls für die anderen Aufgabengruppen.



## Berechnungen zur Spule

Hierbei geht es um die Induktivität einer Spule, also um die Formel

$$L = \mu_o * A * \frac{N^2}{l} \quad (\mu_o = \text{Permeabilitätskonstante})$$

Gesucht ist immer eine fehlende Größe. Die restlichen Daten werden wieder vom Programm per Zufall bestimmt. Dadurch entstehen manchmal irrealer Werte. Dies soll aber nicht weiter stören, da hier die Umstellung von Formeln erlernt werden soll. Die Eingabe von Ergebnissen ist dieselbe wie bereits unter »Widerstandsschaltungen« beschrieben.

## Kondensatorschaltung

Die benötigten Formeln lauten:

$$C = \frac{Q}{U} \quad C_{\text{ges}} = C_1 + C_2$$

Die Schaltung, die wieder in der oberen Bildschirmhälfte erscheint, beinhaltet zwei Kondensatoren und eine Gleichspannungsquelle  $U =$ . Die beiden Kapazitäten der Kondensatoren und die anliegende Spannung sind immer gegeben und werden vom Programm angezeigt. Gesucht sind jeweils abwechselnd die Ladung  $C_1$  oder  $C_2$ , die Gesamtladung oder -kapazität. Die geforderte Maßeinheit ist zu beachten. Um Mißverständnisse zu vermeiden, ist mit  $\mu\text{C}$  Mikrocoulomb und mit  $\mu\text{F}$  Mikrofarad gemeint. Die Ergebniseingabe ist dieselbe wie bereits beschrieben.

## Alle sechs Aufgabengruppen gemischt

Per Zufallsfunktion gesteuert, wählt das Programm einen von 6 Abschnitten aus und springt anschließend dorthin. Anstatt "Noch einmal? (j/n)" erscheint jetzt "Weiter? (j/n)", wobei bei der Eingabe von j wiederum zu einem weiteren Aufgabengebiet verzweigt wird. Andernfalls kehrt man mit n ins Hauptmenü zurück.

## Beenden des Programms

Das Ende des Programms erreicht man im Hauptmenü mit der Eingabe von 9. Der Bildschirm wird gelöscht und die Leistung bewertet. Keine Angst, es gibt keine Schulnoten, sondern nur vier verschiedene Kommentare, die aus dem Verhältnis der richtigen Ergebnisse zur Anzahl der Eingaben ausgegeben werden. Die Leistungsbewertung ist unverbindlich und kann, zum Beispiel in der Abstufung, von dir verbessert und erweitert werden.

Das Programm verabschiedet sich mit

»Bis bald auf ein neues Training!«

## Programmbeschreibung

### Die wichtigsten Programmzeilen kurz erklärt:

- 120 – 170: Voreinstellungen wie Bildschirm gelb, Kleinbuchstaben und oft verwendete Steuerzeichen.
- 210 – 370: Hauptmenü. Verzweigung zu den einzelnen Gebieten.
- 410 – 460: Beenden des Programms. 0.4, 0.7, 0.9 bedeuten <40%, <70%, <90% der Aufgaben richtig gelöst.
- 500: Widerstand per RND (X)-Funktion bestimmen.
- 520 – 550: Einsprungrzeilen der einzelnen Widerstands-Bereiche.
- 560: Wählt die zu bearbeitende Schaltung aus.
- 570 – 580: Variablen zur Darstellung der Schaltung auf dem Bildschirm.
- 590 – 600: Ausgewählte Schaltung auf dem Bildschirm zeichnen.
- 610: Widerstandswerte ausgeben.
- 620: Zu der Schaltung gehörigen Gesamtwiderstand berechnen.
- 630 – 660: Ergebnisseingabe und Abfrage nach weiteren Aufgaben.
- 700 – 1070: Schaltungen
- 1110 – 1250: Formeln zum Gesamtwiderstand.
- 1290 – 1360: Spulenberechnung. Per Zufallsfunktion zu einer Aufgabe verzweigen. Ergebnisseingabe.
- 1380 – 1600: Die einzelnen Aufgaben zur Spulenberechnung.
- 1640 – 1740: Ergebnisseingabe.
- 1750: Einen Zufallswert bestimmen.
- 1790 – 1860: Kondensatorschaltung. Schaltplan auf den Bildschirm zeichnen.
- 1870 – 1890: Gegebene Werte bestimmen und anzeigen.
- 1900: Per Zufallsfunktion zu einer Aufgabe verzweigen.
- 1950 – 2050: Vier Kondensatoraufgaben.
- 2090 – 2120: Gemischte Aufgaben.  
Per Zufall eine von sechs Teilaufgaben auswählen und anschließend verzweigen.

### Tips zum Eingeben des Programms

Die folgenden Hinweise sollen Dir das Eintippen des Listings erleichtern. Als erstes kannst Du alle REM-Zeilen weglassen. Diese sind für den Programmablauf nicht nötig. Viele Zeilen können kopiert werden, indem du eine Zeile eingibst und dann mit dem *Cursor* hochfährst und die Zeilennummer und einige andere Stellen änderst. Anzuwenden wäre dies zum Beispiel bei den Zeilen 250 – 310, 520 – 540, 1380 – 1600. Vergiß aber nicht die Zeilennummer zu ändern, sonst überschreibst du die alte Zeile! Die Zeilen 700 – 1070, also die Schaltpläne der Widerstände, verlangen bei der Eingabe Auf-

merksamkeit. Hier kurz die Zeichen und die entsprechenden Tastenkombinationen, die man eingeben muß:

```
'r' = 'C=' + 'A'      'f' = 'C=' + 'Q'
'7' = 'C=' + 'S'      '4' = 'C=' + 'W'
'U' = 'C=' + 'Z'      '1' = 'C=' + 'E'
'J' = 'C=' + 'X'      'T' = 'C=' + 'R'
'-' = 'SHIFT' + '*'

      'C=' Commodore-Taste
```

Die inversen Zeichen sollen auf dem Bildschirm ein inverses Leerfeld (■) bedeuten. Man erzeugt es mit folgender Tastenkombination:

'CTRL + 9 Leerzeichen CTRL + 0'

Wenn du Lust hast, kannst du die Widerstandsschaltungen sowie alle anderen Aufgaben ergänzen!

Ich wünsche dir viel Freude mit diesem Programm!

(Christoph Bronold)

```
10 rem " ***** "
20 rem " * * "
30 rem " * Physik - Trainer * "
40 rem " * * "
50 rem " ***** "
60 :
70 rem " Copyright 1985 by "
80 rem " moderne verlags gesellschaft"
90 :
95 rem " Autor: Christoph Bronold "
110 :
120 poke53280,7:poke53281,7:rem *"Bild-
schirm gelb
130 printchr$(14):rem *"Kleinbuchstaben
140 ro$=chr$(18) :rem *"RVS ein
150 ch$=chr$(145):rem *"Cursor hoch
160 cl$=chr$(147):rem *"Bildschirm loe-
schen
```

```
170 for i=1to40:ba$=ba$+"-":bs$=bs$+"*":
next i
180 :
190 rem"*** Hauptmenue *****
200 :
210 printchr$(30);cl$;bs$"*      P h y s
   i k - T r a i n e r      *";bs$
220 id=0:rem"* Keine gemischten Aufgaben
230 printchr$(31)"Folgende Gebiete stehe
n zur Verfuegung: "ba$
240 print,"      Widerstandsschaltungen
   :";
250 print,"* 2 *   mit 2 Widerstaenden"
260 print,"* 3 *   mit 3 Widerstaenden"
270 print,"* 4 *   mit 4 Widerstaenden"
280 print,"* 5 *   mit 5 und mehr Widerst
   ."
290 print,"* 6 *   Berechnungen zur Spule
   ":print
300 print,"* 7 *   Kondensatorschaltung":
   print
310 print,"* 8 *   Alle 6 Gebiete gemisch
   t"
320 print,"* 9 *   Ende des Trainings":pr
   int:printba$;
330 print,"Deine Auswahl (2-9) ?":printb
   a$;
340 get t$:if t$=""then 340
350 if t$<"2"or t$>"9"then 340
360 on val(t$)gosub10,520,530,540,550,12
   70,1790,2090,410
370 goto210
380 :
390 rem"*** Leistungsbewertung und
   Programmende      ****
400 :
410 printcl$:ifaf<4then460
420 qu=ar/af:ifqu<0.4then print"Duarfs
   t noch mehr ueben!":goto460
430 ifqu<0.7then print"Das war eine mitt
   elmaessige Leistung!":goto460
440 ifqu<0.9then print"Eine gute Leistun
   g!      ":goto460
450 print"Ausgezeichnet! Du beherrschst
   diesen Stoff sehr gut!"
```

```

460 print:print:print"Bis bald auf ein n
eues Training !":end
470 :
480 rem"*** Widerstandsschaltungen ***
490 :
500 fori=1toaz:r(i)=int(rnd(1)*i*50)+i*8
510 print"R"i="r(i)"Ohm",:next:return
520 am=2:az=2:vk=1:goto560
530 am=4:az=3:vk=3:goto560
540 am=3:az=4:vk=7:goto560
550 am=4:vk=10
560 zf=int(rnd(1)*am)+vk
570 11$="|      |" :12$="|      |"
580 13$="|      |"
590 printcl$ro$"Berechne bitte folgende
Schaltung:":printchr$(129)bs$
600 on zf gosub 700,720,740,760,780,810,
830,850,880,900,940,990,1040
610 print:printbs$:print"Die Widerstaend
e haben die Werte:":print:gosub500
620 on zf gosub110,1120,1130,1140,1150,
1160,1170,1180,1190,1200,1210,1220,1240
630 gosub1640
640 if id=1 then return
650 print:input"Noch einmal? (j/n) ";fr$
:iffr$="j"then560
660 return
670 :
680 rem"*** Schaltungen *****
690 :
695 rem "⏏ = RVS on (CTRL+9)
      ⏏ = RVS off (CTRL+0)"
696 :
700 print"⏏ R1⏏ ⏏ R2⏏ ⏏":return
710 :
720 print"⏏ R1⏏ ⏏":print12$:prin
t"⏏ R2⏏ ⏏":return
730 :
740 print"⏏ R1⏏ ⏏":print12$:prin
t"⏏ R2⏏ ⏏ R3⏏ ⏏":return
750 :
760 print"⏏ R1⏏ ⏏ R2⏏ ⏏ R3⏏ ⏏
":return
770 :

```

```

780 print"┌─┐ R1 ─┐":print12$:prin
t"┌─┐ R2 ─┐":print12$
790 print"└─┐ R3 ─┐":return
800 :
810 print"┌─┐ R1 ─┐":print11$:print"└─┐
R2 ─┐└─┐ R3 ─┐":return
820 :
830 print"┌─┐ R1 ─┐└─┐ R2 ─┐":print12$:
print"└─┐ R3 ─┐└─┐ R4 ─┐":return
840 :
850 print"┌─┐ R1 ─┐└─┐ R2 ─┐":print12$
860 print"└─┐ R3 ─┐└─┐ R4 ─┐":ret
urn
870 :
880 print"┌─┐ R1 ─┐":print11$:print"└─┐
R2 ─┐└─┐ R3 ─┐└─┐ R4 ─┐":return
890 :
900 print"┌─┐ R1 ─┐":az=6:print11$
910 print"└─┐ R2 ─┐└─┐ R3 ─┐":print12$
920 print"└─┐ R4 ─┐└─┐ R5 ─┐└─┐ R6 ─┐":
return
930 :
940 print"┌─┐ R1 ─┐":az=6:print11$
950 print"└─┐ R2 ─┐└─┐ R3 ─┐":print12$
960 print"└─┐ R4 ─┐└─┐ R5 ─┐":prin
t11$
970 print"└─┐ R6 ─┐":return
980 :
990 print"└─┐ R1 ─┐":az=9:pri
nt13$
1000 print"└─┐ R2 ─┐└─┐ R3 ─┐└─┐ R4 ─┐"
:print13$
1010 print"└─┐ R5 ─┐└─┐ R6 ─┐└─┐ R7 ─┐└─┐
R8 ─┐":print" "11$
1020 print"└─┐ R9 ─┐":return
1030 :
1040 print"┌─┐ R1 ─┐└─┐ R2 ─┐":az=9:pri
nt12$
1050 print"└─┐ R3 ─┐└─┐ R4 ─┐":pri
nt13$
1060 print"└─┐ R5 ─┐└─┐ R6 ─┐└─┐ R7
─┐":print11$" "11$
1070 print"└─┐ R8 ─┐└─┐ R9 ─┐":retur
n

```

```

1080 :
1090 rem"*** Formeln *****
1100 :
1110 er=r(1)+r(2):return
1120 er=1/(1/r(1)+1/r(2)):return
1130 er=1/(1/r(1)+1/(r(2)+r(3))):return
1140 er=r(1)+r(2)+r(3):return
1150 er=1/(1/r(1)+1/r(2)+1/r(3)):return
1160 er=1/(1/r(1)+1/r(2))+r(3):return
1170 er=1/(1/(r(1)+r(2))+1/(r(3)+r(4))):
return
1180 er=r(4)+1/(1/(r(1)+r(2))+1/r(3)):re
turn
1190 er=1/(1/r(1)+1/r(2))+r(3)+r(4):retu
rn
1200 er=r(6)+1/(1/(r(4)+r(5))+1/(r(3)+1/
(1/r(1)+1/r(2)))):return
1210 er=r(5)+1/((1/r(4)+1/r(6)+(r(3)+1/(
1/r(1)+1/r(2))))):return
1220 er=1/(1/r(1)+1/(r(2)+r(3)+r(4))+1/(
r(5)+1/(1/r(6)+1/r(9))+r(7)))
1230 er=er+r(8):return
1240 rh=1/(1/(1/r(5)+1/r(8))+1/(1/r(6)+1
/r(9)))
1250 er=1/(rh+1/(r(4)+1/(1/(r(1)+r(2))+1
/r(3))))+r(7):return
1260 :
1270 rem"*** Spulenberechnungen *****
1280 :
1290 print cl$;ro$;"Gesucht sind die res
tlichen Daten zu ei-";
1295 printro$"ner Spule.
":print
1300 uo=12.566e-7:rem "< Permeabili-
taetskonstante"
1310 print"Gegeben sind:":print:printchr
$(129);ba$
1320 on int(rnd(1)*4)+1 gosub 1380,1440,
1500,1560
1330 if id=1 then return
1340 print:input"Moechtest Du es noch
einmal wagen? (j/n)";fr$
1350 if fr$="j"then 1290
1360 return
1370 :

```

```
1380 fk=1e4:gosub1750:a=w:print"Flaeche
A: ",a"cm2"
1390 fk=1e3:gosub1750:n=w:print"Windungs
zahl: ",n
1400 fk=100:gosub1750:l=w:print"Spulenla
enge l: ",l"cm"
1410 print:print"Gesucht ist die Indukti
vitaet L in H"
1420 er=uo*a*n2/1/100:gosub1640:return
1430 :
1440 fk=5:gosub1750:in=w:print"Induktivi
taet L: ",in"H"
1450 fk=50:gosub1750:l=w:print"Spulenlae
nge l: ",l"cm"
1460 fk=1e4:gosub1750:a=w:print"Flaeche
A: ",a"cm2"
1470 print:print"Gesucht ist die Windung
szahl."
1480 er=sqr(in*100*1/a/uo):gosub1640:ret
urn
1490 :
1500 fk=5:gosub1750:in=w:print"Induktivi
taet L: ",in"H"
1510 fk=100:gosub1750:l=w:print"Spulenla
enge l: ",l"cm"
1520 fk=1e3:gosub1750:n=w:print"Windungs
zahl: ",n
1530 print:print"Gesucht ist die Spulenf
laeche in m2."
1540 er=in*1/uo/n2/100:gosub1640:return
1550 :
1560 fk=5:gosub1750:in=w:print"Induktivi
taet L: ",in"H"
1570 fk=1e3:gosub1750:n=w:print"Windungs
zahl: ",n
1580 fk=1e4:gosub1750:a=w:print"Flaeche
A: ",a"cm2"
1590 print:print"Gesucht ist die Spulenl
aenge in m."
1600 er=uo*n2*a/in/1e4:gosub1640:return
1610 :
1620 rem"*** Ergebniseingabe *****"
1630 :
1640 er=int(er*1000)/1000:af=af+1:print:
printba$chr$(5)
```



```

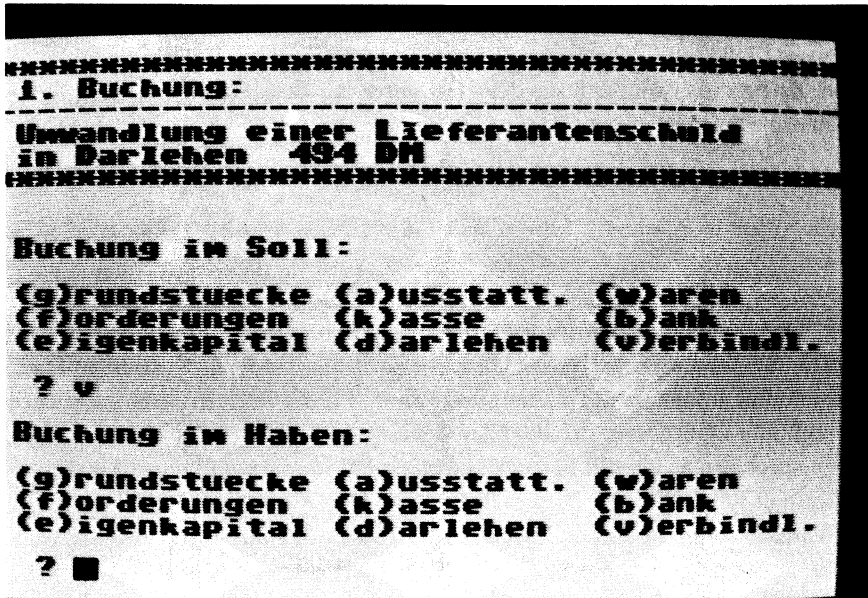
1650 print"Welches Ergebnis hast Du? (↑=
Ergebnis)":input eg$
1660 ifeg$="↑"thenprint"Ergebnis:",er;ch
r$(30):eg$="":return
1670 eg=val(eg$):if(er-.1<eg)and(eg<er+.
1)then1700
1680 printch$
      "

1690 printch$ch$;ch$;chr$(28);"Leider fa
lsch! Ueberpruefung notwendig!":goto1650
1700 print:printchr$(31)"Sehr gut! Du ha
st richtig gerechnet!":print
1710 ar=ar+1:print"Mein Ergebnis zum Ver
gleich:"er:return
1720 :
1730 rem"*** 1 Zufallswert *****
1740 :
1750 w=int(rnd(1)*fk)+1:return
1760 :
1770 rem"*** Kondensatorschaltung *****
1780 :
1790 printcl$;ro$;"          Berechnungen zu
      einer          "
1800 print ro$;"          Kondensatorschaltu
ng          "
1810 print:print"Gegeben ist folgende Sc
haltung:":printchr$(129)bs$
1820 print,"          "
1830 print,"┌C1┌C2┐"
1840 print,"└──┬──┘ U="
1850 print,"└──┬──┘"
1860 print:printbs$
1870 fk=999:gosub1750:c1=w:print"Kapazit
aet C1: ",c1"pF":
1880 fk=999:gosub1750:c2=w:print"Kapazit
aet C2: ",c2"pF"
1890 fk=100:gosub1750:u=w:print"Anliegen
de Span.:",u" V"
1900 on int(rnd(1)*4)+1 gosub1950,1980,2
010,2040
1910 ifid=1thenreturn
1920 input"Nochmal (j/n) ";fr$:iffr$="j"
then1790
1930 return

```

```
1940 :
1950 print:print"Gesucht ist Ladung Q1 v
on C1 uC."
1960 er=c1*u/1e3:gosub1640:return
1970 :
1980 print:print"Gesucht ist Ladung Q2 v
on C2 in uC."
1990 er=c2*u/1e3:gosub1640:return
2000 :
2010 print:print"Gesucht ist die Gesamtl
adung Q in uC";
2020 er=((c1+c2)*u)/1e3:gosub1640:return
2030 :
2040 print:print"Gesucht ist die Gesamtk
apaz. Cges in uF"
2050 er=(c1+c2)/1e3:gosub1640:return
2060 :
2070 rem"*** Gemischte Aufgaben ****"
2080 :
2090 id=1:on (int(rnd(1)*6)+1) gosub520,
530,540,550,1270,1790
2100 print:input"Weiter (j/n) ";fr$
2110 iffr$="j"then2090
2120 id=0:return
```

## 6 Grundlagen der Buchführung



### Programm-Telegramm . . .

- Der Computer stellt die Aufgaben; die Buchungen müssen mitgeschrieben werden.
- Innerhalb der einzelnen Abschnitte ist die Lösung jederzeit abfragbar.
- Die Leistungsbewertung erfolgt durch Kommentare.
- Das Programm kann uneingeschränkt erweitert werden.

### . . . mit Anmerkungen

Damit das Programm nicht schwerfällig wird, habe ich mich auf den einfachen Buchungssatz ohne Mehrwertsteuer und Abschreibung beschränkt. Geübten Programmierern bereitet es keine Schwierigkeiten, das Programm zu erweitern.

Der Computer stellt nur die Aufgaben und verlangt die Eingabe der betreffenden Konten und später jene der Salden. Natürlich könnte das Programm mehr Schreibarbeit abnehmen, aber dann würde es in eine reine Additionsübung ausarten, und die Möglichkeiten, Fehler zu machen, wären äußerst gering.

## Programmablauf

Hat man in mühevoller Arbeit ein Programm eingetippt und startet es zum ersten Mal, kommt es häufig erst einmal zu einem »SYNTAX ERROR IN . . .«. Nur nicht verzweifeln! Vergleiche bitte die eingegebenen Zeilen genau mit denen des Listings im Buch, denn im Programm selbst sind keine Syntax-Fehler.

Nach dem Verbessern der letzten Fehler ist das Programm lauffähig.

Um die Zeit zu überbrücken, die der Computer zur Bearbeitung der Eingabe benötigt, erscheint zunächst

»Eingabe wird bearbeitet. Bitte warten!«

sowie die öfter verwendete Überschrift »GRUNDLAGEN DER BUCHFÜHRUNG«.

Als erstes wirst Du aufgefordert einzugeben, wie viele Buchungen Du bearbeiten möchtest. Danach folgt die Eröffnungsbilanz. Ihre Werte werden mit der Zufallsfunktion 'RND (X)' berechnet und stimmen nicht ganz mit der Realität überein. Dies soll aber nicht weiter stören, da mit diesem Programm nur die Buchungstechnik vermittelt werden soll. Die Darstellung der Bilanzen weicht ebenfalls von der Norm ab. So sind hier keine sogenannten »Buchungsnasen« dargestellt, sondern nur »- -« mit dem Wert Null. Wie man sieht, läßt sich eben nicht alles auf dem Computer darstellen, will man den Aufwand in Grenzen halten. Um mit dem Programm vernünftig arbeiten zu können, ist es notwendig, sich die Eröffnungsbilanz auf einem Zettel zu notieren beziehungsweise und/oder die einzelnen Konten mit den Anfangsbeständen. Mit einem beliebigen Tastendruck nach der Aufforderung

»Weiter = beliebige Taste druecken!«

gelangt man zur ersten Buchungsaufgabe. Der Bildschirm wird gelöscht, und folgendes erscheint:

```
*****
1. Buchung:
-----
(Hier steht jeweils die Buchungsaufgabe)
*****

Buchung im Soll:
(g)rundstuecke (a)usstatt. (w)aren
(f)orderungen (k)asse (b)ank
(e)igenkapital (d)arlehen (v)erbindl.
?
```

Nach dem Fragezeichen muß der Anfangsbuchstabe – nur Kleinbuchstaben sind zulässig! – des entsprechenden Kontos eingegeben werden. Bei falscher Eingabe (z.B. Großbuchstaben) weist das Programm folgendermaßen darauf hin:

»Wie bitte?«

Der *Cursor* springt wieder in die Zeile mit dem Fragezeichen. Nun ist noch die Gegenbuchung im Haben nötig. Diese erfolgt im Prinzip wie die Buchung im Soll.

Die Eingabeschleife kann nur dadurch abgebrochen werden, indem man einfach alle Anfangsbuchstaben durchprobiert. Du sollst natürlich auch auf deinem Zettel die Buchung durchführen. Nun kann erst die nächste Aufgabe in Angriff genommen werden.

Nachdem alle Aufgaben gestellt sind, folgt nun die Eingabe der einzelnen Abschlußsalden. Schließe alle Konten ab und beginne dann erst mit der Eingabe der einzelnen Salden. Dabei erscheint in der zweiten Zeile zwischen zwei Strichen das entsprechende Konto. Mit

»Saldobetrag: (↑ = Ergebnis) ?«

wirst du zur Eingabe deines Ergebnisses aufgefordert. Da das Programm mit ganzen DM-Beträgen rechnet, sind Eingaben mit Kommastellen sinnlos. Wenn

»Leider falsch!«

auf dem Bildschirm erscheint, war dein Ergebnis falsch. Bei richtiger Lösung wird die Bilanz gezeigt. Hast du nach intensiver Suche deinen Fehler nicht gefunden, kann das richtige Ergebnis mit der Eingabe von "↑" abgefragt werden.

Dieses gibt das Programm in der Form von

»Ergebnis: xxxx«

aus, und darunter erscheint die Bilanz.

Diese ist wieder computertypisch dargestellt. Nach einem beliebigen Tastendruck wird mit dem nächsten Konto fortgefahren, und die Saldoeingabe erfolgt in gleicher Weise. Nachdem alle Salden eingegeben wurden, bist du am Ende des Programms angelangt. Nun wird auf dem Bildschirm folgendes ausgegeben:

»Nun bist Du am Ende der Trainingseinheit angelangt.  
Deine Leistung war . . .«

Aus dem Verhältnis der falschen Eingaben zur Gesamtzahl der Eingaben wird deine Leistung bewertet und ein entsprechender Kommentar ausgegeben. Abschließend fragt das Programm mit

»Moechtest Du noch weiter trainieren? (j/n)«

nach einem erneuten Durchlauf, der mit der Eingabe von 'j' erfolgt. Ansonsten ist das Programm nun beendet und verabschiedet sich mit

»Bis bald zur naechsten Trainingseinheit!«

## Programmbeschreibung

### Die wichtigsten Programmzeilen kurz erklärt

- 160 – 290: Definition der Bildschirmfarben, Felder, Farben, Linien.  
330 – 350: Eingabe der Anzahl der Buchungen.  
370: Überlesen der Buchungsaufgaben.  
380 – 440: Aufstellen der Eröffnungsbilanz.  
450 – 510: Ausgabe der Eröffnungsbilanz.  
530: Warten, bis eine Taste gedrückt wird.  
560 ff: Buchungsaufgabe stellen.  
580: l = Schleifenzähler  
an = Anzahl der gestellten Aufgaben.  
af = Anzahl der falschen Eingaben.  
600: az = Anzahl der Buchungsaufgaben.  
DATA-Zeilen so lange durchsuchen, bis die zufällig bestimmte Aufgabe i gefunden ist.  
610: Buchungswert per RND (X)-Funktion bestimmen.  
620: Buchungswert ausgeben und falls  $i > 10$  zum zweiten Buchungsblock verzweigen.  
640: Verzweigung der dazugehörigen Buchung in den Konten.  
650 – 940: Anzahl der Einträge im Sollkonto auf sb erhöhen, in das Gegenkonto die sb-te Buchung eintragen und den Buchungswert abspeichern.  
Dito im Habenkonto durchführen.  
sb\$ = Anfangsbuchstabe des Sollkontos.  
hb\$ = Anfangsbuchstabe des Habenkontos.  
950: Zur 10. – 17. Aufgabenlösung verzweigen.  
960 – 1160: Wie von Zeile 650 – 940.  
1200: Ausgabe von »Buchung im Soll« und Konten.  
1210: Eingabe des Anfangsbuchstabens des Sollkontos.  
1220: Ausgabe von »Buchung im Haben« und Konten.  
1230: Eingabe des Anfangsbuchstabens des Habenkontos.  
1240: Schleifenzähler um 1 erhöhen, und falls noch nicht alle Aufgaben abgearbeitet sind, nach Zeile 590 zurückspringen.  
1270: Hilfsunterprogramme.  
1290: »Eingabe wird bearbeitet . . .« ausgeben.  
1310: Überschrift »Grundlagen . . .« ausgeben.  
1330: Ausgabe der möglichen Konten.  
1370 ff: Saldeneingabe  
1420: Aufsummieren der Beträge der Aktiva-Konten auf der Sollseite.  
1430: Aufsummieren der Beträge der Aktiva-Konten auf der Habenseite.  
1450 – 1460: Abschließen des Kontos.

- 1470 – 1490: Eingabe des Saldobetrages.
- 1500 – 1560: Konto auf dem Bildschirm zeigen.
- 1600 – 1750: Dito für Passiva-Konten.
- 1760 – 1850: Beenden des Programms mit Leistungsbewertung.
- 1890 – 2050: Geschäftsvorfälle in DATA-Zeilen.
- 2060 – 2080: Die einzelnen Konten in DATA-Zeilen gespeichert.

### Erklärungen zu den Feldern k, k\$, az

Will man das Programm ergänzen oder verbessern, ist es notwendig, den Aufbau des Feldes k bzw. k\$ zu verstehen. Folgende Angaben gelten auch für das Feld k\$.

Für k (Nr, Art, S/H, Buchung) gilt:

Nr = Nummer des Kontos, z.B.: 1 = Grundstücke, 2 = Geschäftsausstattung usw.

Ebenso bei Passiva-Konten: 1 = Kapital, 2 = Darlehens.

Art = Art des Kontos: 1 = Aktiva-Konto, 2 = Passiva-Konto.

S/H = Angabe von Soll- oder Habenseite: 1 = Soll, 2 = Haben.

Buchung: Hier steht der Betrag bzw. der Name des Gegenkontos der Buchung.

Bei az (Nr, Art, S/H) verfährt man ebenso.

az = Anzahl der im Konto stehenden Buchungen.

### Einige Tips zur Eingabe des Programms

Die im Programm auftretenden REM-Zeilen, d.h. Programmerläuterungen sind für den Programmablauf nicht nötig und können weggelassen werden.

Viele Zeilen kann man kopieren, indem man eine Zeile eintippt, mit dem Cursor hochfährt, die Zeilennummer und die entsprechenden Stellen ändert. Nicht vergessen: Die Zeilennummer ändern, sonst wird die alte Zeile überschrieben! Dieses Verfahren könnte man insbesondere bei den Zeilen 650 – 1150, 1410 – 1560 und 1600 – 1750 anwenden.

Ich hoffe, daß dir dieses Programm viel Freude bereitet.

(Christoph Bronold)

```
10 rem *****
20 rem * *
30 rem" * Grundlagen der * "
50 rem" * Buchfuehrung * "
60 rem * *
70 rem ***** "
80 :
90 rem" Copyright 1985 by "
100 rem moderne verlags gesellschaft
110 :
120 rem" Autor: Christoph Bronold "
130 :
150 poke53281,1:rem" Hintergrund weiss
160 poke53280,3:rem" Rahmen tuerkis
170 gosub1290 :rem" Kommentar ausgeben
190 dim az(8,2,2) :rem" Anzahl d. Ein-
traege
200 dimk$(8,2,2,20):rem" Kontennamen
210 dimk(8,2,2,20) :rem" Betraege
220 cl$=chr$(147) :rem" Bildschirm
loeschen
230 ch$=chr$(145) :rem" Cursor hoch
240 r$=chr$(28) :rem" rot
250 b$=chr$(31) :rem" blau
260 printchr$(14) :rem" Kleinbuchstaben
265 print cl$
270 fori=1to40:s3$=s3$+"*":s4$=s4$+chr$(
45):s5$=s5$+chr$(192):next
280 s1$=left$(s5$,21)+chr$(178)+left$(s5
$,17)
290 s2$=left$(s5$,21)+chr$(219)+left$(s5
$,17)
300 :
310 rem"*** Programmbeginn *****
320 :
330 gosub1310:print:printb$"Wie viele Bu
chungen moechtest Du bear-"
340 input"beiten (5-50) ";ab
350 ifab<5orab>50thenprintch$ch$:goto340
360 print:print:print"Es folgt die Eroef
fnungsbilanz."
370 read az:fori=1to az:reada$:next
:rem"*** Ueberlesen
```



```

380 reada1:ba=a1:fori=1toa1:readk$(i,1,0
,0):gosub1300
390 w1=int(w*10↑int(ba/2)):k(i,1,1,1)=w1
:ba=ba-1:s1=s1+w1:next
400 fori=1toa1:k$(i,1,1,1)="Anfangsbest.
":az(i,1,1)=1:next
410 reada2:readk$(1,2,0,0)
420 ba=a2-1:fork=2toa2:readk$(k,2,0,0):g
osub1300
430 w2=int(w*10↑ba):k(k,2,2,1)=w2:ba=ba-
1:s2=s2+w2:next:k(1,2,2,1)=s1-s2
440 fori=1toa2:k$(i,2,2,1)="Anfangsbest.
":az(i,2,2)=1:next
450 gosub1310:printchr$(30)"Aktiva      E
roeffnungsbilanz      Passiva"
460 prints1$:fori=1toa1
470 printk$(i,1,0,0);:h$=str$(k(i,1,1,1)
):printtab(21-len(h$))h$;chr$(221);
480 ifk$(i,2,0,0)=""thenk$(i,2,0,0)=""
"
490 printk$(i,2,0,0);:h$=str$(k(i,2,2,1)
):printtab(39-len(h$))h$
500 next:prints2$:h$=str$(s1):printtab(2
1-len(h$))h$;chr$(221);tab(39-len(h$))h$
510 print"====="          "=====" "====="
      "====="
520 print:printb$;"  Weiter = beliebige
Taste druecken!    "
530 poke 198,0:wait 198,1:get wt$
540 gosub1290:rem"*** Tastendruck bestae-
tigen
550 :
560 rem"*** Buchungsschleife *****
570 :
580 l=1:an=0:ar=0
590 print:print:print:printchr$(144)c1$;
s3$;l". Buchung:"
600 restore:readaz:fori=1toint(rnd(1)*17
)+1:readbu$:next:i=i-1
610 bw=int(rnd(0)*1000)+100
620 prints4$bu$" "bw"DM":prints3$;b$:ifi
>10then950
625 :
630 rem"*** Buchung ausfuehren *****
635 :

```

640 onigoto650,680,710,740,770,800,830,860,890,920  
650 sb=az(3,1,1)+1:az(3,1,1)=sb:k\$(3,1,1, sb)=k\$(5,1,0,0):k(3,1,1,sb)=bw  
660 sb=az(5,1,2)+1:az(5,1,2)=sb:k\$(5,1,2, sb)=k\$(3,1,0,0):k(5,1,2,sb)=bw  
670 sb\$="w":hb\$="k":goto1200  
680 sb=az(3,1,1)+1:az(3,1,1)=sb:k\$(3,1,1, sb)=k\$(6,1,0,0):k(3,1,1,sb)=bw  
690 sb=az(6,1,2)+1:az(6,1,2)=sb:k\$(6,1,2, sb)=k\$(3,1,0,0):k(6,1,2,sb)=bw  
700 sb\$="w":hb\$="b":goto1200  
710 sb=az(3,1,1)+1:az(3,1,1)=sb:k\$(3,1,1, sb)=k\$(3,2,0,0):k(3,1,1,sb)=bw  
720 sb=az(3,2,2)+1:az(3,2,2)=sb:k\$(3,2,2, sb)=k\$(3,1,0,0):k(3,2,2,sb)=bw  
730 sb\$="w":hb\$="v":goto1200  
740 sb=az(4,1,1)+1:az(4,1,1)=sb:k\$(4,1,1, sb)=k\$(3,1,0,0):k(4,1,1,sb)=bw  
750 sb=az(3,1,2)+1:az(3,1,2)=sb:k\$(3,1,2, sb)=k\$(4,1,0,0):k(3,1,2,sb)=bw  
760 sb\$="f":hb\$="w":goto1200  
770 sb=az(6,1,1)+1:az(6,1,1)=sb:k\$(6,1,1, sb)=k\$(3,1,0,0):k(6,1,1,sb)=bw  
780 sb=az(3,1,2)+1:az(3,1,2)=sb:k\$(3,1,2, sb)=k\$(6,1,0,0):k(3,1,2,sb)=bw  
790 sb\$="b":hb\$="w":goto1200  
800 sb=az(5,1,1)+1:az(5,1,1)=sb:k\$(5,1,1, sb)=k\$(3,1,0,0):k(5,1,1,sb)=bw  
810 sb=az(3,1,2)+1:az(3,1,2)=sb:k\$(3,1,2, sb)=k\$(5,1,0,0):k(3,1,2,sb)=bw  
820 sb\$="k":hb\$="w":goto1200  
830 sb=az(2,1,1)+1:az(2,1,1)=sb:k\$(2,1,1, sb)=k\$(6,1,0,0):k(2,1,1,sb)=bw  
840 sb=az(6,1,2)+1:az(6,1,2)=sb:k\$(6,1,2, sb)=k\$(2,1,0,0):k(6,1,2,sb)=bw  
850 sb\$="a":hb\$="b":goto1200  
860 sb=az(2,1,1)+1:az(2,1,1)=sb:k\$(2,1,1, sb)=k\$(5,1,0,0):k(2,1,1,sb)=bw  
870 sb=az(5,1,2)+1:az(5,1,2)=sb:k\$(5,1,2, sb)=k\$(2,1,0,0):k(5,1,2,sb)=bw  
880 sb\$="a":hb\$="k":goto1200  
890 sb=az(3,2,1)+1:az(3,2,1)=sb:k\$(3,2,1, sb)=k\$(6,1,0,0):k(3,2,1,sb)=bw

```

900 sb=az(6,1,2)+1:az(6,1,2)=sb:k$(6,1,2
, sb)=k$(3,2,0,0):k(6,1,2,sb)=bw
910 sb$="v":hb$="b":goto1200
920 sb=az(3,2,1)+1:az(3,2,1)=sb:k$(3,2,1
, sb)=k$(5,1,0,0):k(3,2,1,sb)=bw
930 sb=az(5,1,2)+1:az(5,1,2)=sb:k$(5,1,2
, sb)=k$(3,2,0,0):k(5,1,2,sb)=bw
940 sb$="v":hb$="k":goto1200
950 oni-10goto960,990,1020,1050,1080,111
0,1140
960 sb=az(5,1,1)+1:az(5,1,1)=sb:k$(5,1,1
, sb)=k$(4,1,0,0):k(5,1,1,sb)=bw
970 sb=az(4,1,2)+1:az(4,1,2)=sb:k$(4,1,2
, sb)=k$(5,1,0,0):k(4,1,2,sb)=bw
980 sb$="k":hb$="f":goto1200
990 sb=az(6,1,1)+1:az(6,1,1)=sb:k$(6,1,1
, sb)=k$(4,1,0,0):k(6,1,1,sb)=bw
1000 sb=az(4,1,2)+1:az(4,1,2)=sb:k$(4,1,
2,sb)=k$(6,1,0,0):k(4,1,2,sb)=bw
1010 sb$="b":hb$="f":goto1200
1020 sb=az(1,1,1)+1:az(1,1,1)=sb:k$(1,1,
1,sb)=k$(6,1,0,0):k(1,1,1,sb)=bw
1030 sb=az(6,1,2)+1:az(6,1,2)=sb:k$(6,1,
2,sb)=k$(1,1,0,0):k(6,1,2,sb)=bw
1040 sb$="g":hb$="b":goto1200
1050 sb=az(5,1,1)+1:az(5,1,1)=sb:k$(5,1,
1,sb)=k$(6,1,0,0):k(5,1,1,sb)=bw
1060 sb=az(6,1,2)+1:az(6,1,2)=sb:k$(6,1,
2,sb)=k$(5,1,0,0):k(6,1,2,sb)=bw
1070 sb$="k":hb$="b":goto1200
1080 sb=az(6,1,1)+1:az(6,1,1)=sb:k$(6,1,
1,sb)=k$(5,1,0,0):k(6,1,1,sb)=bw
1090 sb=az(5,1,2)+1:az(5,1,2)=sb:k$(5,1,
2,sb)=k$(6,1,0,0):k(5,1,2,sb)=bw
1100 sb$="b":hb$="k":goto1200
1110 sb=az(3,2,1)+1:az(3,2,1)=sb:k$(3,2,
1,sb)=k$(2,2,0,0):k(3,2,1,sb)=bw
1120 sb=az(2,2,2)+1:az(2,2,2)=sb:k$(2,2,
2,sb)=k$(3,2,0,0):k(2,2,2,sb)=bw
1130 sb$="v":hb$="d":goto1200
1140 sb=az(5,1,1)+1:az(5,1,1)=sb:k$(5,1,
1,sb)=k$(2,2,0,0):k(5,1,1,sb)=bw
1150 sb=az(2,2,2)+1:az(2,2,2)=sb:k$(2,2,
2,sb)=k$(5,1,0,0):k(2,2,2,sb)=bw
1160 sb$="k":hb$="d":goto1200
1170 :

```

```
1180 rem"*** Eingabe des Anfangsbuchsta-
      bens der Buchung *****
1190 :
1200 print:printr$;"Buchung im Soll:":an
      =an+1:gosub1330
1210 inputse$:ifse$<>sb$then printr$"Wie
      bitte ?"b$;ch$;ch$:af=af+1:goto1210
1220 print:printb$;"Buchung im Haben:":a
      n=an+1:gosub 1330
1230 se$="":input se$
1235 if se$<>hb$ then print"Wie bitte? "
      b$;ch$;ch$:af=af+1:goto 1230
1240 l=l+1:ifl<=ab then 590
1250 goto 1370
1260 :
1270 rem"*** Hilfsunterprogramme *****
1280 :
1290 print:print"Eingabe wird bearbeitet
      . Bitte warten! ":return
1300 :
1310 printchr$(149)c1$s3$" *      Grundlag
      en der Buchfuehrung      * "s3$b$
1320 print:return
1330 print"(g)rundstuecke (a)usstatt. (w
      )aren"
1340 print"(f)orderungen (k)asse (b
      )ank"
1350 print"(e)igenkapital (d)arlehen (v
      )erbindl.":return
1360 :
1370 rem"*** Saldeneingabe *****
1380 :
1390 gosub1310
1400 print"Gib bitte jetzt die jeweilige
      n Saldo- betraege ein.":print
1410 fori=1toal:prints4$,k$(i,1,0,0):pri
      nts4$:s1=0:s2=0:an=an+1
1420 forj=1toaz(i,1,1):s1=s1+k(i,1,1,j):
      next
1430 fork=1toaz(i,1,2):s2=s2+k(i,1,2,k):
      next
1440 if s2>s1then1460
1450 s=s1-s2:k$(i,1,2,k)="Saldo":k(i,1,2
      ,k)=s:az(i,1,2)=k:goto1470
1460 s=s2-s1:k$(i,1,1,j)="Saldo":k(i,1,1
```

```

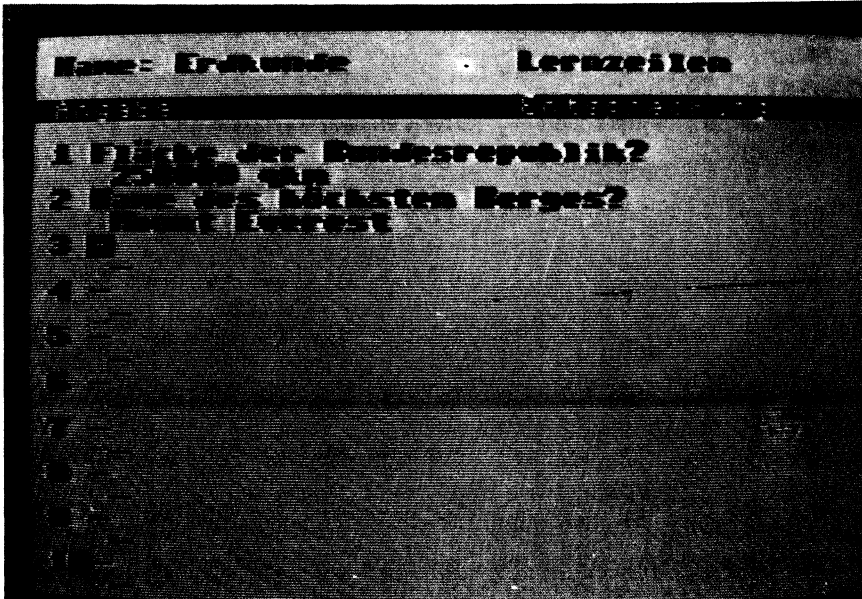
,j)=s:az(i,1,1)=j
1470 input"Saldobetrag: (↑=Ergebnis) ";s
b$
1480 ifsb$="↑"thenprint"Ergebnis:      "s
"DM":af=af+1:goto1500
1490 sb=val(sb$):ifsb<>s thenprintr$"Lei
der falsch! "b$;ch$;ch$:af=af+1:goto1470
1500 print:printchr$(30)"S
      J                      H "s2$
1510 kh=az(i,1,1):ifaz(i,1,2)>khthenkh=a
z(i,1,2):
1520 form=1tokh:h$=str$(k(i,1,1,m))
1530 printk$(i,1,1,m)tab(21-len(h$))h$" |
";
1540 h$=str$(k(i,1,2,m)):printk$(i,1,2,m
)tab(39-len(h$))h$:next
1550 prints2$:prints3$b$:print"Taste dru
ecken !"
1560 wait198,203:gett$:printc1$:next
1570 :
1580 rem"*** Passiva-Konten *****
1590 :
1600 fori=1toa2:printc1$;s4$,k$(i,2,0,0)
:prints4$:s1=0:s2=0:an=an+1
1610 forj=1toaz(i,2,1):s1=s1+k(i,2,1,j):
next
1620 fork=1toaz(i,2,2):s2=s2+k(i,2,2,k):
next
1630 ifs2>s1then1650
1640 s=s1-s2:k$(i,2,2,k)="Saldo":k(i,2,2
,k)=s:az(i,2,2)=k:goto1660
1650 s=s2-s1:k$(i,2,1,j)="Saldo":k(i,2,1
,j)=s:az(i,2,1)=j
1660 input"Saldobetrag: (↑=Ergebnis) ";s
b$
1670 ifsb$="↑"thenprint"Ergebnis:      "s
"DM":af=af+1:goto1690
1680 sb=val(sb$):ifsb<>sthenprintr$"Leid
er falsch !"b$ch$ch$:af=af+1:goto1660
1690 print:printchr$(30)"S
      J                      H "s2$
1700 kh=az(i,2,1):ifaz(i,2,2)>khthenkh=a
z(i,2,2):
1710 form=1tokh:h$=str$(k(i,2,1,m))
1720 printk$(i,2,1,m)tab(21-len(h$))h$" |
";

```

```
1730 h$=str$(k(i,2,2,m)):printk$(i,2,2,m
)tab(39-len(h$))h$:next
1740 prints2$:prints3$:printb$"Taste dru
ecken !"
1750 wait198,203:gett$:printc1$:next
1760 gosub1310:print"Nun bist Du am Ende
der Trainingsein- heit angelangt."
1770 print:printr$"Deine Leistung war ";
:qu=af/an
1780 ifqu<.4 then print"nichts besondere
s. Mehr ueben!":goto1820
1790 ifqu<.6 then print"mittelmaessig!":
goto1820
1800 ifqu<.85 then print"recht gut!":got
o1820
1810 print"ausgezeichnet! Sehr gut!"
"
1820 print:print:printb$"Moechtest Du no
ch weiter trainieren? (j/n)"
1830 wait198,203:gett$
1840 if t$="j"then run
1850 printc1$"Bis bald zur naechsten Tra
iningseinheit!":end
1860 :
1870 rem"*** Geschaefstsvorfaelle *****
1880 :
1890 data 17,"Wareneinkauf gegen bar"
1900 data "Wareneinkauf gegen Scheck"
1910 data "Wareneinkauf auf Ziel"
1920 data "Warenverkauf auf Ziel"
1930 data "Warenverkauf gegen Scheck"
1940 data "Warenverkauf gegen bar"
1950 data "Kauf einer Schreibmaschine ge
gen Scheck "
1960 data "Kauf eines Schrankes gegen ba
r"
1970 data "Bankueberweisung an Lieferant
en"
1980 data "Barzahlung an Lieferanten"
1990 data "Barzahlung eines Kunden"
2000 data "Bankueberweisung eines Kunden
"
2010 data "Kauf eines Grundstuecks gegen
Scheck "
2020 data "Barabhebung vom Bankkonto"
```

2030 data "Bareinzahlung auf Bankkonto"  
2040 data "Umwandlung einer Lieferantens  
chuld in Darlehen"  
2050 data "Aufnahme eines Darlehens; bar  
"  
2060 data 6, "Grundstuecke", "Gebaeudeauss  
t.", "Waren", "Forderungen", "Kasse"  
2070 data "Bank"  
2080 data 3, "Kapital", "Darlehen", "Verbin  
dlich."

## 7 Lernfiler



Die Fläche der BR Deutschland beträgt 248 610 km<sup>2</sup>. Diese Fläche wurde hier aus Lernzwecken gerundet. Das Bildschirmfoto zeigt nur eine von vielen Anwendungsmöglichkeiten dieses Programms!

### Programm-Telegramm

- Lerndatei mit integriertem französischem und italienischem Zeichensatz sowie deutschen Umlauten und ß.
- Geeignet für alle Lernstoffe nach dem Frage- und Antwortprinzip.
- Individuelle Dateneingabe möglich.
- Menügesteuerter Zugriff auf bis zu 15 verschiedene Bildschirme (in der großen Diskettenversion).

### Vom Gehirn auf Diskette – und zurück!

#### Begriffsklärung

**Lernfiler:** Programm, das mnemotechnisch (lerntechnisch) organisierte Dateien erstellt und verwaltet.

**File:** Auf Diskette Speicherbarer Inhalt des vom Programm verwalteten Speichers.



**Lern-Datei** oder **Bildschirm** wird der Inhalt der einzeln zu bearbeitenden Lerngegenstände genannt. Zwischen 1 und 15 davon sind nach entsprechender Auswahl unmittelbar verfügbar, ohne daß auf die Diskette zugegriffen werden müßte.

**Escape-Taste:** Taste links oben, Pfeil nach links. Mit ihrer Hilfe gelangt man aus fast jeder Programmstation ins Menü zurück und kann so die verschiedensten Aktionen abbrechen, ohne den Speicherinhalt (das *File*) zu verändern.

### Pädagogische Überlegungen

Nach allgemein gültigen pädagogischen Bewertungen ist es für die meisten Lernenden notwendig, sich Lernstoffe nicht nur lesend, sondern tätig – in der Regel schreibend – anzueignen. Ein Programm wie der *Lernfiler* ist daher ein Instrument, das die sinnvolle Aktivität und Mitarbeit des Lernenden fordert. Ein guter Teil der Lernerarbeit wird dabei durchaus schon bei der Eingabe des Lernstoffes in die Lerndatei geleistet. Der übende Umgang mit dem Programm festigt über die Zufälligkeit in der Reihenfolge der Abfrage den Kenntnisstand. Die Mitteilung der Zahl richtiger Antworten wirkt sich positiv verstärkend aus. Die Anzeige nicht beherrschter Angabenpaare (z.B. Vokabelgleichungen oder fremdsprachliche Form – Formbestimmung oder Rechtschreibproblem – Lösung) ermöglicht die Festigung des Lernstoffs im Gedächtnis.

Zur Auffrischung und Sicherung des Kenntnisstandes ist es jederzeit leicht möglich, das *Lernfile* wieder von der Diskette zu laden und damit zu arbeiten. So kann man sein Lehrbuch, seine Formen und Tabellen, *Idioms*, Formeln oder Vokabeln kontinuierlich mit dem Programm *Lernfiler* begleiten und den Stoff für spätere Wiederholungen bereithalten.

Eine motivierende Nutzung des Programms könnte es von Zeit zu Zeit auch sein, wenn eine andere Person eine Datei oder ein ganzes *File* erstellt – etwa mit Wörtern, die eine Rechtschreibschwierigkeit enthalten (z.B.: s – ss – ß; vgl. hierzu das *Beispielfile* auf der Diskette). Eine Gestaltung in Art von Kreuzworträtsselfragen bietet sich hier an. Daß dann nur von einer bestimmten Seite zur anderen gefragt werden sollte, versteht sich von selbst.

Andererseits ist es pädagogisch durchaus sinnvoll, daß keine Seite gegen Einsichtnahme geschützt wird, was programmtechnisch möglich wäre, da das Richtige gelernt werden soll und daher verfügbar gehalten werden muß.

Die auf der Diskette enthaltenen *Beispielfiles* zeigen ein kleines Spektrum der mit dem *Lernfiler* gegebenen Möglichkeiten.

### Kurze Programmcharakterisierung

Lernfiler ist das ideale Instrument für alle, die Fremdsprachen lernen oder muttersprachliche Übungen gestalten wollen.

Die Idee der Lerndatei, die sich vom gängigen Vokabelprogramm unterscheiden sollte, fand ihre Umsetzung in einem recht vielfältig einsetzbaren Masken-*Layout* am Bildschirm, das zwar auch das »bloße« Vokabellernen unterstützt, vor allem aber grammatische Strukturen der Deklination und Konjugation »dynamisiert«. Mit dem Lernfiler kommt so etwas wie Bewegung in das statische Kasus- bzw. Personalformensystem. Da sämtliche französischen Sonderzeichen sowie der deutsche Zeichensatz mit Umlauten und ß implementiert sind, eignet sich das Programm für Rechtschreibübungen ebenso wie für das Training in Französisch, Italienisch (zwei Sonderzeichen) bzw. Latein und Englisch.

## Programmablauf

In der vorliegenden Form, die den voll funktionsfähigen Hauptteil eines mit zusätzlichen Routinen ausgestatteten komfortableren »Lernfilers« darstellt, der auf der zu diesem Buch angebotenen Diskette enthalten ist, stellt das Programm die zur Lerndatei-Verwaltung und zum Üben notwendigen Programmteile zur Verfügung.

### 1 Lernstoff eingeben

Nach dem Programmstart wird zunächst die Eingabe aufgerufen. Ein Untermenü verzweigt – je nach Wunsch bzw. Notwendigkeit – in eine von vier Bildschirmmasken.

Die **erste** Maske trägt am oberen Bildschirmrand die Bezeichnung »8 Deklinationszeilen«. In der nächsten Zeile wartet hinter der Bezeichnung »Name:« ein Fragezeichen auf die Eingabe einer Kurzbenennung der in Arbeit befindlichen Lerndatei. Ein reverser Balken trennt diesen schmalen oberen Bereich von der eigentlichen »Datei«. Die diesbezügliche Bildschirmorganisation sieht je vier Doppelzeilen, die am linken Rand mit den Ziffern 1 bis 4 gekennzeichnet sind, in der oberen und unteren Hälfte des verbliebenen freien Bildschirms vor. Die Doppelzeilen sind als »linke« und »rechte« Zeile (etwa wie in einem Vokabelheft) zu verstehen; in die linke wird beispielsweise eingetragen »*le mur*«, in die rechte als Entsprechung »die Mauer« oder »Nominativ Singular mask.«. Die Zeilen erlauben 33 Anschläge, die rechte Zeile liegt um drei Anschläge nach rechts versetzt unter der linken. Durch diese Anordnung konnte eine wenig sinnvolle Beschränkung auf ca. 17 Anschläge in echten rechten und linken Zeilenhälften vermieden werden.

Die beiden deutlich gegeneinander abgesetzten Blöcke mit je vier Doppelzeilen stellen in der oberen Hälfte den Singularbereich und in der unteren den Pluralbereich bereit.

Diese **erste** Maske ist z.B. für Französisch zu wählen, wenn es gilt, Deklinationsformen einzugeben, um sie anschließend zu üben. In entsprechender Weise sind die anderen Masken gestaltet. Die Unterschiede ergeben sich aus der Anzahl der Doppelzeilen bzw. aus der Numerierung dieser Zeilen.

Die **zweite** Maske ist für Latein gedacht, weil hier mit dem Ablativ ein unerläßlicher Kasus zusätzlichen Platz beansprucht. Auf den Vokativ, den sechsten lateinischen Kasus, kann man verzichten, da einzig die o-Deklination im Maskulinum eine eigene Form anbietet (Marcus – Marce; Publius – Publi); diese Ausnahme von der Norm sollte sich auch ohne Programmunterstützung merken lassen. In allen anderen Deklinationen entspricht der Vokativ dem Nominativ.

Während die zweite Maske zehn Doppelzeilen in zwei Fünferblöcken enthält, begnügt sich die **dritte** Maske mit zwei Dreier-Blöcken, da hier Konjugationsformen eingegeben werden sollen. Die Voraussetzungen für die Eingabe der Formen der 1. bis 3. Person in Singular und Plural sind damit gegeben.

Die **vierte** Maske enthält wie die zweite Maske zehn Doppelzeilen, die allerdings als ein Block von 1 bis 10 durchgezählt werden. (Um zusätzlichen Programmieraufwand zu vermeiden, wurde die trennende Freizeile zwischen dem oberen und unteren Block übrigens beibehalten. Eine Auswirkung auf den Programmablauf ergibt sich dadurch nicht.) Hier lassen sich Lerndaten beliebiger Art erfassen. Z.B. »Tod Goethes« – »1832« oder »höchster Berg Afrikas« – »Kilimandscharo«. Auch Vokabeln können bearbeitet werden.

Falls Zeilen in den Masken nicht beschriftet werden, dann setzt das Programm an die Leerstelle einen Punkt (.). Beim Üben wäre entsprechend in freien Zeilen als Antwort der Punkt (.) zu geben.

## 2 Lernstoff von Diskette laden

Wenn mit dem Programm erstellte *Files* auf Diskette vorliegen, können sie mit der LOAD-Funktion aufgerufen und zu Übungen herangezogen werden. Auch hier erlaubt das Inhaltsverzeichnis einen orientierenden Überblick. Nach dem Laden »weiß« der Lernfiler, welche Bildschirmmaske für das *File* gültig ist, da die notwendigen Parameter beim vorherigen Speichern mit in die sequentielle Datei übergeben wurden.

## 3 Überprüfen und Ändern

Sobald der Inhalt der letzten neu eingegebenen Zeile mit **RETURN** übergeben wurde, kehrt das Programm ins Menü zurück. Bevor nun der Speicherinhalt – falls er später noch benötigt wird – abgespeichert wird, kann er über die Editierfunktion kontrolliert und gegebenenfalls berichtigt werden. Auch diesen Programmteil kann man wie jeden anderen mit der Taste »Pfeil nach links« (linke obere Taste) verlassen.

Beim Editieren wird der Speicherinhalt mit der **RETURN**-Taste »durchgeblättert«. Nach Druck auf Taste »a« erscheint unter dem änderungsbedürftigen Eintrag ein Fragezeichen, das auf die Notwendigkeit zur Neueingabe hinweist. **Return** schließt die Eingabe ab, die nächste Zeile wird angezeigt. Dieser Programmteil wird nach der Anzeige der letzten Zeile automatisch mit einem Sprung zum Menü verlassen.

#### 4 Abspeichern

Der Aufruf der SAVE-Funktion eröffnet zunächst die Möglichkeit, das Disketten-Inhaltsverzeichnis anzuschauen. So kann man entweder bewußt einen identischen Namen für ein *File* wählen, um einen Alteintrag zu überschreiben oder auch nur einen geringfügig veränderten Namen verwenden, um den Alteintrag zu erhalten. Natürlich sind außerdem beliebige Neueinträge möglich.

Seinen Namen hat das Programm übrigens daher, daß mit ihm Lern-»Files« auf Diskette abgelegt werden können.

#### 5 Üben des Lernstoffes

Den eigentlichen Lerneffekt soll die Funktion »Üben« bieten. Dieser Programmteil läßt drei Arten der Abfrage zu: erstens »von links nach rechts«, zweitens »von rechts nach links« und drittens zufällig gemischt, wobei auch »rechte« Einträge »links« ausgegeben werden. Vom Zufallsgenerator gesteuert, erscheint in einer Zeile das Fragezeichen und erwartet die Eingabe, deren Inhalt der Übende aus der zugehörigen »Fragezeile« erschließen kann, wenn er den Lernstoff schon beherrscht. Angenommen, dem Programm läge intern vor: »den Mann – Akkusativ Singular mask.«. Dann stünde bei der Abfrage von »links nach rechts« am Schirm:

4 – den Mann  
– ?

In der vom Fragezeichen besetzten Zeile wäre nun einzugeben: Akkusativ Singular mask. Die **RETURN**-Taste wird zur Übergabe der Antwort nicht benötigt, da ein Längenvergleich zwischen erwarteter und tatsächlicher Eingabe erfolgt. Bei Erreichen der »richtigen« Länge wird aus der quantitativen Bewertung eine qualitative, indem das Programm beide *Strings* vergleicht. Falls eine Übereinstimmung vorliegt, erscheint das Fragezeichen an anderer Stelle, bis schließlich nach und nach alle freien Zeilen vom Üben aufgefñllt sind. Sollte eine Nichtübereinstimmung, also ein Fehler, festgestellt werden, so ersetzt das Programm die Fehleingabe durch den korrekten Zeileninhalt, der allerdings in weißer Farbe erscheint, um den Eingriff des Programms zu signalisieren. Danach verläuft die Abfrage weiter in der eben geschilderten Weise.

Die Bedienung der **RETURN**-Taste wertet das Programm im Übungsablauf als Verzicht auf eine Antwort und reagiert wie bei einer Fehleingabe.

#### Bewertung der Leistung

Sobald alle Abfragen bearbeitet sind, bleibt der Schirm einige Sekunden unverändert, um einen Überblick über richtige und korrigierte Antworten zu gewähren. Die folgende Anzeige gibt bekannt, wie viele Fragen korrekt beantwortet wurden. Darüberhinaus werden alle Doppelzeilen-Inhalte am Schirm dargestellt, in denen eine Korrektur

nötig war. Da erst ein Tastendruck von hier ins Menü zurückführt, erhält der Übende Gelegenheit, seine Wissenslücken in aller Ruhe zu bedenken und auszugleichen.

## Kommentierung des Programm-Listings

### Hinweise zur Programmeingabe

1. Da das *Listing* doch recht lang ist, sollte folgende Möglichkeit genutzt werden, um sich das Eintippen wenigstens an zwei Stellen zu erleichtern:

Zeilen 2700 – 3400: Wie sicher allgemein bekannt, erlaubt der komfortable *Basic-Editor* der Commodore Computer das Kopieren von Zeilen, indem man mit dem *Cursor* auf die alte Zeilennummer fährt, eine neue Zahl statt der alten eintippt und **RETURN** drückt. Beim Listen des so behandelten Zeilenbereichs erkennt man, daß die alte Zeile noch existiert. Zusätzlich gibt es die neue Zeile mit identischem Inhalt.

Diese Möglichkeit kann folgendermaßen genutzt werden: Man tippt die Zeile 2700 ein. Nach Abschluß mit **RETURN** geht man mit dem *Cursor* auf die Zeilennummer und ändert sie in 2710. Nun fährt man mit dem *Cursor* auf die »1« bei `.AZ = 1 . .`, ersetzt sie durch die »3« und drückt **RETURN**. Damit ist Zeile 2710 eingegeben. Verfährt man entsprechend – eventuell unter Einsatz von **INST/DEL** – in den folgenden Zeilen, so kann man viel Tipparbeit sparen. Dieselbe Möglichkeit zur Eingabebesleunigung hat man in den Zeilen 4090 bis 4290. In Zeile 4290 darf dann allerdings das »:return« beim Eintippen nicht vergessen werden.

2. Einige wenige Zeilen (z.B. 3810) füllen beim Abtippen zwei Bildschirmzeilen völlig aus. Der *Cursor* wandert hier schon in die dritte Zeile, was der *Basic-Editor* nicht akzeptiert, so daß mit **RETURN** die eben eingegebene Zeile nicht übernommen würde. Notwendige Abhilfe: man schiebt den *Cursor* mit den *Cursortasten* eine Zeile zurück, so daß er wieder auf der zuletzt geschriebenen Doppelzeile blinkt. Nun wird mit **RETURN** die Zeile korrekt übernommen. Eine andere Möglichkeit zur Vermeidung des Problems: Man gibt das *Basic*-Wort `DATA` abgekürzt (»dA«) ein. Schließlich kann man auch auf den Leerraum hinter der Zeilennummer verzichten.

3. **Wichtig!** Sichern Sie das Programm vor dem ersten Start auf Diskette! Die Manipulation am Zeichensatz könnte zum irreparablen Absturz des Programms führen. Zum Austesten des Programms sollte Zeile 300 so geändert werden, daß der Zeichensatz vorerst nicht verändert wird: `. . . : REM: GOSUB 3920`. Nach der Sicherung der zum Laufen gebrachten Programmversion kann dann das `REM` entfernt werden. Jetzt müßte – fehlerfreie Eingabe vorausgesetzt – nach dem nun länger dauernden Programmstart der Sonderzeichensatz zur Verfügung stehen (genauere Angaben später).

Das Programm sollte von jetzt an nur mit **STOP** ohne **RESTORE** gestoppt werden. Das gewohnte **STOP/RESTORE** versetzt den Bildschirm in einen unkontrollierbaren Zustand. In der Regel gibt es Abhilfe, indem noch einmal **STOP/RESTORE** ge-

drückt wird. Nun wird »blind« RUN eingetippt und **RETURN** gedrückt. Das Programm sollte daraufhin ganz normal auf geklärtem Bildschirm starten.

Da das aber nicht zu garantieren ist, muß die Aufforderung zur Programmsicherung vor dem ersten Start unbedingt befolgt werden! Um unbeabsichtigten Programmabbruch ganz auszuschließen, kann man Zeile 260 ergänzen: . . . : POKE 808,234. Damit sind **STOP/RESTORE** und – was hier bedeutungslos bleibt – LIST ausgeschaltet.

## Die Programmstruktur

Zu Beginn werden *Strings* definiert, die als Ersatz der schwer identifizierbaren Commodoresteuerzeichen dienen. Nur in den INPUT-Anweisungen Zeile 1850 und 2040 konnten RVS ON und RVS OFF nicht ersetzt werden.

Die Farbgestaltung (Zeile 260) kann beliebig verändert werden. Bedenken sollte man jedoch, daß die Computer-Korrekturen in ihrer weißen Schrift erkennbar bleiben sollten, ein Kontrast zur übrigen Schrift also gewährleistet sein sollte.

Aus dem Wahl-Menü führt der Verteiler in Zeile 510 zu den gewünschten Programmstationen.

Den größten programmtechnischen Aufwand erfordert die »Eingabe«, weil hier (Zeilen 530 – 1490) vier Bildschirmmasken zu verwalten sind. Zeilen 670 – 740 stellen die erforderlichen Variableninhalte bereit. Die zugehörige Maske wird von den Zeilen 800 – 1010 erstellt. Die Einträge in die drei Dateibereiche »Name, linke Seite, rechte Seite« werden von den Zeilen 1020 bis 1310 organisiert.

Eine INPUT-Routine (Zeilen 1320 – 1490) überwacht in allen drei Bereichen die Tastatureingabe, blendet z.B. die *Cursorsteuerung* aus, positioniert die Zeichen am Bildschirm (etwa Zeile 1470) und verhindert Einträge von mehr als 33 Anschlägen. Die letzten erlaubten fünf Anschläge lösen je einen Warnton aus.

Eine etwas anders arbeitende INPUT-Routine (Zeilen 4530 – 4670) wird von der Änderungsfunktion (Zeilen 1510 – 1750) angesprochen.

Die SAVE-Routine schreibt eine sequentielle Datei auf die Diskette. Als Besonderheit sei erwähnt, daß alle Eingaben, die der Benutzer gestaltet, von G\$ – das ist CHR\$(34), also das Anführungszeichen – eingerahmt werden. So ist es möglich, auch solche Zeichen in den Dateieinträgen zu verwenden, die die Betriebssysteme von Rechner bzw. *Floppy* sonst als Trennzeichen werten – z.B. das Komma. SAVE- und LOAD-Routine gemeinsam bieten den Aufruf des Inhaltsverzeichnisses an. Ebenso gemeinsam fangen beide Programmteile Diskettenfehler mit entsprechender Meldung (Zeilen 4690 – 4750) ab.

Das eigentliche Üben organisieren die Zeilen 2160 – 3570. Auf die Möglichkeit, »linke« und »rechte« Einträge wechselnd abfragen zu lassen, soll kurz eingegangen werden.

Falls die Wechsel-Abfrage gewählt wird, tritt ein Vertauscher-*Flag* (VT(E)) in Aktion. In Zeile 2450 wird, falls AZ = 3, eine Zufallszahl erzeugt (1 oder 2 ist möglich). Wird eine 2 generiert, dann werden für die in Arbeit befindliche Doppelzeile (um die wievielte Zeile es geht, steht in der Variablen E) die Inhalte der linken und rechten Zeilen-

»hälfte« vertauscht. Die Flagge (VT(E)) für die Doppelzeile wird auf 1 gesetzt. Dies ist notwendig, damit nach Beendigung des Übungsdurchgangs oder nach einem Abbruch mit der *Escape*-Taste (Pfeil nach links) die Vertauschung rückgängig gemacht werden kann. Dies geschieht entweder automatisch nach richtigen Einzelantworten in Zeile 3530 durch Rücktausch mit Löschung der Flagge (VT(E) wird 0) oder für alle betroffenen *Strings* nach Abbruch in den Zeilen 3700 – 3740. Erwähnenswert sind auch noch die Zeilen 2640 – 2690, in denen überprüft wird, ob die für die Abfrage in N(W) erzeugte Zufallszahl schon aufgerufen wurde. Dies geschieht in Zeile 2680. Bei Wiederholung einer Zahl wird der Zufallsgenerator in Zeile 2650 nochmals aufgerufen. In den Zeilen 2700 – 3380 werden die je nach gewünschter Abfrageweise vorausgewählten Zeilen angesprungen, wo die Variablenwerte für die jeweils in der Abfrage befindliche Zeile übergeben werden. Anschließend wird zur Abfrage-Routine in den Zeilen 3390 – 3570 verzweigt. In AC\$ wird der Eingabestring bis zur Höchstlänge J (Zeile 3410: J = LEN (A\$ (E))) zusammengesetzt und in den Zeilen 3540 – 3560 bewertet.

Wenn die in Zeile 2640 definierte Schleife (ED ist hierbei die Höchstzahl der von der Maske bereitgehaltenen Zeilen) abgearbeitet ist, folgt die Fehlerbewertung in den Zeilen 3590 – 3680.

Die Routine, die das Inhaltsverzeichnis ermöglicht, wird in den Zeilen 3750 – 3870 bereitgestellt.

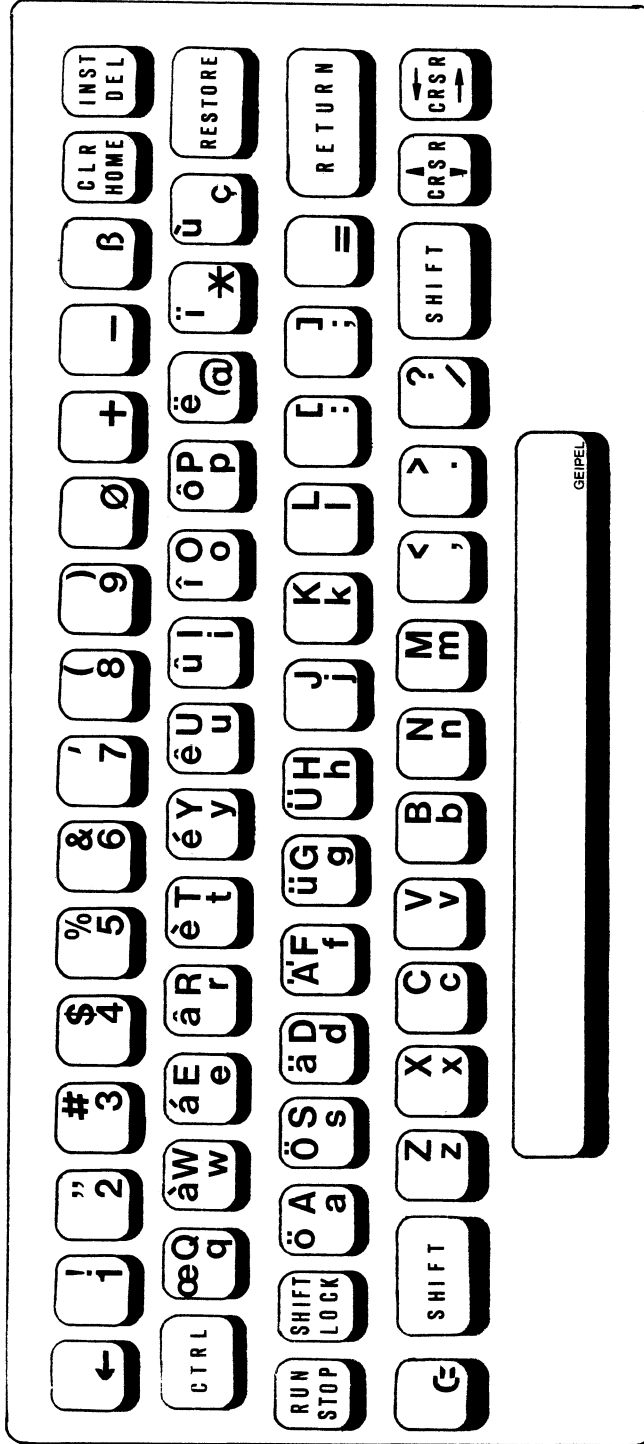
In den Zeilen 3920 – 4510 wird die Änderung des Zeichensatzes vorgenommen.

Welche Zeichen verändert werden, läßt sich von Zeile 4090 – 4290 leicht an den Zahlen ablesen, die mit 8 multipliziert werden. Diese Zahlen stellen die CHR\$-Codes bestimmter Zeichen dar – 115 etwa steht für das im ROM-Zeichensatz enthaltene Herzsymbol. Die Werte, von denen die *Pixel* der neuen Zeichen bestimmt werden, werden jeweils aus den DATA-Zeilen 4310 – 4510 eingelesen.

Die Sonderzeichen können nach dem Programmstart durch bestimmte Tastenkombination abgerufen werden. Ausnahmen stellen die Pfundtaste, die ein ß produziert, und der Potenzpfeil, der das französische Cedecille (ç) auf den Schirm bringt, dar, weil sie ihre neuen Zeichen direkt ausgeben. Die übrigen Sonderzeichen erfordern gleichzeitiges Gedrückthalten der Commodore-Taste. In dieser Kombination erreicht man mit den Tasten QWERTYUIOP@\* den französischen Zeichensatz. Entsprechend ergeben die Tasten ASDFGH die deutschen Umlaute (öÖäÄüÜ), Commodore-Taste plus »g« erzeugt z.B. das kleine ü.

**Tastenbelegung mit Sonderzeichen im Programm »Lernfiler«**

Das Programm arbeitet im Kleinschrift-Modus. Groß- und Kleinbuchstaben sind wie gewohnt zu erreichen. Die Sonderzeichen bringt man mit Hilfe der **C =** -Taste auf den Bildschirm. Sie sind in dieser Übersicht auf den Tasten links oben abgebildet.





## Information zu Programmversionen auf Diskette

In der hier vorliegenden Form ist das Programm auf der parallel zu diesem Buch angebotenen Diskette enthalten. Darüberhinaus bietet diese Diskette einen voll ausgestalteten Lernfiler mit einer Reihe weiterer Funktionen. Vor allem hält das große Programm 15 Bildschirmseiten gleichzeitig im Speicher, die lerntechnisch völlig unabhängig voneinander verwaltet werden. Deshalb kann man den Programmablauf so steuern, daß bei wiederholtem Aufruf eines schon bearbeiteten Übungsschirms nur die fehlerhaft angegebenen Formen bzw. Zeileninhalte zur Abfrage herangezogen werden.

Das Ergebnis der Lernbemühungen wird in einem Prozentwert für alle Fragen und Antworten seit dem jeweiligen Programmstart ausgegeben und gleichzeitig in konkreten Zahlen (»von 10 Antworten waren 9 richtig«) für den aktuellen Einzelschirm.

Auch eine Dokumentation des Speicherinhalts mit einem Drucker ist anwählbar. Die Sonderzeichen werden dabei intern für den Druck aufbereitet. Aus ß wird sz, aus ä wird ae usw.

Dazu kommen die Möglichkeiten, einzelne Schirme innerhalb der 15 zu löschen und dafür andere einzutippen oder von Diskette nachzuladen, die Fehlerwertung auf Null zu setzen, die Farbgestaltung des Bildschirms während des Programmablaufs zu ändern. Als Belohnungen für Lernerfolge sind eine kleine Tonfolge und eine abstrakte Farbgrafik vorgesehen.

(Claus Richter)

```

10 rem *****
20 rem *
30 rem" * Lernfiler *
40 rem *
50 rem *****
60 :
70 rem" Copyright 1985 by
80 rem moderne verlags gesellschaft
90 rem :
100 rem" Autor: Claus Richter
110 :
150 : rem ** vorbereitung **
155 :
160 rem -- bildschirmsteuerung -----
165 :
170 cd$=chr$(17) :rem cursor down
180 ce$=chr$(29) :rem cursor rechts
190 cl$=chr$(157):rem cursor links
200 cc$=chr$(147):rem clear home
210 ro$=chr$(18) :rem reverse on

```

```
220 rf$=chr$(146):rem reverse off
230 cu$=chr$(145):rem cursor up
240 ch$=chr$(19) :rem cursor home
250 :
260 poke53280,0:poke53281,0:poke646,8
270 dimee$(36),e$(11),zu$(36),uu$(36),ae
$(36),a$(36),u$(36),ax$(36):x$=chr$(13)
280 dimac$(36),vt(11):ch=129:g$=chr$(34)
:z1=28:z2=33
290 le$="
    ":rem 33 spaces
300 poke53272,23:gosub3750:gosub3920
310 :
320 : rem ** menuewahlen **
330 :
340 printcc$:print
350 printtab(12)ro$" Lernfiler "rf$
360 printtab(12)ro$"Claus Richter"rf$
370 print:print:fori=1to9:printce$;:next
380 print"Formen eingeben "ro$" 1 "rf$
390 print:fori=1to9:printce$;:next
400 print"Formen einladen "ro$" 2 "rf$
410 print:fori=1to9:printce$;:next
420 print"Formen edieren "ro$" 3 "rf$
430 print:fori=1to9:printce$;:next
440 print"Formen saven "ro$" 4 "rf$
450 print:print:print
460 print:fori=1to9:printce$;:next
470 print"Formen ueben "ro$" 5 "rf$
480 print:print:print
490 printtab(5)ro$" Lernfiler —— Lern
filer "rf$
500 poke198,0:wait198,1:geta$:ifval(a$)<
1orval(a$)>5then500
510 onval(a$)goto520,1950,1500,1760,2150
:rem verteiler
520 printcc$:print
530 printce$;ce$;ro$"Eingeben
    "rf$
540 print:print:print
550 printce$;ce$"Deklinationsschema mit
"ro$" 8"rf$" Formen "ro$" 1 "rf$
560 print:print:print
570 printce$;ce$"Deklinationsschema mit
"ro$"10"rf$" Formen "ro$" 2 "rf$
```

```

580 print:print:print
590 printce$;ce$"Konjugationsschema mit
"ro$" 6"rf$" Formen "ro$" 3 "rf$"
600 print:print:print
610 printce$;ce$"Abfrageschema mit
"ro$"10"rf$" Paarzl. "ro$" 4 "rf$"
620 print:print:print:print
630 printce$;ce$;ro$"Menue
      ←"rf$"
640 poke198,0:wait198,1:geta$:ifval(a$)<
1orval(a$)>4thenifasc(a$)<>95then640
650 ifa$=chr$(95)then340
660 fori=1to10:e$(i)="" :u$(i)="" :next:zo
$=""
670 ifa$="1"thened$=" Deklinations":ed=8
:e1=8:e2=9:j1=4:i1=11:i2=3:i3=21:i4=20
680 ifa$="1"theni5=12:i6=3
690 ifa$="2"thened$="Deklinations":ed=10
:e1=10:e2=11:j1=5:i1=13:i2=1:i3=23
700 ifa$="2"theni4=22:i5=14:i6=1
710 ifa$="3"thened$=" Konjugations":ed=6
:e1=6:e2=7:i1=0:i1=9:i2=5:i3=19
720 ifa$="3"theni4=18:i5=10:i6=5
730 ifa$="4"thened$=" Lern":ed=10
:e1=10:e2=11:j1=5:i1=13:i2=1:i3=23
740 ifa$="4"theni4=22:i5=14:i6=1
750 gosub770:goto1030
760 :
770 rem ** eingabe formen *
780 :
790 poke198,0
800 printcc$" "ed;ed$"zeilen
"
810 print"Name:"
820 printro$"Angabe
Entsprechung"rf$;
830 j=0
840 fori=3toe2step2
850 j=j+1
860 poke211,0:poke214,i:sys58732
870 printcl$;j
880 nexti
890 j=0
900 ifed$=" Lern"thenj=5
910 fori=14toi4step2

```

```
920 j=j+1
930 poke211,0:poke214,i:sys58732
940 printcl$;j
950 nexti
960 fori=3toi4step2:ifi=i1theni=i+i6
970 poke211,2:poke214,i:sys58732
980 print"--"
990 poke211,4:poke214,i+1:sys58732
1000 print"--"
1010 nexti:return
1015 :
1020 rem ** zuordnung eingeben **
1025 :
1030 u$=""
1040 zo$="":z=0:zu$(z)="":l=5:ii=1:poke2
11,5:poke214,ii:sys58732
1050 pokesc,s3:print"?";cl$;:poke204,0:g
osub1330
1060 ifu$="then340
1070 zo$=u$
1080 :
1090 rem ** lernstoff eingeben **
1100 :
1110 rem ** linke seite **
1120 :
1130 i=1
1140 fore=1toel
1150 e$(e)="":u$="":z=0
1160 i=i+2:ifi=i1theni=i+i2
1170 l=3:ii=i:poke211,l:poke214,i:sys587
32
1180 pokesc,s4:print"?";cl$;:poke204,0:g
osub1330
1190 ifu$=chr$(95)then340
1200 e$(e)=u$:gosub1220:nexte
1210 ife>elthengoto340
1220 :
1230 rem ** rechte seite **
1240 :
1250 ife=1theni8=2
1260 z=0:u$(e)="":u$=""
1270 i8=i8+2:ifi8=i5theni8=i8+i6
1280 ii=i8:l=5:pokesc,s5:poke211,l:poke2
14,ii:sys58732
1290 print"?";cl$;:poke204,0:gosub1330
```

```

1300 ifu$=chr$(95)then return
1310 u$(e)=u$:return
1315 :
1320 :      rem ---- inputroutine -----
1325 :
1330 getuu$(z):ifuu$(z)=""then1330
1340 ifpeek(207)<>0then1340
1350 poke204,1:ifuu$(z)=chr$(95)then e=e
1:u$=uu$(z):return
1360 ifuu$(z)=chr$(17)oruu$(z)=chr$(19)or
ruu$(z)=chr$(29)thenuu$(z)="":goto1330
1370 ifuu$(z)=chr$(145)oruu$(z)=chr$(147
)thenuu$(z)="":goto1330
1380 ifuu$(z)=chr$(157)oruu$(z)=chr$(148
)thenuu$(z)="":goto1330
1390 ifuu$(z)=chr$(20)theniflen(u$)>0the
nu$=left$(u$,len(u$)-1)
1400 ifuu$(z)=chr$(20)thenpoke211,1:poke
214,ii:sys58732:printle$;:p=z:p=p-1
1410 ifuu$(z)=chr$(20)thenpoke211,1:poke
214,ii:sys58732:printu$;:ifp<0thenp=0
1420 ifuu$(z)=chr$(20)thenpoke211,1+len(
u$):poke214,ii:sys58732:z=p:goto1290
1430 ifz>28thengosub3880
1440 ifz>33thenuu$(z)=chr$(13)
1450 ifuu$(z)=chr$(13)thenpoke211,1+z:po
ke214,ii:sys58732:print " ":return
1460 u$=u$+uu$(z)
1470 poke211,1+z:poke214,ii:sys58732
1480 printuu$(z);:z=z+1:poke211,1+z:poke
214,ii:sys58732:print"?";:cl$;
1490 poke211,1+z:poke214,ii:sys58732:pok
e204,0:goto1330
1500 :
1510 rem *** edieren / aendern ***
1515 :
1520 ifzo$=""thenprintcc$cd$ro$"Speicher
  leer!"rf$;fori=1to800:nexti:goto340
1530 printcc$;cd$;ce$;ce$;ce$;
1540 print"Speicherinhalt sichten/ aende
rn":printce$;ce$;ce$;
1550 print"Aendern ";ro$" a "rf$;
1560 print" weiter mit Return":print
1570 printzo$

```

```
1580 poke198,0:wait198,1:geta$:ifa$=""then
en1580
1590 ifa$="a"thengosub4530
1600 ifa$=chr$(95) oran$=chr$(95)thena$=
":an$="":goto340
1610 ifa$="a"thenzo$=an$
1620 print
1630 fore=1toed
1640 printe$(e)
1650 poke198,0:wait198,1:geta$
1660 ifa$="a"thengosub4530
1670 ifa$=chr$(95) oran$=chr$(95)thena$="
":an$="":e=ed:goto1740
1680 ifa$="a"thene$(e)=an$:print
1690 printu$(e)
1700 poke198,0:wait198,1:geta$
1710 ifa$=chr$(95) oran$=chr$(95)thena$="
":an$="":e=ed:goto1740
1720 ifa$="a"thenu$(e)="" : gosub4530
1730 ifa$="a"thenu$(e)=an$:print
1740 nexte
1750 goto340
1760 :
1770 :      rem ** save-routine **
1775 :
1780 printcc$;cd$;ce$;ce$;ce$;
1790 print"Diskette wird initialisiert"
1800 close15:open15,8,15,"i0"
1810 gosub4690:rem *** fehlerkanal **
1820 printce$;ce$;ce$;ro$"Save-Routine":
rf$
1830 printcd$;ce$;ce$;ce$;
1840 print"Directory j/n?":poke198,0:wai
t198,1:geta$:ifa$="j"thensys49152
1850 input"File-Name:";fi$
1860 iffi$="then1940
1870 close2:open2,8,2,"`0:"+fi$+",s,w"
1880 print#2,g$;zo$;g$;x$;ed$;x$;ed;x$;e
1;x$;e2;x$;j1;x$;i1;x$;i2;x$;i3;x$;
1890 print#2,i4;x$;i5;x$;i6;x$;
1900 fore=1toed
1910 print#2,g$;e$(e);g$;x$;g$;u$(e);g$;
x$;
```

```

1920 nexte
1930 gosub4690
1940 close2:goto340
1950 :
1960 :      rem ** load-routine **
1965 :
1970 printcc$;cd$;ce$;ce$;ce$;
1980 print"Diskette wird initialisiert":
e=1
1990 close15:open15,8,15,"i0"
2000 gosub4690:rem *** fehlerkanal **
2005 :
2010 printce$;ce$;ce$;ro$"Load-Routine";
rf$
2020 printcd$;ce$;ce$;ce$;
2030 print"Directory j/n?":poke198,0:wai
t198,1:geta$:ifa$="j"thensys49152
2040 input"File-Name: ";fi$
2050 iffi$=chr$(95)then2130
2060 close2:open2,8,2,"0:"+fi$+"s,r"
2070 input#2,zo$,ed$,ed,e1,e2,j1,i1,i2,i
3,i4,i5,i6
2080 rs=st
2090 input#2,e$(e),u$(e)
2100 ifst=64then2130
2110 gosub4690
2120 e=e+1:goto2090
2130 close2:goto340
2140 :
2150 :
2160 :      rem ** formen ueben **
2170 :
2180 ifzo$=""thenprintcc$;cd$;cd$;ce$;ce
$;ce$;ro$;
2190 ifzo$=""thenprint"Speicher leer!";r
f$:fori=1to1000:next:goto340
2200 printcc$;cd$;cd$;tab(7)ed;ed$"forme
n ueben"
2210 printcd$;cd$;cd$;cd$;tab(4)"Abfrage
links -> rechts ";ro$;" 1 ";rf$
2220 printcd$;cd$;tab(4)"Abfrage rechts
-> links ";ro$;" 2 ";rf$
2230 printcd$;cd$;tab(4)"Abfrage wechsel
nd ";ro$;" 3 ";rf$

```

```
2240 poke198,0:wait198,1:geta$:ifval(a$)
<lorval(a$)>goto2240
2250 :
2260 gosub760:rem *** bildschirm-maske
2270 :
2280 fori=1to10:ife$(i)=""thene$(i)=". "
2290 ifu$(i)=""thenu$(i)=". "
2300 next
2310 :
2320 ifa$="1"thenaz=1:goto2360
2330 ifa$="2"thenaz=2:goto2490
2340 ifa$="3"thenaz=3:fore=1toed:vt(e)=0
:next
2350 :
2360 : rem *** linke formen eintragen
2370 :
2380 i=1:poke211,5:poke214,1:sys58732:pr
intzo$
2390 fore=1toel
2400 i=i+2:ifi=i1theni=i+i2
2410 ife=1thenii=2
2420 ii=ii+2:ifii=i5thenii=ii+i6
2430 rr(e)=ii
2440 vt(e)=0:poke211,3:poke214,i:sys5873
2
2450 ifaz=3thenif(int(rnd(0)*2)+1)=2then
er$=e$(e):e$(e)=u$(e):u$(e)=er$:vt(e)=1
2460 printe$(e)
2470 nexte:gosub2610:gosub3580:goto340
2480 :
2490 : rem *** rechte formen eintragen
2500 :
2510 poke211,11:poke214,1:sys58732:print
zo$
2520 i=1
2530 fore=1toel
2540 i=i+2:ifi=i1theni=i+i2
2550 rl(e)=i
2560 ife=1thenii=2
2570 ii=ii+2:ifii=i5thenii=ii+i6
2580 poke211,5:poke214,ii:sys58732
2590 printu$(e)
2600 nexte:gosub2610:gosub3580:goto340
2610 :
```



```
2620 :    rem ** uebungsabfrage **
2630 aw=0
2640 forw=1toed
2650 n(w)=int(rnd(0)*ed)+1
2660 : ifw=1then2700
2670 :    fork=1tow-1
2680 :        ifn(w)=n(k)then2650
2690 :    nextk
2700 ifaz=1anded=10thengosub2790:goto339
0
2710 ifaz=3anded=10thengosub2790:goto339
0
2720 ifaz=2anded=10thengosub2930:goto339
0
2730 ifaz=1anded=8thengosub3030:goto3390
2740 ifaz=3anded=8thengosub3030:goto3390
2750 ifaz=2anded=8thengosub3150:goto3390
2760 ifaz=1anded=6thengosub3250:goto3390
2770 ifaz=3anded=6thengosub3250:goto3390
2780 ifaz=2anded=6thengosub3330:goto3390
2790 :
2800 rem ** abfrage 10 rechts **
2805 :
2810 ifn(w)=1thenp=5:pp=4:e=n(w):a$(e)=u
$(e):return
2820 ifn(w)=2thenp=5:pp=6:e=n(w):a$(e)=u
$(e):return
2830 ifn(w)=3thenp=5:pp=8:e=n(w):a$(e)=u
$(e):return
2840 ifn(w)=4thenp=5:pp=10:e=n(w):a$(e)=
u$(e):return
2850 ifn(w)=5thenp=5:pp=12:e=n(w):a$(e)=
u$(e):return
2860 ifn(w)=6thenp=5:pp=15:e=n(w):a$(e)=
u$(e):return
2870 ifn(w)=7thenp=5:pp=17:e=n(w):a$(e)=
u$(e):return
2880 ifn(w)=8thenp=5:pp=19:e=n(w):a$(e)=
u$(e):return
2890 ifn(w)=9thenp=5:pp=21:e=n(w):a$(e)=
u$(e):return
2900 ifn(w)=10thenp=5:pp=23:e=n(w):a$(e)
=u$(e):return
2910 :
```

```
2920      rem ** abfrage 10 links **
2925 :
2930 ifn(w)=1thenp=3:pp=3:e=n(w):a$(e)=e
$(e):return
2940 ifn(w)=2thenp=3:pp=5:e=n(w):a$(e)=e
$(e):return
2950 ifn(w)=3thenp=3:pp=7:e=n(w):a$(e)=e
$(e):return
2960 ifn(w)=4thenp=3:pp=9:e=n(w):a$(e)=e
$(e):return
2970 ifn(w)=5thenp=3:pp=11:e=n(w):a$(e)=
e$(e):return
2980 ifn(w)=6thenp=3:pp=14:e=n(w):a$(e)=
e$(e):return
2990 ifn(w)=7thenp=3:pp=16:e=n(w):a$(e)=
e$(e):return
3000 ifn(w)=8thenp=3:pp=18:e=n(w):a$(e)=
e$(e):return
3010 ifn(w)=9thenp=3:pp=20:e=n(w):a$(e)=
e$(e):return
3020 ifn(w)=10thenp=3:pp=22:e=n(w):a$(e)
=e$(e):return
3030 :
3040 rem ** abfrage 8 rechts **
3045 :
3050 ifn(w)=1thenp=5:pp=4:e=n(w):a$(e)=u
$(e):return
3060 ifn(w)=2thenp=5:pp=6:e=n(w):a$(e)=u
$(e):return
3070 ifn(w)=3thenp=5:pp=8:e=n(w):a$(e)=u
$(e):return
3080 ifn(w)=4thenp=5:pp=10:e=n(w):a$(e)=
u$(e):return
3090 ifn(w)=5thenp=5:pp=15:e=n(w):a$(e)=
u$(e):return
3100 ifn(w)=6thenp=5:pp=17:e=n(w):a$(e)=
u$(e):return
3110 ifn(w)=7thenp=5:pp=19:e=n(w):a$(e)=
u$(e):return
3120 ifn(w)=8thenp=5:pp=21:e=n(w):a$(e)=
u$(e):return
3130 :
3140 :      rem ** abfrage 8 links **
3145 :
```

```
3150 ifn(w)=1thenp=3:pp=3:e=n(w):a$(e)=e
$(e):return
3160 ifn(w)=2thenp=3:pp=5:e=n(w):a$(e)=e
$(e):return
3170 ifn(w)=3thenp=3:pp=7:e=n(w):a$(e)=e
$(e):return
3180 ifn(w)=4thenp=3:pp=9:e=n(w):a$(e)=e
$(e):return
3190 ifn(w)=5thenp=3:pp=14:e=n(w):a$(e)=
e$(e):return
3200 ifn(w)=6thenp=3:pp=16:e=n(w):a$(e)=
e$(e):return
3210 ifn(w)=7thenp=3:pp=18:e=n(w):a$(e)=
e$(e):return
3220 ifn(w)=8thenp=3:pp=20:e=n(w):a$(e)=
e$(e):return
3230 :
3240 rem ** abfrage 6 rechts **
3245 :
3250 ifn(w)=1thenp=5:pp=4:e=n(w):a$(e)=u
$(e):return
3260 ifn(w)=2thenp=5:pp=6:e=n(w):a$(e)=u
$(e):return
3270 ifn(w)=3thenp=5:pp=8:e=n(w):a$(e)=u
$(e):return
3280 ifn(w)=4thenp=5:pp=15:e=n(w):a$(e)=
u$(e):return
3290 ifn(w)=5thenp=5:pp=17:e=n(w):a$(e)=
u$(e):return
3300 ifn(w)=6thenp=5:pp=19:e=n(w):a$(e)=
u$(e):return
3310 :
3320 :      rem ** abfrage 6 links **
3325 :
3330 ifn(w)=1thenp=3:pp=3:e=n(w):a$(e)=e
$(e):return
3340 ifn(w)=2thenp=3:pp=5:e=n(w):a$(e)=e
$(e):return
3350 ifn(w)=3thenp=3:pp=7:e=n(w):a$(e)=e
$(e):return
3360 ifn(w)=4thenp=3:pp=14:e=n(w):a$(e)=
e$(e):return
3370 ifn(w)=5thenp=3:pp=16:e=n(w):a$(e)=
e$(e):return
```

```
3380 ifn(w)=6thenp=3:pp=18:e=n(w):a$(e)=
e$(e):return
3390 :
3400 :   rem ** getabfrage **
3405 :
3410 ac$="":j=len(a$(e))
3420 poke211,p:poke214,pp:sys58732:print
"?";c1$;
3430 foraf=1toj
3440 getax$(af):ifax$(af)=""then3440
3450 ifax$(af)=chr$(20)thenax$(af)="":af
=j:goto3410
3460 ifax$(af)=chr$(95)thengosub3690:ret
urn:rem menue ueber stringtauscher
3470 ifax$(af)=x$thenaf=j:poke211,p+af:p
oke214,pp:sys58732:print" ";:goto3520
3480 poke211,p-1+af:poke214,pp:sys58732:
printax$(af);
3490 ifaf=jgoto3510
3500 poke211,p+af:poke214,pp:sys58732:pr
int"?";
3510 ac$=ac$+ax$(af)
3520 nextaf
3530 ifvt(e)=1thener$=u$(e):u$(e)=e$(e):
e$(e)=er$:vt(e)=0
3540 ifac$<>a$(e)thenpoke211,p:poke214,p
p:sys58732:printchr$(5);a$(e);chr$(ch)
3550 ifac$<>a$(e)thenpoke198,0:ee=e:aw=a
w+1:fe$(ee)=e$(e)+" "+u$(e)
3560 ifac$=a$(e)thenpoke211,p:poke214,pp
:sys58732:printa$(e)
3570 nextw:az=0:fort=1to2000:nextt:retur
n
3580 :
3590 :   rem ** fehlerliste **
3600 :
3610 ifax$(af)=chr$(95)thenaf=j:w=ed:ret
urn
3620 printcc$;cd$;cd$;cd$;ro$;"Wiederhol
e:";rf$
3630 foree=1toed
3640 printfe$(ee):print
3650 fe$(ee)=""
3660 nextee
```

```
3670 printch$"Von"ed" Formen wuttest du
"(ed-aw)" ";ro$"Taste";rf$
3680 poke198,0:wait198,1:return
3690 :
3700 :   rem ** strings ordnen **
3710 fori=1toed
3720 ifvt(i)=1thener$=e$(i):e$(i)=u$(i):
u$(i)=er$
3730 nexti
3740 return
3750 :   rem *** sys-directory ***
3760 :
3770 fordi=49152to49256
3780 readdr
3790 pokedi,dr
3800 nextdi:return
3810 data169,36,133,251,169,251,133,187,
169,00,133,188,169,01,133,183,169,08,133
3820 data186,169,96,133,185,32,213,243,1
65,186,32,180,255,165,185,32,150,255,169
3830 data00,133,144,160,03,132,251,32,16
5,255,133,252,164,144,208,47,32,165,255
3840 data164,144,208,40,164,251,136,208,
233,166,252,32,205
3850 data189,169,32,32,210,255,32,165
3860 data255,166,144,208,18,170,240,06,3
2,210,255,76,76,192,169,13,32
3870 data210,255,160,02,208,198,32,66,24
6,96
3880 :   rem ** ton **
3890 :
3900 si=54272:pokesi+24,15:pokesi+7,207:
pokesi+8,34:pokesi+13,240
3910 pokesi+11,17:foro=1to200:nexto:poke
si+11,0:return
3920 :   rem *** sonderzeichen ***
3930 :
3940 ad=828
3950 readop$:gosub3990:pokead,op:ad=ad+1
:ifad=876then4030
3960 goto3950
3970 dataa5,14,48,a5,15,48,20,f7,b7,a5,0
1,48,a5,15,c9,d0,90,07,c9,e0,b0,03,a9,31
3980 data2c,a9,34,78,85,01,a0,00,b1,14,a
8,68,85,01,58,68,85,15,68,85,14,4c,a2,b3
```

```
3990 ol$=left$(op$,1):ox$=right$(op$,1)
4000 ol=val(ol$):if asc(ol$)>64then ol=asc
(ol$)-55
4010 ox=val(ox$):if asc(ox$)>64then ox=asc
(ox$)-55
4020 op=ol*16+ox:return
4030 poke785,828and255:poke786,3
4040 m=4096:a=13*m:e=14*m
4050 for i=atoe-1:pokei+m,usr(i):nexti
4060 pokea+24,24:poke56576,148:poke648,1
96
4070 printchr$(14)
4080 b=(peek(a+24)and2)*1024
4090 z=e+b+107*8:gosub4300
4100 z=e+b+115*8:gosub4300
4110 z=e+b+113*8:gosub4300
4120 z=e+b+114*8:gosub4300
4130 z=e+b+99*8:gosub4300
4140 z=e+b+119*8:gosub4300
4150 z=e+b+120*8:gosub4300
4160 z=e+b+98*8:gosub4300
4170 z=e+b+121*8:gosub4300
4180 z=e+b+111*8:gosub4300
4190 z=e+b+100*8:gosub4300
4200 z=e+b+95*8:gosub4300
4210 z=e+b+30*8:gosub4300
4220 z=e+b+28*8:gosub4300
4230 z=e+b+112*8:gosub4300
4240 z=e+b+110*8:gosub4300
4250 z=e+b+108*8:gosub4300
4260 z=e+b+123*8:gosub4300
4270 z=e+b+101*8:gosub4300
4280 z=e+b+116*8:gosub4300
4290 z=e+b+94*8:gosub4300:return
4300 for i=ztoz+7:reada:pokei,a:nexti:ret
urn
4310 data0,0,126,219,223,216,126,0
4320 data48,24,60,6,62,102,62,0
4330 data12,24,60,6,62,102,62,0
4340 data24,102,60,6,62,102,62,0
4350 data48,24,60,102,126,96,60,0
4360 data12,24,60,102,126,96,60,0
4370 data24,102,60,102,126,96,60,0
4380 data24,102,0,102,102,102,62,0
4390 data24,102,0,56,24,24,60,0
```

```
4400 data24,102,60,102,102,102,60,0
4410 data102,0,60,102,126,96,60,0
4420 data0,108,0,56,24,24,60,0
4430 data0,0,60,96,96,60,24,48
4440 data60,102,102,124,102,102,124,96
4450 data102,0,60,102,102,102,60,0
4460 data102,60,102,102,102,102,60,0
4470 data102,0,60,6,62,102,62,0
4480 data102,60,102,126,102,102,102,0
4490 data102,0,0,102,102,102,62,0
4500 data102,0,102,102,102,102,60,0
4510 data48,24,102,102,102,102,62,0
4520 :
4530 : rem ** aenderung eingeben **
4540 :
4550 an$="" :z=0:ae$(z)="" :print"?";cl$;
4560 getae$(z):ifae$(z)=""then4560
4570 ifae$(z)=chr$(17)orae$(z)=chr$(19)or
rae$(z)=chr$(29)thenae$(z)=""
4580 ifae$(z)=chr$(145)orae$(z)=chr$(147
)thenae$(z)=""
4590 ifae$(z)=chr$(157)thenae$(z)=""
4600 ifae$(z)=chr$(20)thenze=peek(214):p
oke211,0:poke214,ze
4610 ifae$(z)=chr$(20)thenprintle$;:poke
211,0:poke214,ze:goto4550
4620 ifz>28thengosub3880
4630 ifz>33thenae$(z)=chr$(13)
4640 ifae$(z)=chr$(13)thenprint" ":retur
n
4650 an$=an$+ae$(z)
4660 printae$(z);:z=z+1:print"?";cl$;
4670 goto4560
4680 :
4690 : rem ** fehlerkanal lesen **
4695 :
4700 input#15,en,em$,et,es
4710 ifen=0then return
4720 printcc$;cd$;cd$;ce$;ce$;ce$;ro$;
4730 print"Disketten-Fehler";rf$
4740 printen;em$;et;es
4750 close2:poke198,0:wait198,1:goto340
```

## Anhang

### Hinweise zur Anwendung der Diskette

Die Diskette enthält mehr Programme als das Buch. Zusätzlich wurde der Datensatz einiger Programme erweitert und deren Qualität gesteigert. Die Diskette erhalten Sie in Ihrem Fachgeschäft oder direkt bei mvg - moderne verlagsgesellschaft mbh.

(ISBN 3-478-99270-2

Preis: DM 58.-)

---

**\*\*\* PROGRAMMDISKETTE \*\*\***  
**NEUE LERNPROGRAMME**  
**Für Commodore 64/128**

---

Die Programme 1 bis 8 auf der Diskette werden mit

LOAD "Programmname",8 (RETURN)

geladen und mit RUN (RETURN) gestartet. Die Dateien 8A-8F können vom Programm 8 GROSSER LERNFILER geladen werden.

Lädt man zuerst das C-64 WEDGE von der DEMODISKETTE C=64, so genügt folgende Eingabe:

↑Programmname

In beiden Fällen kann man statt des gesamten Programmnamens die Zahl und den \* eingeben.

Beispiel: Programm

7 LERNFILER

Laden und Starten:

↑7\* (RETURN)

Die Programme 5 und 7 arbeiten mit Sonderzeichen. Laden Sie nach einem dieser Programme ein weiteres, so schalten Sie den Computer vorher kurz aus. Beim Programm SINUS UND KOSINUS ist immer zuerst Programm 2A\* zu laden.



<b>Diskname: "MVG NEUE LERNPRG" 64 2A</b>	<b>Blöcke</b>
1 MATHEMATIK-TRAINER	52
2A SINUS UND KOSINUS	7
2B SINUS UND KOSINUS	31
3 DEUTSCHE LITERATUR	47
4 DIE STEIGERUNG im Englischen	43
5 PHYSIK-TRAINER	27
6 GRUNDLAGEN DER Buchführung	34
7 LERNFILER	54
8 GROSSER LERNFILER	131
8A RECHTSCHREIBUNG	18
8B AVOIR/ETRE (Frz.)	13
8C PRAESENS/PERFEKT (ital.)	12
8D QUIZ	7
8E BIOGRAFIEN	9
8F GESCHICHTE	6

**Die Programmnummer ist Bestandteil des Programmnamens!**

### **Künstliche Intelligenz in BASIC**

Arbeitsbuch für Heimcomputer (ISBN 3-478-09310-4) DM 29,80

von Mike James

Künstliche Intelligenz ist auch auf dem Heimcomputer programmierbar! Die BASIC-Listings in diesem Buch demonstrieren es: Das Prinzip der Künstlichen Intelligenz ist nicht nur den „großen“ Rechnern vorbehalten.

### **BASIC-Computer Programme**

– **Mathematik für Schule, Wissenschaft und Technik** (ISBN 3-478-09255-7)

– **Physik für Schule, Wissenschaft und Technik** (ISBN 3-478-09260-4)

von Charles Sternberg, je DM 29,80

Diese beiden Bücher sind Sammlungen zahlreicher fachspezifischer Programme, die beim Lösen mathematischer bzw. physikalischer Aufgaben ständig vorkommen. Routine-Berechnungen müssen nicht zeitraubend sein: Diese Listings in allgemeinem BASIC ermöglichen es, sich von vornherein auf die eigentliche Problemstellung zu konzentrieren und effektiver zu arbeiten.

### **Computer-Lösungen für Schule, Studium und Alltag**

Ein mathematisches Arbeitsbuch mit Herleitungen  
Beispiele und Listings für den

– **Commodore 64/128** (ISBN 3-478-09210-8)

– **ZX Spectrum/Plus** (ISBN 3-478-09220-5)

von Rainer und Patrick Gaitzsch, je DM 29,80

15 Problemstellungen zum Mathematikunterricht der Kollegstufe werden ausführlich mit Abänderungen, Verbesserungen einschließlich endgültigem Listing kommentiert. Die Übertragung der mathematischen Fragestellung auf den Alltag macht das Üben und Lernen besonders kurzweilig.

### **Computer-Lösungen für Schule und Studium – Band II**

Ein mathematisches Arbeitsbuch mit Herleitungen  
Beispiele und Listings für den

– **Commodore 64/128** (ISBN 3-478-09280-9)

– **IBM (und kompatible Modelle)** (ISBN 3-478-09290-6)

von Rainer und Patrick Gaitzsch, je DM 29,80

Die Programme dieses Buches entsprechen dem Mathematik- und Physikunterricht der Grund- und Leistungskurse. Sie eignen sich zur unmittelbaren Anwendung im Informatik-Unterricht. Die aktuelle Problemstellung und Simulationen machen das frühere „Pauken“ zur spannenden Unterhaltung.

**mv**  
 **verlag**



## Neue Lernprogramme = neue Perspektiven für das Lernen mit dem Computer

Die „neuen Lernprogramme“ sind eine Auswahl neuer grundlegender Übungsprogramme für das computergestützte Lernen. Anhand konkreter Aufgaben aus dem Unterricht demonstrieren sie, wie vielfältig die Anwendungsmöglichkeiten des Computers beim Lernen sein können.

Die „neuen Lernprogramme“ erleichtern es dem Schüler,

- die Winkelfunktionen zu verstehen und anzuwenden,
- physikalische Berechnungen, an Kondensatorschaltungen und Spulen, vorzunehmen,
- sich die Grundkenntnisse der deutschen Literatur anzueignen,
- die Steigerung im Englischen zu beherrschen,
- die Grundlagen der Buchführung zu trainieren,
- sich Fakten einzuprägen.

Das Programm „Lernfiler“ zeichnet sich durch seine Vielseitigkeit besonders aus: Es eignet sich für alle Lernstoffe nach dem Frage- und Antwortprinzip und ist übertragbar auf alle Fächer, zum Beispiel Biologie, Erdkunde, Geschichte, Latein.

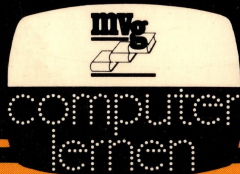
Das „Programm-Telegramm“ beschreibt die einzelnen Programme nach Anwendungsziel und ihren speziellen Möglichkeiten. Programm-Ablauf, Lernstoffeingabe, Variations- und Erweiterungsmöglichkeiten werden ausführlich kommentiert. Auf programmiertechnische Vor- und Nachteile, Besonderheiten im Listing und auf Steuerzeichen wird extra verwiesen.

Die ausführlich kommentierten „neuen Lernprogramme“ bieten jedem die Möglichkeit, den Lernstoff nach verschiedenen Schwierigkeitsgraden abzustimmen, seine programmiertechnische Raffinesse zu schulen.

Die Computer-Bücher der Reihe „mvg-computer-lernen“ folgen einem klaren pädagogischen Konzept: Die Programme sind so kommentiert, daß der Computer-Anwender Aufbau und Ablauf nachvollziehen und erlernen kann. Die Programme für computergestütztes Lernen sind mit Lernstoff erweiterbar und je nach Schwierigkeitsgrad ausbaufähig.

Die „neuen Lernprogramme“ sind auch auf Diskette erhältlich (ISBN 3-478-99270-2).

ISBN 3-478-09270-1



Alle Listings sind auch als Software erhältlich. Sie ersparen sich damit das Eingeben der Programme.

