20,– DM
Ös 170,–
sfr 20,–

SONDERHEFT
Joyce Nr. 2

JOYCE

eine Publikation der PC Schneider International

Sonderheft
Joyce
2/87/88

Programme:
Adreß- und
Archivverwaltung

Hardware:
Zweitlaufwerk

Turbo Pascal:
Grafik-Toolbox!

Tips:
zu dBase, LOGO,
LocoScript, BASIC



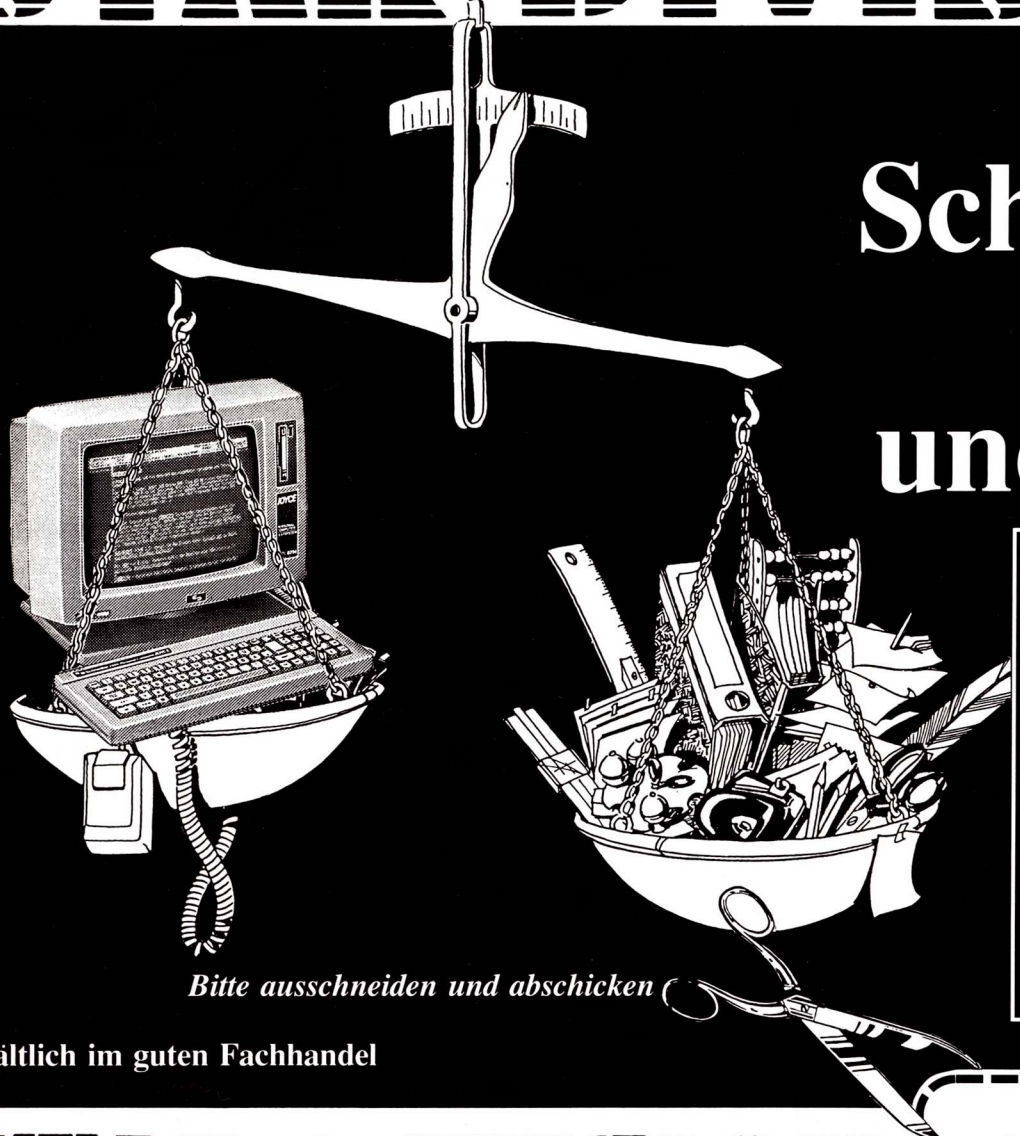
Die Zukunft hat begonnen!

Super-Software von

Günstige Preise

STAR-DIVISION

für Schneider Joyce und CPC



CPC
STAR-WRITER I
DATEI-STAR
Mouse Operating System
STATISTIC-STAR
FIBU-STAR Plus
u.v.m

Joyce
Joyce-Mailing-System
STATISTIC-STAR
STAR-BASE
BUSINESS-STAR
BUISINESS-STAR Plus
FIBU-STAR
u.v.m.

Bitte ausschneiden und abschicken

Erhältlich im guten Fachhandel

STAR-DIVISION

Uelzener Straße 12 • 2120 Lüneburg • Tel.: (0 41 31) 40 25 50

Vertrieb für die Schweiz.
VCS Video-Computershop • Schaffhauserstr. 473 • P.O.Box 103 • CH-8052 Zürich • Tel.:01/3 022600

Vertrieb für Österreich:
Ueberreuther-Media • Alserstr. 24 • A-1091 Wien • Tel.: 4 81 53 80

Copyright by STAR-DIVISION GmbH 1987

Info-Coupon

Ich möchte mehr Informationen über das Software-Lieferprogramm Joyce ☐ CPC ☐

Name, Vorname

Strasse, Nr.

PLZ, Ort

Bitte ausschneiden, auf eine Postkarte kleben oder in einen Briefumschlag stecken und an STAR-DIVISION GmbH schicken.

Impressum

Herausgeber:
Christian Widuch

Chefredakteur:
Stefan Ritter (SR)

Redaktion:
Michael Ebbrecht (ME)

Satz:
Claudia Küllmer, Silvia Führer,
Martina Siebert, Gabriela Joseph

Gestaltung:
Gerd Köberich, Petra Biehl,
Helmut Skoupy, Manuela Eska

Reprografie:
Helmut Skoupy,

Grafik/Illustration:
Heinrich Stiller

Fotografie
Gerd Köberich, Christian Heckmann

Freie Mitarbeiter:
M.Anton, Dr. K. Stratemann, W. Huske,
U. Schmidt, R. L. Suermann, O. Schmidt,
C. Reimann, W. Otternberg, F. Sauerländer
H. Schmidt, J. Brandt, C. Frederking
H. Ackermann, G. Ott, S. Stürmer,
P. Horstmann, M. Meyer, M. Laue, C. Reimann
E. Wirth, B. Breede, A. Schmidt, K. Wiltsch

Anzeigenleitung:
Wolfgang Schnell

Anzeigenverkauf:
Wolfgang Brill
Es gilt die Anzeigenpreisliste Nr. 2a vom 15.10.86

Anschrift Verlag/Redaktion:
DMV Daten & Medien Verlagsgesellschaft mbH
Postfach 250
3440 Eschwege
Telefon: (0 56 51) 87 02

Vertrieb:
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Druck:
Druckerei Jungfer, 3420 Herzberg

Bezugspreise:
Schneider PC International
Joyce Sonderheft 2/87.
Erscheint unregelmäßig.

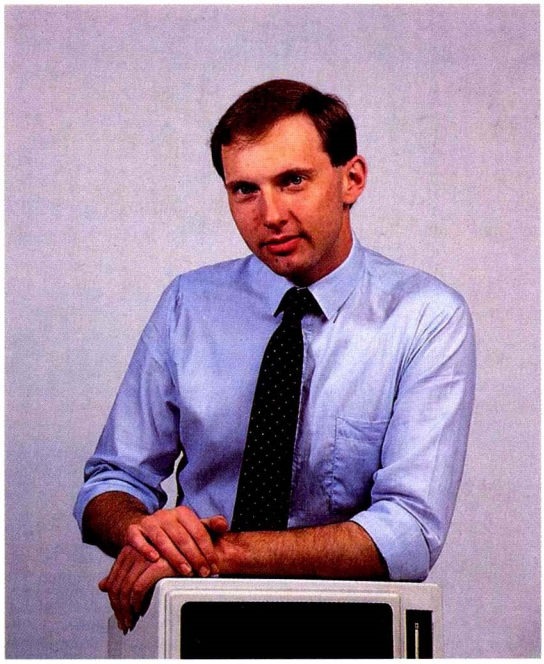
Einzelpreis DM 20,–/sfr. 20,–/ÖS 170,–

Für unverlangt eingesandte Manuskripte und Datenträger sowie Fotos übernimmt der Verlag keine Haftung.

Honorare nach Vereinbarung (die Zustimmung zum Abdruck wird vorausgesetzt).

Das Urheberrecht für veröffentlichte Manuskripte liegt ausschließlich beim Verlag. Nachdruck sowie Vervielfältigung oder sonstige Verwertung von Texten, nur mit schriftlicher Genehmigung des Verlages.

Namentlich gekennzeichnete Fremdbeiträge geben nicht in jedem Fall die Meinung der Redaktion wieder.



Liebe Leserinnen, liebe Leser,

allen Unkenrufen zum Trotz hat sich der Joyce seinen Anteil am Markt bewahrt; im Software- und Zubehörbereich tritt in zunehmendem Maße Qualität an die Stelle des schnellen Geldes. Die Freunde des letzteren haben sich auf den jüngst für die Masse erschlossenen PC-Markt begeben; was zurückbleibt, sind Hersteller und Händler, die in der Betreuung der Joyce-Kunden und der Entwicklung neuer bzw. Verbesserung bestehender Produkte eine sinnvolle Aufgabe sehen. So haben die PCWs die Chance, der Eskalation von Leistung, Geschwindigkeit und Statusdenken zu entgehen, indem die Fähigkeiten der Maschine wirklich ausgereizt werden. Natürlich ist eine MS-DOS-Maschine mit einer höheren Arbeitsgeschwindigkeit ausgestattet, und auf diesem Marktsegment tummeln sich natürlich die Anbieter umfangreicher Softwaresysteme. Die Frage beim Kauf eines Rechners ist jedoch die nach dem Einsatzgebiet, und hier hat der Joyce eine ganz bestimmte Gruppe von Anwendern für sich vereinnahmt. Die Solidität des Konzepts der Hard- und Software ist m.E. einmalig! Joyce ist ein Computer, der bei entsprechender Nutzung keinerlei Wertminderung (außer der rein materiellen) unterliegt. Unter dieser Prämisse ist auch dieses Sonderheft konzipiert worden. Es kann und soll kein Lehrbuch für Einsteiger sein, ebenso ist hierin kein Forum für nur-Programmierer zu sehen. Die ausgewogene Mischung an Erweiterungen und Verbesserungen Ihrer Programmbibliothek und die Erweiterung Ihrer Möglichkeiten der Verwendung des Computers durch Arbeitshilfen für bestehende Softwarelösungen stellen daher den Schwerpunkt der Thematik im Sinne des oben Gesagten dar. Ihre Anregungen zur Verbesserung waren bei der Erstellung des vorliegenden Heftes eine wertvolle Hilfe; ich möchte Sie auch in dieser Ausgabe um Ihre Mithilfe bitten.

Herzlichst

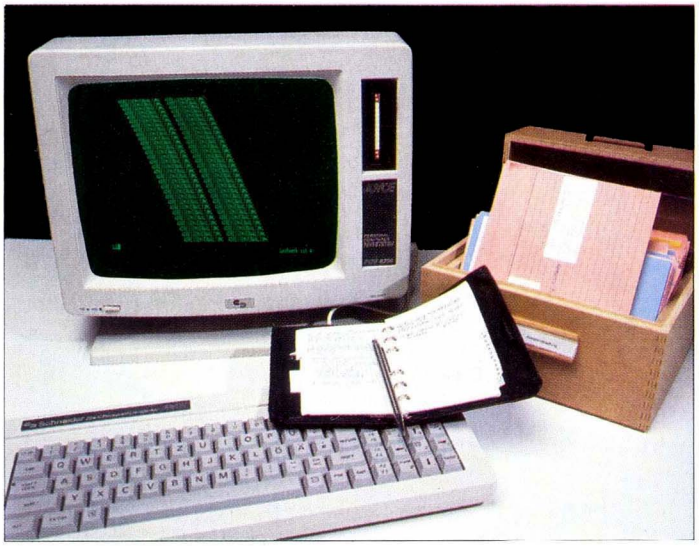
M. Ebbrecht

Michael Ebbrecht
Redakteur Joyce



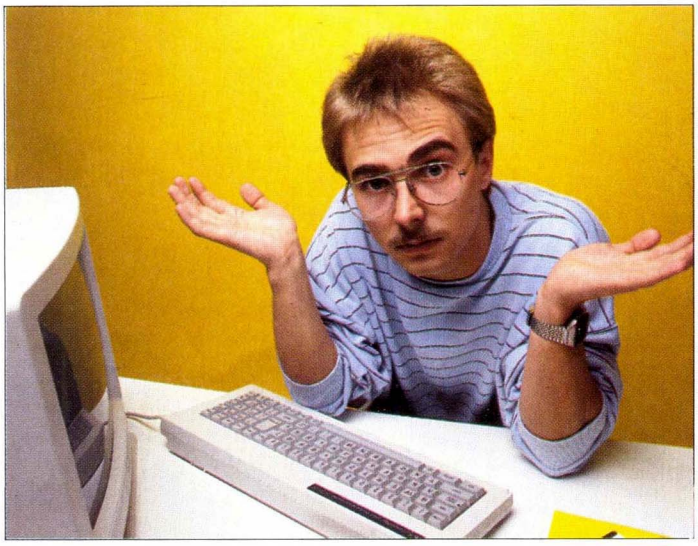
Software, Zubehör und Hardware in einer großen Übersicht. Alles, was das Herz begehrt...

S. 10



Sie mögen keine Briefmarken? Sammeln Sie Adressen mit unserer Adressverwaltung...

S. 20



Stehen Sie vor einer schwierigen Entscheidung? Lassen Sie sich von unserem Entscheidungsgenerator helfen!

S. 43

Berichte:

Der Beitrag ***Ode to Joyce*** ist eine persönliche Stellungnahme zu Werdegang und Zukunft der PCWs .....**S. 6**

Unsere ***Marktübersicht*** enthält Anbieter von Software, Hardware und Zubehör für Joyce .....**S. 10**

Wie hat's der Joyce mit der ***Logik***? Dieser Bericht verschafft Überblick .....**S. 16**

Programme:

Eine mit den JETSAM-Funktionen arbeitende ***Adreßverwaltung*** macht Schluß mit der Zettelwirtschaft .....**S. 20**

Sammeln Sie die Daten Ihrer Schallplatten- oder Videosammlung komfortabel mit unserer ***Archiv-Verwaltung*** .....**S. 24**

Für Aufsteiger: ***PASCOM*** ist ein in Mallard-BASIC geschriebener PASCAL-Compiler .....**S. 28**

Eine Simulation ökologischer (oder anderer) Zusammenhänge ist mit dem ***Kybernetik-Lernspiel*** möglich .....**S. 34**

Das Programm ***CHOICE*** ist ein unentbehrlicher Helfer für die Entscheidungsfindung bei komplizierten Problemen .....**S. 43**

Einen ***Kalender*** druckt das gleichnamige Programm für Sie aus .....**S. 50**

Zur Entspannung bietet sich ***SUBURBIA*** an, angelehnt an ein weltberühmtes Straßenmaklerspiel .....**S. 52**

Eine komplette Turbo-PASCAL-Toolbox stellt Ihnen die ***TURBO-GRAFIK*** zur Verfügung – garantiert ohne GSX .....**S. 60**

Grafmod I – mit neuem Layout.....**S. 121**

Hardware:

Es muß nicht immer ein Original-Zweitlaufwerk sein. Tips und eine Bauanleitung bietet Ihnen der Beitrag ***Joyce-Zweit-Floppy*** .....**S. 76**

Der ***Umbau des Helligkeitsreglers*** ermöglicht es, die Bildschirmhelligkeit von 0 bis 100 zu regeln .....**S. 81**

Tips:

Ordnung in BASIC-Programmlistings bringt der ***LISTER*** ..... **S. 84**

Balkencursor wie unter LocoScript selbst programmieren? Wie's geht, erfahren Sie auf ..... **S. 88**

Mathematische Funktionen aller Art stellt das Programm ***3D-Plot*** dreidimensional am Bildschirm dar .... **S. 94**

Den Ausdruck von Adreßaufklebern vereinfacht das Programm ***Labelprint*** ..... **S. 96**

Suchen Sie ein Register, welches bei der Arbeit stets aufrufbar ist? ***Quickregister*** macht's möglich! ..... **S. 98**

Einiges über Zufallszahlen erfahren Sie im Beitrag ***Random*** auf ..... **S. 102**

Durch geschickte Auswahl der ***Dateinamen*** können Sie unter LocoScript Ordnung in Ihr Directory bringen. Auch Sonderzeichen sind möglich! ..... **S. 103**

Spaltensatz unter LocoScript ist kein Problem mehr, wenn Sie diesen Artikel gelesen haben..... **S. 104**

Komfortable Belegung der Funktionstasten nach Ihren Wünschen schnell und einfach mit ***FKeys*** ..... **S. 106**

dBase:

Viele ***Tips zur Installation von dBase*** von einem Anwender für Anwender ..... **S. 110**

Drucken Sie sich mit vorliegendem BASIC-Programm ein ***dBase-Handbuch*** aus! ..... **S. 116**

LOGO:

Lehrreich und nützlich: Die Bereiche Mathematik und Grafik verbindet der ***LOGO-Funktionenzeichner*** ..... **S. 118**

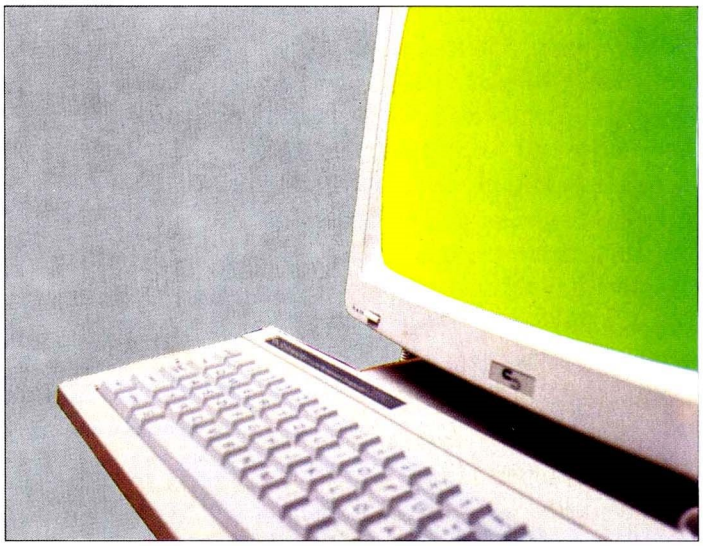
Rubriken:

Editorial ..... **S. 3**
Impressum ..... **S. 3**
Inserentenverzeichnis ..... **S. 117**



Gehen Sie in die Schwarzwaldklinik... Gehen Sie direkt dorthin. Entspannung beim Spiel (1 – 4 Personen)

S. 52



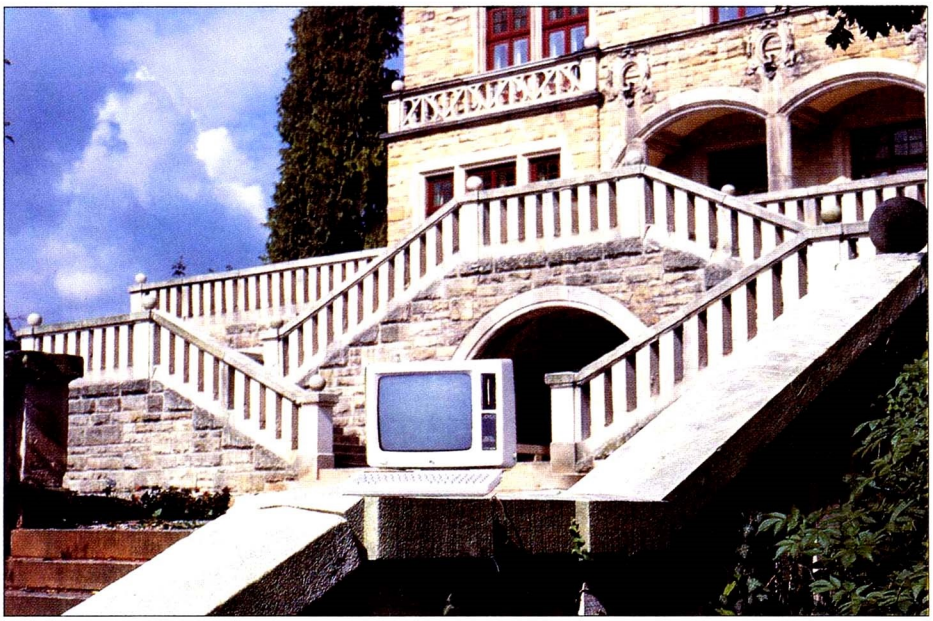
Ganz hell oder ganz dunkel – eine kleine Bastelei erweitert den Regelbereich für die Bildschirmhelligkeit.

S. 81



Zweitlaufwerk preiswert selbst eingebaut. Wie es geht, erfahren Sie auf Seite...

S. 76



ODE TO JOYCE

Keine Sorge: ganz so schwärmerisch, wie die Titelzeile klingt, liest sich dieser Artikel nicht. Da jedoch der PCW der Fama zufolge nach einer Dame bei Amstrad in England benannt wurde, ist dieser sanfte Ausdruck eher eine angemessene Inhaltsbeschreibung als etwa die aggressive Formulierung »Eine Lanze für den Joyce«.

Es ist doch seltsam. Da hat man sich 1985 seinen ersten Computer zugelegt, ohne je an einem solchen Gerät gesessen zu haben, voller Zweifel, ob man damit zurecht käme, aber mutig entschlossen. Etwas verwegen kam man sich vor als Mensch mittleren Alters, und fast auf der Höhe der Zeit. Auslöser für den Kauf war der Werbeslogan »Das komplette deutsche Schreibsystem". Nun ist man eingearbeitet, hat die Hürden anfänglich fehlerhafter Software und unzulänglicher Dokumentation überwunden, hat durch einschlägige Literatur seinen Horizont erweitert, sich den seinerzeit nicht bekannten weiteren Möglichkeiten des Rechners zugewandt, ihn den wachsenden Kenntnissen und Bedürfnissen in Hardware und Software angepaßt, liest Fachzeitschriften, schreibt sogar ein wenig über das Gebiet (dabei wollte man das Ding doch eigentlich nur benutzen!), und was passiert?

Verunsichert wird man. Ständig kommt Neues auf den Markt. 8-Bit Rechner sind out; waren schon überholt als man sich sein Maschinchen zulegte, so hört man; 16-Biter sind in (und fast auch schon wieder ein alter Hut); CP/M ist veraltet, MS-DOS der derzeitige Standard.

Die Fachblätter spiegeln die Entwicklung wieder, im redaktionellen ebenso wie im Anzeigenteil. Scharenweise scheinen die Leute ihren Joyce zu verkaufen. Aufsteiger sind sie, heißt es.

Kann es sein, daß man als Joyceler Angehöriger eines unterentwickelten Stammes oder einer aussterbenden Spezies ist?

Geht Ihnen Ähnliches auch periodisch durch den Sinn? Dann lesen Sie bitte weiter, denn in den nächsten Absätzen finden Sie vielleicht irgendwo den einen Hinweis der Ihnen bestätigt, daß Sie mit ihrem Joyce gar nicht so schlecht dran sind und in absehbarer Zeit auch nicht dastehen werden. Etwas sachlicher wird's auch zugehen, Ehrenwort.

Ich habe mich ein wenig über den Joyce umgehört, bei meinem Redakteur, im Fachhandel und bei der Industrie; habe die einschlägige Literatur noch einmal Revue passieren lassen und mir kritisch angesehen, was sich an meinem eigenen Joyce abspielt. Da kam ein recht buntes Spektrum an Aussagen und Perspektiven zusammen, die eine Betrachtung wert sind.

Zunächst aber müssen wir unser Gedächtnis ein wenig auffrischen:

- Der Joyce entsprang zwar einem gemeinsamen Konzept der Firmen Schneider und Amstrad (Allan Michael Suger Trading), aber die Federführung bei der Entwicklung lag in England.
- Wir befinden uns hier – falls es so etwas gibt – auf einem »Sekundärmarkt«.
- »Joyce« und »Komplettsystem für Textverarbeitung« ist unser Rechner nur hier bei uns, aufgrund eines genialen Marketing-Konzeptes, das mit Produktidentifikation (Name ist besser als Nummer) und eindeutiger Zielrichtung der Werbe-Aussage beim deutschen Verbraucher genau auf den Punkt traf: 70.000 Rechner dieses Typs waren Mitte des Jahres verkauft.
- Jenseits des Ärmelkanals ist er einfach der PCW, ein Personal Computer unter vielen, und doch verkauft er sich drüben auch nicht schlechter. Man sagt – etwas überspitzt – er sei dort in jedem kleinen Eckladen zu finden.

Bei uns hat es Aufregung gegeben, als der Hersteller neben dem Fachhandel auch die Kaufhäuser und Versandketten am Vertrieb beteiligte. Vielleicht ist dieser Vertriebsmodus dem Ruf des Joyce abträglich gewesen (Stichwort »Kaufhaus-Computer«), und sicher hat er – im Verein mit der einseitigen Werbung – die wahren Qualitäten dieses Rechners eher verdeckt als allgemein bekannt gemacht.

Der Fa. Schneider deshalb Minuspunkte anzukreiden, wäre gewiß unfair, denn zwischen dem Joyce und den anderen Serien muß natürlich aus marktechnischen Gründen eine Differenzierung stattfinden. Kritik müßte eher in einem anderen Punkt ansetzen. Fred Köster, Leiter der Schneider Computer Division, konstatierte in einem Interview mit dieser Zeitschrift (Heft 4, S. 19):

»Der Joyce wird als Textsystem gekauft und erst in zweiter Linie als Computer im eigentlichen Sinne genutzt«.

Der erste Teil dieser Aussage ist unzweifelhaft richtig, zumindest für den deutschen Markt. Der zweite Halbsatz hätte in mehrfacher Hinsicht einer Differenzierung bedurft.

Es wäre interessant zu wissen, in wieweit Joyce Besitzer ihr Gerät nach, sagen wir, einem Jahr für andere Zwecke einsetzen. Sicherlich wird Textverarbeitung weiterhin das Hauptgebiet sein und der Prozentsatz sonstiger Anwendungen niedriger liegen als bei anderen PCs. Wäre das ein Wunder bei einer Dokumentation, die geeignet ist, einen Anfänger mutlos zu machen?

Fachblätter und Bücher vermögen natürlich das Defizit auszugleichen, aber es wäre unsinnig zu erwarten, daß ein Computerneuling sich derart engagieren könnte. Auch vor dem Kauf von Fachliteratur liegt eine Hemmschwelle. Die Begleitdokumentation des Joyce ist zumindest nicht dazu angetan, dem Neuling den Wunsch nach Einstieg in die weiteren Möglichkeiten zu wecken, die dieses vorgebliche »Schreibsystem« bietet.

Ein Blick auf den Markt im Ursprungsland belegt, daß unter den dortigen Verhältnissen für den Joyce und seinem Anwender offenbar von vornherein ein anderes Umfeld besteht. Da wird erstens Software in Hülle und Fülle angeboten. Einiges davon – mehr oder minder gut umgesetzt – ist inzwischen auch bei uns erhältlich. Manche der anderen Programmtypen hätten bei uns wohl kaum eine Marktchance, weshalb das finanzielle Risiko einer Eindeutschung nicht eingegangen wird. Das beginnt mit Lernprogrammen für Kinder aller Altersklassen (MAGIC MATHS, 4-8 J.; MATHS MANIA, 8-12 J.; BETTER MATHS, CHEMISTRY, BIOLOGY, 12-16 J.), und reicht über hierzulande unbekannte Software für den Spekulanten (STOCKMARKET) und Wettfreund (POOLSWINNER, COURSEWINNER) bis zu einer Unzahl von Computerspielen.

Ein ähnliches Bild bietet sich bei den Zusatzgeräten. Allein ein so lapidarer Artikel wie ein Druckerständer für den PCW ist drüben gleich in vier Modellen am Markt. Speziell auf den Joyce systemnah geschriebene Software ist auch bei uns angelsächsischen Ursprungs, und bei der Peripherie sieht es ähnlich aus. Markantes Beispiel: zwei der drei jetzt für den Joyce angebotenen Winchesters stammen aus England. Das einheimische Produkt ist zudem durch ein völlig anderes Konzept gekennzeichnet.

Ich möchte nicht an mangelndes Interesse bei den hiesigen Entwicklern glauben und habe herumgefragt, warum offenbar nur in England sowohl Soft- als auch Hardware dem Joyce direkt »auf den Leib geschneidert« werden können. Was ich zur Anwort erhielt, scheint recht logisch: Es liegt keine veröffentlichte Literatur vor über den internen Aufbau des Joyce, wie zum Beispiel die fest vorprogammierten Routinen im ROM, und derartige Systemunterlagen werden, so hat es den Anschein, nur einigen wenigen englischen Firmen zur Verfügung gestellt.

Dieser Zustand wird nicht von Dauer sein, aber im Moment müssen wir wohl darauf hoffen, die englischen Entwicklungen hier in »eingedeutschten« Versionen angeboten zu bekommen, entweder durch die Hersteller selbst, oder durch die Initiative von hiesigen Anbietern, die im Joyce ein auch für die Zukunft ernst zu nehmendes Gerät sehen.

Und da sind die Dinge in Bewegung: So ließ sich beispielsweise ARNOR auf dem deutschen Markt nieder und bietet seine gesamte Produktkette direkt an, darunter mit dem Prowort eine deutsche Version des drüben ungeheuer beliebten Protext. Damit haben die Anwender, welche mit dem LocScript und seinen Zusatzprogrammen nicht ganz zufrieden waren, neben dem guten alten Wordstar jetzt eine weitere Alternative.

Von der ASD-Festplatte, sagte man mir, soll noch in diesem Jahr eine Version mit 40-Megabyte angeboten werden, und von der WEB wird wohl beim Erscheinen dieses Artikels ein neues Modell mit integriertem Backup-Streamer erhältlich sein, ein Stück Hardware, das bislang für keinen anderen PC existierte..

Diese Artikel sind in erster Linie für den Bereich des professionellen Einsatzes gedacht. Daß der Joyce dort offenbar anders gesehen wird als oft angenommen, ergab in ersten Hinweisen die Auswertung des NUTZWARE Fragebogens (Heft 8). 8 % der Einsender, hieß es dort, haben mit der Textverarbeitung »nichts im Sinn«, und 4 % setzen eine Festplatte ein. Nun ist klar, daß die Fragebogen-Aktion das Bild des professionellen Anwenderbereiches nur unvollkommen widerspiegeln kann.

Zunächst einmal sind natürlich nur Leser dieser Zeitschrift erfaßt, und darunter nur solche mit genug Interesse, um sich der Mühe der Beantwortung zu unterziehen. Meine Nachfragen haben denn auch ergeben, daß der Joyce in weitaus größerem Umfang professionell eingesetzt wird als man vermuten sollte, teilweise in Einsatzformen, die man ihm gar nicht zutraut, und mit denen er in Bereiche vorstößt, die man größeren Rechnern vorbehalten glaubt.

Dazu fand sich in Heft 7 das Beispiel der Fa. Schwenckner, die im maritimen Zubehör-Versandhandel zwei Joyce als Terminals in einem schon recht anspruchsvollen Datenverarbeitungssystem einsetzt. Derartige Anlagen mit dem Joyce als Rechner stehen bei Möbelspeditionen, Produktionsbetrieben diverser Branchen, Verlagen und Universitäten, von Norddeutschland bis nach Österreich und in die Schweiz. Da arbeitet der Joyce in Kombination mit Peripherie, deren Anschaffungswert schon mal um ein Mehrfaches über dem des Rechners liegt (z.B. Drucker um 7.500 DM).

Offenbar hat sich der Joyce bei dieser Art Einsatz sehr bewährt, denn es sind in Hamburg Bestrebungen im Gange, in Zusammenarbeit mit ARNOR durch

Entwicklung geeigneter Terminalprogramme den Einsatz des Joyce auch in Mehrplatzsystemen zu ermöglichen, die an Komplexität das oben angeführte System weit übertreffen. Datenfernübertragung für diese Konfigurationen ist im Gespräch, und ebenso ein Scanner als selbständige Peripherie.

Man ist als »Klein-Anwender« zunächst geneigt, den Nutzwert derartig aufwendiger Entwicklungen mit ihrem in Relation zum Joyce sehr hohen Kaufpreis in Frage zu stellen. Aber damit verhält es sich natürlich etwa so wie mit dem Automobilrennsport und den AlltagsPKWs: Es fallen nützliche Beiprodukte an. Die jetzt am Markt befindlichen Festplatten bespielsweise sind eine Erweiterung, die auch für viele nichtprofessionelle Nutzer von unbestreitbarem Interesse ist. Hierzulande sind die Entwickler dem Joyce gegenüber zurückhaltender. Da war zum Beispiel bei einem bekannten deutschen Peripherie-Hersteller ein MS-DOS-Emulator für den Joyce in Arbeit, aber wie mir jetzt gesagt wurde, ist nicht mehr an eine Einführung gedacht. Verständlich, denn warum auch MS-DOS für den Joyce?

Für CP/M existiert ein weltweiter Pool an Software, aus dem auch bei uns Einiges zur Verfügung steht, ausgereifte Produkte zumeist. In der Marktübersicht in Heft 1/87 fallen mir da zunächst dBase II, Multiplan, Wordstar, DR Draw, und DR Graph als fast legendäre Programme ins Auge, und eine Durchsicht des Angebots der letzten Monate erweist, daß die Palette ständig bunter wird. Diverse Programmiersprachen wären hier noch zu nennen und manch brauchbares Programm aus der Public Domain.

Einiges entgeht dem Joyceler natürlich systembedingt. Wer derzeit mit dBase II arbeitet, liebäugelt eventuell mit dBase III. Aber deswegen einen anderen Rechner kaufen mit neuen Anfangsproblemen und dem ganzen Schwanz zusätzlicher Kosten?

Auch das Diskettenformat macht uns derzeit noch manche Software unzugänglich. Wie oft bin ich jedoch froh, daß ich die Floppy Disks, die eben keine sind, nicht mit Samthandschuhen anfassen muß und Möglichkeiten der Format-Konversion bestehen – geeignete Hardware vorausgesetzt – auch. Bei Einsatz einer Festplatte benötigt man die Diskette ohnehin nur noch für den Programmstart und zum Zwecke von Backups, und mit einem Streamer nicht mal mehr hierfür.

Derartige Lösungen werden für die Mehrzahl der Anwender entweder nicht erforderlich sein oder unerschwinglich bleiben. Schauen wir uns deshalb doch einmal an, was wir schon an einem einfachen Joyce oder Joyce Plus haben.

– Der Drucker:

Reichlich langsam ist er, aber das Schriftbild kann sich eigentlich sehen lassen. Und die Einfachheit der Bedienung ist auf Grund der Steuerung außergewöhnlich. Fünf Minuten Anweisung und Demonstration genügten, damit meine neue Auszubildende den Ausdruck diverser mehrseitiger Texte vornehmen konnte. Wo findet man das sonst?

– Das LocoScript:

Ebenfalls ziemlich langsam, jedoch unübertroffen in puncto Bedienungskomfort und vor allem Erlernbarkeit. Dieser Text wird mit Prowort geschrieben, weil die Umsetzung zur Druckvorlage einfacher ist, weil vieles schneller geht und ich die Gelegenheit zur weiteren Einarbeitung nutze. Ohne Grundkenntnisse in CP/M Plus fiele das schwer, während man beim LocoScript rein gar keine benötigt. Meine Helferinnen beispielsweise arbeiten mit einigen der Schablonen aus Jürgen Siebert's Buch »Praktische Textverarbeitung mit Joyce«. Da werden bei jedem Posteingang die Aufklebe-Etiketten angeschaut und festgestellt, daß die unsrigen gegenüber denen unserer Korrespondenten und Lieferanten geradezu wie gedruckt aussehen.

Mit den Zeichendickten 10, 12, 15 und 17, dazu Proportionalschrift und die jeweiligen gedehnten Schriftbreiten, Einsatz von Superscript und was der Möglichkeiten mehr sind, bietet LocoScript eine größere Vielfalt an Gestaltungsmöglichkeiten, als mir von irgendeinem anderen Textverarbeitungsprogramm für PCs bekannt sind. Ein Mangel im Programm gibt da ein besonders gutes Beispiel ab für die positiven Seiten des LocoScript:

Mit 15 oder 17 Zeichen/Zoll ist bekanntlich in NLQ kein Fettdruck möglich. Damit wäre bei anderen Programmen das Thema abgehakt. Nicht so im LocoScript. Man hilft sich, indem man für die zum Fettdruck vorgesehenen Worte auf eine der größeren Schriftbreiten umschaltet und danach wieder zur kleineren Schrift zurückkehrt. Das bedeutet Verwendung unterschiedlicher Schriftbreiten innerhalb ein und derselben Zeile und ist ohne Auflösung der rechten Randbegrenzung in keinem vergleichbaren Schreibsystem möglich.

Ich muß natürlich gestehen, daß mir das unzulängliche Bearbeitungs- und Ausdrucktempo mit LocoScript und dem normalen PCW-Drucker schon oft lästig sind. Das hat mich aber nicht gehindert, einen Buchtext von derzeit mehr als 600 Seiten Din A4, mit Kapiteln von bis zu 60 Kilobyte, mehrmals zu überarbeiten und zur Reproduktion im Offset-Verfahren fertig auszudrucken. Und die restlichen etwa 250 Seiten werden auch noch auf die gleiche Art fertiggestellt werden.

Nach Einführung von Prowort und mit dem bevorstehenden Erscheinen von LocoScript 2, von dessen nicht unerheblichen Verbesserungen auch in der Geschwindigkeit ich mich im Test selbst überzeugen konnte, kann man nun – bei entsprechendem Bedarf und finanziellen Aufwand- endlich eine Vielzahl an Alternativen für die Textverarbeitung entwickeln. Unsere wird so aussehen:

Prowort für lange Texte, Serienbriefe, Rundschreiben; ein schnellerer Fremddrucker für jeden Einsatz von Endlospapier; LocoScript 2 für all die Schriftsätze, bei denen das ansprechende Aussehen wichtig ist und der Umfang nicht zu groß; der Joyce-Drucker für Einzelblattbetrieb.

Textverarbeitung auf dem Joyce kann jetzt vielseitiger betrieben werden als je zuvor. Überdenken Sie einmal, wie sich diese neuen Möglichkeiten in Ihrem Anwendungsbereich auswirken könnten, nehmen die Tatsache hinzu, daß jetzt auch Desktop Publishing mit dem Joyce möglich ist, und beantworten Sie sich dann unter Berücksichtigung der Gesamtinvestition die Frage, ob der Joyce in der Textverarbeitung zeitgemäß ist. Meine persönliche Antwort ist oben vorweggenommen.

Mit der Textverarbeitung, das haben wir gesehen, ist der Joyce nun nicht im geringsten ausgelastet.

Auch ohne Zusatz-Peripherie kann er mehr. Das weiter oben kurz erwähnte Buchprojekt mag als Beispiel dienen. Es ist ein chronologisches Sachbuch über eine Karriere im Jazz, teils »Lesebuch«, teils Nachschlagewerk. Damit derart chronologisch gegliederte Texte als Datensammlung von Nutzen sein können, sind Indizes nötig der Personen, Auftrittsorte, usw. Bei mir sind's deren etwa ein Dutzend.

Automatische Indexerstellung durch Codierung der Begriffe während der Texteingabe ist nur in Textsystemen möglich, die für den Normalanwender unerschwinglich, für den Joyce nicht denkbar sind. Dennoch kann er diese Arbeit wesentlich erleichtern. Ein Datenbank-Verwaltungssystem macht's möglich.

Ich will dieses Thema nicht unnötig ausbreiten, denn der Systeme gibt es für den Joyce unzählige, zur Abdeckung fast jeden denkbaren Bereiches. Wir machen das mit dBase II. Hier beziehe ich meine beruflichen Mitarbeiterinnen mit ein, die ohne jede tiefere Kenntnis von Datenverwaltung beteiligt sind, indem sie die vom Chef selbstgestrickten Programme auf Lauffähigkeit und Benutzerfreundlichkeit überprüfen und – so alles läuft – schon mal die Dateneingabe besorgen. Dabei stellt sich immer wieder heraus, daß dieses kleine Gerät die derzeit gestellten Aufgaben mit durchaus ansprechender Geschwindigkeit und ohne Mucken zu erledigen in der Lage ist.

Halten wir uns nochmals einige den Joyce betreffende Tatsachen vor Augen:

- Mit dem Betriebssystem CP/M Plus haben wir Zugang zur größten Softwarebibliothek der Welt.
- 16-Bit Computer sind nicht immer schneller, da viele der darauf eingesetzten Programme ursprünglich für 8-Biter geschrieben und gar nicht in der Lage sind die höhere Rechnergeschwindigkeit umzusetzen.

Kein Grund also, den Joyce abzuschreiben. Wenn dennoch die Fa. Schneider hier bei uns unter ständiger Erweiterung ihrer Modellpalette den Joyce nur so nebendran weiterlaufen läßt, kann natürlich schon der Eindruck entstehen, dieser Rechner sei ausgereizt. Nichts im Interview, das der DMV-Verlag anläßlich der CeBIT '87 mit Hanns-Werner Laaks aus dem Hause Schneider führte (JOYCE SONDERHEFT 1, S. 72) deutete darufhin, daß am Joyce mit Neuerungen zu rechnen sei.

Das Gegenteil ist der Fall: Im Herbst wird AMSTRAD in England einen neuen PCW mit den Typenbezeichnungen 9256 und 9512 vorstellen. Markantestes Merkmal wird ein Laufwerk A sein, das mit einer Kapazität von 1 Megabyte neue Horizonte eröffnet. Dazu wird ein Typenraddrucker angeboten werden.

Es hat nicht den Anschein, daß die Fa. Schneider diesen Rechner auf dem deutschen Markt anbieten wird, sowenig wie von dieser Seite anscheinend mit dem LocoScript 2 zu rechnen ist.

Aber da gibt es ja Firmen wie Werder, Hamburg, und Weeske, Backnang, die – wie am Beispiel Harddiscs zu sehen – auch für die Zukunft auf den Joyce setzen. Letztlich wird der DMV-Verlag dem PCW weiterhin mehr als ein flüchtiges Interesse widmen, so daß der Joyce auch in Zukunft die seinen Qualitäten angemessene Unterstützung haben wird.

Verdient hat er sie.

(Dr. K. Stratemann)

varDAT II

Menuegesteuerte Benutzeroberfläche für dBASE II

varDAT II ergänzt beim Joyce die Textverarbeitung fast ideal durch unkomplizierte Datenverwaltung mit einer leistungsfähigen Datenbank und erhöht dadurch die Anwendungsmöglichkeiten enorm.

»..Es ist die optimale Ergänzung zu dBASE II.«
Testbericht in PC Schneider International 9/87

varDAT II erlaubt umfangreiche Anwendungen ohne Programmieren, aber auch Realisierung eigener Ideen.

varDAT II eignet sich universell für den Einsatz im Büro, in der Sachbearbeitung, bei Freiberuflern, beim Studium, in der Lagerhaltung, im Verein...

Selbstverständlich auch mit Festplatte einzusetzen.

Preis: 199,– DM, Handbuch im Schuber
bei Verrechnungsscheck, sonst gegen NN (+ 10,– DM)

Direkt von:

SOFTDESIGN Horstmar Konradt
Bleichstr. 25, 4040 Neuss 1
Tel.: 0 21 01 / 27 61 51

BASIC COMPILER

NEW

Superschneller Real-BASIC-Compiler für alle CPCs und JOYCE

- Professionelles Komplettsystem mit Editor, Compiler und Runtimesystem
- Datentypen Real (7-stellig, E+-38), Integer (-32768..+32767), Cardinal (0..65535), Byte (0..255) und String (Länge 0..255)
- Alle Standardbefehle/Funktionen (z.B. READ/DATA, FOR/NEXT, DIR und SIN(x), MID$, RND) inkl.Fehlerbehandlung (!) und <ESC>/<STOP>-Unterdrückung
- n-dimensionale Arrays
- Beliebig lange Variablennamen möglich
- Strukturierte Programmierung mit REPEAT...UNTIL, WHILE...WEND, IF...ELSE IF..END IF, DO..END DO, LOOP..END LOOP etc.
- Labels statt Zeilennummern, z.B. GOTO menu statt GOTO 3760
- Extrem schnell: etwa 1800 - 1900 Realmult's, 6500 - 7500 Cardinalmult's 330'000 GOTO's; allgemein etwa Faktor 5 - 1000
- Sequentielle und Randomdateien (Random nicht CPC 464 CAS)
- Direkte Grafikunterstützung (auch auf JOYCE!)
- Spezialbefehle, wie z.b. freien Speicher auf Disk ermitteln
- Einbindung von Maschinenroutinen leicht möglich
- Leistungstarker und schneller Editor
- Sehr große Programme (über 2200 Zeilen auf JOYCE/6128, über 800 auf 664/464 Disk, auf 464 CAS etwas weniger) möglich durch Auslagerung des Editors und Compilers auf Disk während des Kompilierens und durch Kompilation auf Disk
- Geeignet für kommerzielle Programmierung
- Genau auf den jeweiligen Rechner abgestimmt
- Demoprogramme werden mitgeliefert (nicht für CPC 464 CAS)
- Auf JOYCE auch Mausunterstützung (Reisware/ISS Maus)

unverbind. Preisempf.: CPC-Version 99,– DM
JOYCE-Version 139,– DM

BIALKE - BERENDSEN - REIMANN
Software
Beimoorweg 2 · D-2070 Ahrensburg
☎ 0 41 02 / 4 39 40

BBG



Marktübersicht

Teil I: Hardware und Zubehör
Teil II: Anwenderprogramme
Teil III: Sprachen
Teil IV: Spiele

Bezeichnung	Preis in DM	Bemerkung	Bezugsquelle
Teil I: Hardware und Zubehör			
AMX Mouse	von 279,– bis 327,80	incl. deutsches Handbuch	3,18
Bildschirmfilter	von 59,– bis 98,–		3,4,13,16,
CPS 8256	139,–	Parall./seriell. Port	29
Druckerverlängerungskabel	59,–		29
Echtzeituhrenmodul	138,–		18
Mouse (Electric-Studio)	448,–	kompl. mit Interface	18
Farbband (2 Stk.)	29,50		34
Farbband	19,90		29
Grafpad 3	549,–	Grafik-Tablett	23
Harddisk 10 MB	1.698,–		18
Harddisk 20 MB	1.998,–		18
Harddisk 20 MB	2.298,–		29
Harddisk 20 MB	2.698,–		16
Joyce Mousepack	249,–	Maus u. Treibersoftware	24
Joystick-Set	99,–	Joystick, Interface und Spiel	56
Joystickadapter	98,–		18
Joystickadapter	158,–	mit 3 Kanal Sound Synthesizer	18
Lightpen	278,–	Freihandzeichnen auf Monitor	18

Bezeichnung	Preis in DM	Bemerkung	Bezugsquelle
Papierführung	37,–		29
RAM-Aufrüstsatz auf 512 kB	von 79,– bis 148,–		35,16,29,33,22, 34,3
PACE MICRO	698,–	5,25"-Zweitlaufwerk mit Konvertierungsprogr. CP/M–MS-DOS	64
Schaltplan Joyce	15,–		18
Haftetiketten endlos für 3"-Disketten 100 St.	9,95		60
Schneider Data SD 15	698,–	Typenraddrucker für Joyce (für DM 59,– Aufpreis komfortable Anpassung für Joyce)	17
Schneider Data SD 24	1298,–	24-Nadel-Drucker für Joyce	17
Star Mouse	168,–	mit Grafiksoftware	18
StaticClear	128,–	gegen elektr. Aufladungen	16
Video-Digitiser	367,–	mit Adapter	18
Zweitfloppy FD2	von 548,– bis 698,–		16,13,19,35,22,3
ID-55	589,–	5,25"-Laufwerk	25
5,25" Laufwerk	499,–		26
ID-35	489,–	3,5"-Laufwerk	25
3,5" Laufwerk	458,–		26

Teil II: Anwenderprogramme

Adresscontrol	79,–	Etikettendruck, Serienbriefe	63
Adressdat	48,–	Adressenverwaltung	71
Adressverwaltung	98,–		12
Adressverwaltung	129,–		7
Agenturverwaltung	620,–	Versicherungspaket	5
ALGO-Handwerk	998,–	Paket für Handwerker	46
ALGO-Haus	998,–	Hausverwaltung	46
ALGO-Kart I	79,–	Dateiverwaltung	46
ALGO-Rhytmus I	149,–	Adressen-Briefverwaltung	46
AMX Pagemaker	207,80	Layoutprogramm mit deutschen Handbuch	18
Appartment/Hotel-Abrechnung	1.298,–		30
Arche	79,–	Dateiverwaltung	26
Astrologie	85,–		4
Auftrag	398,–	Auftragsbearbeitung	12
Auftragsbearbeitung	448,–		3
Bäcker-Software	349	Bäcker-Abrechnung	38
Baufinanz-Control II	279,–	Baufinanzierung	63
Baustelle	600,–		58
Bibliomat	185,–	Buchhandelspaket	5
Biorhytmus	45,–		4
BUSINESS-PACK	199,–	Adressverw., Faktura, Lagerverw.	55
Buchhaltung	577,–	Finanzbuchhaltung	7
Business-Star Plus	298,–	Auftragsbearbeitung	9
Carat-Auftrag	398,–	Auftragsbearbeitung	6
Carat-Kasse Plus	168,–	Kassenverwaltung	6
Carat-Lager Plus	278,–	Lagerverwaltung	8
Carat-Lager Plus	288,–	Lagerverwaltung	26
Carat Provision	148,–	Abrechn. für Handelsvertreter	6
CardBase	148,–	Vereinsverwaltung	6
COMAC-Kasse	128,–	Einnahme-Überschußrechnung	60,61
COMAC-LV	398,–	Leistungsverzeichnis	60,61
ComPack	798,–	Lager, Auftragsbearb., Faktura u. Fibu	37
Cracker	198,–	Tabellenkalkulation	17
Datamat	298,–	Dateiverwaltung	10
Datei Star 2.0	98,–	Dateiverwaltung	9
Datenbank	198,–		7
Datenrem	68,–	Dateiverwaltung	15
dB Faktur	798,–	Fakturierung (nur mit dBase)	39
dB Fibu	798,–	Fibu (nur mit dBase)	39
dB Kunden	398,–	Kundenverw. (nur mit dBase)	39
dB Lager	398,–	Lagerverw. (nur mit dBase)	39
dB Liquidat	798,–	Arzt-Abrechnung (nur mit dBase)	39

Bezeichnung	Preis in DM	Bemerkung	Bezugsquelle
dB Triat	498,–	Triathlonauswert. (nur mit dBase)	39
dBase II	von 179,– bis 199,–	Dateiverwaltung	3,11,26
Dentafox	998,–	Abrechnung f. zahntechn. Labore	2
Dentist	Anfrage	Zahnärzte-Abrechnung	2
Dictionary-Set	99,–		55
D.M.S	128,–	Ergänzung Business-Star	60
DR Draw	von 179,– bis 199,–	Zeichenprogramm	3,26
DR Graph	von 179,– bis 199,–	Diagrammerstellung	3,26
Einnahme-Überschuß-Rechnung	398,–		30
ELOCA	198,–	Kassenverwaltung	41
EASYFAKT	500,–	Fakturierung	47
EASYFIBU	500,–	Finanzbuchhaltung	47
Etatgraph	58,–	Haushaltsbuch mit Grafik	15
Faktura/Lager/Datei	178,–		7
Faktura-Standard	94,–	Fakturierung	63
Faktura-Control II	169,–	Erweiterung von Faktura-Standard	63
Fakturem	98,–	Fakturierung	18
Fakturierung	94,–		33
Fibu-FOX	198,–	Finanzbuchhaltung	2
Fibuking	136,–	Finanzbuchhaltung	15
Fibu Star plus	298,–	Finanzbuchhaltung	9
Finanzbuchhaltung	194,–		33
Finanzbuchhaltung	278,–		12
Finanzbuchhaltung	499,–		42
Finanzmathematik	95,–		7
Fleet Street Editor	von 219,– bis 279,–	Desktop Publishing System	11,35,29
Gehirn-Trust	198,–	elektr. Kurzzeitsgedächtnis	51
Getreide-Abrechnung	398,–	f. Landwirte	60
Handwerk I	998,–	Aufmaß,Kalkulation, Angebote Rechnungen für Handwerksbetriebe	5
Hausverwaltung	570,–	Paket für Hauswirte	7
Holzlager	298,–	Lagerverwaltung	60
Immobilienvermittlung	498,–		48
Immobilienvermittlung	149,–		63
Joyce-Adress	96,–	Adressverwaltung	6
Joyce-Phono-Set	339,–	RS-232 Schnittstelle, Akustikkoppler und Zubehör	29
Joyce-Programmsammlung Vol.1	59,–	Character Designer, Funktionsplotter	21
Joyce-Programmsammlung Vol.2	49,–	univ. Dateiverwaltung	21
Kunden-Faktura-Lager	298,–		54
KORTEX	198,–	Benutzer-Schnittstelle	59
Leistungsstatistik	Anfrage		43
LISAMED	990,–	Liquidation, Statistik und Adressenverw. f. Ärzte	67
Literdat	39,–	Verwaltung von Büchern, Schall platten, Videos	71
Loco 15	59,–	ASCII-Umwandlung für SD 15	17
Loco 24	Anfrage	ASCII-Umwandlung für SD 24	17
LocoMail	128,–	Serienbriefe mit LocoScript	37
LocoPost	84,–	Serienbriefe, Adressverwaltung	71
LocoScript-Einführung	29,50	Diskette u. Buch	37
MERCATOR	399,–	Finanzbuchaltung	40
MICA	198,–	CAD	1
Multi-Database & Toolkit	49,–	Dateiverwaltung	31
Multiplan	von 179,– bis 199,–	Tabellenkalkulation	3,11,26
NewsDesk International	178,–	Desktop Publishing-System	16
Orgafibu	274,–	Fibu 50 Konten	50
Orgafibu	297,–	Fibu 100 Konten	50
Orgafakt	548,–	Fakt./Adressverw./Mahnw.	50
Polyfont	75,–	Schriftmuster für Polyprint	45
Polymail	175,–	Serienbriefe/Adressverw. für Polyprint	45
Polyplot	175,–	Geschäftsgraphiken	45

Bezeichnung	Preis in DM	Bemerkung	Bezugsquelle
Polyprint	175,–	Textverarbeitung	45
Praktische Textverarbeitung	89,–	Ergänzung Locoscript	21
Privatliquidation	Anfrage	Arzt-Abrechnung	43
Profirem	178,–	Faktura, Lager u. Adressen	15
Prompt	69,–	Dateiverwaltung	18
Prompt-Druck	39,–	Ergänzung Locoscript	18
PROWORT	von 239,– bis 249,–	Textverarbeitung	69,66
Public Domain-Software	je 30,–		62
Quick-Bill 2.0	168,–	Angebotsrechnung	16
Quick-Calc 2.1	148,–	Finanzbuchhaltung	16
Quick-Cine 2.0	98,–	Videoarchiv	16
Quick-Data 3.0	228,–	GSX-Geschäftssoftware	16
Quick-Data-WEB	498,–	GSX-Geschäftssoftware für 20 MB Festplatte WEB	16
Quick-Mahn 3.0	68,–	Mahn-Modul	16
Quick-Name 2.0	128,–	Adressverwaltung	16
Quick-Stat 3.0	78,–	Statistik-Modul	16
Quick-Zahl 3.0	119,–	Zahlkarten-Modul	16
RH-Dat	89,–	Adressverwaltung	1
Schneider-Soft	1990,–	Schreiner-Abrechnung	2
Standardleistungsbuch für Bauwesen	Anfrage		13
Star Base	198,–	Dateiverwaltung	9
Star Mail 2.0	98,–	Ergänzung Locoscript	9
Statistikprogramm	169,–	Regressions-, Varianz- und Faktoranalyse	71
SuperCalc2	198,–	Kalkulationsprogramm (engl. Anleitung)	37
Super Copy	89,–	Kopierprogramm	18
Supertype	128,–	Schriftarten für Locoscript	16
Tas	198,–	Rel. Datenbank/Programmiersprache mit Compiler	65
Tasword 8000	von 78,– bis 148,–	Textverarbeitung	18,57



Die neuen Herbstprogramme sind da!

COMAC-LITBOX 3.0 — professionelle Karteikartenverwaltung . . DM 98.—
- individuelle Gestaltung jeder Karteikarte
- maximal 540 (!) Zeichen pro Karteikarte
- Anlage einer speziellen Stichwörterkartei
- komfortable Suchroutinen (max. 9 Suchwörter gleichzeitig)
- umfangreiche Sortiermöglichkeiten
- mehrere Druckoptionen usw.usw.

COMAC-LIQUI — Privat-Liquidation für Ärzte . . . . . DM 348.—
- beliebige Eingabe der GOÄ-Daten
- schneller Zugriff auf Rechnungsdateien
- selbstrechnendes Rechnungsformular
- individuelle Steigerungssätze
- Laden, Speichern und Vernichten von Rechnungen
- einfachste Bedienung durch Hilfszeilen usw.usw.

COMAC-D.M.S. — Direktwerbung/Serienbriefsystem . . . . . DM 128.—
- schnelle Eingabe der Kundenadressen
- individuelle Brieftext-Erstellung
- komfortable Suchmöglichkeiten
- selektives Verknüpfen von Adressen mit Brieftexten
- Verwendung von Formularen
- datenkompatibel zu BUSINESS-STAR usw.usw.

unsere Erfolgsprogramme:

COMAC-KASSE— Einnahmen-Überschußrechnung . . . . . DM 128.—
- Festlegung beliebig vieler Konten
- nachträgliche Änderung von Einzelbuchungen
- Umsatzsteuervoranmeldung usw.usw.

COMAC-LV — Leistungsverzeichnis f. Architekten/Ingenieure . . . . . DM 398.—
- frei erstellbare Leistungstexte
- individuelle Anlage der einzelnen Positionen
- nach Erstellung noch Korrekturmöglichkeiten usw.usw.

Alle Programme werden mit einem umfangreichen Handbuch geliefert.

Verlangen Sie unseren Sonderprospekt JOYCE-Software!

CMZ-Verlag Winrich C.-W. Clasen
Borgswiese 9-11, 4650 Gelsenkirchen 2
Telef. Bestellannahme rund um die Uhr: 0209 - 777896

Bezeichnung	Preis in DM	Bemerkung	Bezugsquelle
Turbo-Adress	von 149,– bis 169,–	Adressverwaltung	18,1
Turbo Faktum	199,–	Faktura	18
Universelles Büro	199,–	Büroverwaltung	52
varDAT II	198,–	Benutzeroberfläche für dBase II	53
Vereinsverwaltung	198,–	nur Joyce+	1
Vereinsverwaltung	248,–		3
Vereinsverwaltung	249,–		19
Verbentrainer	59,–		18
Videoarchivverwaltung	98,–		7
Vokabeltrainer	59,–		26
Vokabeltrainer	69,–		18
Vokabi	58,–	Vokabeltrainer	15
Wärmebedarfsrechnung	248,–		7
Wärmebedarfsrechnung	298,–		8
Window Adresse	98,–	Adressverwaltung	6
Window Kasse	98,–	Kassenverwaltung	6
Window Lager	148,–	Lagerverwaltung	6
Wordstar	von 149,– bis 199,–	Textverarbeitung	26,11,3
Teil III: Sprachen			
CBasic Compiler	von 139,– bis 174,–		33,11,6
COM	139,–	Kommunikationsprogramm	49
COMAL-80	148,–		36
FORTH	139,–	Forth-Compiler	49
Mac Lin	350,–	Programmkerngenerator	22
MACRO	139,–	Makroassembler	49
Maxam II	239,–	Assembler/Disassmbler/Monitor	69
MI-C	445,–	mehr. Versionen	14
Microsoft-Basic	98,–		30
Nevada Cobol	von 109,– bis 168,–		19,35
Nevada-Cobol-Compiler	135,–		19
Nevada Fortran	von 109,– bis 168,–		19,35
Nevada-Fortran-Compiler	135,–		19
Oxford Pascal	89,–		20
PROFI	139,–	Assemlerprogrammsammlung	49
Small C	99,–	Entwicklungssystem	33
Turbo-Lader Business	148,–		33
Turbo-Lader Grundpaket	138,–		33
Turbo-Lader Science	189,–		33
Turbo Pascal	von 199,– bis 225,–	ohne Grafik	3,11,35
Turbo Pascal	von 279,– bis 285,–	mit Grafik	11,18
Turbo-Toolbox	125,–		18
Turbo-Toolbox	225,–		33
Turbo-Tutor	104,–	englisch/deutsch	33
XBC	139,–	Basic Compiler	68
Teil IV: Spiele			
ACE	59,90	Flugsimulator	56
Aftershock	69,90	Adventure	18
Annals of Rome	69,90		32
Batman	von 39,– bis 59,–	Action-Adventure	18,32
Bounder	39,90		32
Colossus Chess 4.0	49,90	Schachprogramm	11
Cyrus Chess	von 49,90 bis 69,–	Schachprogramm	18,32
Cyrus II-3D	128,–	Schachproramm	37
3D Clock Chess	von 49,– bis 79,90	Schachprogramm	3,18,20

Bezeichnung	Preis in DM	Bemerkung	Bezugsquelle
Fairlight	59,–	Action-Adventure	18
Football Fortunes	59,90	Strategiespiel	56
Fourth protocol	49,90	Action-Spiel	20
Frank Bruno's Boxing	56,90	Action-Spiel	56
Hitchhikers	99,–	Textadventure	18
Infocom-Adventures	je 79,90		56
Jewels of Darkness	49,90	Adventure	20
Leader Board (Golf)	59,90	Simulation	56
Lord of the Rings	99,–	Adventure	18
Lotto 6 aus 49	59,–		4
Pawn	69,90		32
Ping Pong	149,90		55
S.A.D.Raid	59,–	Action	18
Space invaders	49,90	Action	32
Starglider	von 69,90 bis 79,90	Weltraumspiel mit Vektorgrafik	11,20
Strike Force Harrier	von 54,90 bis 69,90		20,32
Super-Lotto RUN 49	99,–		31
Tomahawk	59,90	Hubschraubersimulation	11
Trivial Pursuit	59,90	Frage/Antwortspiel	56

Adressen für Joyce Soft- und Hardware

1. GfA Systemtechnik	Heerdter Sandberg 30	4000 Düsseldorf 11
2. M. Kempenich	Zetastr. 13	6220 Rüdesheim
3. Schauties	Bachstr. 52	7980 Ravensburg
4. mükra	Schöneberger Str. 5	1000 Berlin 42
5. KMS-Soft Krautwald	Julius-Leber-Straße 28	2000 Hamburg 50
6. Micro Market Worms	Joh.-Westermann-Platz 1	4780 Lippstadt
7. Computer Studio	Am Rätschenbach 9	8058 Erding
8. WHS-Soft	Hohenzollernstr. 9	7447 Aichtal
9. Star Division	Uelzener Str. 12	2120 Lüneburg
10. Data Becker	Merowinger Str. 30	4000 Düsseldorf 1
11. Waldeck	Tulpnstr. 30	2870 Delmenhorst
12. Byte me	Wilhelmstr. 7	5240 Betzdorf
13. G+W	Hartmeyerstr. 50	7400 Tübingen
14. H. Rose EDV	Bogenstr. 32	4390 Gladbeck
15. Van der Zalm	Schieferstätte	2949 Wangerland
16. Werder	Bramfelder Ch. 215	2000 Hamburg 71
17. Schneider Data	Rinder Markt 8	8050 Freising
18. Weeske	Potsdamer Ring	7150 Backnang
19. Computer Max	Postfach 1461	7920 Heidenheim
20. M. Naujoks	Rottmannstr. 40	6900 Heidelberg
21. DMV-Verlag	Fuldaer Str. 6	3440 Eschwege
22. SFK	Delsterner Str. 23	5800 Hagen
23. PiZie Data	Mittelstr. 61	4322 Sprockhövel
24. Gerdes	Heidegartenstr. 36	5300 Bonn 1
25. Innowave Data	Kurt-Schuhmacher Str. 26	3000 Hannover
26. TG-Soft	Offersdorf 5	8491 Rimbach
27. Boll Computer- und Software-Vertrieb	Wilhelmsthaler Str. 26 B	3501 Ahnatal-Weimar
28. Syndrom Computer GmbH	Edwaldstr. 181	4352 Herten
29. Vortex-Versand	Falterstr.	7101 Flein
30. ESCHCOMP-SYSTEM	Ernst Scheithauer Postfach	8391 Pelesrent
31. Interlogik-Peter Herzog	Grabenweg 30	8240 Schönau am Königssee
32. Waldeck-Software	Tulpenstr. 30	2870 Delmenhorst
33. R. Schuster Electronic	Obere Münsterstr. 33-35	4620 Castrop-Rauxel
34. ABD Computer	Zettachring 12	7000 Stuttgart 80
35. G Computerstore	Hochstr. 11	8500 Nürnberg
36. COMAL-Vertrieb	Derek Belz	2270 Utersum/Föhr
37. Schneider AG	Silvastraße 1	8939 Türkheim
38. ABS COMPUTERLAND	Huglfinger Str.4	8121 Polling
39. Computerladen Braun	Kreuzstr. 9	8940 Memmingen
40. Computer Studio Offenbach	Kl. Biergrund 17	6050 Offenbach
41. Computer Team Cuxhaven	Am Hauptbahnhof	2190 Cuxhaven
42. DAST-Mobil	Saarbrückerstr. 56	4670 Lünen
43. Effekta Chr. Walls	Am Wiggert 9c	4500 Osnabrück
44. Franzis-Software	Karlstr. 37-41	8000 München
45. " " " "		
46. A.H.W. Goscg oHG	Grindelallee 138	2000 Hamburg 13
47. Gruse Elektronik	Stieglitzweg 1	3300 Braunschweig
48. Hailer & Gensberger	Albstr. 18	7317 Wendlingen
49. Holtkötter	Albert-Schweitzer-Ring 15	2000 Hamburg 70
50. ORGASOFT	Graneggstraße 43	7732 Niedereschach
51. Richter's	Thürmchenswall 3	5000 Köln
52. Schmidt-Soft	Blumenstr. 15	5600 Wuppertal 1
53. SOFTDESIGN	Bleichstr. 25	4040 Neuss
54. Vertretung der Industrie	Brachvogelweg 9	8037 Olching
55. ZS-Soft	Postfach 2361	8240 Berchtesgaden
56. Sunshine Software	Auf dem Schilde 14	5270 Gummersbach
57. Husoft	Halde 15	7463 Rosenfeld
59. Gerald Keil	Obere Hochstraße 136	6652 Bexbach/Saa
60. Büro für Software Entwicklung	Sonnenstr. 43	5270 Gummersbach
61. CMZ-Verlag Winrich C.-W. Clasen	Borgswiese 9-11	4650 Gelsenkirchen 2
62. Martin Kotulla	Grabbestraße 9	8500 Nürnberg 90
63. Hashagen-EDV	Eckstraße 11	6113 Babenhausen
64. Ralf Hille Datentechnik	Im Lübbering 19	4322 Sprockhövel 2
65. Wolfgang Henschke Datentechnik	Aidlinger Weg 6	7034 Gärtringen
66. Arnor (Deutschland)	Hans-Henny-Jahnn-Weg	2000 Hamburg 76
67. LISAMED P. u. S. Bosch	Auf dem Oelsfeld 24	5300 Bonn/Bad Godesberg
68. BBG Software	Beimoorweg 2-4	2070 Ahrensburg
69. Pr8 Soft Klaus-M. Pracht	Postfach 500	8702 Margetshöchheim
70. Heschcom	Weichselgartenstr. 13	8000 München 71
71. K.Heil & E.Kaefer	Marburger Str. 7	6000 Frankfurt 90

Logische Operationen mit JOYCE

Alltag und Logik

Folgende Situation: Ein Lehrer steht vor einer gemischten Klasse und sagt: »Alle Jungen und Mädchen stehen jetzt auf«! Was passiert? Alle Schüler, ob Junge oder Mädchen werden aufstehen. Was daran so besonderes ist? Es ist logisch falsch. Kein Schüler hätte aufstehen dürfen. Denn niemand kann zugleich männlichen und weiblichen Geschlechts sein. Das ist eben der Unterschied zwischen Umgangssprache und logischer Sprache.

Verknüpfungsoperatoren

Die meisten Programmiersprachen kennen Verknüpfungsoperatoren wie AND, OR und NOT. Verknüpft man zwei Aussagen (Variablen, Elemente) miteinander, so erhält man eine wahre oder falsche Aussage. Es handelt sich um eine zweiwertige Boolesche Algebra, die auch Aussagen-Algebra genannt wird.

Aussagenlogik

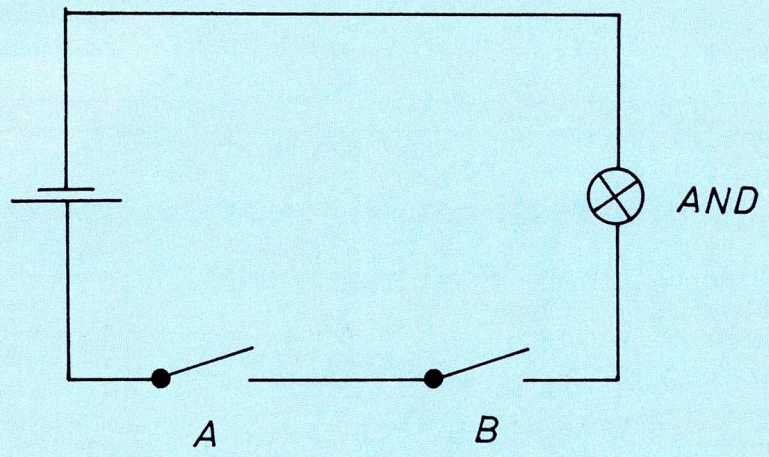
Die Aussagenalgebra wird auch als Aussagenlogik, formale Logik oder Junktorenalgebra bezeichnet. Sie wurde begründet von dem griechischen Gelehrten ARISTOTELES (* 384 v.u.Z. 322 v.u.Z.).

Unter einer Aussage wird eine sprachliche oder mathematische Formulierung verstanden, die als falsch oder wahr bewertet werden kann. Bei Computern findet man häufig T für TRUE (wahr) und F für FALSE (falsch). So ist die Aussage »Madrid ist die Hauptstadt von Schweden« falsch. Denn Madrid ist die Hauptstadt von Spanien (die von Schweden ist übrigens Stockholm). Die Aussage »2 + 2 = 4« ist nach Adam Riese (und Eva Zwerg) wahr. Beide Seiten der Gleichung haben denselben Wahrheitsgehalt. In der Logik gibt es nur die zwei Wahrheitswerte T oder F. Ein bißchen richtig oder ein bißchen falsch gibt es nicht. Genausowenig wie ein »bißchen schwanger« im Alltag. Solche Zweiwertigkeit liegt nicht bei allen Aussagen vor. Beispielsweise ist die Aussage »Dieser Satz, den Sie hier lesen, ist falsch« nicht zweiwertig.

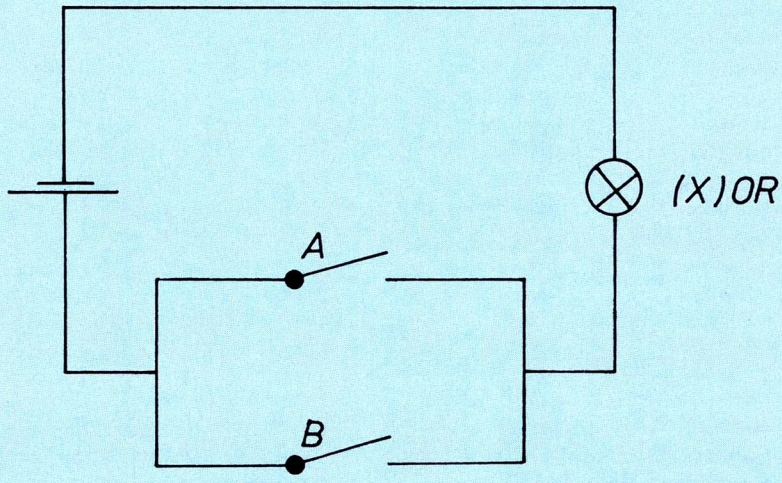
AND und OR verknüpfen zwei Aussagen miteinander. NOT verneint eine Aussage. Übersicht 1 stellt die Verknüpfungen dar, wie sie auf JOYCE unter DR.LOGO oder dBaseII möglich sind.

Natürlich auch mit anderen Programmiersprachen wie beispielsweise Pascal.

LOGISCHE VERKNÜPFUNGEN und, oder, entweder-oder AN STROMKREIS-BEISPIELEN



Die UND-Verknüpfung: Nur wenn die Schalter A UND B geschlossen sind, brennt die Lampe



Die ODER-Verknüpfung: Wenn einer der beiden Schalter A ODER B oder beide Schalter geschlossen sind, brennt die Lampe

EXKLUSIVES-ODER: Wenn ENTWEDER der Schalter A ODER der Schalter B geschlossen ist, brennt die Lampe

Die AND-Verknüpfung anhand eines Beispiels aus dem Alltag eines Computer-Freaks auf der Suche nach Nahrung ergibt folgende Übersicht:

Situation		
A	**B**	**A AND B**
hungrig?	**Restaurant vorhanden?**	**Handlung?**
ja	**ja**	**essen gehen!**
nein	**ja**	**braucht nichts essen**
ja	**nein**	**kann nichts essen**
nein	**nein**	**ist eh' egal**

Programmierlogik

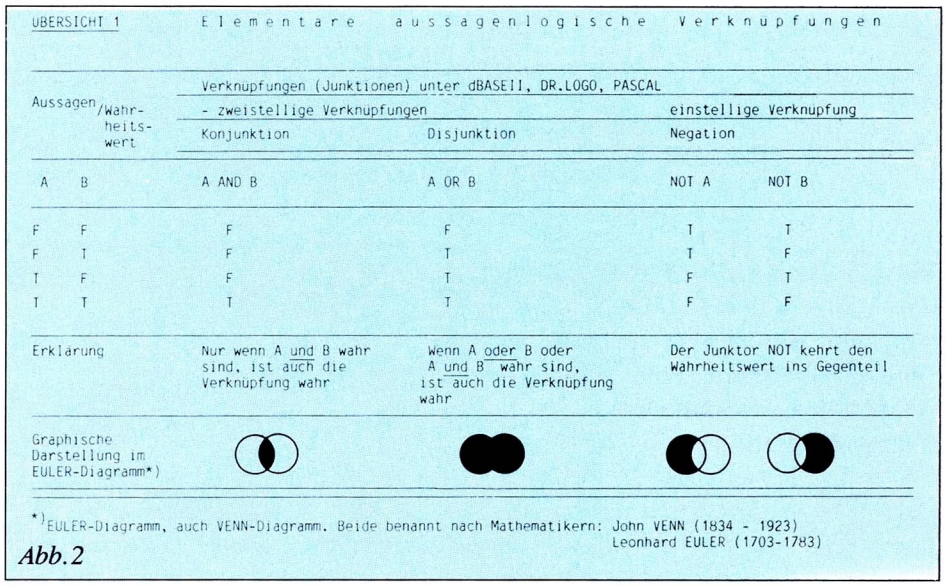
Sehen wir uns die Verhältnisse in einem Programm an: Bei einer Abfrage in einem Menü soll ein 'j' und ein 'J' für »Ja«, ein 'n' und ein 'N' für »Nein« akzeptiert werden. Löst die dBase-Kommandodatei logik#1.cmd das Problem? Nein. Denn die auf der Tastatur einzugebende Antwort kann nicht zugleich ein 'j' UND ein 'J' sein, wie es das Programm mit .AND. fordert. In den CASE-Zeilen ist der Verknüpfungsoperator .OR. einzusetzen! Probieren Sie es aus!

Mit dem Beispiel der »Zwitter« in der Schulklasse und diesen beiden Programmbeispielen dürfte der Unterschied zwischen der UND und der ODER-Verknüpfung deutlich geworden sein. Die Stromkreisbeispiele in der Zeichnung verdeutlichen dies.

Die NICHT-Verknüpfung kehrt den Wahrheitswert einer Aussage in ihr Gegenteil. Ist A TRUE, so ist NOT A logisch FALSE, hat A den Wahrheitsgehalt FALSE ist NOT A folglich TRUE. Aus dem Alltag: Ein nicht schlechter Schüler, ist ein guter Schüler. Ein nicht schönes Mädchen ist logischerweise ein Häßliches.

Unlogisches MALLARD-Basic?

Das Basic von JOYCE kennt noch mächtigere Verknüpfungsoperatoren als AND, OR und NOT. Übersicht 2 stellt sie dar. Sie sind aber nicht vom Typ boolean. Das heißt, sie ergeben nicht die Werte TRUE oder FALSE.



ÜBERSICHT 1 Elementare aussagenlogische Verknüpfungen

Aussagen	Wahrheitswert	Verknüpfungen (Junktionen) unter dBASEII, DR.LOGO, PASCAL			
		- zweistellige Verknüpfungen		einstellige Verknüpfung	
		Konjunktion	Disjunktion	Negation	
A	B	A AND B	A OR B	NOT A	NOT B
F	F	F	F	T	T
F	T	F	T	T	F
T	F	F	T	F	T
T	T	T	T	F	F
Erklärung		Nur wenn A und B wahr sind, ist auch die Verknüpfung wahr	Wenn A oder B oder A und B wahr sind, ist auch die Verknüpfung wahr	Der Junktor NOT kehrt den Wahrheitswert ins Gegenteil	
Graphische Darstellung im EULER-Diagramm*)					

*) EULER-Diagramm, auch VENN-Diagramm. Beide benannt nach Mathematikern: John VENN (1834 - 1923) Leonhard EULER (1703-1783)

Abb. 2

Sind Sie deshalb nicht logisch? Keineswegs: Vielmehr wird die Verknüpfung bitweise überprüft: Es werden jeweils die an der gleichen Stelle stehenden Bits der dualen Darstellung der Aussage entsprechend der verlangten Verknüpfung verglichen.

AND im 8bit-Dualcode ergibt als Verknüpfungsergebnis 0, falls nicht beide Bits 1 sind:

```
A          1001  1101
B          1110  0011
---------------------
A AND B    1000  0001
```

Erläuterungen zu den anderen Verknüpfungsoperatoren im JOYCE Handbuch 2 (in der Ausgabe von 1985 auf Seite 113).

Wozu kann man nun diese bitweise Verknüpfung benutzen? Geben Sie unter MALLARD-Basic einmal ein: PRINT (200 AND 60). Das Ergebnis ist 8, denn:

```
200      11001000
60       00111100
---------------
         00001000
```

Mit der AND-Verknüpfung können in einem Byte Bits gezielt gelöscht werden. Umgekehrt sind mit der OR Verknüpfung gezielte Bits in einem Byte zu setzen. Mit entsprechenden Masken-Zahlen sind so schöne Sprite-Programmierungen möglich. Beim JOYCE sind diese Grafikfähigkeiten noch kaum getestet, geschweige denn ausgeschöpft worden.

Das Programm LOGIK#1.BAS fragt, ob Sie logisch denken. Es akzeptiert die Eingabe von j, J, n und N als Antwort. Das Programm LOGIK#2.BAS fragt ebenfalls, ob Sie logisch denken. Geben sie etwas anderes als den erlaubten Wertevorrat {j, J, n, N} ein, so »hängt sich das Programm auf«. Sicherlich wissen sie als Logiker, warum! Übrigens: Mit der (STOP)-Taste kommen Sie zurück zu BASIC.

Im Programm LOGIK#1.BAS können sie in den Zeilen 40 und 50 oder in 80 und 90 den Verknüpfungsoperator OR auch durch XOR ersetzen. XOR ist das sogenannte ENTWEDER-ODER (EXCLUSIVE-OR). Im Unterschied zum OR ist das Verknüpfungsergebnis nur dann 1, wenn nur eines der beiden verknüpften Elemente 1 ist, nicht wenn beide gleich sind.

Also: ENTWEDER »Junge« ODER »Mädchen« hätte der Schulmeister richtig sagen müssen. Wenn ENTWEDER A ODER B wahr sind, ist C (A X OR B) wahr; sind A und B beide wahr, so ist C (A XOR B) falsch. Traurig und fröhlich zugleich geht logisch nicht.

Folgernde Verknüpfungen

Die Implikation IMP (=>) ist eine WENN-DANN-Verknüpfung. Sie ist ein Sonderfall der Subjunktion (A -> B). Sie läßt sich in anderen Programmiersprachen mit (NOT A OR B) simulieren. Folgende Tabelle zeigt dies:

AB	NOT A	(NOT A OR B)	(A=>B)	AB	A IMP B
FF	T	T	T	10	1
FT	T	T	T	01	0
TF	F	F	F	11	0
TT	F	T	T	00	0

Die Implikation IMP (=>) ist eine stets wahre Subjunktion (->). Man sagt auch: Sie ist eine tautologische Subjunktion (griechisch tautologia = Wiederholung von etwas Gesagtem mit anderen Worten).

Tautologien in der Sprache sind beispielsweise der »kastrierte Wallach«, der »weiße Schimmel« und der »schwarze Neger«. Aus A folgt B, auch: A impliziert B.

Übrigens: Eine stets falsche Aussageform heißt Kontradiktion: [A AND NOT(B OR A)].

Die Äquivalenz EQV (<=>) ist eine stets wahre, tautologische Bijunktion (<->). A <=> B ist gleichwertig mit der Aussage [(NOT A AND NOT B) OR (A AND B)]. Aus A folgt B und umgekehrt, auch: B ist genau dann wahr, wenn A wahr ist. »Ein logischer Mensch ist ein Logiker«. Mathematische Beispiele: (x^2-1=0 <=> x = 1); [x^2-1=0 <=> (x=1) v (x=-1)]. Verknüpfungsoperatoren sind stets in Verbindung mit den Vergleichsoperatoren zu sehen.

Folgende Tabelle stellt die wichtigsten Vergleichsoperatoren dar:

ÜBERSICHT 2 Aussagenlogische Verknüpfungen unter MALLARD-BASIC auf JOYCE

Aussagen/Wahrheitswert	Verknüpfungen (Junktionen) unter MALLARD-BASIC: Antivalenz >-<	Äquivalenz <==>	Implikation ==>
A B	A XOR B	A EQV B	A IMP B
0 0	0	0	0
0 1	1	1	0
1 0	1	1	1
1 1	0	0	0
Erklärung	Nur, wenn entweder A oder B wahr ist, ist auch die Verknüpfung wahr. (Das Ergebnis ist 1, falls nicht beide bits gleich sind)	Aus A folgt B und umgekehrt (Das Ergebnis ist 1, falls das rechte bit nicht 1 und das linke bit nicht 0 ist)	Aus A folgt B (Das Ergebnis ist 1, falls das rechte bit nicht 1 und das linke bit nicht 0 ist)
Graphische Darstellung im EULER/VENN-Diagramm *)			

*) zur Namensgebung der Diagramme siehe Anmerkung zu ÜBERSICHT 1.

Abb. 3

Operator	DR. LOGO	dBaseII	MALLARD BASIC
=	gleich	gleich	gleich
<	kleiner	kleiner	kleiner
<>		ungleich	ungleich
#		ungleich	
<=		kleiner gleich	kleiner gleich
>=		größer gleich	größer gleich

(MALLARD BASIC akzeptiert auch: => oder =<)

Besonders bei der Abfrage in Datenbank-Abfrage- und Programmiersprachen, wie sie z.B. dBaseII auf dem JOYCE bietet, wird deren Bedeutung deutlich: LIST FOR Umsatz einer entsprechenden Quelldatei listet alle Datensätze aus, deren numerisches Feld »Umsatz« größer als Hundert (DM) und deren logisches Feld Stammkunde den Wert TRUE hat. Wenn Sie aus einer Adressendatei die Adressen des Postleitzahlbereichs 3 und 4 suchen, müssen sie in dBase wie folgt verknüpfen: LIST FOR PLZ = 3 .OR. PLZ = 4. Mit LIST FOR PLZ = 3 .AND. PLZ = 4 bekommen Sie garnichts geLISTet, weil ein Wohnort nicht zugleich dem Postleitzahlbereich 3 und 4 angehören kann. Genauso logisch »denkt« auch MALLARD-Basic, d.h. der Interpreter erwartet, daß Sie als Programmierer so logisch denken. Tut er's nicht, so ist er entweder kaputt oder Sie sollten JOYCE ihrem Händler zurückgeben. (G.Ott)

```
LISTING >LOGIK£1<, REMARK = >REM<.

<89> 10 REM "LOGIK£1"
< 0> 20 PRINT CHR$(27)+"E"+CHR$(27)+"H"
<64> 30 INPUT "Denken Sie logisch - Ja oder Nein ein (J
     /N)"; Antwort$
<10> 40 IF Antwort$="j" OR Antwort$="J" THEN PRINT "Sie
      gaben ein ";Antwort$;" f|r >Ja<"
<52> 50 IF Antwort$="n" OR Antwort$="N" THEN PRINT "Sie
      gaben ein ";Antwort$;" f|r >Nein<"
< 7> 60 IF Antwort$<>"j" AND Antwort$<>"J" AND Antwort$
     <>"n" AND Antwort$<>"N"..................... THEN PR
     INT "Falsche Eingabe: Nicht aus dem Wertevorrat!"
<48> 70 INPUT "Weiterer Programmlauf? - Ja oder Nein (J
     /N)"; Antwort$
<49> 80 IF Antwort$="J" OR Antwort$="j" THEN GOTO 20
<28> 90 IF Antwort$="N" OR Antwort$="n" THEN GOTO 120
<62> 100 IF Antwort$<>"j" AND Antwort$<>"J" AND Antwort
     $<>"n" AND Antwort$<>"N"..................... THEN P
     RINT "Falsche Eingabe: Nicht aus dem Wertevorrat!"
<72> 110 GOTO 70
<27> 120 END
```

```
LISTING >LOGIK£2<, REMARK = >REM<.

<89> 10 REM "LOGIK£2"
<83> 20 cls$=CHR$(27)+"E"+CHR$(27)+"H"
<56> 30 INPUT "Denken Sie logisch - Ja oder Nein"; Antw
     ort$
<87> 40 IF Antwort$<>"j" AND Antwort$<>"n" AND Antwort$
     <>"J" AND Antwort$<>"N"..................... THEN GO
     TO 40
<90> 50 IF Antwort$="j" OR Antwort$="J" THEN PRINT cls$
     ;"Angeber!"
<34> 60 IF Antwort$="n" OR Antwort$="N" THEN PRINT cls$
     ;"Hier lernen Sie es!"
<10> 70 GOTO 30
<35> 80 END
```

```
*** logik#1.cmd

DO WHILE T
ACCEPT 'Denken Sie logisch? - Ja oder Nein (J/N)' TO Antwort
ERASE

      DO CASE
         CASE Antwort='j' .OR. Antwort='J'
              @ 0,0 SAY 'Angeber'
         CASE Antwort='n' .OR. Antwort='N'
              @ 0,0 SAY 'Hier lernen Sie es!'
         OTHERWISE
              @ 0,0 SAY 'Die Eingabe stammt nicht aus dem Wertevorrat!'

      ENDCASE
ENDDO
```

DMV präsentiert:

Joyce-Programmsammlung Vol. 1

***Hochwertige Software** zu **Niedrigpreisen** finden Joyce-Anwender jetzt im Rahmen einer Programmsammlung in der Angebotspalette des DMV-Verlages.*

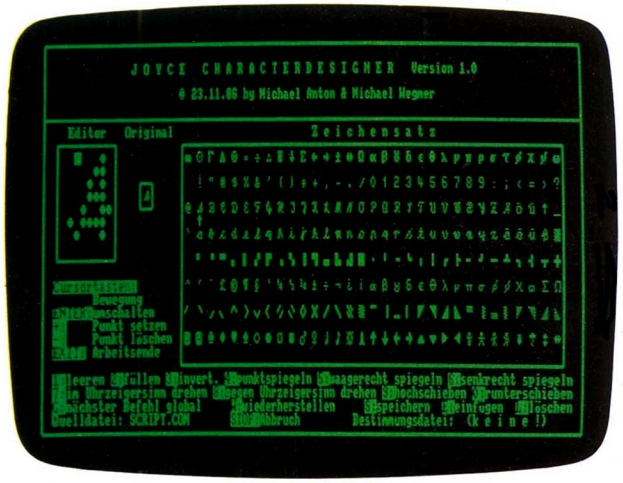
Jede Ausgabe aus dieser Reihe enthält eines oder zwei Programme, die aus verschiedenen Anwendungsgebieten kommen. Diese Serie erscheint in unregelmäßiger Reihenfolge und wird als komplettes Programmpaket mit 3"-Diskette und Bedienungsanleitung ausgeliefert.

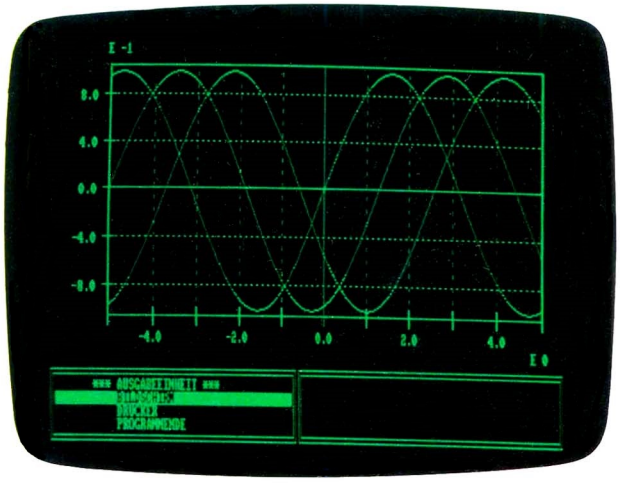
CHARACTER DESIGNER – DAS Werkzeug zur Erstellung eigener Zeichensätze und MGX-Mathematical Graphics Extension- ein Funktionenplotter mit außergewöhnlichen Leistungsmerkmalen.

1. Der Character Designer

Funktion: Komfortable Erstellung eigener Zeichensätze auf PCW 8256/8512 und deren Darstellung am Bildschirm! Ausdruck von ASCII-Files in diesem Zeichensatz unter CP/M Plus

Leistungsumfang: CD.COM ist der Character-Designer, der Editierung oder Neuerstellung von Zeichensätzen und deren Speicherung erlaubt. Zeichensätze können als COM-File abgespeichert werden, d.h. auch unter CP/M aufgerufen und somit aktiviert werden. **CD-PRINT** druckt vorformatierte ASCII-Texte auf dem Joyce-Drucker oder anderen Druckern in dem gewünschten Zeichensatz aus.
CRAZY, ORIGINAL, LOCOCHAR und **SCRIPT** sind mitgelieferte Zeichensätze. **SCRIPT** ist ein Schreibschrift-Zeichensatz und wurde freundlicherweise von der Firma Gerdes Imperial Software zur Verfügung gestellt. **SETUP.COM** erlaubt als Zugabe die Vorwahl einiger Systemparameter, z.B. die der seriellen Schnittstelle, der Tastaturgeschwindigkeit und der Floppy-Steprate.





2. MGX

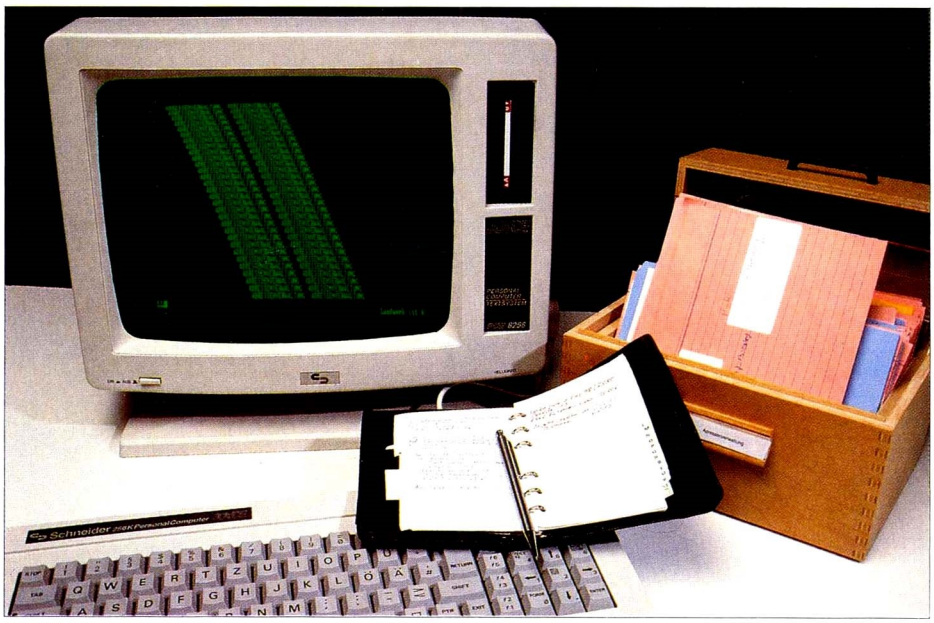
Funktion: Graphische Darstellung von mathematischen Funktionen und beliebigen Meßreihen auf Bildschirm oder im Großformat auf dem Drucker

Leistungsumfang: Neben den arithmetischen Grundfunktionen sind auch weitere Funktionen darstellbar, die z.B. unter Mallard-Basic nicht zur Verfügung gestellt werden. Es können mehrere Funktionen und Meßreihen (diese wiederum mit mehreren Meßwerten gleichzeitig dargestellt werden). Die X-undYBereiche können manuell oder automatisch bestimmt werden, ebenso ist die Darstellung der Graphen mit **X/Y-Gitter, X oder Y-Gitter** oder ohne Gitter möglich. Es stehen **drei Auflösungsstufen** zur Verfügung, wobei mit Erhöhung der Auflösung auch die Arbeitsgeschwindigkeit abnimmt. Letztendlich kann noch eine **beliebige Überschrift** in die Funktion/Meßreihe eingefügt werden.
Bei der Druckausgabe steht als Standard das **A-4-Format** zur Verfügung; dieses ist jedoch in weiten Grenzen frei variierbar.

Die JOYCE Programmsammlung VOL.I ist beim Verlag zum Preis von DM 59,– erhältlich.
Coupon auf Seite 115 ausfüllen und senden an:

DMV-Verlag · Redaktion PC International
Fuldaer Str. 6 · 3440 Eschwege · Postfach 250

Das hier vorgestellte Programm nützt die großartigen Möglichkeiten der Jetsam-Version des Mallard-Basic aus. Der »kleine« Joyce entwickelt damit ungeahnte Möglichkeiten. Im Vergleich zu Programmen, die sequentielle Dateien zur Speicherung von Informationen benutzen, ist die Verarbeitung in dieser Version nicht durch die begrenzte Kapazität des Arbeitsspeichers behindert. Man hat einen direkten Diskettenzugriff, wodurch die gesamte Diskettenkapazität zur Verfügung steht. Ein weiterer Vorteil ist, daß man an seinen gespeicherten Daten Veränderungen vornehmen kann. Das Faszinierende aber ist der Zugriff auf Daten über Schlüssel.



Personal- und Adreßverwaltung

realisiert mit JETSAM

In diesem Programm werden persönliche Daten (Namen und Geburtsdaten) und Adressen erfaßt. Die Familiennamen, der Geburtsmonat, das Geburtsjahr, die ersten Ziffern der Postleitzahl und der Wohnort werden zu Jetsam-Schlüsseln definiert.

Jetsam-Schlüssel sind Begriffe, die in einer gesonderten Datei (Index-Datei) gesammelt werden und Zugriffsmöglichkeiten auf die Nummern der Sätze in einer Direktzugriffsdatei darstellen. Damit besteht die Möglichkeit, Informationen unter verschiedenen Stichwörtern abzuspeichern und zu lesen. Dieses Programm listet z.B. nach Eingabe der Ziffer 10 nach entsprechender Abfrage alle jene Personen auf, die im Oktober geboren sind. Auf Geburtstagsglückwünsche kann man sich also in Ruhe vorbereiten. Für eine angemessene Berücksichtigung von besonderen Geburtstagen (50/60/65) kann man die Ziffer 27 eingeben, und alle Mitarbeiter, die im Jahr 1987 ihren 60. Geburtstag feiern, werden aufgelistet. Will man wissen, wer in welchem Ort seinen Wohnsitz hat, gibt man den Ort ein und erhält eine Liste der dort als wohnhaft verzeichneten Personen. Die Nummer eines Ortsteils wird gesondert erfaßt. Für einen Geschäftsmann kann es für die Kundenbetreuung sinnvoll sein, seine Adreßdatei nach Postleitzahl-Bereichen ordnen zu lassen, um sicher zu gehen, keinen Kunden zu vergessen.

Je nach Bedarf kann der ausgewiesene Bereich weiter oder enger gefaßt werden. Voreingestellt sind im Programm die ersten beiden Ziffern der Postleitzahl, durch Veränderung der Variablen »b« in Zeile 120 läßt sich dieses durchführen. Wird die Variable mit 3 belegt, werden engere, durch Belegung mit 1 weitere Bezirke durchsucht. Vor Eröffnung einer entsprechenden Datei sollte allerdings die Wahl nach den jeweiligen Bedürfnissen fallen. Eine spätere Veränderung gewährleistet nicht mehr den kompletten Zugriff.

Neben diesen vorgegebenen Schlüsseln lassen sich bei der Eingabe neuer Daten vier weitere Schlüsselbegriffe hinzufügen. Hier sind für den Anwender vielfältige individuelle Möglichkeiten gegeben, persönliche Daten zu ordnen. Vom Grad der Verwandtschaft über persönliche Merkmale und soziale Kennzeichen bis hin zu betriebsorganisatorischen Informationen.

Der Platz für die Eingabe dieser Begriffe ist auf 20 Zeichen ausgelegt. Wer diesen Platz nicht benötigt, kann durch Änderung der Variablen »a« in Zeile 120 eine Kürzung vornehmen. Damit wird Speicherkapazität gespart.

Die Eingabe erfolgt über vorgegebene Felder, die in den Zeilen 570 – 700 definiert werden. Sie sind so eingerichtet,

```
********** Personalverwaltung **********

A C H T U N G - Beenden des Programms immer über ENDE - sonst VERLUST aller Daten!!!

        AUSWAHL:

                        SUCHEN:       S
                        HINZUFÜGEN:   H
                        DRUCKEN:      D
                        VERÄNDERN:    V
                        ENDE:         E
```

Abb. 1: Das Hauptmenue der Adressverwaltung

```
********** Suchen **********

        AUSWAHL

                        NAME:           N
                        GEBURTSJAHR:    J
                        GEBURTSMONAT:   M
                        ORT:            O
                        PLZ-Bereich:    P
                        SCHLÜSSEL:      S
```

Abb. 2: Diese Optionen bietet das Suchen-Menue...

daß die Informationen in dem zur Verfügung stehenden Satz untergebracht werden können. Wenn ein Benutzer andere Daten als die vorgegebenen erfassen will, kann er hier Änderungen vornehmen. Der Feldname kann verändert werden, ohne daß dieses an anderer Stelle im Programm Konsequenzen hätte. Die Eingaben werden in 2600 ff in Sätze geschrieben und als Schlüssel definiert. Versucht man, bei der Eingabe Komma und Semikolon zu verwenden, wird die Annahme mit Piepston verweigert. Dieses ist so vorgesehen, weil ein Komma in Strings von Basic nicht akzeptiert wird und ein Semikolon zu Schwierigkeiten beim Lesen der Informationen führt. Alle anderen Zeichen sind erlaubt, auch die Verwendung von Anführungsstrichen, die von sequentiellen Dateien innerhalb eines Strings verübelt würden. Die Eingabe wird durch die EXIT-Taste beendet. Solange diese Betätigung nicht erfolgt, kann zwischen den Eingabefeldern mit Hilfe der Cursor-Tasten in alle Richtungen und mit der RETURN-Taste nach unten gesprungen werden. Tippfehler können durch Überschreiben korrigiert werden.

```
********************************** Daten hinzufügen **********************************

Name     :Mustermann.....                    Vorname  :Erika.....
                                             Geb.-Dat.:13.6.63.
Strasse  :Holzweg 13.....
Ort      :4711 :Musterstadt.....     OT:22
Telefon  :0123/456789....
Info     :.....

Verschlüsselung:
1. Schlüssel:.....                           3. Schlüssel:.....
2. Schlüssel:.....                           4. Schlüssel:.....

                    :Nach Abschluß des Eintrags EXIT-Taste drücken:
```

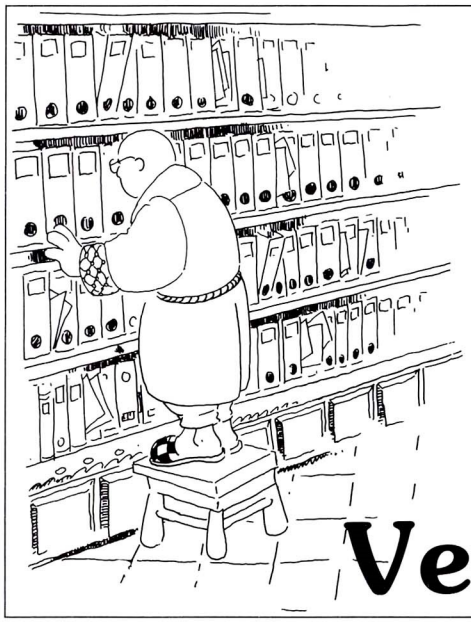
Abb. 3: Die Gestaltung der Datensatzeingabe...

Neben der Bildschirmausgabe der gesuchten Informationen besteht die Möglichkeit, Listen nach den beschriebenen Merkmalen sowie eine komplette Namensliste aller Personen ausdrucken zu lassen. Um ständig Aktualität der Verwaltung zu garantieren, lassen sich Veränderungen vornehmen. Nach Eingabe eines Namens und der Abfrage nach Richtigkeit werden alle Informationen zu dieser Person gelöscht. Danach läßt sich ein neuer Satz mit aktuellen Daten aufnehemen. In der Jetsam-Version lassen sich zwar auch teilweise Veränderungen ohne völlige Löschung vornehmen, aber der Weg der Löschung scheint einfacher und im Hinblick auf Übersichtlichkeit sinnvoller zu sein. Zum Abschluß soll noch auf eine Besonderheit aufmerksam gemacht werden: Indizierte Dateien müssen immer geschlossen werden, bevor sie erneut benutzt werden. Deshalb erscheint über der zentralen Abfrage des Programms der Hinweis auf das ordnungsgemäße Ende über die E-Taste. Sollte das Programm einmal anders als auf diese Weise beendet werden, muß unbedingt ein »close file%« bzw. »close 1« eingegeben werden. Die einfache »close«-Anweisung genügt nicht! Für denjenigen, der bisher nicht mit indizierten Dateien gearbeitet hat, ist ein solches Verfahren zunächst ungewöhnlich. (F. Sauerländer)

```
LISTING >ADRESSEN<, REMARK = >'<.

<94> 10 '*** Adressen- und Personalverwaltung ***
<78> 20 :
<21> 30 '*** Bildschirmaufbau ***
< 1> 40 GOSUB 400
<81> 50 :
<72> 100 '*** Datei bestimmen und Variablen belegen ***
<16> 110 :
<27> 120 a=20:b=2
<55> 130 file%=1
< 5> 140 druck=8793:ein=234:aus=239
<83> 150 filename$="adressen.rnd":dateiname$="adressen.
     ind"
<26> 160 :
<51> 170 '*** Datei festlegen ***
<30> 180 :
<95> 190 IF FIND$(filename$)<>"" AND FIND$(dateiname$)<
     >""THEN BUFFERS 12:OPEN "K",file%,filename$,datein
     ame$,2:GOTO 250
<61> 200 match$="WwNn":PRINT FNkopf$(" Dateisuche "):PR
     INT;bell$:PRINT:PRINT:PRINT:PRINT:PRINT TAB(
     20);"Die benötigte Datei ist nicht auf dieser Disk
     ette.":PRINT:PRINT TAB(20)"Diskette wechseln oder
     neue Datei einrichten? (W/N)":GOSUB 900
<35> 210 IF antwort=1 OR antwort=2 THEN GOSUB 1000
<16> 220 IF antwort=3 OR antwort=4 THEN GOSUB 800:GOTO
     190
<68> 230 RESET
< 1> 240 GOTO 190
<84> 250 FIELD file%,15 AS rec.name$,15 AS rec.vorname$
     ,4 AS rec.plz$,20 AS rec.adresse$,30 AS rec.strass
     e$,15 AS rec.tel$,8 AS rec.geb$,19 AS rec.info$
<27> 260 :
<76> 270 befehl=0
<31> 280 WHILE befehl <9
<35> 290 PRINT FNkopf$(" Personalverwaltung "):PRINT:PR
     INT TAB(3);"A C H T U N G - Beenden des Programms
     immer über ENDE - sonst VERLUST aller Daten!!!":PR
     INT:PRINT:PRINT
<16> 300 PRINT:PRINT:PRINT TAB(20);"AUSWAHL:":PRINT:PRI
     NT TAB(40);"SUCHEN:..... S":PRINT:PRINT TAB(40);"
     HINZUFÜGEN:.. H":PRINT:PRINT TAB(40);"DRUCKEN:....
     . D":PRINT:PRINT TAB(40);"VERÄNDERN:... V":PRINT:P
     RINT TAB(40);"ENDE:..... E"
<84> 310 match$="SsHhDdVvEe":GOSUB 900:befehl = antwort
<77> 320 ON befehl GOSUB 2000,2000,2600,2600,3600,3600,
     4000,4000:GOSUB 1000
<87> 330 WEND
<24> 340 :
<40> 350 '*** Abschluß ***
<28> 360 :
<53> 370 CLOSE file%:PRINT cls$:PRINT curs.an$
<41> 380 END
<34> 390 :
<28> 400 '*** Bildschirmaufbau ***
<19> 410 :
<96> 420 escape$=CHR$(27):bell$=CHR$(7):cls$=escape$+"E
     "+escape$+"H":crlf$=CHR$(13)+CHR$(10)
<23> 430 DEF FNplaz$(x,y)=escape$+"Y"+CHR$(y+32)+CHR$(x
     +32)
<67> 440 DEF FNkopf$(titel$) = cls$+STRING$((88-LEN(tit
     el$))/2,"*")+""+titel$+""+STRING$((88-LEN(titel$))
     /2,"*")+crlf$+crlf$
<56> 450 curs.rechts$=escape$+"C":curs.links$=escape$+"
     D":curs.an$=escape$+"e":curs.aus$=escape$+"f":PRIN
     T curs.aus$
<47> 460 DIM ein$(20),eingabe$(20),nom$(20),x(20),y(20)
     ,lg(20),pr(20),feld(20)
<71> 470 stzchn=6:FOR n=1 TO stzchn
<51> 480 READ stzchn%(n):NEXT
<69> 490 READ feld:
<15> 500 FOR n=1 TO feld
<56> 510 READ nom$(n),x(n),y(n),lg(n)
<42> 520 NEXT
```

```
<77> 530 RETURN
<26> 540 :
<98> 550 DATA 13,30,6,1,31,27
<68> 560 DATA 15
<48> 570 DATA "Name.... ",1,4,15
<85> 580 DATA "Vorname. ",50,4,15
<49> 590 DATA "Geb.-Dat.",50,6,8
<48> 600 DATA "Strasse. ",1,8,30
<38> 610 DATA "Ort..... ",1,10,4
<10> 620 DATA " ",15,10,18
<68> 630 DATA "OT",40,10,2
<80> 640 DATA "Telefon. ",1,12,15
<97> 650 DATA "Info.... ",1,14,19
< 7> 660 DATA "Verschlüsselung",1,20,0
<42> 670 DATA "1. Schlüssel",1,22,20
<12> 680 DATA "2. Schlüssel",1,24,20
<19> 690 DATA "3. Schlüssel",50,22,20
<71> 700 DATA "4. Schlüssel",50,24,20
<66> 710 DATA ":Nach Abschluß des Eintrags EXIT-Taste d
     rücken",20,27,0
<24> 720 :
<23> 800 '*** leere Datei erstellen ***
<23> 810 :
<19> 820 PRINT FNkopf$(" Erstellung einer neuen Datei "
     ):PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT TAB(37
     );"Bitte warten."
<19> 830 RESET:BUFFERS 12: CREATE file%,filename$,datei
     name$,2:
<15> 840 FIELD file%,10 AS rec.test$:LSET rec.test$=SPA
     CE$(10):ergebnis=ADDREC(file%,0,6,SPACE$(10))
<62> 850 CLOSE file%:PRINT:PRINT:PRINT:PRINT TAB(37);"D
     atei erzeugt."
<86> 860 RETURN
<35> 870 :
<45> 900 '*** Taste holen *****
<24> 910 :
<93> 920 antwort$ = INKEY$
<58> 930 WHILE antwort$=""
<82> 940 antwort$=INKEY$
<97> 950 WEND
<69> 960 antwort = INSTR(match$,antwort$): IF antwort =
      0 THEN GOTO 920
<89> 970 RETURN
<38> 980 :
<65> 1000 '*** auf Taste warten ***
< 2> 1010 :
<32> 1020 PRINT:PRINT"Weiter mit beliebiger Taste"
<46> 1030 WHILE INKEY$=""
<23> 1040 WEND
<93> 1050 RETURN
<17> 1060 :
<24> 2000 '*** Unterroutinen ***
< 3> 2010 :
<19> 2020 '*** Suchen ***
< 9> 2030 :
<40> 2040 z=0:l=1:befehl=0:WHILE befehl<13
<52> 2050 PRINT FNkopf$(" Suchen "):PRINT:PRINT TAB(20)
     ;"AUSWAHL":PRINT:PRINT TAB(40);"NAME:.............
      N":PRINT:PRINT TAB(40);"GEBURTSJAHR:..... J":PRI
     NT:PRINT TAB(40);"GEBURTSMONAT:..... M":
< 5> 2060 PRINT:PRINT TAB(40);"ORT:.............. O":PR
     INT:PRINT TAB(40);"PLZ-Bereich:..... P":PRINT:PRI
     NT TAB(40);"SCHLÜSSEL:........ S"
<38> 2070 match$="NnJjMmOoPpSs":GOSUB 900:befehl=antwor
     t
<23> 2080 ON befehl GOSUB 2110,2110,2160,2160,2210,2210
     ,2260,2260,2320,2320,2370,2370:befehl=0:GOTO 2100
<39> 2090 WEND
<81> 2100 RETURN
<74> 2110 PRINT curs.an$:PRINT:PRINT:INPUT "Gesuchten N
     amen eingeben: ",name$
<16> 2120 PRINT curs.aus$:PRINT FNkopf$(" Personaldaten
      "):PRINT:PRINT TAB(10);"Unter dem Namen ";UPPER$(
     name$);" sind folgende Daten gespeichert:":PRINT:n
     $=name$+SPACE$(15):schl1$=UPPER$(LEFT$(n$,15))
< 0> 2130 erg=SEEKKEY(file%,0,0,schl1$):IF erg=0 THEN G
     ET file%:GOSUB 2440
<18> 2140 IF erg<>0 THEN PRINT:PRINT"Keine Eintrag vorh
     anden.":RETURN
<59> 2150 GOSUB 2430:RETURN
<35> 2160 PRINT curs.an$:PRINT:PRINT:INPUT "Gesuchtes G
     eburtsjahr eingeben (2 Ziffern): ",schl7$
<92> 2170 PRINT curs.aus$:PRINT FNkopf$(" Personaldaten
      "):PRINT:PRINT TAB(10);"Mit gesuchtem Jahr sind f
     olgende Personen verzeichnet: ":PRINT
<87> 2180 erg=SEEKKEY(file%,0,3,schl7$):IF erg=0 THEN G
     ET file%:GOSUB 2440
<73> 2190 IF erg<>0 THEN PRINT:PRINT"Kein Eintrag vorha
     nden.":RETURN
<46> 2200 GOSUB 2430:RETURN
<46> 2210 PRINT curs.an$:PRINT:INPUT"Gesuchten Geburtsm
     onat eingeben (2 Ziffern): ",schl8$
<60> 2220 PRINT curs.aus$:PRINT FNkopf$(" Personaldaten
      "):PRINT:PRINT TAB(10);"Mit gesuchtem Monat sind
     folgende Personen verzeichnet: ":PRINT
<33> 2230 erg=SEEKKEY(file%,0,4,schl8$):IF erg=0 THEN G
     ET file%:GOSUB 2440
<60> 2240 IF erg<>0 THEN PRINT:PRINT"Kein Eintrag vorha
     nden.":RETURN
<61> 2250 GOSUB 2430:RETURN
<46> 2260 PRINT curs.an$:PRINT:INPUT"Ort eingeben: ",sc
     hl$
<94> 2270 PRINT curs.aus$:PRINT FNkopf$(" Personaldaten
      "):PRINT:PRINT TAB(10);"Folgende Personen sind ve
     rzeichnet:":PRINT
<35> 2280 such$=schl$+SPACE$(20):s$=UPPER$(LEFT$(such$,
     18))
<72> 2290 erg=SEEKKEY(file%,0,1,s$):IF erg=0 THEN GET f
     ile%:GOSUB 2440
<50> 2300 IF erg<>0 THEN PRINT:PRINT"Kein Eintrag vorha
     nden.":RETURN
<51> 2310 GOSUB 2430:RETURN
<37> 2320 PRINT curs.an$:PRINT:PRINT"PLZ-Bereich (";b;:
     INPUT" Ziff.) eingeben: ",schl9$
<21> 2330 PRINT curs.aus$:PRINT FNkopf$(" Personaldaten
      "):PRINT:PRINT TAB(10);"Folgende Personen sind un
     ter dem PLZ-Bereich ";schl9$;" verzeichnet:":PRINT
     :schl$=LEFT$(schl9$,b)
<40> 2340 erg=SEEKKEY(file%,0,5,schl$):IF erg=0 THEN GE
     T file%:GOSUB 2440
<65> 2350 IF erg<>0 THEN PRINT:PRINT"Kein Eintrag vorha
     nden.":RETURN
<66> 2360 GOSUB 2430:RETURN
<95> 2370 PRINT curs.an$:PRINT:PRINT:INPUT "Schlüssel e
     ingeben: ",schl$
<35> 2380 PRINT curs.aus$:PRINT FNkopf$(" Personaldaten
      "):PRINT:PRINT"Unter dem Stichwort ";UPPER$(schl$
     );" sind folgende Personen verzeichnet: ":PRINT
<44> 2390 such$=schl$+SPACE$(a):s$=UPPER$(LEFT$(such$,a
     ))
<75> 2400 erg=SEEKKEY(file%,0,2,s$):IF erg=0 THEN GET f
     ile%:GOSUB 2440
<55> 2410 IF erg<>0 THEN PRINT:PRINT"Kein Eintrag vorha
     nden.":RETURN
<56> 2420 GOSUB 2430:RETURN
<24> 2430 erg=SEEKNEXT(file%,0):IF erg=0 THEN GET file%
     :GOSUB 2440:GOTO 2430 ELSE RETURN
< 8> 2440 PRINT TAB(10);"Name:";TAB(20);rec.name$;" ";r
     ec.vorname$;"Geb.-Dat.: ";rec.geb$:PRINT TAB(10);"
     Adresse: ";TAB(20);rec.strasse$:PRINT TAB(20);rec.
     plz$;" ";rec.adresse$:PRINT TAB (10);"Tel.: ";TAB(
     20);rec.tel$;TAB (40);rec.info$:PRINT
<47> 2450 z=z+1:IF z=4*l THEN GOSUB 1000:l=l+1:PRINT FN
     kopf$(" Personaldaten "):PRINT
< 6> 2460 RETURN
<29> 2470 :
<32> 2480 :
<13> 2510 :
<23> 2600 '*** Satz hinzufügen ***
<15> 2610 :
<35> 2620 PRINT FNkopf$(" Daten hinzufügen ")
<35> 2630 ende=0
<16> 2640 ergebnisa=0:ergebnisb=0:ergebnisc=0:ergebnisd
     =0:ergebnise=0:ergebnisf=0:ergebnisg=0:ergebnish=0
     :ergebnisi=0
<27> 2650 GOSUB 3030
<14> 2660 name$=eingabe$(1):vorname$=eingabe$(2)
<86> 2670 geb$=eingabe$(3):strasse$=eingabe$(4)
<31> 2680 plz$=eingabe$(5):adresse$=eingabe$(6)+eingabe
     $(7)
<12> 2690 tel$=eingabe$(8):info$=eingabe$(9)
< 5> 2700 schl1$=UPPER$(name$):schl2$=UPPER$(eingabe$(6
     ))
<86> 2710 schl3$=UPPER$(LEFT$(eingabe$(11),a)):schl4$=U
     PPER$(LEFT$(eingabe$(12),a))
<25> 2720 schl5$=UPPER$(LEFT$(eingabe$(13),a)):schl6$=U
     PPER$(LEFT$(eingabe$(14),a))
<31> 2730 schl7$=RIGHT$(eingabe$(3),2):schl8$=MID$(eing
     abe$(3),4,2)
<20> 2740 schl9$=LEFT$(plz$,b):such$=SPACE$(a)
<14> 2750 LSET rec.info$=info$
<58> 2760 LSET rec.name$=name$
<61> 2770 LSET rec.vorname$=vorname$
<24> 2780 LSET rec.adresse$=adresse$
<17> 2790 LSET rec.strasse$=strasse$
<36> 2800 LSET rec.geb$=geb$
<92> 2810 LSET rec.tel$=tel$
<67> 2820 LSET rec.plz$=plz$
<72> 2830 pruefen=INSTR(schl1$,SPACE$(15)):IF pruefen=1
      THEN PRINT FNkopf$(" F e h l e r "):PRINT:PRINT:P
     RINT TAB(15);"Es wurde kein Name eingetragen. Bitt
     e die Eintragung wiederholen.":PRINT:PRINT:PRINT:R
     ETURN
<61> 2840 ergebnisa = ADDREC(file%,0,0,schl1$)
<40> 2850 nummer=FETCHREC(file%)
<11> 2860 ergebnisb=ADDKEY(file%,0,1,schl2$,nummer)
< 1> 2870 pruefen=INSTR(schl3$,such$):IF pruefen=1 THEN
      GOTO 2890
< 0> 2880 ergebnisc=ADDKEY(file%,0,2,schl3$,nummer)
<13> 2890 pruefen=INSTR(schl4$,such$):IF pruefen=1 THEN
      GOTO 2910
<29> 2900 ergebnisd=ADDKEY(file%,0,2,schl4$,nummer)
<93> 2910 pruefen=INSTR(schl5$,such$):IF pruefen=1 THEN
      GOTO 2940
<86> 2920 ergebnise=ADDKEY(file%,0,2,schl5$,nummer)
<43> 2930 pruefen=INSTR(schl6$,such$):IF pruefen=1 THEN
```

```
     GOTO 2960
<44> 2940 ergebnisf=ADDKEY(file%,0,2,schl6$,nummer)
<29> 2950 pruefen=INSTR(schl7$,SPACE$(2)):IF pruefen=1
     THEN GOTO 2970
<33> 2960 ergebnisg=ADDKEY(file%,0,3,schl7$,nummer)
<86> 2970 pruefen=INSTR(schl8$,SPACE$(2)):IF pruefen=1
     THEN GOTO 2990
<22> 2980 ergebnish=ADDKEY(file%,0,4,schl8$,nummer)
< 6> 2990 pruefen=INSTR(schl9$,SPACE$(2)):IF pruefen=1
     THEN GOTO 3010
<63> 3000 ergebnisi=ADDKEY(file%,0,5,schl9$,nummer)
<19> 3010 IF ergebnisa=0 AND ergebnisb=0 AND ergebnisc=
     0 AND ergebnisd=0 AND ergebnise=0 AND ergebnisf=0
     AND ergebnisg=0 AND ergebnish=0 AND ergebnisi=0 TH
     EN PRINT FNkopf$(" Kontrolle "):PRINT:PRINT:PRINT:
     PRINT"Neue Einträge geschrieben":RETURN
<33> 3020 PRINT cls$:PRINT FNkopf$(" F e h l e r "):PRI
     NT:PRINT"Die letzten Einträge wurden nicht korrekt
      verarbeitet. Das Programm ist beendet und kann ne
     u gestartet werden.":GOTO 350
<33> 3030 feldnr=1
<83> 3040 FOR n=1 TO feld
< 1> 3050 PRINT FNplaz$(x(n),y(n));
<43> 3060 PRINT nom$(n);":";STRING$(lg(n),"."):eingabe$
     =STRING$(lg(n)," ")
<52> 3070 eingabe$(n)=STRING$(lg(n)," ")
< 9> 3080 NEXT
<62> 3090 x=x(feldnr):y=y(feldnr)
< 9> 3100 flg=lg(feldnr)
<24> 3110 x=x+LEN(nom$(feldnr))+1
<98> 3120 ein$=eingabe$(feldnr)
<10> 3130 GOSUB 3210
<79> 3140 eingabe$(feldnr)=ein$
<96> 3150 IF ende THEN RETURN
<18> 3160 IF oben THEN feldnr=feldnr-1:IF feldnr=0 THEN
      feldnr=1
<17> 3170 IF unten THEN feldnr=feldnr+1:IF feldnr>feld
     THEN feldnr=feld
<56> 3180 unten=0:oben=0
<63> 3190 GOTO 3090
<87> 3200 PRINT bell$;:GOTO 3250
< 8> 3210 posit=x
<34> 3220 PRINT FNplaz$(x,y);
<83> 3230 lg=1
<16> 3240 PRINT curs.an$;
<72> 3250 zchn$=INKEY$:IF zchn$="" THEN 3250
< 5> 3260 zchn%=ASC(zchn$)
<40> 3270 FOR n=1 TO stzchn:IF zchn%<>stzchn%(n) THEN N
     EXT
<24> 3280 ON n GOTO 3350,3350,3380,3400,3420,3430
<84> 3290 kor=1
<73> 3300 GOSUB 3440
<63> 3310 IF kor=0 THEN 3200
<24> 3320 IF posit>=x+flg THEN 3200
<61> 3330 ein$=LEFT$(ein$,posit-x)+zchn$+RIGHT$(ein$,fl
     g-(posit-x)-1)
<73> 3340 PRINT zchn$;:posit=posit+1:GOTO 3250
<95> 3350 unten=1
<54> 3360 PRINT curs.aus$;
< 8> 3370 RETURN
<53> 3380 IF posit=x+flg THEN 3200
<64> 3390 PRINT curs.rechts$;:posit=posit+1:GOTO 3250
<65> 3400 IF posit=x THEN 3200
<30> 3410 PRINT curs.links$;:posit=posit-1:GOTO 3250
<72> 3420 oben=1:GOTO 3360
<10> 3430 ende=1:GOTO 3360
<18> 3440 IF zchn%<44 THEN RETURN
<45> 3450 IF zchn%>59 THEN RETURN
<42> 3460 IF zchn%>44 AND zchn%<59 THEN RETURN
<57> 3470 kor=0:RETURN
<33> 3480 :
<54> 3600 '*** Ausdruck ***
<16> 3610 :
<72> 3620 z=0:l=1:befehl=0:WHILE befehl<11
<58> 3630 PRINT FNkopf$(" Drucken "):PRINT:PRINT TAB(20
     );"Ausdruck von Listen nach:":PRINT:PRINT TAB(40);
     "NAMEN:.......... N":PRINT:PRINT TAB(40);"sämtl.
     NAMEN:...... L"
<12> 3640 PRINT:PRINT TAB(40);"ORT:.............. O":PR
     INT:PRINT TAB(40);"PLZ-Bereich:...... P":PRINT:PRI
     NT TAB(40);"SCHLÜSSEL:........ S"
<50> 3650 match$="NnLlOoPpSs":GOSUB 900:befehl=antwort
<26> 3660 ON befehl GOSUB 3690,3690,3730,3730,3760,3760
     ,3810,3810,3850,3850:befehl=0:GOTO 3680
<46> 3670 WEND
<23> 3680 POKE druck,aus:RETURN
<82> 3690 POKE druck,aus:PRINT curs.an$:PRINT:PRINT:INP
     UT "Gesuchten Namen eingeben: ",name$
<20> 3700 PRINT curs.aus$:POKE druck,ein:PRINT TAB(10);
     "Unter dem Namen ";UPPER$(name$);" sind folgende D
     aten gespeichert:":PRINT:n$=name$+SPACE$(15):schl1
     $=UPPER$(LEFT$(n$,15))
<17> 3710 erg=SEEKKEY(file%,0,0,schl1$):IF erg=0 THEN G
     ET file%:GOSUB 3910 ELSE RETURN
<98> 3720 GOSUB 3900:RETURN
<29> 3730 POKE druck,ein:PRINT TAB(10);"Namensliste ":P
     RINT
<32> 3740 erg=SEEKRANK(file%,0,0):IF erg=0 THEN GET fil
     e%:GOSUB 3930 ELSE RETURN
<78> 3750 erg=SEEKNEXT(file%,0):IF erg=0 OR erg=101 THE
     N GET file%:GOSUB 3930:GOTO 3750 ELSE RETURN
<33> 3760 POKE druck,aus:PRINT curs.an$:PRINT:INPUT"Ort
      eingeben: ",schl$
<49> 3770 PRINT curs.aus$:POKE druck,ein:PRINT TAB(10);
     "Folgende Personen wohnen in ";UPPER$(schl$);":":P
     RINT
<46> 3780 such$=schl$+SPACE$(20):s$=UPPER$(LEFT$(such$,
     18))
<83> 3790 erg=SEEKKEY(file%,0,1,s$):IF erg=0 THEN GET f
     ile%:GOSUB 3910 ELSE RETURN
<94> 3800 GOSUB 3900:RETURN
<39> 3810 POKE druck,aus:PRINT curs.an$:PRINT:PRINT"PLZ
     -Bereich (";b;:INPUT" Ziff.) eingeben: ",schl$
<92> 3820 PRINT curs.aus$:POKE druck,ein:PRINT TAB(10);
     "Folgende Personen haben ihren Wohnort im PLZ-Bere
     ich ";schl$;":":PRINT
< 6> 3830 erg=SEEKKEY(file%,0,5,schl$):IF erg=0 THEN GE
     T file%:GOSUB 3910 ELSE RETURN
<21> 3840 GOSUB 3890:RETURN
<88> 3850 POKE druck,aus:PRINT curs.an$:PRINT:PRINT:INP
     UT "Schlüssel eingeben: ",schl$
<80> 3860 PRINT curs.aus$:POKE druck,ein:PRINT TAB(10);
     "Unter dem Stichwort ";UPPER$(schl$);" sind folgen
     de Personen verzeichnet: ":PRINT
<49> 3870 such$=schl$+SPACE$(a):s$=UPPER$(LEFT$(such$,a
     ))
< 9> 3880 erg=SEEKKEY(file%,0,2,s$):IF erg=0 THEN GET f
     ile%:GOSUB 3910 ELSE RETURN
<22> 3890 GOSUB 3900:RETURN
< 6> 3900 erg=SEEKNEXT(file%,0):IF erg=0 THEN GET file%
     :GOSUB 3910:GOTO 3900 ELSE RETURN
<10> 3910 PRINT TAB(10);"Name:";TAB(20);rec.name$;" ";r
     ec.vorname$;"Geb.-Dat.: ";rec.geb$:PRINT TAB(10);"
     Adresse: ";TAB(20);rec.strasse$:PRINT TAB(20);rec.
     plz$;" ";rec.adresse$:PRINT TAB (10);"Tel.: ";TAB(
     20);rec.tel$;TAB (40);rec.info$:PRINT
< 5> 3920 RETURN
<16> 3930 PRINT TAB(10);rec.name$;rec.vorname$
<11> 3940 RETURN
<91> 4000 '*** Verändern ***
< 5> 4010 :
< 5> 4020 PRINT FNkopf$(" Verändern von Einträgen "):PR
     INT:PRINT:PRINT:PRINT:PRINT
<96> 4030 PRINT curs.an$:PRINT:PRINT TAB(25);"Die Verän
     derung erfolgt über den Namen.":PRINT:PRINT TAB(20
     )"Bitte den Namen eingeben und RETURN drücken. ":P
     RINT TAB(40):INPUT":",sign$
<36> 4040 PRINT curs.aus$:a$=sign$+SPACE$(15):schl$=UPP
     ER$(LEFT$(a$,15)):PRINT:PRINT
<92> 4050 ergebnis=SEEKKEY(file%,0,0,schl$)
<75> 4060 IF ergebnis=0 THEN GET file%:PRINT TAB(10);re
     c.name$;rec.vorname$;rec.adresse$
<94> 4070 IF ergebnis<>0 THEN PRINT FNkopf$(" F e h l e
      r "):PRINT:PRINT:PRINT:PRINT:PRINT TAB(20);"Unter
      dem eingegebenen Namen sind keine Daten vorhanden
     .":PRINT:PRINT:PRINT:PRINT:RETURN
<58> 4080 match$="JjNn":PRINT:PRINT:PRINT"Ist dieses da
     s gesuchte Material? (J/N)":GOSUB 900
<66> 4090 IF antwort=1 OR antwort=2 THEN GOTO 4120
<59> 4100 ergeb=SEEKNEXT(file%,0):IF ergeb=0 THEN GET f
     ile%:PRINT TAB(10);rec.name$;rec.vorname$;rec.adre
     sse$:GOTO 4080
<31> 4110 IF ergeb<>0 THEN PRINT "Es sind unter dem ges
     uchten Namen keine weiteren Daten vorhanden":RETUR
     N
<39> 4120 erg=DELKEY(file%,0)
<77> 4130 identa$=rec.name$:identb$=rec.vorname$
<44> 4140 ergebnis=SEEKRANK(file%,0,1):IF ergebnis=0 TH
     EN GET file%:IF rec.name$=identa$ AND rec.vorname$
     =identb$ THEN IF DELKEY(file%,0) THEN GOTO 4140
<94> 4150 GOSUB 4230:
<77> 4160 ergebnis=SEEKRANK(file%,0,2):IF ergebnis=0 TH
     EN GET file%:IF rec.name$=identa$ AND rec.vorname$
     =identb$ THEN IF DELKEY(file%,0) THEN GOTO 4160
< 1> 4170 GOSUB 4230:
<11> 4180 ergebnis=SEEKRANK(file%,0,3):IF ergebnis=0 TH
     EN GET file%:IF rec.name$=identa$ AND rec.vorname$
     =identb$ THEN IF DELKEY(file%,0) THEN GOTO 4180
< 7> 4190 GOSUB 4230:
<87> 4200 ergebnis=SEEKRANK(file%,0,4):IF ergebnis=0 TH
     EN GET file%:IF rec.name$=identa$ AND rec.vorname$
     =identb$ THEN IF DELKEY(file%,0) THEN GOTO 4200
<47> 4210 GOSUB 4230
<54> 4220 GOTO 4270
< 2> 4230 WHILE ergebnis=0 OR ergebnis=101
<31> 4235 ergebnis=SEEKNEXT(file%,1):IF ergebnis>101 TH
     EN RETURN
< 5> 4240 GET file%:IF rec.name$=identa$ AND rec.vornam
     e$=identb$ THEN IF DELKEY(file%,0) THEN GOTO 4240
<33> 4250 WEND
< 4> 4260 RETURN
<80> 4270 match$="JjNn":PRINT:PRINT:PRINT"Soll ein neue
     r Satz aufgenommen werden? (J/N)":GOSUB 900
<94> 4280 IF antwort=1 OR antwort=2 THEN GOSUB 2600
<13> 4290 RETURN
< 8> 4300 :
```



Archiv-Verwaltung

Wer beruflich oder privat Sammlungen aller Art verwaltet, kennt die Mühen, die diese Arbeit mit sich bringt. Vielfach scheitern die Versuche, sich ein Archiv anzulegen, in dem die Informationen über Loseblattsammlungen, Videofilme, Toncassetten usw. gespeichert sind, an dem zeitlichen Aufwand für das Ausfüllen und Führen von Karteien, an den Schwierigkeiten, eine systematische und praktikable Ordnung aufzubauen. Befragungen im Kollegen- und Bekanntenkreis ergeben immer wieder Auskunft über das Scheitern guter Vorsätze. Der Joyce-Besitzer, der mit solchen Sammlungen beschäftigt ist, denkt deshalb an die Fähigkeiten und Einsatzmöglichkeiten seines Computers, bei einem solchen Vorhaben behilflich zu sein.

Die mit großem Engagement betriebene Arbeit der Entwicklung eines Programms zur Archiv-Verwaltung mittels sequentieller Dateien wurde jäh enttäuscht, als beim 135. Eintrag die Meldung »Memory full in ... « auf dem Bildschirm erschien. Die Kapazität des Arbeitsspeichers reicht beim Joyce also nicht aus, umfangreichere Informationen mit Hilfe sequentieller Dateien zu verarbeiten. Diese Enttäuschung veranlaßte mich, mich mit dem im Joyce-Handbuch 2 beschriebenen Möglichkeiten der Direktzugriffsdateien zu beschäftigen. Für meine Zwecke schien die Verarbeitung von Informationen über Schlüssel-Zugriff (JetsamVersion) ideal zu sein. Ich wollte Signaturen und Informationstexte über meine Sammlungsobjekte speichern und um auf diese über Schlagwörter zugreifen zu können. Durch Eingabe des Begriffs »Krimi« wollte ich z.B. aus der Sammlung von Videofilmen alle Cassetten-Nummern und die Titel der Kriminalfilme aufgelistet bekommen. Bei Eingabe von »Tatort« sollten aus dieser Liste nur die Tatort-Krimis, bei Eingabe von »Schimansky« nur die Tatort-Krimis mit Götz George aufgelistet werden. Diese Anforderungen erfüllt das hier vorgestellte Programm. Es war allerdings ein mühsamer Weg dorthin, denn leider wird die Hilfestellung des Handbuchs bei der Entwicklung von Programmen mit Jetsam-Version spärlich. Sind für die anderen Möglichkeiten der Verwendung von Dateien noch Programmbeispiele abgedruckt, fehlen sie hier völlig. Die Ausführungen beschränken sich auf wenige Erläuterungen. »Das große Joyce Buch« von Dullin und Straßenburg läßt schon den BASIC-Interessierten weitgehend allein, für die Arbeit mit Jetsam werden keine Informationen vermittelt. Zur Handhabung des Programms folgende Erläuterungen: Man kann verschiedene Archive getrennt voneinander verwalten. Das Programm richtet nach Wahl die entsprechenden Dateien ein. Die Einrichtung erfolgt durch die Zeilen 800-860. Vorgesehen ist die Eingabe von Signaturen, eines Informationstextes und von bis zu 6 Schlagwörtern. Es muß mindestens die Signatur und das erste Schlagwort eingegeben werden, damit eine Speicherung erfolgen kann.

Das Eingabefeld ist so formatiert, daß die Einträge nur in der Länge angenommen werden, wie sie gespeichert werden können. Die Länge der Schlagwörter ist auf 20 Zeichen begrenzt. Eine Reduzierung kann vorgenommen werden, etwa um Speicherkapazität zu sparen, wenn in Zeile 140 der Wert der Variablen »a« entsprechend verändert wird. Die Länge der Schlagwörter ließe sich bis auf 31 Zeichen verändern, denn Jetsam akzeptiert Schlüssel bis zu dieser Länge.

Aus Gründen der Sparsamkeit und Schnelligkeit bei der Suche nach diesen Schlüsseln ist jedoch eine Kürzung zu empfehlen. Will man Veränderungen an den Eingabefeldern vornehmen, sollte man bedenken, daß 10 Zeichen für die Signatur und 116 Zeichen für den Informationstext die obere Grenze sind. Übrige Veränderungen können

```
********** Archivverwaltung **********

Auswahl des Archivs:

                    Loseblattsammlung:  L
                    Video:              V
                    Bibliothek:         B
                    Ton:                T
```

Abb. 1: Das Hauptmenü der Archiv-Verwaltung

```
********** Materialsuche **********

Zum Schlagwort CASABLANCA liegen folgende Materialien vor:

    V 121/87        Ein alter Klassiker der Filmgeschichte
                    S/W 103 min.

Weiter mit beliebiger Taste
```

Abb. 2: Die Materialsuche kann sehr flexibel gestaltet werden...

in den Zeilen 560-660 vorgenommen werden. In Zeile 560 wird die Anzahl der Eingabefelder, in den übrigen Zeilen die Benennung, die Bildschirmposition und ihre Länge bestimmt.

Die Eingabe der Informationen erfolgt über die Eingabefelder, Tippfehler sind durch Überschreiben zu korrigieren. Solange die Eingabe nicht beendet wurde, kann zwischen den Feldern mit den Cursor-Tasten in alle Richtungen und mit der RETURN-Taste nach unten gesprungen werden.

Ein weiteres Kennzeichen bei der Eingabe von Zeichen ist, daß Komma und Semikolon vom Programm nicht angenommen werden, da eine Verwendung dieser beiden Zeichen Probleme beim Lesen ergeben.

Der Abschluß der Eingabe erfolgt über die EXIT-Taste. An der zentralen Stelle der Auswahlmöglichkeiten erscheint der Hinweis auf das ausschließliche Beenden des Programms über die E-Taste. Diesen Hinweis zu beachten, ist sehr wichtig, da die indizierten Dateien ordentlich geschlossen werden müssen, was bei den sequentiellen Dateien z.B. nicht der Fall sein muß. Sollte aufgrund eines Fehlers das Programm einmal selbst enden bzw. durch STOP beendet werden, muß zur Sicherung der Daten unbedingt ein »close file%« oder »close 1« eingegeben werden. Ein unspezifizierter »CLOSE«-Befehl genügt nicht!

Es ist sinnvoll, um den Verlust der Daten zu verhindern, zusätzlich eine Sicherheitskopie der Daten anzufertigen. Wenn Informationen gesucht werden, wird man gefragt, ob eine Liste der vorhandenen Schlagwörter ausgewiesen werden soll. Auf diese Weise hat man ständig eine Übersicht über die gespeicherten Zugriffsmöglichkeiten.

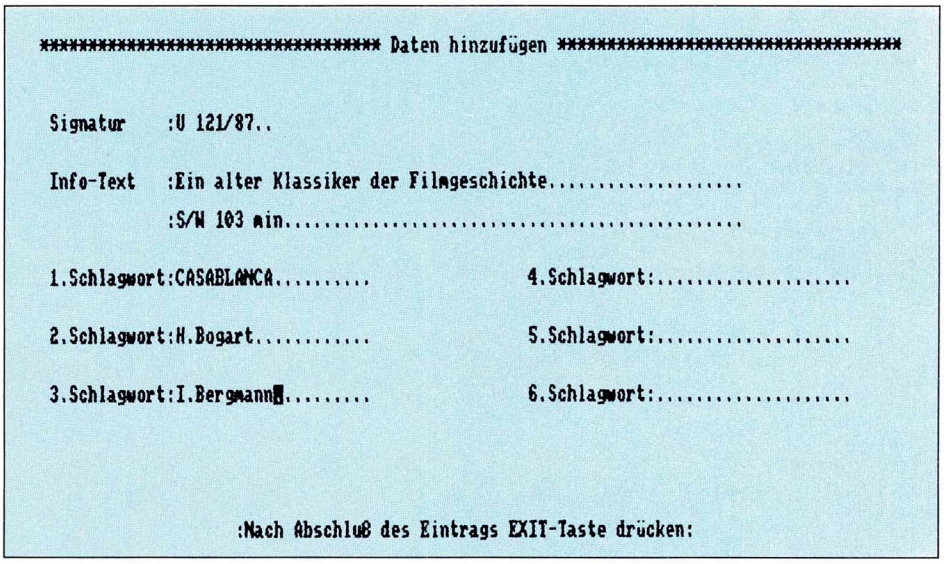


Abb. 3: Die Eingabemaske der Archiv-Verwaltung...

Wenn das gesuchte Schlagwort eingegeben wird, listet das Programm alle unter diesem Stichwort geführten Signaturen und Informationstexte auf. Dieses ist das eigentliche Herzstück der Archiv-Verwaltung.

Daneben gibt es die Möglichkeiten, die Liste der Schlagwörter, die unter dem Stichwort verzeichneten Materialien und eine vollständige Liste aller Signaturen und Texte ausdrucken zu lassen. Die Möglichkeit der Veränderung von Eintragungen ist für eine Archiv-Verwaltung von großer Bedeutung, um sie ständig auf einem aktuellen Stand zu halten. Mit Programmen, die mit sequentiellen Dateien arbeiten, gibt es diesbezüglich große Probleme.

Die indizierten Dateien lassen sich dagegen ständig verändern. Ich habe in diesem Programm die Möglichkeit gewählt, einen Satz mit allen seinen Schlüsseln zu löschen. Danach läßt sich eine neuer Satz aufnehmen. Das Löschen beginnt mit der Eingabe der Signatur, worauf nach Bestätigung der richtigen Auswahl die Löschung vorgenomen wird.

Mit diesem Programm lassen sich sehr umfangreiche Archive verwalten. Der Arbeitsspeicher begrenzt nicht mehr die Möglichkeiten. Sie sind nur durch die Speicherkapazität der Diskette eingeschränkt.

(Friedhelm Sauerländer)

```
LISTING >ARCHIV<, REMARK = >'<.

<16> 10 '*** Archivverwaltung ***
<78> 20 :
<21> 30 '*** Bildschirmaufbau ***
< 1> 40 GOSUB 400
<81> 50 :
<82> 60 '*** Datei auswählen und Variablen belegen ***
<83> 70 :
< 1> 80 befehl=0:WHILE befehl<9
<65> 90 PRINT FNkopf$(" Archivverwaltung "):PRINT:PRINT
<79> 100 PRINT:PRINT"Auswahl des Archivs:":PRINT:PRINT
     TAB(35);"Loseblattsammlung:.. L":PRINT:PRINT TAB(3
     5);"Video:..... V":PRINT:PRINT TAB(35);"B
     ibliothek:..... B":PRINT:PRINT TAB(35);"Ton:..
     ..... T"
<16> 110 match$="LlVvBbTt":GOSUB 900:befehl=antwort
<19> 120 ON befehl GOSUB 690,690,710,710,730,730,750,75
     0:GOTO 140
```

```
<85> 130 WEND
<18> 140 a=20:REM Länge des Schlagwortes (1-20 Zeichen)
<59> 150 file%=1
<26> 160 :
<51> 170 '*** Datei festlegen ***
<30> 180 :
<95> 190 IF FIND$(filename$)<>"" AND FIND$(dateiname$)<
     >""THEN BUFFERS 12:OPEN "K",file%,filename$,datein
     ame$,2:GOTO 250
<61> 200 match$="WwNn":PRINT FNkopf$(" Dateisuche "):PR
     INT;bell$:PRINT:PRINT:PRINT:PRINT:PRINT TAB(
     20);"Die benötigte Datei ist nicht auf dieser Disk
     ette.":PRINT:PRINT TAB(20)"Diskette wechseln oder
     neue Datei einrichten? (W/N)":GOSUB 900
<35> 210 IF antwort=1 OR antwort=2 THEN GOSUB 1000
<16> 220 IF antwort=3 OR antwort=4 THEN GOSUB 800:GOTO
     190
```

```
<68> 230 RESET
< 1> 240 GOTO 190
<92> 250 FIELD file%,10 AS rec.sign$,116 AS rec.info$
<27> 260 :
<76> 270 befehl=0
<31> 280 WHILE befehl <9
<40> 290 PRINT FNkopf$(" Archivverwaltung "):PRINT:PRIN
     T TAB(3);"A C H T U N G - Beenden des Programms im
     mer über ENDE - sonst VERLUST aller Daten!!!":PRIN
     T:PRINT:PRINT
<16> 300 PRINT:PRINT:PRINT TAB(20);"AUSWAHL:":PRINT:PRI
     NT TAB(40);"SUCHEN:..... S":PRINT:PRINT TAB(40);"
     HINZUFÜGEN:.. H":PRINT:PRINT TAB(40);"DRUCKEN:....
     . D":PRINT:PRINT TAB(40);"VERÄNDERN:... V":PRINT:P
     RINT TAB(40);"ENDE:........ E"
<84> 310 match$="SsHhDdVvEe":GOSUB 900:befehl = antwort
<21> 320 ON befehl GOSUB 2000,2000,2400,2400,3400,3400,
     3700,3700:GOSUB 1000
<87> 330 WEND
<24> 340 :
<40> 350 '*** Abschluß ***
<28> 360 :
<53> 370 CLOSE file%:PRINT cls$:PRINT curs.an$
<41> 380 END
<34> 390 :
<28> 400 '*** Bildschirmaufbau ***
<19> 410 :
<96> 420 escape$=CHR$(27):bell$=CHR$(7):cls$=escape$+"E
     "+escape$+"H":crlf$=CHR$(13)+CHR$(10)
<23> 430 DEF FNplaz$(x,y)=escape$+"Y"+CHR$(y+32)+CHR$(x
     +32)
<67> 440 DEF FNkopf$(titel$) = cls$+STRING$((88-LEN(tit
     el$))/2,"*")+""+titel$+""+STRING$((88-LEN(titel$))
     /2,"*")+crlf$+crlf$
<56> 450 curs.rechts$=escape$+"C":curs.links$=escape$+"
     D":curs.an$=escape$+"e":curs.aus$=escape$+"f":PRIN
     T curs.aus$
<69> 460 stzchn=6:FOR n=1 TO stzchn
< 9> 470 READ stzchn%(n)
<53> 480 NEXT
<69> 490 READ feld:
<15> 500 FOR n=1 TO feld
<56> 510 READ nom$(n),x(n),y(n),lg(n)
<42> 520 NEXT
<77> 530 RETURN
<26> 540 :
<98> 550 DATA 13,30,6,1,31,27
<13> 560 DATA 10
<60> 570 DATA "Signatur... ",1,4,10
<27> 580 DATA "Info-Text.. ",1,7,58
<21> 590 DATA "..... ",1,9,58
<97> 600 DATA "1.Schlagwort",1,12,20
<95> 610 DATA "2.Schlagwort",1,15,20
<93> 620 DATA "3.Schlagwort",1,18,20
<86> 630 DATA "4.Schlagwort",50,12,20
<87> 640 DATA "5.Schlagwort",50,15,20
<88> 650 DATA "6.Schlagwort",50,18,20
<48> 660 DATA ":Nach Abschluß des Eintrags EXIT-Taste d
     rücken",20,25,0
<33> 670 :
<24> 680 '*** Dateinamen zuweisen ***
<27> 690 filename$="archiv1.rnd":dateiname$="archiv1.in
     d"
<73> 700 RETURN
<32> 710 filename$="video1.rnd":dateiname$="video1.ind"
<77> 720 RETURN
<42> 730 filename$="bibl1.rnd":dateiname$="bibl1.ind"
<81> 740 RETURN
<21> 750 filename$="ton1.rnd":dateiname$="ton1.ind"
<85> 760 RETURN
<34> 770 :
<23> 800 '*** leere Datei erstellen ***
<23> 810 :
<19> 820 PRINT FNkopf$(" Erstellung einer neuen Datei "
     ):PRINT:PRINT:PRINT:PRINT:PRINT:PRINT TAB(37
     );"Bitte warten."
<19> 830 RESET:BUFFERS 12: CREATE file%,filename$,datei
     name$,2:
<79> 840 FIELD file%,10 AS rec.test$:LSET rec.test$=SPA
     CE$(10):ergebnis=ADDREC(file%,0,2,SPACE$(10))
<62> 850 CLOSE file%:PRINT:PRINT:PRINT:PRINT TAB(37);"D
     atei erzeugt."
<86> 860 RETURN
<35> 870 :
<45> 900 '*** Taste holen *****
<24> 910 :
<93> 920 antwort$ = INKEY$
<58> 930 WHILE antwort$=""
<82> 940 antwort$=INKEY$
<97> 950 WEND
```

```
<69> 960 antwort = INSTR(match$,antwort$): IF antwort =
      0 THEN GOTO 920
<89> 970 RETURN
<38> 980 :
<65> 1000 '*** auf Taste warten ***
< 2> 1010 :
<32> 1020 PRINT:PRINT"Weiter mit beliebiger Taste"
<46> 1030 WHILE INKEY$=""
<23> 1040 WEND
<93> 1050 RETURN
<17> 1060 :
<24> 2000 '*** Unterroutinen ***
< 3> 2010 :
<19> 2020 '*** Suchen ***
< 9> 2030 :
<22> 2040 befehl=0:WHILE befehl<3
<97> 2050 PRINT cls$:PRINT FNkopf$(" Materialsuche "):P
     RINT:PRINT"Wünschen Sie eine Schlagwortübersicht?
     (j/n)"
<21> 2060 match$="JjNn":GOSUB 900:befehl=antwort
<53> 2070 ON befehl GOSUB 2250,2250:GOTO 2090
<36> 2080 WEND
<28> 2090 PRINT curs.an$:PRINT:PRINT:INPUT "SCHLAGWORT
     eingeben: ",name$
<82> 2100 PRINT curs.aus$:n$=name$+SPACE$(30)
< 2> 2110 z=0:l=1
<65> 2120 schl$=UPPER$(LEFT$(n$,a))
<91> 2130 PRINT FNkopf$(" Materialsuche "):PRINT
<21> 2140 ergebnis=SEEKKEY(file%,0,1,schl$)
<18> 2150 IF ergebnis=0 THEN PRINT "Zum Schlagwort ";UP
     PER$(name$);" liegen folgende Materialien vor:":PR
     INT
<97> 2160 IF ergebnis=0 THEN GET file%:GOSUB 2230
<45> 2170 IF ergebnis<>0 THEN PRINT:PRINT"Zum Schlagwor
     t ";UPPER$(name$);" liegen keine Materialien vor."
     :PRINT:PRINT:PRINT:RETURN
<64> 2180 ergebnis=SEEKNEXT(file%,0)
<26> 2190 IF ergebnis=0 THEN GET file%:GOSUB 2230:z=z+1
<55> 2200 IF z=6*l THEN GOSUB 1000:l=l+1:PRINT FNkopf$(
     " Materialsuche "):PRINT TAB(5);"Signatur:";TAB(20
     );"Information:":PRINT
<80> 2210 IF ergebnis=0 THEN GOTO 2180 ELSE RETURN
<10> 2220 :
<73> 2230 PRINT TAB(5);rec.sign$;TAB(20);LEFT$(rec.info
     $,58):PRINT TAB(20);RIGHT$(rec.info$,58):PRINT:RET
     URN
<16> 2240 :
<32> 2250 PRINT FNkopf$(" Schlagwortregister ")
<22> 2260 :
<22> 2270 z=0:l=1
<25> 2280 ergebnis=SEEKRANK(file%,1,1)
<78> 2290 k$=FETCHKEY$(file%):PRINT k$:z=z+1
<85> 2300 ergebnis=SEEKSET(file%,1)
<87> 2310 IF ergebnis=101 THEN k$=FETCHKEY$(file%):PRIN
     T k$:z=z+1
<51> 2320 IF z=25*l THEN GOSUB 1000:l=l+1:PRINT FNkopf$
     (" Schlagwortregister ")
<57> 2330 IF ergebnis=101 THEN GOTO 2300 ELSE RETURN
<18> 2340 :
<19> 2400 '*** Satz hinzufügen ***
<28> 2410 PRINT FNkopf$(" Daten hinzufügen ")
<28> 2420 ende=0
<85> 2430 ergebnisa=0:ergebnisb=0:ergebnisc=0:ergebnisd
     =0:ergebnise=0
<41> 2440 GOSUB 2850
<40> 2450 sign$=eingabe$(1)
<24> 2460 info1$=eingabe$(2)
<59> 2470 info2$=eingabe$(3)
< 5> 2480 info$=info1$+info2$
<54> 2490 schlag1$=eingabe$(4)
<65> 2500 schlag2$=eingabe$(5)
< 5> 2510 schlag3$=eingabe$(6)
<44> 2520 schlag4$=eingabe$(7)
<83> 2530 schlag5$=eingabe$(8)
<23> 2540 schlag6$=eingabe$(9)
<79> 2550 LSET rec.sign$=sign$
<13> 2560 LSET rec.info$=info$
<64> 2570 schl1$=UPPER$(LEFT$(schlag1$,a))
<10> 2580 schl2$=UPPER$(LEFT$(schlag2$,a))
<55> 2590 schl3$=UPPER$(LEFT$(schlag3$,a))
<72> 2600 schl4$=UPPER$(LEFT$(schlag4$,a))
<18> 2610 schl5$=UPPER$(LEFT$(schlag5$,a))
<63> 2620 schl6$=UPPER$(LEFT$(schlag6$,a))
< 7> 2630 schl7$=UPPER$(LEFT$(sign$,10))
<25> 2640 such$=SPACE$(a)
<26> 2650 pruefen=INSTR(schl1$,such$):IF pruefen=1 THEN
      PRINT FNkopf$(" F e h l e r "):PRINT:PRINT:PRINT:
     PRINT:PRINT:PRINT TAB(10);"Das 1. Schlagwort wurde
      nicht eingetragen. Bitte den Eintrag wiederholen.
```

```
     ":PRINT:PRINT:PRINT:RETURN
<63> 2660 pruefen=INSTR(schl7$,SPACE$(10)):IF pruefen=1
      THEN PRINT FNkopf$(" F e h l e r "):PRINT:PRINT:P
     RINT:PRINT:PRINT:PRINT:PRINT TAB(10);"Die Signatur
      wurde nicht eingetragen. Bitte den Eintrag wieder
     holen.":PRINT:PRINT:PRINT:RETURN
<62> 2670 ergebnisa = ADDREC(file%,2,1,schl1$)
<45> 2680 nummer=FETCHREC(file%)
<43> 2690 pruefen=INSTR(schl2$,such$):IF pruefen=1 THEN
      GOTO 2710
<90> 2700 ergebnisb=ADDKEY(file%,0,1,schl2$,nummer)
<64> 2710 pruefen=INSTR(schl3$,such$):IF pruefen=1 THEN
      GOTO 2730
<48> 2720 ergebnisc=ADDKEY(file%,0,1,schl3$,nummer)
<14> 2730 pruefen=INSTR(schl4$,such$):IF pruefen=1 THEN
      GOTO 2750
<34> 2740 ergebnisf=ADDKEY(file%,0,1,schl4$,nummer)
<63> 2750 pruefen=INSTR(schl5$,such$):IF pruefen=1 THEN
      GOTO 2770
<91> 2760 ergebnisg=ADDKEY(file%,0,1,schl5$,nummer)
<13> 2770 pruefen=INSTR(schl6$,such$):IF pruefen=1 THEN
      GOTO 2790
<49> 2780 ergebnish=ADDKEY(file%,0,1,schl6$,nummer)
< 2> 2790 ergebnisd=ADDKEY(file%,0,0,schl7$,nummer)
< 0> 2800 ergebnise=CONSOLIDATE(file%)
<27> 2810 IF ergebnisa=0 AND ergebnisb=0 AND ergebnisc=
     0 AND ergebnisd=0 AND ergebnise=0 AND ergebnisf=0
     AND ergebnisg=0 AND ergebnish=0 THEN PRINT cls$:PR
     INT:PRINT "Neue Einträge geschrieben":RETURN
<26> 2820 PRINT:PRINT"Die letzten Einträge wurden nicht
      korrekt verarbeitet. Das Programm ist beendet und
      kann neu gestartet werden.":GOTO 350
< 5> 2830 RETURN
<28> 2840 :
<54> 2850 feldnr=1
< 5> 2860 FOR n=1 TO feld
<22> 2870 PRINT FNplaz$(x(n),y(n));
<64> 2880 PRINT nom$(n);":";STRING$(lg(n),"."):eingabe$
     =STRING$(lg(n)," ")
<73> 2890 eingabe$(n)=STRING$(lg(n)," ")
< 2> 2900 NEXT
<55> 2910 x=x(feldnr):y=y(feldnr)
<30> 2920 flg=lg(feldnr)
<45> 2930 x=x+LEN(nom$(feldnr))+1
<20> 2940 ein$=eingabe$(feldnr)
<33> 2950 GOSUB 3030
< 1> 2960 eingabe$(feldnr)=ein$
<18> 2970 IF ende THEN RETURN
<39> 2980 IF oben THEN feldnr=feldnr-1:IF feldnr=0 THEN
      feldnr=1
<38> 2990 IF unten THEN feldnr=feldnr+1:IF feldnr>feld
     THEN feldnr=feld
<30> 3000 unten=0:oben=0
<30> 3010 GOTO 2910
<91> 3020 PRINT bell$;:GOTO 3070
<10> 3030 posit=x
<36> 3040 PRINT FNplaz$(x,y);
<85> 3050 lg=1
<18> 3060 PRINT curs.an$;
<76> 3070 zchn$=INKEY$:IF zchn$="" THEN 3070
< 7> 3080 zchn%=ASC(zchn$)
<42> 3090 FOR n=1 TO stzchn:IF zchn%<>stzchn%(n) THEN N
     EXT
<54> 3100 ON n GOTO 3170,3170,3200,3220,3240,3250
<58> 3110 kor=1
<77> 3120 GOSUB 3260
<67> 3130 IF kor=0 THEN 3020
<28> 3140 IF posit>=x+flg THEN 3020
<63> 3150 ein$=LEFT$(ein$,posit-x)+zchn$+RIGHT$(ein$,fl
     g-(posit-x)-1)
<77> 3160 PRINT zchn$;:posit=posit+1:GOTO 3070
<97> 3170 unten=1
<56> 3180 PRINT curs.aus$;
<10> 3190 RETURN
<29> 3200 IF posit=x+flg THEN 3020
<40> 3210 PRINT curs.rechts$;:posit=posit+1:GOTO 3070
<69> 3220 IF posit=x THEN 3020
<34> 3230 PRINT curs.links$;:posit=posit-1:GOTO 3070
<76> 3240 oben=1:GOTO 3180
<14> 3250 ende=1:GOTO 3180
<20> 3260 IF zchn%<44 THEN RETURN
<47> 3270 IF zchn%>59 THEN RETURN
<44> 3280 IF zchn%>44 AND zchn%<59 THEN RETURN
<59> 3290 kor=0:RETURN
< 7> 3300 :
<50> 3400 '*** Ausdruck ***
<12> 3410 :
< 3> 3420 PRINT FNkopf$(" Ausdruck ")
<41> 3430 befehl=0:WHILE befehl<7
<69> 3440 PRINT:PRINT:PRINT TAB(20);"Auswahl:":PRINT:PR
     INT TAB(35);"Schlagwortregister:.... S":PRINT:PRIN
     T TAB(35);"Materialien:.......... M":PRINT:PRINT
     TAB(35);"Liste:................. L"
<10> 3450 match$="SsMmLl":GOSUB 900:befehl=antwort
<34> 3460 ON befehl GOSUB 3480,3480,3550,3550,3650,3650
     :befehl=0:RETURN
<42> 3470 WEND
<60> 3480 PRINT FNkopf$(" Ausdruck des Schlagwortregist
     ers ")
<33> 3490 ergebnis=SEEKRANK(file%,1,1)
<17> 3500 k$=FETCHKEY$(file%):LPRINT TAB(10);k$
<93> 3510 ergebnis=SEEKSET(file%,1)
< 7> 3520 IF ergebnis=101 THEN k$=FETCHKEY$(file%):IF L
     EFT$(k$,2)<>"üü" THEN LPRINT TAB(10);k$
<95> 3530 IF ergebnis=101 THEN GOTO 3510 ELSE RETURN
<23> 3540 :
<46> 3550 PRINT FNkopf$(" Ausdruck vorhandener Material
     ien "):PRINT:
<86> 3560 PRINT curs.an$:PRINT:PRINT:INPUT "SCHLAGWORT
     eingeben:. ",name$
<75> 3570 LPRINT TAB(10);"Zum Schlagwort ";UPPER$(name$
     );" liegen folgende Materialien vor: ":LPRINT:PRIN
     T curs.aus$
<97> 3580 schl$=UPPER$(LEFT$(name$+SPACE$(20),a))
<45> 3590 ergebnis=SEEKKEY(file%,0,1,schl$)
<57> 3600 IF ergebnis=0 THEN GET file%:LPRINT TAB(10);r
     ec.sign$:LPRINT TAB(10);LEFT$(rec.info$,58):LPRINT
      TAB(10);RIGHT$(rec.info$,58)
<54> 3610 ergebnis=SEEKNEXT(file%,0)
<63> 3620 IF ergebnis=0 THEN GET file%:LPRINT TAB(10);r
     ec.sign$:LPRINT TAB(10);LEFT$(rec.info$,58):LPRINT
      TAB(10);RIGHT$(rec.info$,58)
<56> 3630 IF ergebnis=0 THEN GOTO 3610 ELSE RETURN
<25> 3640 :
<96> 3650 PRINT FNkopf$(" Liste nach Signaturen ")
<25> 3660 ergebnis=SEEKRANK(file%,0,0):IF ergebnis=0 TH
     EN GET file%:LPRINT TAB(10);rec.sign$:LPRINT TAB(1
     0);LEFT$(rec.info$,58):LPRINT TAB(10);RIGHT$(rec.i
     nfo$,58)
<46> 3670 ergebnis=SEEKSET(file%,0):IF ergebnis=101 THE
     N GET file%:LPRINT TAB(10);rec.sign$:LPRINT TAB(10
     );LEFT$(rec.info$,58):LPRINT TAB(10);RIGHT$(rec.in
     fo$,58)
<52> 3680 IF ergebnis=101 THEN GOTO 3670 ELSE RETURN
<40> 3690 :
< 5> 3700 '*** Verändern ***
<18> 3710 :
<18> 3720 PRINT FNkopf$(" Verändern von Einträgen "):PR
     INT:PRINT:PRINT:PRINT:PRINT
<35> 3730 PRINT curs.an$:PRINT:PRINT TAB(25);"Die Verän
     derung erfolgt über die Signatur.":PRINT:PRINT TAB
     (20)"Bitte die Signatur eingeben und RETURN drücke
     n. ":PRINT TAB(40):INPUT":",sign$
<24> 3740 PRINT curs.aus$:a$=sign$+SPACE$(10):schl$=UPP
     ER$(LEFT$(a$,10)):PRINT:PRINT
< 6> 3750 ergebnis=SEEKKEY(file%,0,0,schl$)
<38> 3760 IF ergebnis=0 THEN GET file%:PRINT TAB(10);re
     c.sign$;TAB(25);LEFT$(rec.info$,58):PRINT TAB(25);
     RIGHT$(rec.info$,58)
<82> 3770 IF ergebnis<>0 THEN PRINT FNkopf$(" F e h l e
      r "):PRINT:PRINT:PRINT:PRINT TAB(20);"Mater
     ial mit der eingegebenen Signatur ist nicht vorhan
     den.":PRINT:PRINT:PRINT:PRINT:RETURN
<71> 3780 match$="JjNn":PRINT:PRINT:PRINT"Ist dieses da
     s gesuchte Material? (J/N)":GOSUB 900
<41> 3790 IF antwort=1 OR antwort=2 THEN GOTO 3820
<88> 3800 ergebnis=SEEKNEXT(file%,0):IF ergebnis=0 THEN
      GET file%:PRINT TAB(10);rec.sign$;TAB(25);rec.inf
     o$
<12> 3810 IF ergebnis=0 THEN GOTO 3780 ELSE RETURN
<52> 3820 erg=DELKEY(file%,0)
<34> 3830 identa$=rec.sign$:identb$=LEFT$(rec.info$,20)
< 0> 3840 ergebnis=SEEKRANK(file%,0,1):IF ergebnis=0 TH
     EN GET file%:IF rec.sign$=identa$ AND LEFT$(rec.in
     fo$,20)=identb$ THEN IF DELKEY(file%,0) THEN GOTO
     3840
<47> 3850 WHILE ergebnis=0 OR ergebnis=101
<15> 3860 ergebnis=SEEKNEXT(file%,1):IF ergebnis>101 TH
     EN GOTO 3890
<63> 3870 GET file%:IF rec.sign$=identa$ AND LEFT$(rec.
     info$,20)=identb$ THEN IF DELKEY(file%,0) THEN GOT
     O 3870
<53> 3880 WEND
<64> 3890 match$="JjNn":PRINT:PRINT"Soll ein neuer Satz
      aufgenommen werden? (J/N)":GOSUB 900
<94> 3900 IF antwort=1 OR antwort=2 THEN GOSUB 2400
< 2> 3910 RETURN
<25> 3920 :
```

PASCOM



PASCOM ist ein Mini-Pascalcompiler zum Kennenlernen der Programmiersprache PASCAL oder zum Erstellen kleinerer Maschinenspracheprogramme. Damit diese Programme in Mallard-BASIC eingebunden werden können, erstellt PASCOM nach dem Compilieren auf Wunsch ein BASIC-Programm, das das Maschinenprogramm aus DATA-Zeilen generiert.

Ein Compiler ist ein Programm, welches eine Hochsprache wie Basic, Pascal oder andere, die dem Benutzer den komfortablen Dialog mit dem Computer ermöglichen, in Maschinencode übersetzt, der direkt vom jeweiligen Prozessor ausgeführt werden kann. Vorteil eines compilierten Programms ist die sehr hohe Abarbeitungsgeschwindigkeit. Bei PASCOM stehen folgende Befehle zur Verfügung:

PROGRAM
PROCEDURE
BEGIN
CLRSCR
INVERSE/NORMAL
INVERSE SCREEN/NORMAL SCREEN
+/-
SHR/SHL
GOTOXY
CHR
WRITE/WRITELN
READ/READLN
POKE/POKEX
PEEK/PEEKX
FOR-TO-DO
FOR-DOWNTO-DO
IF = THEN ELSE
IF > THEN ELSE
REPEAT-UNTIL
HALT
END

PASCOM arbeitet mit 29 Integervariablen (a-ü), denen er Zahlen als Ganzzahl oder Hexadezimalzahl zuordnen und die er beliebig addieren oder subtrahieren kann.

Das Source-File (das Programm, welches Sie schreiben) kann entweder zusätzlich zu PASCOM im Speicher erstellt werden, dann muß jede Zeilennummer kleiner als 20000 sein und die Zeile nach der Zeilennummer mit einem Apostroph beginnen, oder auf Diskette als ASCII-File vorliegen. Dieses ASCII-File kann im Prinzip mit jedem Editor (Wordstar, Locoscript Modus 'ASCII-File erstellen', BASIC etc.) erstellt werden und beliebig lang sein. Auch beim ASCII-File kann, muß jedoch nicht, jede Zeile mit einer Zeilennummer und einem Apostroph beginnen.

Pro Zeile darf grundsätzlich nur ein Befehl stehen. Am Ende jeder Zeile erwartet PASCOM ein ';' (Ausnahmen: BEGIN, REPEAT, ELSE, DO). Es sollten keine zusätzlichen Leerstellen innerhalb der Zeile gesetzt werden, nur am Zeilenanfang werden sie ignoriert. Es folgt eine kurze Beschreibung der Befehle. 'var', 'var1' etc. bedeuten Variable, die auch durch Integerzahlen (dezimal oder hexadezimal) ersetzt werden können.

PROGRAM name;

Obligate erste Zeile jedes PASCAL-programmes, wobei 'name' der Programmname ist.

PROCEDURE name;

Definiert ein Unterprogramm, bei PASCAL 'Procedure' genannt, mit dem Prozedurnamen 'name'. Bei PASCOM sind bis zu 20 Prozeduren möglich, die alle vor dem Beginn des Hauptprogramms definiert werden müssen. Im späteren Programm werden die Prozeduren durch ihren Namen 'name' aufgerufen.

BEGIN

Leitet eine Befehlsfolge ein. Steht als Anfang von Prozeduren, IF-THEN-ELSE Bedingungen, FOR-TO/DOWNTO DO Schleifen und zu Beginn des Hauptprogramms.

END

Beendet die letzte mit BEGIN eingeleitete Befehlsfolge. Das 'END' des Hauptprogramms muß 'END.' lauten (Abschluß durch Punkt).

var1:=SHR=SHL(var2);

Schiebt eine Zahl bitweise eine Stelle nach Rechts oder Links. Im Klartext heißt das: SHL multipliziert eine Zahl mit zwei, wobei das Ergebnis nicht größer als 65535 sein darf. SHR teilt eine Zahl durch zwei. Wenn 'var2' dabei ungerade ist, wird das Ergebnis abgerundet.

CLRSCR;

Löscht den Bildschirm und bringt den Cursor in die obere linke Ecke.

GOTOXY(var1,var2);

Führt die nächste Textausgabe mit CHR oder WRITE oder die nächste Eingabe mit READLN an der angegebenen Stelle aus. Dabei bedeutet 'var1' eine Zeile von 1 bis 32, 'var2' eine Spalte von 1 bis 90.

INVERSE;/NORMAL;

INVERSE stellt jede Textausgabe auf dem Bildschirm invers dar, bis ein Befehl NORMAL folgt.

INVERSE/NORMAL SCREEN;

INVERSE SCREEN invertiert den Bildschirm, also stellt dunkle Schrift auf hellem Grund dar.NORMAL SCREEN schaltet wieder auf normale Darstellung um.

CHR(var);

Gibt das Zeichen mit dem ASCII-Wert 'var' auf dem Bildschirm aus.

WRITE/WRITELN(var/'Text');

Mit dem WRITE-Befehl lassen sich entweder Text oder Variablen auf dem Bildschirm ausgeben. Der Text muß in Apostrophs eingeschlossen sein, sonst wird er als Variable interpretiert. Die Variablen werden in hexadezimaler Form ausgegeben. WRITELN (Write-Line) führt nach der Ausgabe einen Zeilenumbruch aus.

READ(var);

Liest ein Zeichen von der Tastatur und ordnet dessen ASCII-Wert der Variablen 'var' zu.

READLN(var);

Liest eine Ganzzahl (0..65535) oder eine Hexadezimalzahl (Präfix '&H' wie im Mallard-BASIC) ein und ordnet sie der Variablen 'var' zu. Die Eingabe wird mit 'RETURN' abgeschlossen.

POKE/POKEX(var1,var2);

Schreibt den Wert der Variablen 'var2' an die Speicherstelle 'var1'. POKEX (für Extended) beschreibt zusätzlich die nächste Speicherstelle (var1+1) mit dem High-Byte von 'var2'.

var1:=PEEK/PEEKX(var2);

Der Variablen 'var1' wird der Inhalt der Speicherstelle 'var2' zugeordnet. Bei PEEKX erfolgt die Zuordnung einer 16-Bit-Integer-Zahl aus den beiden Speicherstellen 'var2' und 'var2+1'.

IF var1=var2/var1>var2 THEN BEGIN END; ELSE

Ermöglicht bedingt auszuführende Befehle. Die komplementären Vergleichsoperatoren ('<>', '<=' werden durch die ELSE Bedingung abgedeckt). Nach BEGIN folgt eine Reihe von Befehlen, die ausgeführt wird, wenn die Bedingung wahr ist, und mit END abgeschlossen werden muß. Falls ein ELSE folgen soll, muß es in derselben Zeile wie das END stehen. In der nächsten Zeile beginnt dann (mit BEGIN) die Reihe von Befehlen, die nur dann ausgeführt werden soll, wenn die Bedingung unwahr ist. Zum Abschluß folgt wieder ein END.

REPEAT UNTIL var1=var2;

Die REPEAT-UNTIL-Schleife wiederholt einen Programmteil mehrfach, bis die Bedingung nach UNTIL erfüllt

```
10 ' program Primzahl;
20 ' begin
30 '   repeat
40 '     clrscr;
50 '     gotoxy(10,1);
60 '     writeln('Dieses Programm
         druckt Primzahlen aus!');
70 '     write('Bitte geben Sie
         den Grenzwert ein : ');
80 '     readln(z);
90 '     z:=shr(z);
100 '    x:=20000;
110 '    y:=z+x;
120 '    for a:=x to y do
130 '    begin
140 '      poke(a,0);
150 '    end;
160 '    for a:=1 to z do
170 '    begin
180 '      b:=a+x;
190 '      c:=peek(b);
200 '      if c=0 then begin
210 '        c:=shl(a);
220 '        c:=c+1;
230 '        write(c);
240 '        chr(&h20);
250 '        b:=b+c;
260 '        if b>c then begin
270 '          for b:=b to y do
280 '          begin
290 '            poke(b,&hc7);
300 '            b:=b+c;
310 '            b:=b-1;
320 '          end;
330 '        end;
340 '      end;
350 '    end;
360 '    read(z);
370 '  until z=32;
380 'end.
```

Abb. 1: Ein Pascal-Beispielprogramm zum Ausdruck von Primzahlen

ist. (Achtung, daß sie auch wirklich erfüllt wird;)

FOR var1:=var2 TO/DOWNTO var3 DO BEGIN

Führt einen Programmteil so oft aus, wie in einer vorgegebenen Zahl angegeben, festgelegt durch die Differenz zwischen 'var2' und 'var3'. Dabei wird die Kontrollvariable 'var1' bei TO in jedem Schleifendurchgang um eins erhöht, bei DOWNTO erniedrigt. Abgeschlossen wird die Schleife mit END. (Indem man in der Schleife die Kontrollvariable ändert, sind auch andere Schrittweiten als eins möglich, (siehe Beispielprogramm)

HALT;

Bewirkt einen Rücksprung in die vorhergehende Ebene. In einer Prozedur springt man an dieser Stelle zurück ins Hauptprogramm, im Hauptprogramm zurück ins BASIC. (Nützlich beim Programmtesten)

Addition, Subtraktion

Die Rechenfunktionen erfolgen nach dem gleichen Syntax wie die Variablenzuordnungen:

var1:=var2+/-var3;

'var1' und 'var2' müssen Variablen sein, 'var3' kann entweder Variable oder Integerzahl sein. Integerzahl heißt bei PASCOM immer Hexadezimalzahl (z.B. &HFF9E) oder dezimale Ganzzahl von 0 bis 65535.

Fehlermeldungen

Natürlich kann bei einem MiniCompiler wie PASCOM nicht jeder Fehler ausführlich kommentiert werden. Die meisten gravierenden Fehler werden jedoch mit einer kurzen Fehlermeldung, die sich selbst erklärt, und dem Abbruch des Compilierens belohnt.

Die Programmerstellung läuft so ab, daß zuerst das Source-File erstellt wird. Dies ist, wie oben erwähnt, in BASIC im Kernspeicher möglich, dann sollte PASCOM vorher geladen werden. Eine andere Möglichkeit ist die Erstellung eines ASCII-Files z.B. mit Locoscript (aber Achtung: dessen Zeichensatz ist bei den Sonderzeichen und Umlauten nicht mit dem CP/M-Zeichensatz identisch!). Pascal-Programme, die mit dem Mallard-Basic-Editor erstellt worden sind, müssen, wie schon erwähnt, mit einem ' beginnen und nach Fertigstellung mit *SAVE "Programmname",a* abgespeichert werden. Nach dem Aufruf von PASCOM wird zuerst gefragt, ob das Sourcefile im RAM steht oder auf Diskette vorliegt. Wenn letzteres zutrifft, fragt PASCOM nach Programmnamen und Startadresse für das Maschinenprogramm. So können auch längere Maschinenprogramme erstellt werden. (Aber Achtung: Startadressen unter &HB000 lassen PASCOM nicht genügend Platz, und wenn Adressen ab &HF606 überschrieben werden, wird das BDOS zerstört. Dann hilft meist nur noch ausschalten...). Wenn das Sourcefile im RAM steht, setzt PASCOM die Startadresse automatisch auf &HE800 und beginnt mit dem Compilieren.

Nach getaner Arbeit bietet PASCOM die Möglichkeit, das Source-File auf Diskette zu speichern, wenn es dort nicht schon vorlag, und einen BASIC-Loader zu erstellen, im Klartext:

DATA-Zeilen. Die Sourcefiles erhalten das Suffix '.PAS', die BASIC-Loader '.BLD'.
Source-Files lassen sich mit Hilfe des MERGE-Befehls an PASCOM ankoppeln und können auf diese Weise schnell im Speicher weiter editiert werden. Nach dem Compilieren werden Startadresse und Länge des Maschinenprogramms sowie der freie Speicherplatz für Maschinencode angezeigt. Das Maschinenprogramm wird dann mit 'CALL adr1' aufgerufen.

Noch ein Tip für die fleißigen Abtipper: die Kommentarzeilen können entfallen, sie stellen keine Sprungziele dar.
Das Beispielprogramm (Abb.1) druckt Primzahlen aus. Es zeigt die Syntax der meisten verfügbaren Befehle. Und nun viel Spaß beim Programmieren; der erste 'Absturz' steht unmittelbar bevor...

(Olaf Schmidt)

```
LISTING >PASCOM  <, REMARK = >'<.

<47> 20000 '*******************************
<72> 20010 '**. Pascal-Compiler. PASCOM. **
< 1> 20020 '**.... (C) Olaf Schmidt..... **
<85> 20030 '**..... 3000 Hannover..... **
<63> 20040 '*******************************
<19> 20050 '
< 4> 20060 '*** Init ***
<76> 20070 DEFINT a-z:adr1=&HE800:GOSUB 20160
<43> 20080 bmax=20:DIM b$(bmax),pn$(20):DEF USR=&HF5FE
<25> 20090 GOSUB 20270:GOSUB 20310:GOSUB 20360
<77> 20100 DEF FNa$(z$)=RIGHT$(z$,LEN(z$)-1)
<74> 20110 va=adr1-60:za=&H7A96
<79> 20120 bf$=";":z$="0123456789 '"
<20> 20130 GOSUB 21010:GOSUB 21620
< 2> 20140 GOSUB 20400:GOSUB 20570:GOTO 20140
<26> 20150 '*** Disc/RAM ***
< 6> 20160 INPUT"Source-File auf Diskette oder im RAM (
     D/R) ";a$
<24> 20170 IF UPPER$(a$)<>"D" THEN MEMORY adr1-61:RETUR
     N
<29> 20180 LINE INPUT"Dateinamen eingeben : ";nam$
< 5> 20190 IF nam$="" THEN nam$="work"
< 9> 20200 IF INSTR(nam$,".")=0 THEN nam$=nam$+".PAS"
<98> 20210 PRINT UPPER$(nam$)" ";:IF nam$=".PAS" THEN 2
     0180
<13> 20220 IF FIND$(nam$)="" THEN PRINT"Nicht vorhanden
     ":FILES:PRINT:GOTO 20180
<18> 20230 LINE INPUT"Startadresse angeben ( >&HB000 )
     : ";a$
<53> 20240 adr1=UNT(VAL(a$)):IF adr1=0 THEN adr1=&HE800
<84> 20250 MEMORY adr1-61:disc=1:OPEN"i",#1,nam$:RETURN
<21> 20260 '*** def usr ***
<19> 20270 b=&HF5FE:RESTORE 20290:FOR adr=b TO b+7
<38> 20280 READ a:POKE adr,a:NEXT:RETURN
<11> 20290 DATA 35,126,14,0,113,43,119,201
<30> 20300 '*** def Befehle ***
< 8> 20310 RESTORE 20330:FOR f=1 TO bmax
< 6> 20320 READ b$(f):NEXT:RETURN
<74> 20330 DATA write,procedure,+,-,shr,shl,clrscr,inve
     rse,normal,gotoxy,chr
<73> 20340 DATA read,poke,peek,for,repeat,if,halt,begin
     ,end
<37> 20350 '*** Stackpointer retten ***
<53> 20360 RESTORE 20380:FOR f=0 TO 14:READ a
<40> 20370 POKE adr1+f,a:NEXT:adr=adr1+f+3:RETURN
<74> 20380 DATA 62,6,50,128,0,38,0,108,57,235,33,144,0,
     249,213
<89> 20390 '*** Befehle/RAM ***
<39> 20400 IF disc THEN 20530
<33> 20410 zl=UNT(PEEK(za)+256*PEEK(za+1))
<89> 20420 zn=UNT(PEEK(za+2)+256*PEEK(za+3))
<94> 20430 IF PEEK(za+5)<>182 THEN 24330
<72> 20440 IF zn>19999 THEN 22720
<63> 20450 w=UNT(za+6):za=UNT(za+zl)
<54> 20460 ba=UNT(VARPTR(bf$)):POKE ba,zl-7
<12> 20470 POKE ba+1,w AND 255
<30> 20480 POKE ba+2,USR(w)
<46> 20490 PRINT TAB(2)zn;TAB(12)bf$
<57> 20500 WHILE LEFT$(bf$,1)=" ":bf$=FNa$(bf$):WEND
< 7> 20510 a$=LOWER$(bf$):RETURN
<37> 20520 '*** Befehle/Disc ***
<94> 20530 LINE INPUT #1,bf$:PRINT bf$
<79> 20540 b$=LEFT$(bf$,1):IF INSTR(z$,b$) THEN bf$=FNa
     $(bf$):GOTO 20540
<23> 20550 a$=LOWER$(bf$):RETURN
<23> 20560 '*** Verzweigung ***
<66> 20570 FOR b=1 TO bmax:IF INSTR(a$,b$(b))=0 THEN NE
     XT:GOTO 21220
<10> 20580 ON b GOTO 21810,21050,21300,21300,21390,2139
     0,21450,21470,21470,21550,21660
<66> 20590 b=b-11:ON b GOTO 22030,22110,22190,22250,224
     50,22530,22690,21170,22710
<63> 21000 '*** program ***
<27> 21010 GOSUB 20400
<93> 21020 IF INSTR(a$,"progr")=0 THEN 24390
<94> 21030 n$=MID$(bf$,9,LEN(bf$)-9):RETURN
<53> 21040 '*** procedure ***
< 8> 21050 IF basis THEN 24370
<28> 21060 pr=pr+1:pn$(pr)=RIGHT$(a$,LEN(a$)-10)
<31> 21070 pa(pr)=adr:GOSUB 20400:beg=beg+1
<16> 21080 IF INSTR(a$,"begin")=0 THEN 24360
<84> 21090 GOSUB 20400:IF LEFT$(a$,3)<>"end" THEN GOSUB
      20570:GOTO 21090
<70> 21100 POKE adr,201:adr=adr+1:beg=beg-1:RETURN
<84> 21110 '*** procedure-Aufruf ***
<79> 21120 FOR b=1 TO 20:IF a$<>pn$(b) THEN NEXT:GOTO 2
     4340
<67> 21130 a=adr:POKE a,205:w=pa(b)
<19> 21140 POKE a+1,w AND 255:POKE a+2,USR(w)
<64> 21150 adr=adr+3:RETURN
<36> 21160 '*** begin ***
<76> 21170 IF beg THEN 24380
<50> 21180 IF basis THEN 24340
<50> 21190 w=adr:basis=w:POKE adr1+15,195
<29> 21200 POKE adr1+16,w AND 255
<83> 21210 POKE adr1+17,USR(w):RETURN
<76> 21220 '*** Variablen einlesen ***
<43> 21230 IF a$="" THEN RETURN
<23> 21240 o=INSTR(a$,":="):IF o=0 THEN 21120
<71> 21250 o$=MID$(a$,o+2,6):GOSUB 24020:w2=w
<38> 21260 GOSUB 24090:o$=LEFT$(a$,1)
<19> 21270 GOSUB 24020:IF var THEN 24230
<40> 21280 GOTO 24370
<90> 21290 '*** Addition, Subtraktion ***
<69> 21300 o=INSTR(a$,":=")+2:o$=MID$(a$,o,1)
<55> 21310 GOSUB 24020:IF var=0 THEN 24350
<86> 21320 w2=w:o$=MID$(a$,o+2,6):GOSUB 24020:GOSUB 241
     40
<94> 21330 w1=w:IF var THEN GOSUB 24160 ELSE GOSUB 2412
     0
<10> 21340 IF INSTR(a$,"-") THEN GOSUB 24300:GOTO 21360
<11> 21350 POKE adr,25:adr=adr+1
<73> 21360 o$=LEFT$(a$,1):GOSUB 24020:IF var THEN 24230
< 9> 21370 GOTO 24350
<22> 21380 '*** shr/shl ***
<40> 21390 o$=LEFT$(a$,1):GOSUB 24020:IF var=0 THEN 243
     50
<67> 21400 w1=w:GOSUB 24010:w2=w:GOSUB 24090
<43> 21410 a=adr:POKE a,203:POKE a+1,60:POKE a+3,29
<29> 21420 IF INSTR(a$,"shl") THEN POKE a+1,37:POKE a+3
     ,20
< 3> 21430 POKE a+2,203:adr=a+4:w=w1:GOTO 24230
<87> 21440 '*** clrscr ***
<31> 21450 o$=CHR$(27)+"E"+CHR$(27)+"H":GOTO 21840
<12> 21460 '*** inverse/normal (screen) ***
<42> 21470 IF INSTR(a$,"screen") THEN 21510
<29> 21480 w=27:GOSUB 21680
<38> 21490 IF INSTR(a$,"inv") THEN w=ASC("p") ELSE w=AS
     C("q")
<35> 21500 GOTO 21680
<87> 21510 IF INSTR(a$,"normal") THEN 21530
<53> 21520 o$=CHR$(27)+"b?"+CHR$(27)+"c"+CHR$(0):GOTO 2
     1840
<43> 21530 o$=CHR$(27)+"c?"+CHR$(27)+"b"+CHR$(0):GOTO 2
     1840
< 6> 21540 '*** gotoxy ***
<11> 21550 w=27:GOSUB 21680:w=ASC("Y"):GOSUB 21680
<95> 21560 GOSUB 24050:GOSUB 21580
<79> 21570 w=w2:var=var2
<61> 21580 IF var=0 THEN w=w+31:GOTO 21680
<75> 21590 GOSUB 24210:a=adr:POKE a,198:POKE a+1,31
```

```
<80> 21600 adr=adr+2:GOTO 21690
<35> 21610 '*** def chr ***
<92> 21620 ch=adr:RESTORE 21640:FOR f=0 TO 12:READ a
< 5> 21630 POKE adr+f,a:NEXT:adr=adr+f:RETURN
<13> 21640 DATA 229,213,197,95,14,2,205,5,0,193,209,225
     ,201
<66> 21650 '*** chr ***
<77> 21660 GOSUB 24010
<43> 21670 IF var THEN GOSUB 24210:GOTO 21690
<30> 21680 GOSUB 24190
<58> 21690 a=adr:POKE a,205:POKE a+1,ch AND 255
<57> 21700 POKE a+2,USR(ch):adr=a+3:RETURN
<55> 21710 '*** def write ***
<41> 21720 hex=adr+2:RESTORE 21790:FOR f=0 TO 7
<72> 21730 READ a:POKE adr+f,a:NEXT:a=adr+f:w=a+4
<92> 21740 POKE a,w AND 255:POKE a+1,USR(w):adr=a+2
<38> 21750 FOR f=0 TO 7:READ a:POKE adr+f,a:NEXT:adr=ad
     r+f
<62> 21760 a=adr+3:POKE adr,a AND 255:POKE a-2,USR(a):a
     dr=adr+2
<29> 21770 FOR f=0 TO 11:READ a:POKE adr+f,a:NEXT:adr=a
     dr+f
<98> 21780 a=adr:POKE a,ch AND 255:POKE a+1,USR(ch):adr
     =a+2:RETURN
<30> 21790 DATA 24,32,124,180,40,5,229,205,225,125,245,
     15,15,15,15,205,241,230,15,198,48,254,58,56,2,198,
     7,195
< 8> 21800 '*** write ***
<40> 21810 o=INSTR(a$,"'"):IF o=0 THEN 21910
<18> 21820 o$=MID$(bf$,o+1,LEN(bf$)-o-3)
< 7> 21830 IF INSTR(a$,"writeln") THEN o$=o$+CHR$(10)+C
     HR$(13)
<27> 21840 o$=o$+CHR$(36):w1=adr+11:merk=adr+9
<24> 21850 POKE adr,14:POKE adr+1,9:adr=adr+2:GOSUB 241
     20
<73> 21860 a=adr:POKE a,205:POKE a+1,5:POKE a+2,0
<65> 21870 POKE a+3,195:adr=adr+5
<97> 21880 FOR f=1 TO LEN(o$):h$=MID$(o$,f,1):POKE adr+
     f,ASC(h$)
< 1> 21890 NEXT:adr=adr+f:a=merk:POKE a,adr AND 255
<64> 21900 POKE a+1,USR(adr):RETURN
<29> 21910 GOSUB 24010:w2=w:GOSUB 24140:IF hex=0 THEN G
     OSUB 21720
<13> 21920 POKE adr,205:POKE adr+1,hex AND 255
<98> 21930 POKE adr+2,USR(hex):adr=adr+3
<71> 21940 IF INSTR(a$,"writeln")=0 THEN RETURN
<16> 21950 w=10:GOSUB 21680:w=13:GOTO 21680
<59> 21960 '*** def readln ***
<94> 21970 re=adr+2:RESTORE 22000:FOR f=0 TO 66
<78> 21980 READ a:POKE adr+f,a:NEXT:adr=adr+f
< 5> 21990 RETURN
<42> 22000 DATA 24,65,14,10,17,128,0,205,5,0,17,129,0,3
     8,0,108,26,79,19,26,254,38,40,20,230,15,213,22,0,9
     5,25,209,13,200,41,213
<14> 22010 DATA 93,84,41,41,25,209,24,230,19,13,13,19,2
     6,254,58,56,2,198,9,230,15,41,41,41,41,133,111,13,
     32,237,201
<15> 22020 '*** read ***
<97> 22030 GOSUB 24010:IF INSTR(a$,"readln") THEN 22070
<58> 22040 a=adr:POKE a,14:POKE a+1,6:POKE a+2,30
<83> 22050 POKE a+3,253:POKE a+4,205:POKE a+5,5:POKE a+
     6,0
<41> 22060 POKE a+7,111:POKE a+8,38:POKE a+9,0:adr=adr+
     10:GOTO 24230
<42> 22070 IF re=0 THEN GOSUB 21970
<35> 22080 a=adr:POKE a,205:POKE a+1,re AND 255
<74> 22090 POKE a+2,USR(re):adr=adr+3:GOSUB 24230:GOTO
     21950
<71> 22100 '*** poke/pokex ***
<80> 22110 GOSUB 24050:IF INSTR(a$,"pokex") THEN 22150
<28> 22120 w=w2:IF var2 THEN GOSUB 24210 ELSE GOSUB 241
     90
<80> 22130 w=w1:IF var=0 THEN 24250
<25> 22140 w2=w1:GOSUB 24140:POKE adr,119:adr=adr+1:RET
     URN
<47> 22150 w1=w2:IF var2 THEN GOSUB 24160 ELSE GOSUB 24
     120
<29> 22160 w2=w:GOSUB 24090
< 2> 22170 a=adr:POKE a,115:POKE a+1,35:POKE a+2,114:ad
     r=adr+3:RETURN
< 3> 22180 '*** peek/peekx ***
<36> 22190 o$=LEFT$(a$,1):GOSUB 24020:IF var=0 THEN 243
     50
```


CPC 6128 ● Joyce ● Joyce Plus

Unsere bahnbrechende Benutzer-Schnittstelle trägt einen neuen Namen:

MENUMATE

Unverändert bleiben die Leistungsmerkmale:

- Menüs nach eigenem Bedarf, in unbegrenztem Umfang
- Meldungen, Hilfsanweisungen, Informationstexte
- Einbau von beliebigen Kommandofolgen (z.B. Programmaufrufe)
- reichhaltige Formatierung einzelner Menüs und Texte
- Fenster-Technik, Pull-Down-Menüs, Schreibtisch-Effekte usw.
- Menü-gesteuerter Editor: Fehlbedienung ausgeschlossen

»Wenn Sie sich mit dem Programm einmal vertraut gemacht haben, sind Sie in der Lage, ganze Handlungsabläufe eines kleinen bis mittleren Betriebes damit zu steuern.«
– PC Schneider International 7/87, Seite 122

»Die mögliche Menüvernetzung mehrerer Anwenderprogramme macht MenuMate meiner Ansicht nach besonders geeignet, Mitarbeiter mit Bildschirm- und Computerarbeit auszusöhnen.«
– Joyce News 3/87, Seite 4 – 6

Informationsbroschüre kostenlos, unverbindlich
DEMO-Version DM 25,–* (wird beim späteren Lizenzerwerb voll angerechnet)
Benutzungslizenz DM 198,–* (unverbindliche Preisempfehlung)

Beim Fachhandel oder direkt von:

Gerald Keil
SOFTWARE-ENTWICKLUNG
UND VERTRIEB

Obere Hochstraße 136 – 6652 BEXBACH/Saar

* Nachnahme (nur BRD) zzgl. DM 3,50. BITTE COMPUTERTYP ANGEBEN

FÜR NIX GIBT'S NIX

Aber für gute Programme und Tips & Tricks umso mehr!

Für den Programmhit des Monats

1000,–



und für den Top-Tip des Monats ganze

500,–

Das sind doch gute Argumente, Ihr Programm auch einmal zum Hit des Monats werden zu lassen.

Probieren geht über Studieren.

Bitte richten Sie Ihre Einsendungen an:

DMV Verlag
Fuldaer Str. 6
3440 Eschwege

MACHEN SIE IHREN JOYCE ZUM FAKTURIER-AUTOMATEN:

Unsere praxiserprobten Faktura-Programme für ein oder zwei Laufwerke sparen viel Zeit und Geld im Büro und lassen Ihre Rechnungen individuell und gleichzeitig professionell aussehen weil:

- Sie das Formular selbst gestalten können
- bis zu 9 verschiedene Steuerschlüssel je Zeile möglich sind,
- beliebige Texte zur Erläuterung und Werbung eingeschoben werden können
- Ihre Stammkunden zum Einblenden ins Formular gespeichert bleiben
- Ihre gespeicherten Artikel selbst bei 8-stelligen alphanumerischen Art.-Nummern innerhalb einer Sekunde in der Rechnung sind,
- Ihr Lagerbestand automatisch kontrolliert wird,
- Sie die gesamte Rechnung auf dem Bildschirm beliebig oft korrigieren können,
- gleichzeitig auch Gutschriften, Lieferscheine, Auftragsbestätigungen und Angebote erstellt werden können,
- eine Offene-Post-Liste und Konten ausgedruckt werden und säumige Kunden automatisch gemahnt werden können,
- das Programm in »C« geschrieben und deshalb enorm schnell ist,
- das Programm für den JOYCE betriebsbereit geliefert wird,
- Sie diese Leistungen zu Preisen von DM 94,– bis DM 189,– erhalten.

Detaillierte Programmbeschreibungen kostenlos, Demo-Diskette 30,– DM, volle Anrechnung bei Kauf.

Unser weiteres JOYCE-Lieferprogramm in Kürze: ADRESS-JOYCE 39,– ● ADRESS-CONTROL 79,– ● SPORTTABELLEN 65,– ● IMMOBILIENVERMITTLUNG 149,– ● BAUFINANZIERUNG 279,– ● M&T FINANZBUCHHALTUNG 184,– ● WORDSTAR, MULTIPLAN, DBASE II je 189,– ● Drucker-Farbband 22,–

Vers. per NN zzgl. DM 5,– (Ausland 10,–) Porto/Verp. oder im Fachhandel.

Hashagen - EDV

Eckstr.11 · 6113 Babenhausen 3
Tel. (0 60 73) 6 19 93

```
<63> 22200 w1=w:GOSUB 24010:w2=w:GOSUB 24090
<61> 22210 a=adr:POKE a,94:IF INSTR(a$,"peekx")=0 THEN
     POKE a+1,22:POKE a+2,0:GOTO 22230
<83> 22220 POKE a+1,35:POKE a+2,86
<20> 22230 adr=adr+3:GOTO 24270
<50> 22240 '*** for-to/downto do ***
<31> 22250 o=INSTR(a$,"="):o$=MID$(a$,o-2,1)
<74> 22260 GOSUB 24020:IF var=0 THEN 24350
<16> 22270 w1=w:o$=MID$(a$,o+1,6):GOSUB 24020
<61> 22280 w2=w:schl=schl+1:GOSUB 24090
<44> 22290 IF INSTR(a$,"downto") THEN dt(schl)=1 ELSE d
     t(schl)=0
<26> 22300 IF dt(schl) THEN POKE adr,35 ELSE POKE adr,4
     3
<50> 22310 adr=adr+1:w=w1:GOSUB 24230:o=INSTR(a$,"to")
< 6> 22320 o$=MID$(a$,o+3,6):GOSUB 24020:lop(schl)=adr
<88> 22330 w2=w:GOSUB 24160:GOSUB 24090
<71> 22340 IF dt(schl)=0 THEN POKE adr,19 ELSE POKE adr
     ,27:POKE adr+1,43:adr=adr+1
<50> 22350 adr=adr+1:GOSUB 24270:GOSUB 24300
<77> 22360 IF dt(schl) THEN POKE adr,210:ELSE POKE adr,
     218
<78> 22370 adr=adr+3
<28> 22380 GOSUB 20400:IF INSTR(a$,"begin")=0 THEN 2436
     0
<57> 22390 GOSUB 20400:IF LEFT$(a$,3)<>"end" THEN GOSUB
      20570:GOTO 22390
<32> 22400 w=lop(schl):a=adr:adr=a+3:POKE a,195
<18> 22410 POKE a+1,w AND 255:POKE a+2,USR(w)
< 5> 22420 a=lop(schl)+16+dt(schl):POKE a,adr AND 255:P
     OKE a+1,USR(adr)
<37> 22430 schl=schl-1:RETURN
<87> 22440 '*** repeat-until ***
<55> 22450 un=un+1:loop(un)=adr
<63> 22460 GOSUB 20400:IF LEFT$(a$,5)<>"until" THEN GOS
     UB 20570:GOTO 22460
<32> 22470 o$=MID$(a$,7,1):GOSUB 24020:w1=w:IF var=0 TH
     EN 24350
<63> 22480 o=INSTR(a$,"="):o$=RIGHT$(a$,LEN(a$)-o):GOSU
     B 24020:w2=w
< 3> 22490 GOSUB 24090:GOSUB 24160:GOSUB 24300
<27> 22500 a=adr:w=loop(un):POKE a,194:POKE a+1,w AND 2
     55:POKE a+2,USR(w)
<72> 22510 un=un-1:adr=adr+3:RETURN
<35> 22520 '*** if-then ***
<13> 22530 IF INSTR(a$,"begin")=0 THEN 24360
<60> 22540 bd=bd+1:o=INSTR(a$,"="):gl(bd)=1:ungl(bd)=0
<59> 22550 IF o=0 THEN o=INSTR(a$,">"):gl(bd)=0:ungl(bd
     )=1
<88> 22560 o$=MID$(a$,o-1,1):GOSUB 24020:IF var=0 THEN
     24350
<67> 22570 w1=w:o$=MID$(a$,o+1,6):GOSUB 24020:w2=w
< 2> 22580 GOSUB 24090:GOSUB 24160:GOSUB 24300
<51> 22590 a=adr:IF gl(bd) THEN POKE a,194 ELSE POKE a,
     210
<80> 22600 adr=a+3:m(bd)=adr-2
<97> 22610 GOSUB 20400:IF LEFT$(a$,3)<>"end" THEN GOSUB
      20570:GOTO 22610
<52> 22620 IF INSTR(a$,"else")=0 THEN 22660
< 6> 22630 a=adr+3:POKE m(bd),a AND 255:POKE m(bd)+1,US
     R(a)
<89> 22640 POKE adr,195:adr=adr+3:m(bd)=adr-2:GOSUB 204
     00
<59> 22650 IF INSTR(a$,"begin") THEN 22610 ELSE GOTO 24
     360
<20> 22660 POKE m(bd),adr AND 255:POKE m(bd)+1,USR(adr)
<34> 22670 bd=bd-1:RETURN
<81> 22680 '*** halt ***
<51> 22690 POKE adr,201:adr=adr+1:RETURN
<56> 22700 '*** end ***
<62> 22710 PRINT"Programm.... : "n$
<73> 22720 PRINT"Anfang..... : &H"HEX$(adr1)
<30> 22730 PRINT"Länge..... :"adr-adr1
<33> 22740 PRINT"frei..... :"&HF605-adr
<67> 22750 IF a$<>"end." THEN PRINT"Das Programm wurde
     nicht mit END. beendet"
<20> 22760 IF schl THEN PRINT"Es wurden nicht alle Schl
     eifen beendet"
<92> 22770 IF bd THEN PRINT"Es wurden nicht alle IF-THE
     N beendet"
<89> 22780 IF un THEN PRINT"Es wurden nicht alle REPEAT
     -UNTIL beendet"
<80> 22790 IF basis=0 THEN PRINT"Das Programm wurde nic
     ht mit BEGIN; eröffnet"
```

```
<34> 22800 PRINT"Compiling ends...":PRINT
<28> 22810 a=adr:POKE a,225:POKE a+1,249
<86> 22820 POKE a+2,201:adr=a+2:CLOSE:GOTO 24420
<14> 24000 '*** variable ***
<40> 24010 h=INSTR(bf$,"(")+1:o$=MID$(bf$,h,6)
<50> 24020 h=ASC(o$):var=0:IF h<97 OR h>126 THEN w=UNT(
     VAL(o$)):RETURN
<77> 24030 h=h-97:w=UNT(va+2*h):var=1:RETURN
<20> 24040 '*** 2 Variablen ***
<18> 24050 h=INSTR(bf$,","):o$=RIGHT$(a$,LEN(a$)-h)
<97> 24060 GOSUB 24020:var2=var:w2=w
<12> 24070 GOSUB 24010:w1=w:var1=var:RETURN
<52> 24080 '*** Ladebefehle ***
<47> 24090 IF var THEN 24140
<92> 24100 a=adr:POKE a,33:POKE a+1,w2 AND 255
<68> 24110 POKE a+2,USR(w2):adr=adr+3:RETURN
<12> 24120 a=adr:POKE a,17:POKE a+1,w1 AND 255
<55> 24130 POKE a+2,USR(w1):adr=adr+3:RETURN
< 8> 24140 a=adr:POKE a,42:POKE a+1,w2 AND 255
<84> 24150 POKE a+2,USR(w2):adr=adr+3:RETURN
<12> 24160 a=adr:POKE a,237:POKE a+1,91
<21> 24170 POKE a+2,w1 AND 255:POKE a+3,USR(w1)
<97> 24180 adr=adr+4:RETURN
<66> 24190 a=adr:POKE a,62:POKE a+1,w AND 255
<38> 24200 adr=adr+2:RETURN
<44> 24210 a=adr:POKE a,58:POKE a+1,w AND 255
<86> 24220 POKE a+2,USR(w):adr=adr+3:RETURN
<27> 24230 a=adr:POKE a,34:POKE a+1,w AND 255
<94> 24240 POKE a+2,USR(w):adr=adr+3:RETURN
<90> 24250 a=adr:POKE a,50:POKE a+1,w AND 255
< 3> 24260 POKE a+2,USR(w):adr=adr+3:RETURN
<54> 24270 a=adr:POKE a,237:POKE a+1,83
<28> 24280 POKE a+2,w1 AND 255:POKE a+3,USR(w1)
< 5> 24290 adr=adr+4:RETURN
<70> 24300 a=adr:POKE a,183:POKE a+1,237
<70> 24310 POKE a+2,82:adr=adr+3:RETURN
<57> 24320 '*** Fehler ausgeben ***
<89> 24330 PRINT"' fehlt in"zn:GOTO 24400
<72> 24340 PRINT"Syntaxfehler":GOTO 24400
<23> 24350 PRINT"Keine Variablen":GOTO 24400
<62> 24360 PRINT"BEGIN erwartet":GOTO 24400
<22> 24370 PRINT"Keine Prozedur nach BEGIN mehr erlaubt
     ":GOTO 24400
<90> 24380 PRINT"Eine Prozedur wurde nicht mit END abge
     schlossen":GOTO 24400
< 8> 24390 PRINT"Programmname fehlt":GOTO 24400
<53> 24400 PRINT"Compiling ends...":END
<51> 24410 '*** Source-File speichern ***
<56> 24420 IF disc THEN 24580
<48> 24430 INPUT"Source-File speichern (J/N) ";a$
<57> 24440 IF UPPER$(a$)<>"J" THEN 24580
<34> 24450 LINE INPUT"Dateinamen eingeben : ";nam$
<73> 24460 h=INSTR(nam$,"."):IF h=0 THEN nam$=nam$+".PA
     S"
<63> 24470 nam$=UPPER$(nam$):PRINT"Saving "nam$"..."
<51> 24480 za=&H7A96:OPEN"o",#1,nam$
<73> 24490 zl=UNT(PEEK(za)+256*PEEK(za+1))
<92> 24500 zn=UNT(PEEK(za+2)+256*PEEK(za+3))
<16> 24510 IF zn>19999 THEN CLOSE:GOTO 24580
<62> 24520 w=UNT(za+6):za=UNT(za+zl)
<53> 24530 ba=UNT(VARPTR(bf$)):POKE ba,zl-7
<62> 24540 POKE ba+1,w AND 255:POKE ba+2,USR(w)
<90> 24550 bf$=STR$(zn)+" '"+bf$
<46> 24560 PRINT #1,bf$:GOTO 24490
<41> 24570 '*** Basic-Lader erstellen ***
<46> 24580 INPUT"Basic Lader speichern (J/N) ";a$
<95> 24590 IF UPPER$(a$)<>"J" THEN END
<20> 24600 LINE INPUT"Dateinamen eingeben : ";nam$
<49> 24610 h=INSTR(nam$,"."):IF h=0 THEN nam$=nam$+".BL
     D"
<48> 24620 nam$=UPPER$(nam$):PRINT"Saving "nam$"...":zn
     =30:OPEN"o",#1,nam$
<97> 24630 a$=" 10 restore 30:for i=&h"+HEX$(adr1)+" to
      &h"+HEX$(adr)
<80> 24640 PRINT #1 a$:a$=" 20 read a:poke i,a:next"
<58> 24650 PRINT #1,a$:PRINT "Zeile "CHR$(27)"j";
< 0> 24660 FOR i=adr1 TO adr STEP 16:a$=STR$(zn)+" data
      &h"+HEX$(PEEK(i))
<43> 24670 FOR j=1 TO 15:IF i+j>adr THEN 24690
< 9> 24680 a$=a$+",&h"+HEX$(PEEK(i+j)):NEXT
<20> 24690 PRINT CHR$(27)"k"zn;:PRINT #1,a$:zn=zn+10:NE
     XT:CLOSE:END
<47> 24700 FOR i=adr1 TO adr:PRINT HEX$(i)" "HEX$(PEEK(
     i)),:NEXT
```



Die führende Schneider-Zeitschrift gibt es jeden Monat neu und aktuell bei Ihrem Zeitschriftenhändler oder vom:

DMV Daten- und Medien-Verlagsgesellschaft mbH
Postfach 250 · Fuldaer Str. 6
3440 Eschwege · Tel. (0 56 51) 87 02



Kybernetik-Lernspiel

Im Rahmen meines Berufes als Lehrer bin ich seit geraumer Zeit auf der Suche nach Materialien zum Thema Kybernetik, um meinen Schülern im Arbeitslehreunterricht, Schwerpunkt »Messen-Steuern-Regeln« (Klassenstufe 10), die weit über die rein technische Dimension der Regelung hinausgehende ökologische Bedeutung des Prinzips »Regelkreis« wenigstens ansatzweise näherzubringen. Nicht zuletzt hat uns doch unsere Unfähigkeit, unsere Umwelt als komplexes System miteinander verkoppelter Regelkreise zu begreifen und zu behandeln, in die augenblickliche ökologische Krise gebracht. Spezialisiertes Wissen ohne Zusammenschau, verbunden mit dem anscheinend unbesiegbaren Drang, alles zu verwirklichen, was technologisch machbar erscheint, und nicht zuletzt die rücksichtslose und äußerst kurzsichtige Durchsetzung ökonomischer Interessen bringen jahrtausendelang funktionierende Ökosysteme in wenigen Jahrzehnten an die Grenzen ihrer Regulationsfähigkeit!

Aufklärung tut also not, trotz oder gerade wegen des fortgeschrittenen Standes von Wissenschaft und Technik. Grundsätzlich müßte diese Aufklärungsarbeit schon im Elternhaus durch die Vermittlung gewisser Grundhaltungen bei den Kindern zumindest vorbereitet werden, damit später Schule und Ausbildung darauf aufbauen können. Aber wenn's »die Erwachsenen« auch nicht wissen und danach handeln? Bekanntlich fallen Äpfel nicht weit vom Stamm!

Kurz und gut, ein Bekannter zeigte mir eines Tages das Spiel »Ökolopoly« von Frederic Vester, dem mit seinen Forschungsarbeiten, Büchern und Ausstellungen (z.B. »Unsere Welt ein vernetztes System« und »Neuland des Denkens«) das Verdienst zukommt, sich um diese ökologischkybernetische Aufklärungsarbeit zu bemühen. Da mir das Spiel nach etlichen Spielrunden zur Bereicherung des Unterrichtes geeignet erschien, beschloß ich dessen Umsetzung in ein Computerprogramm (Laut Spielbeschreibung wurde es auch schon an mehreren Universitäten computerisiert, was mir aber nichts nützte, da ich keinen Zugang zu diesen Programmen hatte und diese wohl kaum auf dem Joyce laufen würden.).

Natürlich kann man auch den Pappcomputer mit seinen Bildern und Einstellscheiben direkt und sozusagen »zum Anfassen« einsetzen. Mich reizte jedoch die Umsetzung in ein Programm aus zweierlei Gründen:
Erstens konnte ich so üben zu programmieren, und zweitens behandele ich das Thema »Computer« ohnehin im Unterricht, soweit das ohne schuleigene Rechner eben möglich ist und etwas mit »Messen-Steuern-Regeln« zu tun hat. Damit hätte ich dann gleich ein positives Anwendungsbeispiel für den Einsatz von Computern:
Simulation vernetzter Regelungssysteme zur Abschätzung der Auswirkungen von Eingriffen in die Natur usw. Gesagt – getan. Zuerst entstand das Programm in einfacher Form für meinen Taschencomputer PC1500 von Sharp. Dann legte ich mir einen JOYCE zu, der natürlich ganz andere Möglichkeiten als ein Taschencomputer bietet. Das Resultat etlicher Programmierabende und -nächte liegt Ihnen jetzt als »Kybernetik-Lernprogramm« vor.

Programmbeschreibung

Das Programmlisting ist so gestaltet, daß die Funktion der einzelnen Programmteile durch die jeweils vorangestellte Kommentarzeile charakterisiert wird (s. Programmteile-Übersicht). Zusammen mit der Variablenliste sollte das Programm dadurch nachvollziehbar sein.
Im Folgenden noch einige Hinweise: Damit nicht immer dieselbe Zufallszahlenfolge nach dem Laden des Programmes erzeugt wird, wird in Zeile 50 die Speicherzelle FBF8H ausgelesen, in der im BCD-Format die Sekunden der Uhrzeit abgelegt werden (die Minuten stehen in FBF7H, die Stunden in FBF6H). So können immerhin sechzig verschiedene Zahlenfolgen automatisch mit dem RANDOMIZE-Kommando initiiert werden. Die Zahl, die nach dem Starten des Programmes kurzzeitig angezeigt wird, ist diese Sekundenzahl im Moment des Zugriffes mittels der PEEK-Funktion. Es ist vielleicht von allgemeinem Interesse zu erfahren, daß der sog. SCB (System Controll Block), zu dem die oben genannten Adressen gehören, beim Joyce bei der Adresse FB9CH beginnt.

Im SCB stehen verschiedene Flags und Parameter für den CCP, das BDOS und andere Systemteile. Näheres hierzu finden Sie im CP/M Plus Programmer's Guide von Digital Research (beim Verlag Markt&Technik erschienen). Um andere Informationen aus dem SCB zu erhalten, müssen Sie nur den entsprechenden Offset zur Anfangsadresse FB9CH addieren. Vorsicht beim eventuellen Ändern von Parametern! Ein Systemabsturz kann die Folge sein!

Die Berechnung der neuen Werte der einzelnen Systemvariablen infolge der jeweiligen Ausgangswerte und der getätigten Investitionen erfolgt sequentiell (ab Zeile 1890) in 18 Schritten. Dazu wird der Feldvariablentabelle %(x) jeweils eine Gruppe von 51 Zahlen aus den Datazeilen ab Zeile 1190 zugeordnet (UP Zeile 1560). Die Feldelemente stellen dann Funktionswerte y=f(x) dar, wobei x im Indexwert repräsentiert wird. Ein Vorteil dieser »Strategie« gegenüber einer langen Reihe von IF..THEN-Entscheidungen liegt in der Überschaubarkeit der einzelnen Funktionswerte in den Datazeilen und ermöglicht auch leicht ein experimentelles Ändern derselben. Das jeweils erste Element mit dem Index 0 stellt eine Prüfziffer dar. Sollte das Programm während der Berechnung abbrechen, so stimmt die Anzahl der Elemente in den Datazeilen (51) nicht mehr. Nicht benötigte Elemente einer Datazeile (bei Maximalwerten < 50) müssen mit Null belegt werden. Näheres zur Mathematik des Spieles findet man in der Ökolopoly-Spielanleitung (<C>Ravensburger Spiele). In den Programmausgängen zu CP/M (Zeile 2290) bzw. BASIC (Zeile 2320) können Sie nach eigenem Wunsch »Abschiedskommandos« einsetzen. Beim Verlassen von BASIC wird das Programm sicherheitshalber auf der RAM-Disk (M:) gespeichert.

Bei der Bestimmung eines Zufallsereignisses (UP Zeile 3060) wird durch Vergleich mit den bisher eingetretenen Ereignissen sichergestellt, daß innerhalb von 155 Spielrunden (»Jahren«) jedes der 31 möglichen Ereignisse nur einmal eintreten kann.

Die Bewertungsformel (Zeile 4410) kann erst ab der 10. Spielrunde eingesetzt werden. 30 Spielrunden sollten nicht überschritten werden. Die Bewertung erscheint mir recht grob und könnte vielleicht verfeinert werden.

Spielablauf

Um vor Spielbeginn die benötigten Dateien auf die RAM-Disk zu kopieren, geben Sie zweckmäßigerweise das Kommando

SUBMIT KYBERNET

ein. Neben den zu kopierenden Dateien müssen sich auf der Diskette auch die Dateien PIP.COM und SUBMIT.COM sowie KYBERNET.SUB befinden. Letztere enthält folgende Kommandos:

```
pip
<m:=a:basic.com
<m:=a:setkeys.com
<m:=a:kybernet.bas
<m:=a:keys.kyb
<m:=a:kyber1.txt
<m:=a:kyber2.txt
<
m:
setkeys keys.kyb
basic kybernet
```

SETKEYS KEYS.KYB belegt die Tasten Nr. 1(Form) und 7(Gitter) mit ^M (RETURN), so daß Sie den in Sonderheft 1/87 beschriebenen JOYceSTICK bei der Eingabe benutzen können.

```
07 N S "^M" Taste 1 RETURN
01 N S "^M" Taste 2     "
```

Sodann werden BASIC und KYBERNET.BAS geladen und das Programm wird gestartet.

Rufen Sie die Option 7 auf. Auf dem Bildschirm werden dann eine Kurzanleitung und anschließend eine Übersicht über das Wirkungsgefüge aus gegeben (s. Abb. Nr.1 und 2).

Nach der Rückkehr ins Menü wählen Sie den Spielmodus. Es erscheinen die numerische und die analoge Darstellung der Systemvariablenwerte im oberen sowie das Investitionsmenü im unteren Fenster (s. Hardcopy Nr.3). Benutzen Sie nun die Cursortasten, um einen der zur Auswahl stehenden Bereiche anzuwählen, indem Sie ihn invers darstellen und RETURN drücken. Der gewählte Bereich erscheint unten links nochmals. Nun geben Sie die zu investierende Punktzahl ein (bei der »Produktion« evtl. auch mit Minuszeichen) und drücken wiederum RETURN. Die investierte Punktzahl erscheint dann rechts neben dem angewählten Bereich. Wählen Sie dann den nächsten Bereich, usw. Sie können jederzeit die investierte Punktzahl durch Überschreiben korrigieren. Wollen Sie die Punktverteilung beenden, wählen Sie die Option '** weiter **'. Wollen Sie die Kurzanleitung studieren, eine Korrektur vornehmen oder das Spiel

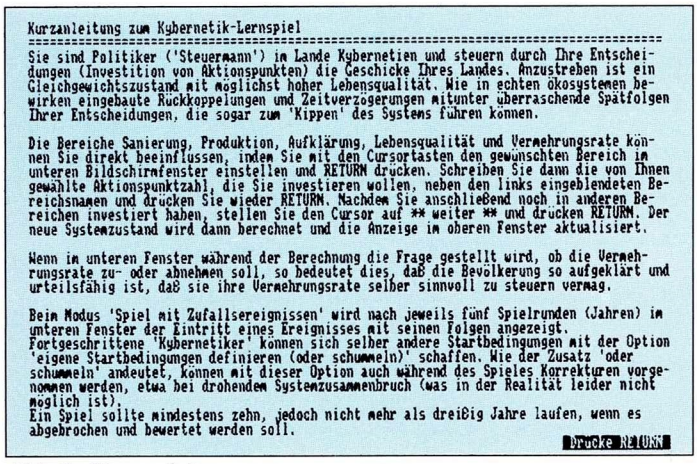


Abb.1: Kurzanleitung

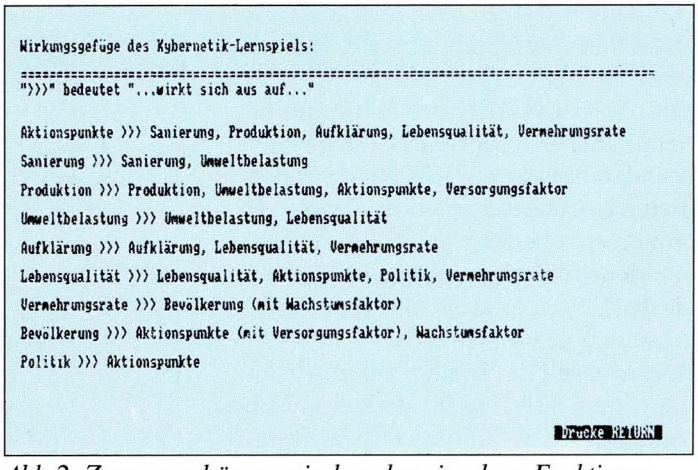


Abb.2: Zusammenhänge zwischen den einzelnen Funktionen

abbrechen, so wählen Sie die Option '*zum Menü*' an. Von dort gelangen Sie über die Option 4 zurück ins laufende Spiel.

Sollten Sie z.B. durch falsche Eingabe eine REDO-Aufforderung bekommen, die den Bildrahmen zerstört, so gehen Sie ins Hauptmenü und können mit der Option 6 einen neuen Bildaufbau durchführen, ohne daß das Spiel dadurch zerstört wird.

Beim Eintritt eines Zufallsereignisses werden im unteren Fenster als Auswirkung Korrekturwerte für einige Systemvariablen angezeigt. Diese werden automatisch eingerechnet und sind in der folgenden aktualisierten Anzeige des Punktstandes im oberen Fenster berücksichtigt.

Neben der numerischen Anzeige wird der Punktstand auch analog durch die Position eines Pfeiles in einem leuchtenden Band dargestellt. Der Punktstand des Vorjahres wird durch einen Punkt innerhalb des Bandes dargestellt. So läßt sich die Entwicklungstendenz auf einen Blick erfassen.

Alle Systemänderungen, Ereignisse und Korrekturen werden protokolliert. So können Sie ein Spiel jederzeit nachvollziehen. Legen Sie das DIN-A4-Blatt nach der Aufforderung im unteren Fenster bei Spielbeginn so in die Zuführung, daß der linke Blattrand mit der linken Verstärkungsrippe auf dem Klappdeckel fluchtet. Ist für ein neues Spiel noch genügend Platz auf dem Papier, so drücken Sie einfach nur die RETURN- oder ENTER-Taste.

Sollten Sie vom Hauptmenü aus die Option 3 wählen, so müssen Sie die Korrekturabfrage für alle Systemvariablen durchlaufen. Bei Werten, die nicht geändert werde sollen, drücken Sie einfach die RETURN-Taste.

Sie sollten ihre Entscheidungen stets daraufhin überprüfen, ob sie realistisch sind, es sei denn, sie wollen übertreiben. So könnten Sie durch Investition sämtlicher Aktionspunkte über vier Runden hinweg ausschließlich in den Bereich »Lebensqualität« das Spielende mit einer scheinbar positiven Schlußmeldung abwürgen. Scheinbar positiv deshalb, weil es sich bei diesem Spiel nicht um ein »Wettrennen« handelt, sondern weil das System innerhalb der gesetzten Grenzwerte und zwischen zehn und dreißig Jahren »Regierungszeit« unter Wahrung eines Fließgleichgewichtes auf ein möglichst hohes Lebensqualitätsniveau gehoben werden soll.

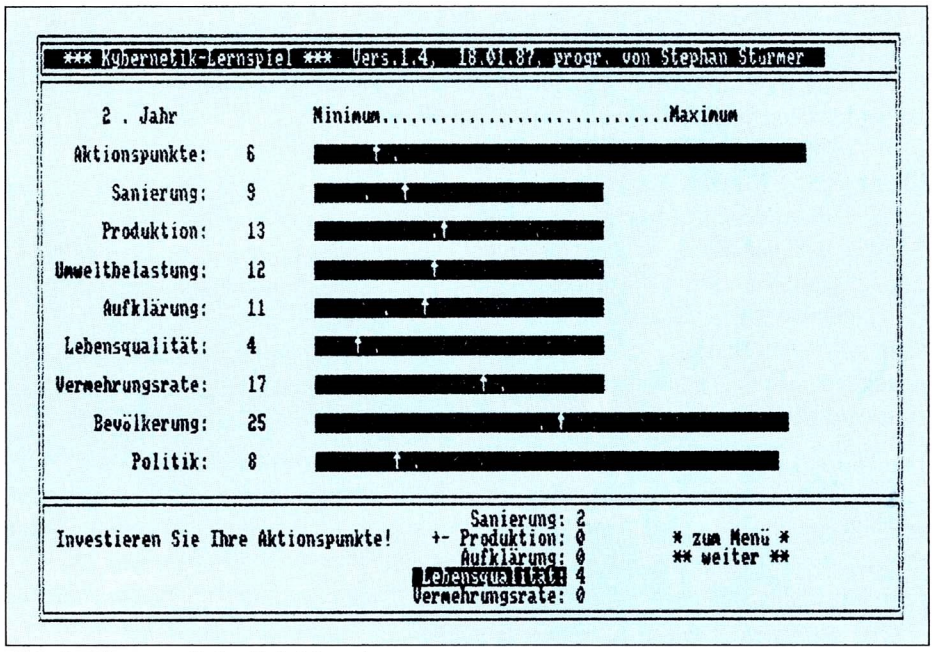


Abb.3: Eine mögliche Spielsituation...

Programmänderungen

Sie können gut mit dem Programm experimentieren, indem Sie z.B. die Funktionswerte in den Datazeilen (ab Zeile 1190) manipulieren. Achten Sie jedoch darauf, daß jede Gruppe immer genau 51 Elemente enthalten muß! Sie können das Spiel auf andere vernetzte Systeme übertragen, indem Sie sich selber entsprechende Systemvariablen mit den zugehörigen Grenzwerten und Funktionen definieren und das Programm entsprechend modifizieren. Sie können sich eigene Zufallsereignisse ausdenken und die vorgegebenen damit ersetzen. Die Variablenliste und die Übersicht über die Programmteile soll dabei eine Hilfe sein.

(S.Stürmer)

Programmteile-Übersicht zu KYBERNET.BAS

```
  10 '********** Kybernetik-Lernspiel**********
  20 '********** nach einer Idee von Frederic Vester **********
  40 '********** Initialisierung
 280 '********** Bildrahmen
 400 '********** Menü
 650 '********** Schalter für Zufallsereignisse
 680 '********** Spielstandausgabe im oberen Fenster
 840 '********** Investitionsmenü im unteren Fenster
1180 '********** Datazeilen
1190 'Auswirkung der Sanierung auf die Umweltbelastung (1)
1210 'Auswirkung der Sanierung auf sich selber (2)
1230 'Auswirkung der Produktion auf sich selber (3)
1250 'Auswirkung der Produktion auf die Umweltbelastung (4)
1270 'Auswirkung der Umweltbelastung auf sich selber (5)
1290 'Auswirkung der Umweltbelastung auf die Lebensqualität (6)
1310 'Auswirkung der Aufklärung auf sich selber (7)
1330 'Auswirkung der Aufklärung auf die Lebensqualität (8)
1350 'Auswirkung der Aufklärung auf die Vermehrungsrate (9)
1370 'Auswirkung der Lebensqualität auf sich selber (10)
1390 'Auswirkung der Lebensqualität auf die Vermehrungsrate (11)
1410 'Auswirkung der Lebensqualität auf die Politik (12)
1430 'Auswirkung der Vermehrungsrate auf die Bevölkerung (13)
1450 'Auswirkung der Bevölkerung auf die Lebensqualität (14)
1470 'Auswirkung der Bevölkerung auf die Aktionspunkte (15)
1490 'Auswirkung der Politik auf die Aktionspunkte (16)
1510 'Auswirkung der Produktion auf die Aktionspunkte (17)
1530 'Auswirkung der Lebensqualität auf die Aktionspunkte (18)
1560 '********** UP Tabelle einlesen
1620 '********** UP letzten Stand speichern
1650 '********** UP oberes Fenster löschen
1680 '********** UP unteres Fenster löschen
```

```
1710 '********** UP Protokoll vorbereiten
1780 '********** UP Stand ausdrucken
1810 '********** UP Investitionen ausdrucken
1840 '********** UP Berechnung
2280 '********** Ausgang zu CP/M
2310 '********** Ausgang zu BASIC
2340 '********** UP Endschalter setzen
2370 '********** UP auf Grenzwertüberschreitung prüfen
2950 '********** UP ins Kommentarfenster schreiben
3000 '********** UP Kommentar ausdrucken
3060 '********** UP Zufallsereignisbestimmung
3200 '********** UP warten auf RETURN
3230 '********** UP Ereignisse
4410 '********** UP Meldung unter 10 Runden
4450 '********** UP Bewertung
4760 '********** UP Werte ändern (z.B. Startbedingungen)
4870 '********** UP Korrektureingabe
4920 '********** UP Korrekturausdruck
```

Variablenliste zu KYBERNET.BAS

ak%: Aktionspunktzahl der letzten Runde
akt%: aktuelle Aktionspunktzahl
au%: Aufklärungspunktzahl der letzten Runde
auf%: aktuelle Aufklärungspunktzahl
beep$ Piepton
be%: Bevölkerungszahl der letzten Runde
bev%: aktuelle Bevölkerungszahl
bilanz%: Bewertungszahl nach der Formel in Zeile 4410
change%: Schalter wird bei Änderung der Punktzahlen gesetzt (Menüoption 3)
coff$: Escapesequenz für 'Cursor unsichtbar'
con$: Escapesequenz für 'Cursor sichtbar'
cs$: Escapesequenz für 'Bildschirm löschen'
dagewesen%(x): enthält die Nummern bereits eingetretener Zufallsereignisse
ende%: Endschalter für Abbruch des Spieles bei Grenzwertüberschreitung
ereignis%: schaltet Zufallereignisse ein oder aus (Spielart)
ereignrunde%: zählt die Anzahl der eingetretenen Zufallsereignisse
fzeile$(x): enthält eine von vier Textzeilen zur Ausgabe im unteren Fenster
getc$: Escapesequenz für 'Cursorposition speichern'
home$: Escapesequenz für 'Cursor in linke obere Ecke'
i: Zählvariable (FOR...TO...NEXT)
j: Zählvariable (FOR...TO...NEXT)
ja%: Jahr der letzten Runde
jahr%: aktuelle Spielrundenanzahl, eine Runde entspricht einem Jahr
le%: Lebensqualitätspunktzahl der letzten Runde
leb%: aktuelle Lebensqualitätspunktzahl
loc$: Escapesequenz für Cursorpositionierung (FNloc$)
neu%: enthält den Korrekturwert bei Änderung der Punktzahlen (Menüoption 3)
p: Zeilenzähler (Investitionsmenü im unteren Fenster)
po%: Politikpunktzahl der letzten Runde
pol%: aktuelle Politikpunktzahl
prc$: Escapesequenz für Positionierung des Cursors an gespeicherter Position
pr%: Produktionspunktzahl der letzten Runde
pro%: aktuelle Produktionspunktzahl
q: Spaltenzähler (Investitionsmenü im unteren Fenster)
runde%: Rundenzähler, bei runde% = 5 tritt ein Ereignis ein
sa%: Sanierungspunktzahl der letzten Runde
san%: aktuelle Sanierungspunktzahl
spiel%: Schalter zur Auswahl der Menüoptionen vor oder während des Spieles
stoff$: Escapesequenz für 'Statuszeile AUS'
tabelle%(x): enthält den x zugeordneten Funktionswert (s.auch Datazeilen)
temp%(x): enthält die gerade in Bereich x investierte Punktezahl
um%: Umweltpunktzahl der letzten Runde
umw%: aktuelle Umweltbelastungspunktzahl
v$(x,y): Strings für Punktverteilungsmenü (s.Zeile 190260)
ve%: Vermehrungsrate der letzten Runde
ver%: aktuelle Vermehrungsrate
vers%: Versorgungsfaktor
wachs%: Wachstumsfaktor
wahl%: enthält die Kennziffer für eine best. Menüoption
xoff$: Escapesequenz für 'inverse Darstellung AUS'
xon$: Escapesequenz für 'inverse Darstellung EIN'
z: enthält den ASCII-Wert der gedrückten Cursorsteuerungstaste (Invest.-Menü)
zahl%: Zufallszahl zur Auswahl eines Ereignisses
zufall%: Startwert für RANDOMIZEKommando
zuletzt$: enthält den zuletzt invers dargestellten Begriff im Menü

5.25' Zweitlaufwerk
Qualitätslaufwerk incl. Gehäuse
Adapter und Einbauanleitung
1 MB Kapazität, voll FD-2 kompatibel
Preissenkung: 449,-

vortex 20 MB Festplatte f. Joyce 2150,-

Joyce MousePack von Gerdes 228,-

Speichererweiterung auf 512 KB 79,-

DÜSI-Vokabeltrainer
Prof. Lernsystem für Fremdsprachen
(siehe Testbericht CPC-Magazin 6/87)
59,-
DÜSI-Verbentrainer
für unregelm. engl. Verben
49,-

DR-Draw prof. Zeichenprogramm 179,-
DR-Graph für Geschäftsgraphik 179,-

dBase II / WordStar / Multiplan je 179,-

Turbo Pascal 215,-

M&T Finanzbuchhaltung
Kompl. Buchführungssystem incl.
Journal, Kostenstellen, BWA, Bilanz
G+V, Umsatzst. Voranmeldung, etc
189,-

CPC-COM
prof. DFÜ-Programm für Terminalemul.
Dateiübertragung, etc
versch. Datenformate
alle Paramter über Menü einstellbar
139,-

SUPERANGEBOT!!!
Papierführung für Joyce-Drucker 25,-
(Lieferung solange Vorrat reicht)

WINDOW ADRESS
Die Superadressverwaltung für Joyce!!
75,-

TG-Soft
Thomas Gmach
Offersdorf 5
8491 Rimbach
Tel. (0 99 41) 37 65

SIGNUM
Computer-Bücher



Dieses Buch ist der unentbehrliche Ratgeber für alle Joyce-Anwender. Es führt von Grund an die Textverarbeitung LocoScript heran, nimmt die Schwellenangst und gibt wertvolle Tips für die Praxis.

Autor: Joachim Seidler
Umfang: 160 Seiten/zahlr. Abb.
Preis: DM 38,–

Überall, wo es Computerbücher gibt.

Signum Medien Verlag
Elsenheimerstraße 59
8000 München 21
Tel. 089-5705892



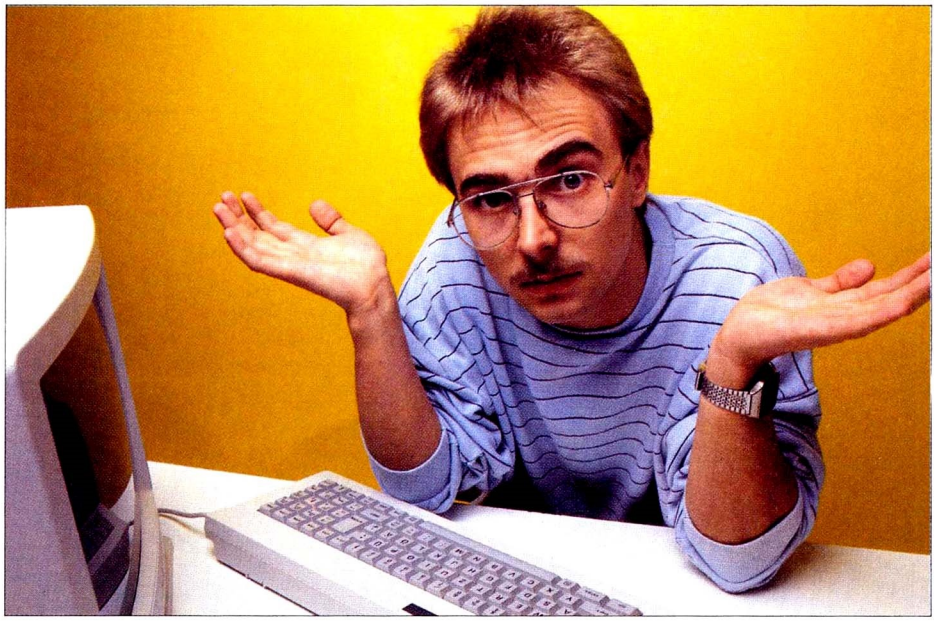
```
LISTING >KYBERNET<, REMARK = >'<.

<29> 10 '********** Kybernetik-Lernspiel.....
      **********
<27> 20 '********** nach einer Idee von Frederic Vester
      **********
< 3> 30 '
<94> 40 '********** Initialisierung
<97> 50 zufall%=VAL(HEX$(PEEK (&HFBF8))):PRINT zufall%:
     RANDOMIZE zufall%
<54> 60 DEFINT a-z
<34> 70 DIM v$(4,1): DIM temp%(4):. DIM tabelle%(50): D
     IM fzeile$(3): DIM dagewesen%(30)
<38> 80 vers%=1: wachs%=2: akt%=8: san%=1: pro%=12: umw
     %=13: auf%=4: leb%=10: ver%=20: bev%=21:pol%=11:en
     de%=0:ereignis%=0:zahl%=0:runde%=0:ereigrunde%=0:s
     piel%=0
<57> 90 DEF FNloc$(x,y)=CHR$(27)+"Y"+CHR$(31+y)+CHR$(31
     +x)
<83> 100 xon$=CHR$(27)+"p": xoff$=CHR$(27)+"q"
<46> 110 cs$=CHR$(27)+"E": con$=CHR$(27)+"e": coff$=CHR
     $(27)+"f": stoff$=CHR$(27)+"0"
<22> 120 beep$=CHR$(7): home$=CHR$(27)+"H": getc$=CHR$(
     27)+"j": prc$=CHR$(27)+"k"
<94> 130 kopf$=". *** Kybernetik-Lernspiel ***. Vers.1.
     4,. 18.01.87, progr. von Stephan Stürmer. "
<47> 140 top$=CHR$(134)+STRING$(88,CHR$(138))+CHR$(140)
     +CHR$(133)+SPACE$(88)+CHR$(133)
<66> 150 mit1$=CHR$(135)+STRING$(88,CHR$(138))+CHR$(141
     )
< 3> 160 mit2$=CHR$(133)+SPACE$(28)+CHR$(133)+SPACE$(28
     )+CHR$(133)+SPACE$(30)+CHR$(133)
<11> 170 bot$=CHR$(131)+STRING$(88,CHR$(138))+CHR$(137)
<13> 180 akt$=". Aktionspunkte: "
< 0> 190 san$="..... Sanierung: ":v$(0,0)="..... Sanier
     ung:"
<72> 200 pro$=".... Produktion: ":v$(1,0)=". +- Produkt
     ion:"
<85> 210 umw$="Umweltbelastung: "
<21> 220 auf$=".... Aufklärung: ":v$(2,0)=".... Aufklär
     ung:"
<95> 230 leb$=" Lebensqualität: ":v$(3,0)=" Lebensquali
     tät:"
<54> 240 ver$="Vermehrungsrate: ":v$(4,0)="Vermehrungsr
     ate:"
<39> 250 bev$="... Bevölkerung: ":v$(2,1)=". ** weiter
     **. "
<14> 260 pol$="....... Politik: ":v$(1,1)=". * zum Menü
      *. "
<33> 270 '
<14> 280 '********** Bildrahmen
<86> 290 PRINT CHR$(27);"0";coff$;cs$;home$;FNloc$(0,0)
     ;top$;mit1$;
<10> 300 FOR y=5 TO 25
<23> 310 PRINT FNloc$(1,y);CHR$(133);FNloc$(90,y);CHR$(
     133);
<53> 320 NEXT y
<18> 330 PRINT FNloc$(1,25);mit1$;
<29> 340 FOR y=27 TO 31
<31> 350 PRINT FNloc$(1,y);CHR$(133);FNloc$(90,y);CHR$(
     133);
<61> 360 NEXT y
<45> 370 PRINT FNloc$(1,31);bot$
<38> 380 PRINT xon$;FNloc$(2,2);kopf$;xoff$
<38> 390 '
<87> 400 '********** Menü
<81> 410 IF spiel%=0 THEN PRINT FNloc$(3,5);"1 - Spiel
     ohne Zufallsereignisse"
< 4> 420 IF spiel%=0 THEN PRINT FNloc$(3,7);"2 - Spiel
     mit Zufallsereignissen"
<36> 430 PRINT FNloc$(3,9);"3 - eigene Startbedingungen
      definieren (oder schummeln)"
<68> 440 IF spiel%=1 THEN PRINT FNloc$(3,11);"4 - zurüc
     k ins laufende Spiel"
<14> 450 IF spiel%=1 THEN PRINT FNloc$(3,13);"5 - Spiel
      abbrechen und bewerten"
<39> 460 PRINT FNloc$(3,15);"6 - neuer Bildaufbau (z.B.
      nach Fehlermeldungen)"
<51> 470 PRINT FNloc$(3,17);"7 - Kurzanleitung und Wirk
     ungsgefüge-Übersicht "
<72> 480 PRINT FNloc$(3,19);"8 - zurück zu BASIC"
<52> 490 PRINT FNloc$(3,21);"9 - zurück zu CP/M"
<72> 500 PRINT FNloc$(3,27);xon$;" Bitte wählen Sie! ";
     xoff$
<76> 510 PRINT beep$;:wahl%=ASC(INPUT$(1))-48
<81> 520 WHILE wahl% > 9 OR wahl% < 1
<88> 530.. wahl%=ASC(INPUT$(1))-48
<91> 540 WEND
<59> 550 IF spiel%=0 THEN ON wahl% GOTO 680,660,580,510
     ,510,280,600,2320,2290
<14> 560 ON wahl% GOTO 510,510,580,570,590,280,600,2320
     ,2290
<85> 570 RETURN
<77> 580 GOSUB 1620: GOSUB 4760: GOSUB 1650: GOSUB 1680
     : GOSUB 4920: GOTO 410
<77> 590 IF jahr% < 10 THEN GOSUB 4420: RETURN: ELSE GO
     SUB 4460: COMMON RESET: GOTO 60
<94> 600 PRINT cs$;home$;: TYPE M:KYBER1.TXT
<44> 610 GOSUB 3200
<34> 620 PRINT cs$;home$;: TYPE M:KYBER2.TXT
<39> 630 GOSUB 3200: GOTO 270
<31> 640 '
<48> 650 '********** Schalter für Zufallsereignisse
< 1> 660 ereignis%=1
<37> 670 '
< 8> 680 '********** Spielstandausgabe im oberen Fenste
     r
<74> 690 spiel%=1: GOSUB 1710: GOSUB 1620: GOSUB 1780
<35> 700 GOSUB 1650
< 7> 710 PRINT FNloc$(7,5);jahr%;". Jahr"
<42> 720 PRINT FNloc$(30,5);"Minimum.....
     .....Maximum"
< 7> 730 akt%=akt%+neu%:PRINT FNloc$(3,7);akt$;FNloc$(2
     2,7);akt%:PRINT FNloc$(30,7);xon$+SPACE$(51);FNloc
     $(ak%+30,7);".";FNloc$(akt%+30,7);"^";xoff$: IF ch
     ange%=1 THEN RETURN
<33> 740 san%=san%+neu%:PRINT FNloc$(3,9);san$;FNloc$(2
     2,9);san%:PRINT FNloc$(30,9);xon$+SPACE$(30);FNloc
     $(sa%+30,9);".";FNloc$(san%+30,9);"^";xoff$: IF ch
     ange%=1 THEN RETURN
<13> 750 pro%=pro%+neu%:PRINT FNloc$(3,11);pro$;FNLOC$(
     22,11);pro%:PRINT FNloc$(30,11);xon$+SPACE$(30);FN
     loc$(pr%+30,11);".";FNloc$(pro%+30,11);"^";xoff$:
     IF change%=1 THEN RETURN
<57> 760 umw%=umw%+neu%:PRINT FNloc$(3,13);umw$;FNLOC$(
     22,13);umw%:PRINT FNloc$(30,13);xon$+SPACE$(30);FN
     loc$(um%+30,13);".";FNloc$(umw%+30,13);"^";xoff$:
     IF change%=1 THEN RETURN
<73> 770 auf%=auf%+neu%:PRINT FNloc$(3,15);auf$;FNLOC$(
     22,15);auf%:PRINT FNloc$(30,15);xon$+SPACE$(30);FN
     loc$(au%+30,15);".";FNloc$(auf%+30,15);"^";xoff$:
     IF change%=1 THEN RETURN
<73> 780 leb%=leb%+neu%:PRINT FNloc$(3,17);leb$;FNLOC$(
     22,17);leb%:PRINT FNloc$(30,17);xon$+SPACE$(30);FN
     loc$(le%+30,17);".";FNloc$(leb%+30,17);"^";xoff$:
     IF change%=1 THEN RETURN
<63> 790 ver%=ver%+neu%:PRINT FNloc$(3,19);ver$;FNLOC$(
     22,19);ver%:PRINT FNloc$(30,19);xon$+SPACE$(30);FN
     loc$(ve%+30,19);".";FNloc$(ver%+30,19);"^";xoff$:
     IF change%=1 THEN RETURN
<52> 800 bev%=bev%+neu%:PRINT FNloc$(3,21);bev$;FNLOC$(
     22,21);bev%:PRINT FNloc$(30,21);xon$;SPACE$(49);FN
     loc$(be%+30,21);".";FNloc$(bev%+30,21);"^";xoff$:
     IF change%=1 THEN RETURN
```

```
<27> 810 pol%=pol%+neu%:PRINT FNloc$(3,23);pol$;FNLOC$(
     22,23);pol%:PRINT FNloc$(30,23);xon$+SPACE$(48);FN
     loc$(po%+30,23);".";FNloc$(pol%+30,23);"^";xoff$:
     IF change%=1 THEN RETURN
<98> 820 PRINT FNloc$(3,27);"Investieren Sie Ihre Aktio
     nspunkte!"
<31> 830 '
<19> 840 '********** Investitionsmenü im unteren Fenste
     r
<91> 850 FOR y=0 TO 4
<36> 860 FOR x=0 TO 1
<88> 870 PRINT FNloc$(40+x*25,y+26);v$(y,x)
<14> 880 IF x=0 THEN PRINT FNloc$(56,y+26);temp%(y)
<62> 890 NEXT x
<55> 900 NEXT y
<90> 910 p=1:q=1:PRINT coff$;:GOTO 1060
<19> 920 z=ASC(INPUT$(1))
<55> 930 IF z=30 THEN p=p+1: IF q=1 THEN IF p=6 THEN p=
     1
<43> 940 IF z=30 THEN IF q=2 THEN IF p>3 THEN p=2
<92> 950 IF z=13 THEN GOTO 1080
<66> 960 IF z=31 THEN p=p-1: IF q=1 THEN IF p=0 THEN p=
     5
< 1> 970 IF z=31 THEN IF q=2 THEN IF p<2 THEN p=3
< 6> 980 IF z=6 THEN q=q+1: IF p<2 THEN p=2
<83> 990 IF z=6 THEN IF q=3 THEN q= 1
<54> 1000 IF z=6 THEN IF p>3 THEN p=3
< 6> 1010 IF z=6 THEN IF q=3 THEN q= 1
<58> 1020 IF z=1 THEN q=q-1: IF p<2 THEN p=2
<35> 1030 IF z=1 THEN IF q=0 THEN q=2
<11> 1040 IF z=1 THEN IF p>3 THEN p=3
<41> 1050 IF z=1 THEN IF q=0 THEN q=2
<19> 1060 PRINT prc$;zuletzt$;FNloc$(40+(q-1)*25,p+25);
     getc$;xon$;v$(p-1,q-1);xoff$:zuletzt$=v$(p-1,q-1)
<47> 1070 GOTO 920
<33> 1080 IF p=3 AND q=2 THEN GOSUB 1840:IF ereignis%=1
      THEN GOSUB 3060: GOSUB 2370: GOSUB 1780: GOTO 710
     :ELSE GOSUB 1780: GOTO 710
<11> 1090 IF p=2 AND q=2 THEN GOSUB 1650: GOSUB 1680: G
     OSUB 400: GOTO 700
<74> 1100 PRINT FNloc$(3,29);SPACE$(35)
<76> 1110 PRINT FNloc$(3,29);xon$;v$(p-1,0);xoff$;:INPU
     T temp%(p-1)
<97> 1120 temp%(0)=ABS(temp%(0)):temp%(2)=ABS(temp%(2))
     :temp%(3)=ABS(temp%(3)):temp%(4)=ABS(temp%(4))
<84> 1130 PRINT FNloc$(3,29);SPACE$(20)
<52> 1140 PRINT FNloc$(56,p+25);temp%(p-1)
<86> 1150 IF (temp%(0)+ABS(temp%(1))+temp%(2)+temp%(3)+
     temp%(4)) > akt% THEN PRINT FNloc$(3,29);xon$;"Kon
     to überzogen! Bitte korrigieren!";xoff$;beep$
<46> 1160 GOTO 920
< 7> 1170 '
<46> 1180 '********** Datazeilen
<35> 1190 'Auswirkung der Sanierung auf die Umweltbelas
     tung (1)
<77> 1200 DATA 1, 0,-1,-1,-1,-1,-1,-1,-2,-2,-2,. -2,-2,
     -3,-3,-3,-3,-3,-4,-4,-4,. -5,-5,-5,-6,-6,-7,-7,-8,
     -9,0,. 0,0,0,0,0,0,0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
<54> 1210 'Auswirkung der Sanierung auf sich selber (2)
<59> 1220 DATA 2, 0,0,0,0,0,0,0,0,0,0,. 0,0,0,0,0,0,0,0,0
     ,0,0,. 0,-1,-1,-2,-2,-3,-4,-5,-6,0,. 0,0,0,0,0,0,0,0
     ,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
<30> 1230 'Auswirkung der Produktion auf sich selber (3
     )
<27> 1240 DATA 3, 0,0,0,0,0,0,1,1,1,1,1,. 1,1,1,1,1,1,1,1,1
     ,1,1,. 1,2,2,2,2,2,0,0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,.
      0,0,0,0,0,0,0,0,0,0,0
<76> 1250 'Auswirkung der Produktion auf die Umweltbela
     stung (4)
<34> 1260 DATA 4, 0,0,0,0,1,1,1,1,2,2,. 2,2,3,3,4,4,5,5
     ,6,6,. 7,7,8,9,10,12,14,18,22,0,. 0,0,0,0,0,0,0,0,0,
     0,0,. 0,0,0,0,0,0,0,0,0,0,0
<51> 1270 'Auswirkung der Umweltbelastung auf sich selb
     er (5)
< 1> 1280 DATA 5, 0,0,0,-1,-1,-1,-1,-1,-1,-1,. -1,-1,-1
     ,-2,-2,-2,-2,-2,-2,-3,. -3,-3,-3,-4,-4,-3,-2,-1,0,
     0,. 0,0,0,0,0,0,0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
<28> 1290 'Auswirkung der Umweltbelastung auf die Leben
     squalität (6)
<27> 1300 DATA 6, 0,0,0,0,0,0,0,-1,-1,-1,. -2,-2,-2,-2,
     -3,-3,-3,-4,-4,-5,. -5,-6,-7,-8,-10,-12,-14,-18,-2
     5,0,. 0,0,0,0,0,0,0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
<68> 1310 'Auswirkung der Aufklärung auf sich selber (7
     )
<91> 1320 DATA 7, 0,0,-1,-1,-1,0,0,0,0,0,. 0,0,0,0,1,1,
     1,1,1,1,. 2,2,2,1,1,1,1,1,1,0,0,. 0,0,0,0,0,0,0,0,0,
     0,. 0,0,0,0,0,0,0,0,0,0,0
<7> 1330 'Auswirkung der Aufklärung auf die Lebensqual
     ität (8)
<86> 1340 DATA 8, -2,-2,-2,-2,-2,-1,-1,-1,-1,0,. 1,1,1,
     2,2,2,2,3,3,3,. 3,4,4,4,4,5,5,6,6,0,. 0,0,0,0,0,0,
     0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
<54> 1350 'Auswirkung der Aufklärung auf die Vermehrung
     srate (9)
<38> 1360 DATA 9, 1,0,0,0,0,0,0,0,0,1,. 1,1,1,2,2,2,2,2
     ,3,3,. 3,3,3,4,4,4,4,5,5,0,. 0,0,0,0,0,0,0,0,0,0,.
      0,0,0,0,0,0,0,0,0,0,0
<72> 1370 'Auswirkung der Lebensqualität auf sich selbe
     r (10)
<45> 1380 DATA 10, 0,1,1,0,0,0,0,0,0,1,. 1,1,2,1,1,0,0,
     -1,-1,-1,. -1,-1,-2,-2,-2,-1,-1,-1,0,0,. 0,0,0,0,0
     ,0,0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
<64> 1390 'Auswirkung der Lebensqualität auf die Vermeh
     rungsrate (11)
<35> 1400 DATA 11, -15,-8,-6,-4,-3,-2,-1,0,1,2,. 2,2,1,
     1,1,1,1,1,1,1,. 0,0,0,0,0,0,0,0,0,0,0,. 0,0,0,0,0,0,
     0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
<48> 1410 'Auswirkung der Lebensqualität auf die Politi
     k (12)
<79> 1420 DATA 12, -10,-8,-6,-3,-2,-1,-1,-1,-1,0,. 0,1,
     1,1,1,1,1,1,1,1,. 2,2,2,3,3,3,4,4,5,0,. 0,0,0,0,0,
     0,0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
<34> 1430 'Auswirkung der Vermehrungsrate auf die Bevöl
     kerung (13)
<90> 1440 DATA 13, -4,-4,-3,-3,-3,-2,-2,-2,-2,-1,. -1,-
     1,-1,-1,0,1,1,1,1,1,. 2,2,2,2,2,3,3,3,3,0,. 0,0,0,
     0,0,0,0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
<88> 1450 'Auswirkung der Bevölkerung auf die Lebensqua
     lität (14)
<65> 1460 DATA 14, 0,0,0,0,0,0,0,0,0,0,. 0,0,0,0,0,-1,-
     1,-1,-1,-1,-1,. -1,-1,-1,-1,-1,-2,-2,-2,-2,-2,. -3,-3
     ,-3,-3,-3,-3,-3,-3,-4,-4,. -4,-4,-5,-5,-6,-7,-8,-1
     0,0,0
<45> 1470 'Auswirkung der Bevölkerung auf die Aktionspu
     nkte (15)
<88> 1480 DATA 15, 0,0,0,0,0,0,0,0,0,1,. 1,1,1,1,1,1,2,
     2,2,2,. 2,3,3,3,3,4,4,4,4,5,. 5,5,5,6,6,6,6,7,7,7,
     . 7,8,8,8,8,9,9,9,0,0
< 0> 1490 'Auswirkung der Politik auf die Aktionspunkte
      (16)
<16> 1500 DATA 16, -5,-2,-1,-1,-1,-1,-1,-1,0,0,. 0,0,0,
     0,0,0,0,0,1,1,. 1,1,1,1,1,1,1,1,1,1,. 1,1,2,2,2,2,
     2,2,2,2,. 2,3,3,3,3,3,3,3,0,0
<19> 1510 'Auswirkung der Produktion auf die Aktionspun
     kte (17)
<63> 1520 DATA 17, -4,-3,-2,-2,0,2,2,3,3,3,. 4,4,4,5,5,
     5,6,6,6,7,. 7,8,8,9,9,10,10,11,11,0,. 0,0,0,0,0,0,
     0,0,0,0,. 0,0,0,0,0,0,0,0,0,0,0
< 8> 1530 'Auswirkung der Lebensqualität auf die Aktion
     spunkte (18)
<92> 1540 DATA 18, -6,-4,-2,0,0,1,1,1,2,2,. 2,2,2,2,2,2
     ,2,3,3,3,. 3,3,4,4,4,4,5,5,5,0,. 0,0,0,0,0,0,0,0,0
     ,0,. 0,0,0,0,0,0,0,0,0,0,0
< 9> 1550 '
<16> 1560 '********** UP Tabelle einlesen
<50> 1570 FOR j=0 TO 50
<80> 1580 READ tabelle%(j)
<21> 1590 NEXT j
<90> 1600 RETURN
<98> 1610 '
<46> 1620 '********** UP letzten Stand speichern
<27> 1630 ja%=jahr%:ak%=akt%:sa%=san%:pr%=pro%:um%=umw%
     :au%=auf%:le%=leb%:ve%=ver%:be%=bev%:po%=pol%: RET
     URN
< 8> 1640 '
<15> 1650 '********** UP oberes Fenster löschen
<68> 1660 FOR i=5 TO 23 STEP 2: PRINT FNloc$(3,i);SPACE
     $(80):NEXT i: RETURN
<17> 1670 '
<16> 1680 '********** UP unteres Fenster löschen
< 0> 1690 FOR i=0 TO 4: PRINT FNloc$(3,26+i);SPACE$(87)
     :NEXT i: RETURN
<97> 1700 '
<81> 1710 '********** UP Protokoll vorbereiten
<28> 1720 GOSUB 1680:PRINT FNloc$(3,27);"Bitte Papier e
     inlegen!": GOSUB 3200
<36> 1730 ZONE 10: LPRINT CHR$(15);CHR$(10);CHR$(14); "
     Protokoll zum Kybernetik-Lernspiel";CHR$(20): LPRI
     NT
<13> 1740 LPRINT "Jahr","Ereignis","Aktions-","Sanie-",
     "Produk-","Umweltbe-","Aufklä-","Lebens-","Vermeh-
     ","Bevölke-","Politik"
<66> 1750 LPRINT "","Nr.","punkte","rung","tion","lastu
     ng","rung","qualität","rungsrate","rung"
```

```
<11> 1760 RETURN
<19> 1770 '
<52> 1780 '********** UP Stand ausdrucken
<67> 1790 LPRINT jahr%,zahl%,akt%,san%,pro%,umw%,auf%,l
     eb%,ver%,bev%,pol%: zahl%=0: RETURN
< 0> 1800 '
< 4> 1810 '********** UP Investitionen ausdrucken
<91> 1820 LPRINT "..... Investition: ","",temp%(0),temp
     %(1),"",temp%(2),temp%(3),temp%(4):RETURN
< 9> 1830 '
<92> 1840 '********** UP Berechnung
< 8> 1850 GOSUB 1620: akt%=akt%-(temp%(0)+ABS(temp%(1))
     +temp%(2)+temp%(3)+temp%(4))
<28> 1860 san%=san%+temp%(0): pro%=pro%+temp%(1): auf%=
     auf%+temp%(2):leb%=leb%+temp%(3):ver%=ver%+temp%(4
     ): jahr%=jahr%+1
<17> 1870 GOSUB 1680: PRINT FNloc$(3,29);"Der neue Syst
     emzustand wird berechnet. Bitte warten..."
<29> 1880 GOSUB 1810: GOSUB 2370
<26> 1890 RESTORE 1200
<69> 1900 GOSUB 1560: umw%=umw%+tabelle%(san%): GOSUB 2
     370: RESTORE 1220: IF tabelle%(0) <> 1 THEN STOP
<81> 1910 GOSUB 1560: san%=san%+tabelle%(san%): GOSUB 2
     370: RESTORE 1240: IF tabelle%(0) <> 2 THEN STOP
<46> 1920 GOSUB 1560: pro%=pro%+tabelle%(pro%): GOSUB 2
     370: RESTORE 1260: IF tabelle%(0) <> 3 THEN STOP
<25> 1930 GOSUB 1560: umw%=umw%+tabelle%(pro%): GOSUB 2
     370: RESTORE 1280: IF tabelle%(0) <> 4 THEN STOP
<59> 1940 GOSUB 1560: umw%=umw%+tabelle%(umw%): GOSUB 2
     370: RESTORE 1300: IF tabelle%(0) <> 5 THEN STOP
<10> 1950 GOSUB 1560: leb%=leb%+tabelle%(umw%): GOSUB 2
     370: RESTORE 1320: IF tabelle%(0) <> 6 THEN STOP
<46> 1960 GOSUB 1560: auf%=auf%+tabelle%(auf%): GOSUB 2
     370: RESTORE 1340: IF tabelle%(0) <> 7 THEN STOP
<50> 1970 GOSUB 1560: leb%=leb%+tabelle%(auf%): GOSUB 2
     370: RESTORE 1360: IF tabelle%(0) <> 8 THEN STOP
<91> 1980 GOSUB 1560: IF tabelle%(0) <> 9 THEN STOP
<62> 1990 IF auf% > 20 THEN GOSUB 1680: PRINT FNloc$(3,
     29);"Soll die Vermehrungsrate steigen (+) oder fal
     len (-)? ": GOTO 2010.
<70> 2000 ver%=ver%+tabelle%(auf%): GOSUB 2370: RESTORE
      1380: GOTO 2040
<65> 2010 z=ASC(INPUT$(1)): IF z=43 THEN GOTO 2000
<62> 2020 IF z=45 THEN ver%=ver%-tabelle%(auf%): GOSUB
     2370: RESTORE 1380: GOTO 2040
<26> 2030 GOTO 2010
<96> 2040 GOSUB 1680: GOSUB 1560: leb%=leb%+tabelle%(le
     b%):GOSUB 2370: RESTORE 1400: IF tabelle%(0) <> 10
      THEN STOP
<89> 2050 GOSUB 1560: ver%=ver%+tabelle%(leb%): GOSUB 2
     370: RESTORE 1420: IF tabelle%(0) <> 11 THEN STOP
<29> 2060 GOSUB 1560: pol%=pol%+tabelle%(leb%): GOSUB 2
     370: RESTORE 1440: IF tabelle%(0) <> 12 THEN STOP
<34> 2070 GOSUB 1560: bev%=bev%+(tabelle%(ver%)*wachs%)
     : GOSUB 2370: RESTORE 1460: IF tabelle%(0) <> 13 T
     HEN STOP
<41> 2080 GOSUB 1560: leb%=leb%+tabelle%(bev%): GOSUB 2
     370: RESTORE 1480: IF tabelle%(0) <> 14 THEN STOP
<11> 2090 IF bev%=1 THEN wachs%=0
<21> 2100 IF 1<bev% THEN IF bev%<15 THEN wachs%=1
<32> 2110 IF 14<bev% THEN IF bev%<36 THEN wachs%=2
<33> 2120 IF 35<bev% THEN IF bev%<49 THEN wachs%=3
<19> 2130 GOSUB 1560: akt%=akt%+(tabelle%(bev%)*vers%):
      RESTORE 1500: IF tabelle%(0) <> 15 THEN STOP
<20> 2140 GOSUB 1560: akt%=akt%+tabelle%(pol%): RESTORE
      1520: IF tabelle%(0) <> 16 THEN STOP
<43> 2150 GOSUB 1560: akt%=akt%+tabelle%(pro%): RESTORE
      1540: IF tabelle%(0) <> 17 THEN STOP
< 5> 2160 IF pro%=1 THEN vers%=-4
<93> 2170 IF pro%=2 THEN vers%=-3
<41> 2180 IF 2<pro% THEN IF pro%<5 THEN vers%=-2
<77> 2190 IF 4<pro% THEN IF pro%<7 THEN vers%=-1
<28> 2200 IF 6<pro% THEN IF pro%<9 THEN vers%=0
<42> 2210 IF 8<pro% THEN IF pro%<30 THEN vers%=1
<22> 2220 GOSUB 1560: akt%=akt%+tabelle%(leb%): IF tabe
     lle%(0) <> 18 THEN STOP
<90> 2230 IF akt% > 50 THEN akt% = 50
<89> 2240 IF akt% < 0 THEN GOSUB 2370
<82> 2250 temp%(0)=0:temp%(1)=0:temp%(2)=0:temp%(3)=0:t
     emp%(4)=0
< 2> 2260 RETURN
<10> 2270 '
<40> 2280 '********** Ausgang zu CP/M
<43> 2290 PRINT con$;: SAVE "m:kyber": SYSTEM
<90> 2300 '
<68> 2310 '********** Ausgang zu BASIC
<61> 2320 PRINT con$;: STOP
< 0> 2330 '
< 8> 2340 '********** UP Endschalter setzen
<71> 2350 ende%=1: GOSUB 3200: RETURN
< 9> 2360 '
< 0> 2370 '********** UP auf Grenzwertüberschreitung pr
     üfen
<17> 2380 IF san% < 1 THEN RESTORE 2390: GOSUB 2950:GOS
     UB 3000: GOSUB 2340
<86> 2390 DATA "Das Ökosystem ist leider zusammengebroc
     hen, weil der Lebensraum während Ihrer"
<76> 2400 DATA "Regierungszeit total vergewaltigt wurde
     !","",""
< 5> 2410 IF san% > 29 THEN RESTORE 2420: GOSUB 2950: G
     OSUB 3000: GOSUB 2340
<61> 2420 DATA "Der Lebensraum Ihres Landes ist aufgrun
     d Ihrer weisen Führung und des Einsatzes"
<52> 2430 DATA "von angepaßter Technologie völlig intak
     t. Dies führt zu einer völlig anderen"
<44> 2440 DATA "Gesellschaftsform. Das Spiel ist beende
     t.",""
<80> 2450 IF pro% < 1 THEN RESTORE 2460: GOSUB 2950: GO
     SUB 3000: GOSUB 2340
<44> 2460 DATA "Ihre Vernachlässigung des Produktionsbe
     reiches hat den Zusammenbruch der Wirtschaft"
<63> 2470 DATA "bewirkt.","",""
<22> 2480 IF pro% > 29 THEN RESTORE 2490: GOSUB 2950: G
     OSUB 3000: GOSUB 2340
<60> 2490 DATA "Ihre Mißwirtschaft führte zu Überkapazi
     täten und Konsumterror!"
<23> 2500 DATA "Das Spiel ist beendet.","",""
<37> 2510 IF umw% < 0 THEN RESTORE 2520: GOSUB 2950: GO
     SUB 3000: GOSUB 2340
<29> 2520 DATA "Ungestörte Ökosysteme sind die Folge Ih
     rer Regierung. Die weitere Entwicklung"
<50> 2530 DATA "ist mit den hier geltenden Regeln nicht
      mehr beschreibbar.","",""
<91> 2540 IF umw% > 29 THEN RESTORE 2550: GOSUB 2950: G
     OSUB 3000: GOSUB 2340
< 9> 2550 DATA "Die Umweltbelastung ist derartig hoch,
     daß die natürliche Regulation völlig"
<83> 2560 DATA "zusammengebrochen ist.","",""
<81> 2570 IF auf% < 1 THEN RESTORE 2580: GOSUB 2950: GO
     SUB 3000: GOSUB 2340
<42> 2580 DATA "Sie haben die Aufklärung der Bevölkerun
     g über kybernetische Zusammenhänge"
<89> 2590 DATA "völlig versäumt, so daß Kurzsichtigkeit
      und unvernetztes Denken vorherrschen."
<23> 2600 DATA "Das Spiel ist beendet.",""
<25> 2610 IF auf% > 29 THEN RESTORE 2620: GOSUB 2950: G
     OSUB 3000: GOSUB 2340
<55> 2620 DATA "Dank Ihrer weitreichenden Aufklärungsar
     beit herrschen in der Bevölkerung"
<93> 2630 DATA "große kybernetische Einsicht und hoher
     Informationsstand."
<35> 2640 DATA "Das Spiel ist beendet.",""
<37> 2650 IF leb% < 1 THEN RESTORE 2660: GOSUB 2950: GO
     SUB 3000: GOSUB 2340
< 0> 2660 DATA "Die Lebensqualität ist derart miserabel
     , daß Hunger und Not herrschen und"
<30> 2670 DATA "die Sozialstruktur zusammengebrochen is
     t.","",""
<80> 2680 IF leb% > 29 THEN RESTORE 2690: GOSUB 2950: G
     OSUB 3000: GOSUB 2340
<73> 2690 DATA "Sie haben Ihrem Volk höchstes Wohlbefin
     den und sinnerfülltes Leben"
<21> 2700 DATA "ermöglicht. Das Spiel ist beendet.","",
     ""
<94> 2710 IF ver% < 1 THEN RESTORE 2720: GOSUB 2950: GO
     SUB 3000: GOSUB 2340
<52> 2720 DATA "Aufgrund Ihrer Regierungsentscheidungen
      ist die Geburtenrate so stark rück-"
<47> 2730 DATA "läufig, daß ein Systemzusammenbruch unv
     ermeidlich ist.","",""
<35> 2740 IF ver% > 29 THEN RESTORE 2750: GOSUB 2950: G
     OSUB 3000: GOSUB 2340
<20> 2750 DATA "Aufgrund Ihrer Regierungsentscheidungen
      ist die Geburtenrate so stark ange-"
< 8> 2760 DATA "stiegen, daß zur Bevölkerungsexplosion
     gekommen ist.","",""
<75> 2770 IF bev% < 1 THEN RESTORE 2780: GOSUB 2950: GO
     SUB 3000: GOSUB 2340
<76> 2780 DATA "Sie herrschen über ein menschenleeres L
     and. Die Bevölkerung ist ausgestorben!"
<67> 2790 DATA "Das Spiel ist aus.","",""
<98> 2800 IF bev% > 48 THEN RESTORE 2810: GOSUB 2950: G
     OSUB 3000: GOSUB 2340
< 9> 2810 DATA "Extreme Bevölkerungsdichte und total üb
     erbevölkerte Ballungsgebiete haben"
```

```
<72> 2820 DATA "Ihr Land unregierbar gemacht. Danken Si
     e ab!","",""
<81> 2830 IF pol% < 1 THEN RESTORE 2840: GOSUB 2950: GO
     SUB 3000: GOSUB 2340
<47> 2840 DATA "Sie sind ein miserabler Regierungschef
     und werden duch einen Staatsstreich"
<56> 2850 DATA "entmachtet.","",""
<23> 2860 IF pol% > 47 THEN RESTORE 2870: GOSUB 2950: G
     OSUB 3000: GOSUB 2340
<56> 2870 DATA "Als überaus fähiger Kybernetiker und St
     aatsmann erfreuen Sie sich höchster"
<24> 2880 DATA "Anerkennung und Beliebtheit im Volke!",
     "",""
<94> 2890 IF akt% < 0 THEN RESTORE 2900: GOSUB 2950: GO
     SUB 3000: GOSUB 2340
<69> 2900 DATA "Aufgrund Ihrer Mißwirtschaft ist die St
     aatskasse leer! Damit sind Sie handlungs-"
<22> 2910 DATA "unfähig geworden. Das Spiel ist aus.","
     ",""
<39> 2920 IF ende%=1 THEN GOSUB 1780: COMMON RESET: GOT
     O 60
< 7> 2930 RETURN
<15> 2940 '
<18> 2950 '********** UP ins Kommentarfenster schreiben
< 7> 2960 GOSUB 1680:FOR i=0 TO 3:READ fzeile$(i)
<79> 2970 PRINT FNloc$(3,27+i);fzeile$(i): NEXT i
<22> 2980 RETURN
<30> 2990 '
<31> 3000 '********** UP Kommentar ausdrucken
<89> 3010 FOR i=0 TO 3: IF fzeile$(i) <> "" THEN LPRINT
      fzeile$(i)
<80> 3020 NEXT i
<89> 3030 RETURN
<97> 3040 '
< 1> 3050 '
<57> 3060 '********** UP Zufallsereignisbestimmung
<94> 3070 runde%=runde%+1: IF runde% = 5 THEN runde% =
     0: GOSUB 1780: ELSE zahl%=0:RETURN
<29> 3080 zahl%=(ROUND(RND*30)+1)
<95> 3090 dagewesen%(ereigrunde%)=zahl%
<20> 3100 IF ereigrunde%=0 THEN GOTO 3130
<98> 3110 FOR i=0 TO (ereigrunde%-1): IF zahl%=dagewese
     n%(i) THEN GOTO 3080
<82> 3120 NEXT i
<75> 3130 ereigrunde%=ereigrunde%+1: IF ereigrunde%>30
     THEN ereigrunde%=0
<64> 3140 ON zahl% GOSUB 3260,3290,3340,3380,3420,3450,
     3480,3520,3560,3590,3630,3660,3710,3760,3790,3820,
     3860,3890,3930,3960,3990,4040,4080,4120,4160,4200,
     4240,4280,4310,4360,4390
<52> 3150 GOSUB 2950
<78> 3160 GOSUB 3000
< 8> 3170 GOSUB 3200
< 7> 3180 RETURN
<15> 3190 '
<27> 3200 '********** UP warten auf RETURN
< 8> 3210 PRINT FNloc$(75,30);xon$;" Drücke RETURN ";xo
     ff$;beep$:warte%=ASC(INPUT$(1)):IF warte%=13 THEN
     GOSUB 1680:RETURN:ELSE GOTO 3210
<95> 3220 '
<78> 3230 '********** UP Ereignisse
<58> 3240 DATA "Die gestaffelte Erhöhung des Strompreis
     es mit erhöhter Abnahme machte energie-"
<20> 3250 DATA "arme Technologien für die Industrie att
     raktiv: Sanierung +1","",""
<90> 3260 RESTORE 3240:san%=san%+1:RETURN
<52> 3270 DATA "Vereinfachte Anwendung von Verhütungsmi
     tteln (unter Sanktionierung durch den"
<77> 3280 DATA "Papst) erleichtert die Geburtenkontroll
     e: Vermehrungsrate -2","",""
<36> 3290 RESTORE 3270:ver%=ver%-2:RETURN
<12> 3300 DATA "Die konsequente Anwendung der Umweltsch
     utzgesetzgebung hat zum Übergang auf"
<13> 3310 DATA "Recycling-Methoden auf breiter Front ge
     führt. Abwasser- und Abfallmengen wurden"
<51> 3320 DATA "dadurch spürbar reduziert:"
<21> 3330 DATA"Umweltbelastung -2, Lebensqualität +1, A
     ktionspunkte +1"
<73> 3340 RESTORE 3300:umw%=umw%-2:leb%=leb%+1:akt%=akt
     %+1:RETURN
<33> 3350 DATA "Die angespannte Finanzlage der Großstäd
     te führt zum zunehmenden Verkauf von"
<27> 3360 DATA "Grünflächen und Erholungsgebieten:"
<14> 3370 DATA "Lebensqualität -3, Politik -1, Aktionsp
     unkte -1",""
<15> 3380 RESTORE 3350:leb%=leb%-3:pol%=pol%-1:akt%=akt
     %-1:RETURN
<10> 3390 DATA "Durch Umstellung auf moderne Kompostier
     ungsverfahren kann ein Teil des Abfalls"
<23> 3400 DATA "wiederverwendet werden. Der verringerte
      Kunstdüngereinsatz spart Energie und die Land-"
<58> 3410 DATA "wirtschaft muß weniger Pestizide einset
     zen: Umweltbelastung -2, Aktionspunkte +2",""
<37> 3420 RESTORE 3390:umw%=umw%-2:akt%=akt%+2:RETURN
<23> 3430 DATA "Durch anhaltende Inflation sind Ihre Ak
     tionspunkte weniger wert geworden:"
< 0> 3440 DATA "Aktionspunkte -2","",""
<49> 3450 RESTORE 3430:akt%=akt%-2:RETURN
< 5> 3460 DATA "Durch eine technische Panne sind in ein
     er Produktionsanlage giftige Chemikalien"
<36> 3470 DATA "in großem Maßstab in die Umwelt gelangt
     : Umweltbelastung +3","",""
<81> 3480 RESTORE 3460:umw%=umw%+3:RETURN
<91> 3490 DATA "Die plötzliche Verknappung einiger esse
     ntieller Rohstoffe durch unüberlegten"
<71> 3500 DATA "Raubbau zeigt ihre Auswirkungen in eine
     r Wirtschaftskrise:"
<52> 3510 DATA "Aktionspunkte -2, Politik -1",""
<12> 3520 RESTORE 3490:akt%=akt%-2:pol%=pol%-1:RETURN
<39> 3530 DATA "Da ein umweltfreundliches Energieerzeug
     ungsverfahren entwickelt wurde, ist die"
<83> 3540 DATA "Zeit günstig für Investitionen in alter
     native Techniken:"
<11> 3550 DATA "Sanierung +2, Aktionspunkte +1",""
<27> 3560 RESTORE 3530:san%=san%+2:akt%=akt%+1:RETURN
<48> 3570 DATA "Die Ausführung neuer Umweltgesetze wurd
     e durch eine lobbyhörige Bürokratie"
<34> 3580 DATA "verzögert: Umweltbelastung +2, Politik
     -2","",""
<59> 3590 RESTORE 3570:umw%=umw%-2:pol%=pol%-2:RETURN
<19> 3600 DATA "Die Einführung einer hohen Umweltabgabe
      führt dazu, daß immer mehr Industrie-"
<50> 3610 DATA "unternehmen auf energieärmere und umwel
     tfreundlichere Verfahren umsteigen:"
<96> 3620 DATA "Sanierung +2",""
<15> 3630 RESTORE 3600:san%=san%+2:RETURN
<36> 3640 DATA "Die geplante Einführung getrennter Krei
     släufe für Trink- und Brauchwasser"
<10> 3650 DATA "stößt in der Bevölkerung auf Widerstand
     : Aktionspunkte -1, Politik -1","",""
<46> 3660 RESTORE 3640:akt%=akt%-1:pol%=pol%-1:RETURN
<74> 3670 DATA "Durch die Entwicklung der Telekommunika
     tion ist eine breitere Streuung der"
<73> 3680 DATA "Arbeitsplätze, insbesondere im Dienstle
     istungsbereich, möglich. Der Berufsver-"
<98> 3690 DATA "kehr wird dadurch entflochten und spürb
     ar verringert:"
<80> 3700 DATA "Umweltbelastung -2, Lebensqualität +1,
     Aktionspunkte +1"
<31> 3710 RESTORE 3670:umw%=umw%-2:leb%=leb%+1:akt%=akt
     %+1:RETURN
< 4> 3720 DATA "Der Autoverkehr kostet Staat und Gemein
     den in Ballungsgebieten mehr als er an"
<59> 3730 DATA "an Kfz-Steuern einbringt. Trotzdem geht
      der Ausbau öffentlicher Verkehrsmittel"
<13> 3740 DATA "aufgrund kurzsichtiger Überlegungen wei
     ter zurück:"
<14> 3750 DATA "Umweltbelastung +2, Lebensqualität -2,
     Aktionspunkte -1"
<51> 3760 RESTORE 3720:umw%=umw%+2:leb%=leb%-2:akt%=akt
     %-1:RETURN
<98> 3770 DATA "Eine Verschärfung der internationalen L
     age hat erhöhte Rüstungsausgaben zur"
< 4> 3780 DATA "Folge: Aktionspunkte -3, Produktion +1"
     ,"",""
<23> 3790 RESTORE 3770:akt%=akt%-3:pro%=pro%+1:RETURN
<47> 3800 DATA "Es wurde ein neues energiesparendes Ver
     fahren zur Aluminiumgewinnung entdeckt:"
<68> 3810 DATA "Sanierung +2","",""
<46> 3820 RESTORE 3800:san%=san%+2:RETURN
<56> 3830 DATA "Die Einsatzmöglichkeiten für automatisc
     he Prozeßsteuerung in der Produktion"
<60> 3840 DATA "haben sich durch neue Anwendungen der M
     ikroelektronik erweitert:"
<58> 3850 DATA "Produktion +1, Lebensqualität -2 (wegen
      vorübergehender Arbeitslosigkeit)",""
<96> 3860 RESTORE 3830:pro%=pro%+1:leb%=leb%-2:RETURN
<50> 3870 DATA "Die Wirtschaft erleidet durch Wechselku
     rsänderunen Einbrüche auf Exportmärkten:"
<61> 3880 DATA "Produktion -2, Lebensqualität -1 (Arbei
     tslosigkeit)","",""
<15> 3890 RESTORE 3870:pro%=pro%-2:leb%=leb%-1:RETURN
<36> 3900 DATA "Nach der Aufdeckung eines Nahrungsmitte
     lskandals (Hormone im Fleisch) ist die"
<17> 3910 DATA "öffentliche Meinung günstig für Umstell
     ungen in der Landwirtschaft in Richtung"
```

```
<76> 3920 DATA "auf ökologischen Landbau: Sanierung +3,
      Aktionspunkte -1",""
<35> 3930 RESTORE 3900:san%=san%+3: akt%=akt% -1:RETURN
<17> 3940 DATA "Die Verwaltung hat große Summen in Schl
     achtprämien zur Verringerung der Milch-"
< 9> 3950 DATA "produktion investiert, die keinen Erfol
     g gezeitigt haben: Aktionspunkte -1","",""
<25> 3960 RESTORE 3940:akt%=akt%-1:RETURN
<31> 3970 DATA "Ein Großprojekt, bei dessen Planung öko
     logische Zusammenhänge mißachtet wurden,"
<48> 3980 DATA "zieht schwere Umweltbelastungen nach si
     ch: Umweltbelastung +3","",""
<86> 3990 RESTORE 3970:umw%=umw%+3:RETURN
<17> 4000 DATA "Wegen der wachsenden Resistenz bakterie
     ller Krankheitserreger, ausgelöst durch"
<49> 4010 DATA "weltweiten Antibiotikamißbrauch in der
     Massentierhaltung, läßt sich eine schwere"
<60> 4020 DATA "Darmerkrankung nicht behandeln und brei
     tet sich epidemisch aus:"
<60> 4030 DATA "Lebensqualität -3, Produktion -2, Bevöl
     kerung -2"
<25> 4040 RESTORE 4000:leb%=leb%-3:pro%=pro%-2:bev%=bev
     %-2:RETURN
<76> 4050 DATA "Die ständig wachsenden Beiträge zu über
     staatlichen Organisationen, in denen"
<73> 4060 DATA "veraltete Strukturen mit wachsendem Auf
     wand künstlich am Leben erhalten werden,"
<58> 4070 DATA "belasten zunehmend die Volkswirtschaft:
      Aktionspunkte -2, Politik -1",""
<10> 4080 RESTORE 4050:akt%=akt%-2:pol%=pol%-1:RETURN
<84> 4090 DATA "Nach einer chemischen Umweltkatastophe
     im Ausland finden Sanierungsmaßnahmen"
<74> 4100 DATA "und verschärfte Umweltschutzbestimmunge
     n Rückhalt in der öffentlichen Meinung:"
<11> 4110 DATA "Sanierung +3, Aktionspunkte -1, Umweltb
     elastung -1",""
<56> 4120 RESTORE 4090:san%=san%+3:akt%=akt%-1:umw%=umw
     %-1:RETURN
<56> 4130 DATA "Die politischen Verhältnisse in zwei wi
     chtigen Rohstofflieferländern haben sich"
<26> 4140 DATA "grundlegend geändert und zu drastischer
      Verteuerung der Rohstoffimporte geführt:"
<92> 4150 DATA "Produktion -4, Aktionspunkte -3",""
<42> 4160 RESTORE 4130:pro%=pro%-4:akt%=akt%-3:RETURN
<22> 4170 DATA "In einem Land, das zu den bedeutendsten
      Rohstofflieferanten gehört, ist ein"
<74> 4180 DATA "Bürgerkrieg ausgebrochen und hat die Ro
     hstoffproduktion praktisch zum Erliegen"
<58> 4190 DATA "gebracht: Produktion -3, Aktionspunkte
     -2",""
<20> 4200 RESTORE 4170:pro%=pro%-3:akt%=akt%-2:RETURN
<94> 4210 DATA "Es wurden neue, der Biologie des Mensch
     en und der Funktionsweise seines Gehirns"
< 2> 4220 DATA "besser angepaßte Lehr- und Lernmethoden
      entwickelt:"
<24> 4230 DATA "Aufklärung +3, Aktionspunkte -1",""
<33> 4240 RESTORE 4210:auf%=auf%+3: akt%=akt%-1:RETURN
<23> 4250 DATA "Durch neue Organisationsformen und Fort
     schritte im integrierten Pflanzenschutz"
<42> 4260 DATA "ist der ökologische Landbau wirtschaftl
     icher und attraktiver geworden:"
<95> 4270 DATA "Sanierung +2, Lebensqualität +1",""
<64> 4280 RESTORE 4250:san%=san%+2:leb%=leb%+1:RETURN
<44> 4290 DATA "Die Absenkung des Grundwasserspiegels f
     ührt nach einem trockenen Sommer zu"
<80> 4300 DATA "schweren Ernteeinbußen: Produktion -2,
     Umweltbelastung +2, Aktionspunkte -2","",""
<13> 4310 RESTORE 4290:pro%=pro%-2:umw%=umw%+2:akt%=akt
     %-2:RETURN
<48> 4320 DATA "Die Investitionen in ein Großprojekt, d
     as durch Wandel der wirtschaftlichen"
< 5> 4330 DATA "Bedingungen im Laufe seiner zehnjährige
     n Planungs- und Laufzeit bis zur Fertig-"
<17> 4340 DATA "stellung unwirtschaftlich geworden ist,
      sind verloren:"
<20> 4350 DATA "Aktionspunkte -2, Politik -1"
< 7> 4360 RESTORE 4320:akt%=akt%-2:pol%=pol%-1:RETURN
<39> 4370 DATA "Die Abwertung der DM verbilligt die hei
     mischen Produkte auf dem Weltmarkt und"
< 3> 4380 DATA "führt zu einem Exportboom: Produktion +
     3","",""
<29> 4390 RESTORE 4370:pro%=pro%+3:RETURN
<94> 4400 '
<57> 4410 '********** UP Meldung unter 10 Runden
<51> 4420 GOSUB 1680: PRINT FNloc$(3,27);"Eine Bewertun
     g ist erst ab 10 gespielten Runden sinnvoll. Spiel
     en Sie weiter!":GOSUB 3200
<98> 4430 RETURN
< 7> 4440 '
<37> 4450 '********** UP Bewertung
<53> 4460 bilanz%=ROUND((((pol%-10)+(3*leb%))*10)/(jahr
     %+3))
<97> 4470 IF bilanz% < 0 THEN RESTORE 4480: GOSUB 2950:
      GOSUB 3000: GOSUB 3200: RETURN
< 7> 4480 DATA "Nachdem Sie nun durch Ihre Entscheidung
     en ganzen Teilen der Bevölkerung die Lebens-"
<98> 4490 DATA "grundlage entzogen haben, sollten Sie s
     ich mit den Grundlagen kybernetischen Denkens"
< 6> 4500 DATA "noch einmal gründlich vertraut machen.
     Oder wollten Sie nur die Grenzen der Entschei-"
<78> 4510 DATA "dungsfreiheit kennenlernen, wo letztlic
     h alle Freiheiten auf Null sinken?"
< 7> 4520 IF bilanz% > -1 THEN IF bilanz% < 6 THEN REST
     ORE 4530: GOSUB 2950: GOSUB 3000: GOSUB 3200: RETU
     RN
<96> 4530 DATA "Um zu einem guten Steuermann zu werden,
      müssen Sie lernen, vorausschauender zu denken"
<86> 4540 DATA "und Wechselwirkungen zu berücksichtigen
     . Dann werden Ihre Entscheidungen weniger un-"
<30> 4550 DATA "reflektiert bzw. weniger heftig, und Si
     e lernen kurzsichtige, übersteuerte Reaktionen"
<86> 4560 DATA "vermeiden."
<45> 4570 IF bilanz% > 5 THEN IF bilanz% < 11 THEN REST
     ORE 4580: GOSUB 2950: GOSUB 3000: GOSUB 3200: RETU
     RN
<28> 4580 DATA "Sie haben sich für den Anfang nicht sch
     lecht geschlagen, denken aber wahrscheinlich"
<70> 4590 DATA "noch zu kurzsichtig. Sie können daher d
     as Ergebnis mit etwas weitsichtigeren Ent-"
<41> 4600 DATA "scheidungen im nächsten Spiel sicher ve
     rbessern.","",""
<11> 4610 IF bilanz% > 10 THEN IF bilanz% < 16 THEN RES
     TORE 4620: GOSUB 2950:GOSUB 3000: GOSUB 3200: RETU
     RN
<17> 4620 DATA "Sie sind mit den Rückwirkungen des Simu
     lationsspieles einigermaßen zurecht gekommen,"
<50> 4630 DATA "wenn auch im Schnitt die Ausgangsbeding
     ungen nicht gerade umwerfend verbessert worden"
< 1> 4640 DATA "sind. Immerhin haben Sie eine stabile G
     leichgewichtslage erreicht, die auch für die"
<42> 4650 DATA "Zukunft eine gewisse Lebensqualität sic
     hert."
<16> 4660 IF bilanz% > 15 THEN IF bilanz% < 21 THEN RES
     TORE 4670: GOSUB 2950: GOSUB 3000: GOSUB 3200: RET
     URN
<96> 4670 DATA "Sie haben das Ziel mit Bravour erreicht
     . Durch kluge Entscheidungen und vorausschau-"
<42> 4680 DATA "endes Steuern haben Sie den Zustand des
      Systems um einiges verbessert. Sie dürfen"
<82> 4690 DATA "sich als Kandidat für den 'Klub der kyb
     ernetischen Denker' fühlen.","",""
<54> 4700 IF bilanz% > 20 THEN RESTORE 4710: GOSUB 2950
     : GOSUB 3000: GOSUB 3200: RETURN
< 3> 4710 DATA "Sie sind ein wahres Genie! Der 'Klub de
     r kybernetischen Denker' nimmt Sie mit großem"
<44> 4720 DATA "Vergnügen in seine Reihen auf. Wenn nur
      Leute wie Sie an den entscheidenden Stellen"
<62> 4730 DATA "säßen, würden die kommenden Generatione
     n mit größerer Freude an die Taten ihrer"
<93> 4740 DATA "Vorfahren zurückdenken."
<16> 4750 '
<86> 4760 '********** UP Werte ändern (z.B. Startbeding
     ungen)
<29> 4770 change%=1: GOSUB 1650
<68> 4780 FOR j=1 TO 9
< 4> 4790 neu%=0
<90> 4800 ON j GOSUB 710,740,750,760,770,780,790,800,81
     0
<94> 4810 GOSUB 4870
<33> 4820 ON j GOSUB 730,740,750,760,770,780,790,800,81
     0
<12> 4830 NEXT j
<67> 4840 change%=0:neu%=0
<13> 4850 RETURN
<21> 4860 '
<28> 4870 '********** UP Korrektureingabe
<20> 4880 GOSUB 1680: PRINT FNloc$(3,27);"Um wieviele P
     unkte soll der letzte Wert nach oben oder unten ko
     rrigiert werden ?": PRINT FNloc$(75,30);xon$;" Drü
     cke RETURN ";xoff$;beep$
<53> 4890 PRINT FNloc$(3,29);:INPUT; "Korrekturwert (+
     oder -): ",neu%
< 0> 4900 RETURN
< 8> 4910 '
<13> 4920 '********** UP Korrekturausdruck
<77> 4930 LPRINT "....... Korrektur: ",akt%,san%,pro%,u
     mw%,auf%,leb%,ver%,bev%,pol%: RETURN
```



Dieser Beitrag bietet zweierlei: Dem Anwender ein Programm zur Entscheidungsfindung, aufgeteilt in einen zufälligen und einen subjektiven Teil. Dem Programmierer werden anhand eines ausgezeichnet dokumentierten Turbo-Pascal-Listings Lösungswege für das angegebene Thema sowie ein Beispiel für strukturierte Programmierung gegeben. Die Konvertierung in Mallard-Basic zu Lernzwecken wird empfohlen...

CHOICE

eine vielseitige Entscheidungshilfe

Zur Idee

Wer stand nicht schon mal vor dem Problem, sich nicht entscheiden zu können? Entweder kann man sich einfach nicht zu etwas »durchringen« oder die Fülle der Entscheidungskriterien ist zu groß, um Vergleiche auf einer vernünftigen Basis zu ziehen. Da es mir auch schon so ging und ich auch gerade Zeit und Idee für eine Lösung dieses Problems hatte, wollte ich dieser Angelegenheit zunächst mit Hilfe von TURBO-PASCAL durch die Anwendung und Auswertung von Zufallszahlen begegnen. Aber schon aus den ersten Reaktionen von »Kritikern« wurde mir klar, daß derartige »Entscheidungen« nur als »Jux« anzusehen sind, bzw. man sich nur dann nach ihnen richten könnte, wenn einem wirklich nicht der Sinn nach Analyse und Bewertung steht, sondern eine schnelle und unbeeinflußte Auswahl gefragt ist. So besteht letztendlich auch nur rund ein Viertel dieses Programms aus dem für die Zufallszahlen zuständigen Teil. Mein Ziel, diesen Entscheidungsgenerator zu schreiben, habe ich dann so verfolgt: Zum einen mußte ich mir Gedanken über den grundsätzlichen und ganz allgemeinen Ablauf einer Entscheidung machen. Anschließend war zu klären, wie ein Computer der JOYCE-Größe dieses Verfahren rechnerisch umsetzen kann. Erschwerend kam dann die Verknüpfung beider Programmteile hinzu und das Problem, daß mir bei der Erstellung schon bald ständig ein »Compiler Overflow« gemeldet wurde, so daß ich Ihnen nun ein Programm vorlege, das den TURBO-PASCAL- Compiler (zumindest meinen) fast völlig ausfüllt. Mit Änderungen müßten Sie also vorsichtig sein.

Kritik

Ich bin für jede Verbesserung und Kritik dankbar, insbesondere an einer effektiven Umsetzung in TURBO-Pascal. Aber mich interessiert auch, ob Sie meine Grundsatzüberlegungen teilen. Generell sollte der Anwender vielleicht eher ein analytisch / abwägend denkender Mensch sein, oder er ist einfach nur neugierig auf dieses Programm und möchte es testen. Ich kenne nämlich auch nette Leute, die mich nur kopfschüttelnd angesehen haben und auch überzeugend erklärten, für ihre Entscheidungen noch immer selbst sorgen zu können, spontan zu sein; auf jeden Fall sich aber niemals 'etwas von einem Computer sagen zu lassen'. Ich hoffe, daß Sie den Computer und dieses Programm wirklich als das sehen, was sie sind: Hilfsmittel, nichts weiter. Vor einer blinden »Computergläubigkeit» sei hier ausdrücklich gewarnt! *Sie* bewerten, *Sie* haben immer noch die Entscheidung nach der Auswertung, und *Sie* können letztlich auch das Programm bzw. die darin enthaltenen Vorgaben verändern! Ihnen wird also nichts vorgeschrieben, sondern nur aufgrund Ihrer Eingaben vorgeschlagen. Wenn Sie das Ganze unter diesem Aspekt betrachten können, werden Sie den Ergebnissen auch den angemessenen Stellenwert zukommen lassen!

Vorüberlegungen

Im Folgenden (und im Programm) nenne ich den Gegenstand der Entscheidung kurzerhand ein »Problem«. In des Wortes eigentlicher Bedeutung muß dieser keineswegs zum Problemfall geworden sein. Ich wollte aber eine kurze und allgemeine Bezeichnung haben.

Die Wegbereiter der Entscheidung sind die »Kriterien«. Sie dienen dazu, das »Problem« nach verschiedenen Gesichtspunkten zu beurteilen. Und schließlich gibt es noch bestimmte Rangordnungen unter den Kriterien, die durch eine »Gewichtung« festzulegen sind.

Wie verläuft eine Entscheidung?

Zuerst möchte ich einen Unterschied zwischen der »spontanen« und der »fundierten« Entscheidung machen. Die spontane Entscheidung hat in den meisten Fällen eher wenige »Probleme« zu bearbeiten und sie verläuft eben sehr schnell. Dabei gelangen nur ganz wenige Kriterien zum Einsatz, oftmals sind es nur persönliche Vorurteile bzw. die so-

genannten, mehr oder weniger bewußten »Vorlieben« für oder gegen etwas. Diese Entscheidung verläuft also oft nicht objektiv und in vielen Fällen ist dies sogar erwünscht.

Es ist eine schnelle Entscheidung, und wenn Sie von mehreren Personen z.B. jeweils eine solche erwarten würden, bekämen Sie sicherlich ganz unterschiedliche Resultate. Begründungen dazu sind auch sehr selten, es sei denn, Sie oder Ihre Testpersonen verfügen über ein sicheres und schnelles Urteilsvermögen. Nun gibt es aber unter den Spontanen auch solche Entscheidungen, die nicht nur schnell und unbegründet sein müssen sondern dazu auch noch relativ unwichtig sind. Zum Beispiel nur die Frage, was man heute abend mal unternehmen oder welchen Spielfilm man sich ansehen könnte. Beiden Fragen liegt natürlich schon eine Vorauswahl zugrunde, aus der man sich aber partout nicht so schnell entscheiden kann, da im Grunde die Vorlieben ziemlich gleichverteilt sind. Und genau diesen Entscheidungs-Typ könnte man aus den o.a. Gründen auch dem Computer anvertrauen.

Für diesen Fall ist die Auswahl nach Zufallszahlen ein mögliches Mittel der Entscheidung. Dagegen habe ich den Verlauf einer fundierten Entscheidung folgendermaßen analysiert: Zunächst sammelt man alle diejenigen »Probleme« zusammen, die sich überhaupt gemeinsam bearbeiten lassen. Sie müßten schon einigermaßen zusammenpassen. Zum Beispiel kann man sich sinnvoll zwischen verschiedenen Autos entscheiden oder z.B. auch zwischen den eigenen Hobbys. Aber ziemlich unpassend wäre wohl eine Entscheidung zwischen Autos und Hobbys.

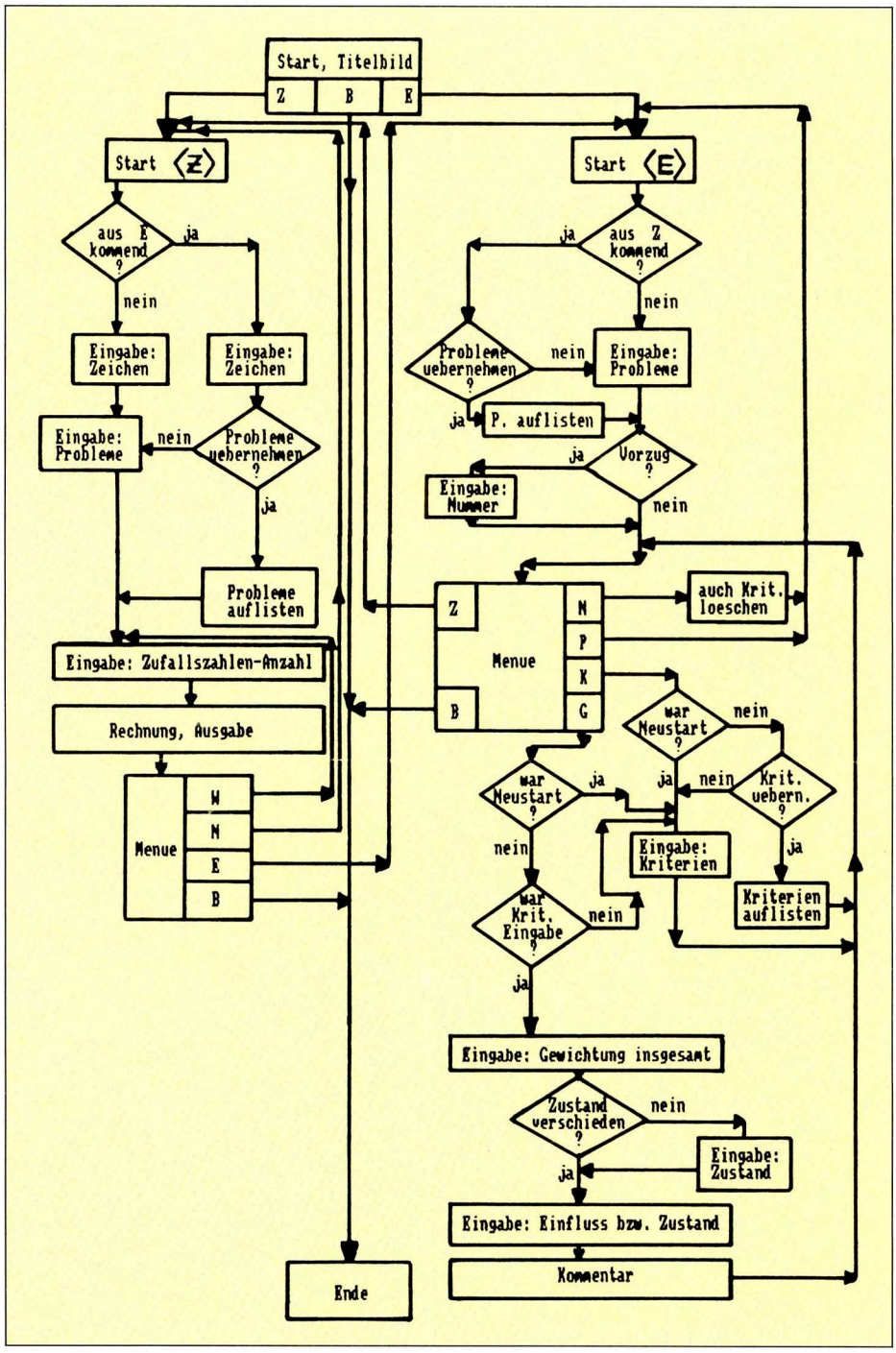


Abb.1: der Programmablauf von Choice

Zu welchem Ergebnis will man gelangen?

Sicherlich geht es doch um die Auswahl des Besten bzw. des den eigenen Vorgaben am besten entsprechenden »Problems«, also doch wohl z.B. des »optimalen« Autos, was soviel heißen soll, daß es sich dabei nicht immer um das »objektiv Beste«, d.h. von der Fachwelt oder der Allgemeinheit bevorzugte und anerkannte, Auto handeln muß.

Der persönliche »Geschmack« kann ganz anders entwickelt sein. Allerdings gibt es auch die Möglichkeit, daß jemand genau das objektiv beste Ergebnis erzielen möchte und seinen persönlichen Einfluß zurückstellt. Beides sollte das Programm leisten können.

Und wie ist meiner Meinung nach nun eine derartige Entscheidung zu führen? Es müssen geeignete Kriterien her, die somit erst einzelne Aspekte des »Problems« untersuchbar machen. Geht es z.B. um technische »Probleme« (z.B. Autos), dann bieten sich dafür deren Daten an, z.B.: Preis, Verbrauch, Ausstattung, Umweltfreundlichkeit, usw. Allerdings ist nun zu unterscheiden, ob man die objektive oder die persönliche Entscheidung (s.o.) sucht.

Bei Objektivität sollten auch nur die Daten Verwendung finden, bei persönlicher Auswahl kann man auch noch die sogenannten »Geschmacksfragen« hinzunehmen, z.B.: Farbe, Form, Fahreindruck, usw. Auf diese Trennung ist unbedingt zu achten, Vermischungen sind nicht ratsam und führen zu falschen Ergebnissen. Kriterien für die Ausübung von Hobbys (z.B. Sportarten) können z.B. sein: das Wetter, die zur Verfügung stehende Zeit, der voraussichtliche Zeitbedarf, momentane Motivation, der erwartete Spaß, die Kosten, u.a.m.

Wichtig ist also, daß die Kriterien überhaupt in einer geeigneten Position zu dem »Problem« stehen. Weiterhin sind sie so auszuwählen und zu formulieren, daß sie für jedes der »Probleme« eine wie auch immer geartete Aussagekraft ha-

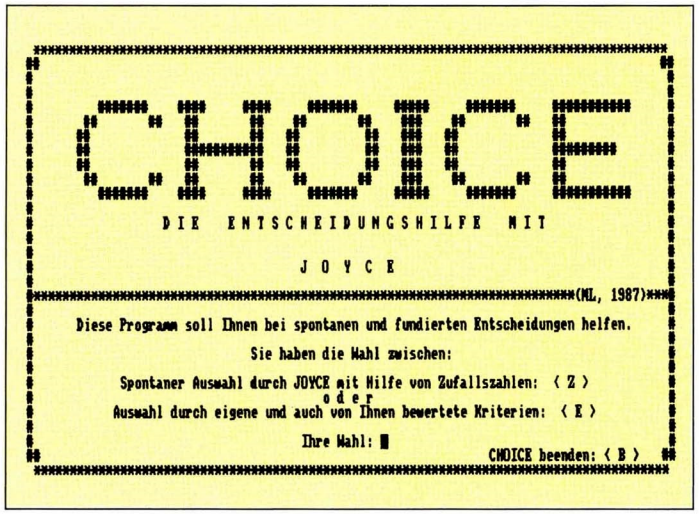


Abb.2: das Titelbild des Programms

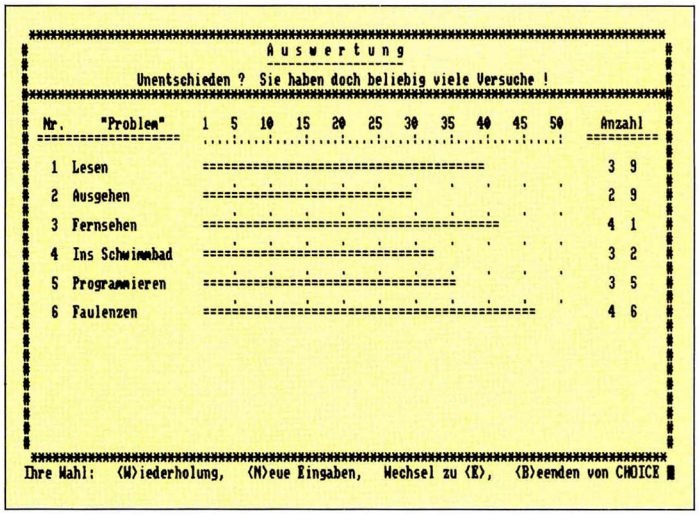


Abb.3: eine Zufallsentscheidung...

ben, denn sonst hat die Entscheidung ja keinen vergleichenden Charakter. Und den »letzten Schliff« bekommt die Entscheidung dann, wenn man sich auch noch Gedanken über die persönlich/objektive Wichtigkeit des Kriteriums im Vergleich mit den übrigen ausgewählten Kriterien gemacht hat. Dabei braucht man die »Probleme« nicht unbedingt »im Hinterkopf« zu haben; es sollte generell festgestellt werden, welchen Stellenwert man selbst (oder ein objektiver Betrachter) dem jeweiligen Entscheidungsgrund einräumt. Man wird sich sozusagen entweder über die eigene »Werteskala« klar oder man gewichtet doch schon ein wenig in Hinblick auf das allgemeine Thema der »Probleme«.

Zum Beispiel: Beim Thema Autokauf ist wohl unter objektiver Betrachtung das Kriterium »Preis« bzw. »Gegenwert« sehr wichtig; die jeweilige eigene Finanzlage kann aber bei persönlicher Einschätzung zu einer anderen Wertung führen, z.B. »weniger wichtig«.

Losgelöst vom Thema Autokauf kann die persönliche Einschätzung von »Gegenwert« wiederum anders ausfallen. Das entspräche dann eher der allgemeinen Einstellung des Betreffenden z.B. zu der Frage, wie sehr derjenige bei seinen Entscheidungen auf »Gegenwert« achtet oder nicht. Um nun endgültig zum Entscheidungsprozeß zu kommen, muß man nun noch den aktuellen Zustand des Kriteriums in Bezug zum jeweiligen »Problem« kennen.

Es ist vorteilhaft, dabei möglichst objektiv vorzugehen, denn damit hat man sowohl die Grundlage für eine eventuell beabsichtigte »objektive« als auch für eine nachvollziehbare »persönliche« Entscheidung.

Wie könnte man die Vorgaben auswerten?

Da man bei größerer Anzahl von »Problemen« und Kriterien nur sehr umständlich mit den Wörtern hantieren kann, greift man am besten zu Zahlen. Dabei bringt man die allgemeine Gewichtung des Kriteriums (z.B. in Punkten von 0 bis 3) mit dem ermittelten Zustand des Kriteriums (z.B. von −3 bis +3) für jedes »Problem« in Beziehung zueinander, z.B. durch Multiplikation. Anschließend werden diese Produkte für jedes »Problem« summiert. Das »Problem« mit der höchsten Punktzahl (eine positive Zahl ist 'höher' als eine vom Betrag her gleich große negative Zahl) müßte dann das Auszuwählende sein. Zusätzlich kann man am Vorzeichen des Ergebnisses auch erkennen, welchen »Wert« das Problem für den Fragesteller hat: eher besser (+), schlechter (−) oder neutral (0 Punkte). Eine Besonderheit habe ich mir noch ausgedacht: Eine ganz entscheidende subjektive Bewertung. (Die ist bei objektiver Auswertung natürlich nicht anzuwenden.) Man stellt sich die Frage, *wie* man selber den »Einfluß« des jeweiligen Kriteriums in dessen vorher bestimmten Zustand auf das »Problem« beurteilt. Damit wird die Möglichkeit geschaffen, ganz unterschiedlich und ganz persönlich zu urteilen.

Zum Beispiel: Es geht um das »Problem« »Sport treiben«, und das gerade anstehende Kriterium dafür ist »Wetter«, eingestuft als »wichtig«. Der Zustand des Wetters ist einigermaßen objektiv als »schlecht« (nach einem Blick durchs Fenster) festgestellt worden. Nun lautet die Frage: "Wie bewerte ich den Einfluß vom schlechtem Wetter auf »Sport treiben"« Falls ich z.B. Tennis spielen wollte, was wohl doch auch objektiv eher gutes Wetter erfordern würde, antworte ich mit »negativ«. Das wäre wohl eine übliche Antwort, die jeder geben würde.

Aber nun gibt es eben auch die Möglichkeit für ungewöhnlichere Antworten. Ich kann nun bestimmen, daß ich im Gegenteil »schlechtes Wetter« geradezu »positiv« für »Sport treiben« finde, falls ich z.B. bei solchem (evtl. regnerischen) Wetter gerne joggen würde. Für die zahlenmäßige Umsetzung dieser Besonderheit wähle ich wiederum positive und negative Faktoren für die Multiplikation mit dem vorherigen Produkt aus Gewicht und Zustand. Da ich diese letzte Entscheidung ausschlaggebend für die Gesamtauswertung halte, soll auch das Vorzeichen dieser Faktoren bestimmend für das Ergebnis sein!

Nun hat man meiner Meinung nach als »Bewertender« die Möglichkeit, alle Aspekte einer vollständigen Entscheidung zu berücksichtigen.

Zur Umsetzung in das Programm

Alle angesprochenen Annahmen finden sich im Programm CHOICE wieder. Nach dem gemeinsamen Titelbild kann in zwei grundsätzliche Programmteile (oder auch gleich zum Ende) verzweigt werden. Dabei stehen die Zeichen <X> für »Taste x drücken«. Bei Buchstaben ist Groß- oder Kleinschreibung möglich. Der eine Teil, zuständig für die spontaneren Entscheidungen mit Hilfe der Zufallszahlen, wird nur noch mit < Z für die eigenen Eingaben mit < E >. Beide Teile verfügen über Eingaberoutinen für die »Probleme«. Es ist dabei ganz egal, in welchem Programmteil sie

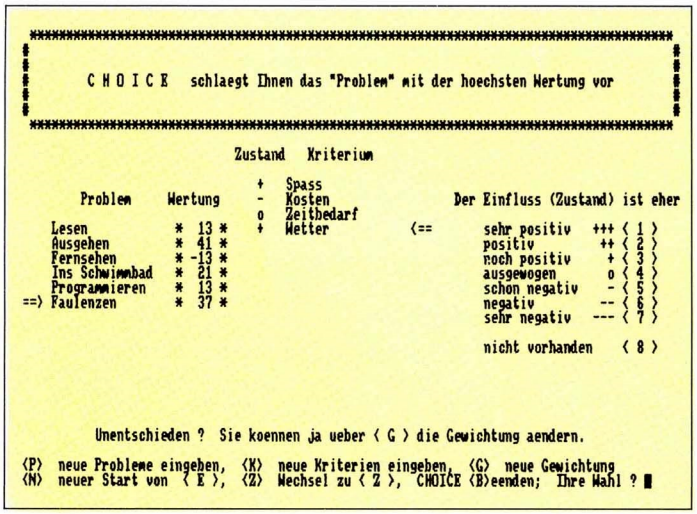


Abb.4: Joyce macht einen Vorschlag...

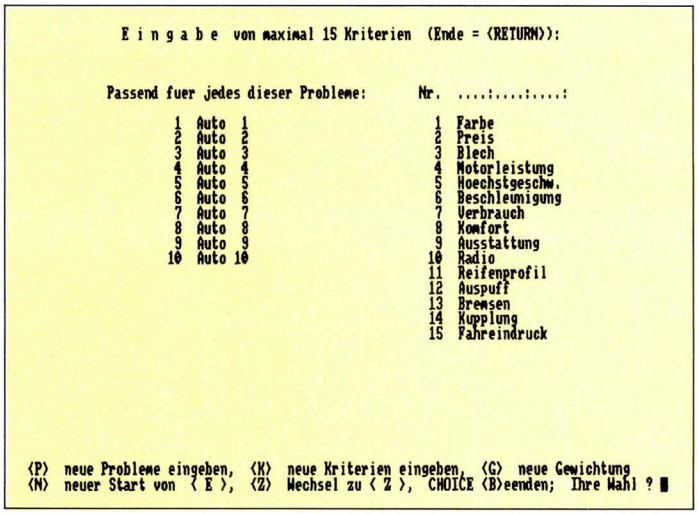


Abb.5: 10 Probleme können mit 15 Kriterien versehen werden

eingegeben werden. Da ein beliebiger Wechsel in den jeweils anderen Teil aus den Menüs möglich ist, wird der Anwender auch jeweils gefragt, ob die »Probleme« übernommen werden sollen (Siehe auch Abb. 1). Falls ja, werden sie aufgelistet, falls nein oder bei erstmaliger Eingabe erscheint ein Lineal mit der Länge von 15 Zeichen, unter dem man dann beliebige Texte oder Zeichen eingeben kann. Ein Hinausschreiben über den rechten Rand ist nicht weiter tragisch, außer vielleicht, daß diese Zeichen nicht gespeichert werden. Jedes »Problem« ist mit <RETURN> registriert und danach mit einer Nummer versehen worden. Das Ende der Eingabe wird durch nochmaliges <RETURN> erreicht. Maximal 10 »Probleme« nimmt das Programm entgegen, begrenzt durch Speicherplatz und Darstellbarkeit auf dem Bildschirm in Teil <Z>. Zwar ist der Bildschirmaufbau in <E> großzügiger bemessen, aber alle »Probleme« sollen ja auch in <Z> erscheinen können. Nur in Teil <Z> wird man vorher noch nach einem beliebigen Zeichen für den Aufbau des Balkendiagramms gefragt. Dessen Eingabe wird ebenfalls mit <RETURN> abgeschlossen. Die Eingabe der Kriterien für Teil <E> erfolgt analog zu den »Problemen«. Nachdem man in <Z> die Anzahl der Zufallszahlen mit <RETURN> eingegeben hat, erfolgt auch schon die Auswertung und Darstellung in Form von Balken. Zwar werden maximal nur 50 Zeichen pro Balken dargestellt, aber als Richtwert hat sich etwa die Eingabe von 100 je »Problem« bewährt, um längere Balken zu erzielen. Die Zufallszahlen werden nämlich vor der Sortierung und Summierung noch stark durch »MOD«-Anweisungen »gesiebt«. Es steht dem Anwender frei, auch sehr geringe Anzahlen einzugeben, z.B. nur 200 bei z.B. 10 »Problemen«. Dann werden die Balken entsprechend kürzer, eventuell entfällt dabei auch gar kein Element auf ein oder mehrere »Probleme«.

In der untersten Bildschirmzeile erscheint das zu <Z> gehörende Menü. Die Eingabe von <B> (ohne RETURN) beendet CHOICE; durch <N> startet man Teil <Z> von Anbeginn neu und muß das Zeichen sowie die »Probleme« neu bestimmen. Das läßt sich durch <W> vermeiden, dann wird man nur noch nach der Anzahl der Zufallszahlen gefragt, wobei die letzte Zahl in Erinnerung gebracht wird. <RETURN> genügt, und die Auswertung erfolgt auf Wunsch auch mit dieser Zahl. Diese schnelle Wiederholungsmöglichkeit soll bewußt auch den »Spieltrieb« unterstützen sowie auch glaubhaft machen, daß sich je Auswertung doch vieles verändern kann. Vielleicht gewinnt der Anwender dadurch generell eine ausreichend kritische Einstellung zu allen zufallsbehafteten Prozessen.

In Teil <E> ist das Menü der zentrale Anlaufpunkt. Nach dem Beginn von <E>, auch nach Verlassen von <Z> und (nach Bedarf) der Eingabe der »Probleme« gibt es nun die Möglichkeit, eventuell schon ein »Problem« im voraus zu bevorzugen. Oftmals gibt es ja schon eine mehr oder weniger bewußte Vorliebe für einen dieser »Kandidaten«. Das Programm rechnet dann am Schluß einen konstant hinzu. Nach der Bevorzugung (oder auch nicht) gelangt man dann in das Menü. Auch hier ist durch <B> die sofortige Beendigung und durch <Z> der Übergang in den Teil <Z> möglich. Käme man dann aus <Z< wieder zurück, wären die »Probleme« und Kriterien noch gespeichert. Nach der Anwahl von <P> oder <K> erscheint dann jeweils die Frage nach Übernahme der »Probleme« oder der Kriterien. Dann folgt wiederum das Menü. <N> führt zu einem Neustart von <E>, wobei die »Probleme« und Kriterien indirekt gelöscht werden durch Umstellen zweier logischer Variablen auf »false«. Durch <G> gelangt man in den abschließenden Teil der Gewichtung und der Auswertung. Eine interne Abfrage verhindert den Einstieg ohne Kriterien, um die Eingabe der Probleme kommt man ja nicht herum. Wie angesprochen, sind die Kriterien in ihrer Wichtigkeit für den Anwender zu beurteilen. Darauf folgt die Zustandsbeschreibung. Spätestens hier sollte man sich entschieden haben, ob der Zustand *eines* Kriteriums auch für jedes der »Probleme« Gültigkeit hat. Das ist nämlich bei der Eingabe gleichartiger »Probleme« oft nicht möglich, wie z.B. beim Autokauf. Hier ergibt sich ja der Sinn der Auswertung erst aus der Verschiedenheit (z.B. ist das Kriterium »Preis« mit Sicherheit von »Problem« »Auto« zu »Auto« unterschiedlich. Im Grunde zielt diese Frage (meiner Meinung nach) auf die Wahl einer eher »objektiveren« (also: gleichartige »Probleme« vorhanden) oder einer »subjektiveren« Entscheidung ab.

Ich möchte nämlich die persönlichere Entscheidung dann im dritten Teil der Gewichtung fördern (siehe »Einfluß«). Da aber die »Einfluß«- Frage (wie schon angesprochen) nicht für die »objektive« Auswahl gelten kann, jedoch aber in diesem Teil die komfortabelste Darstellung aller Parameter einschließlich des Durchlaufens aller gewichteten Kriterien je »Problem« gegeben ist, sollte auch die notwendige Zustandsbeschreibung (objektiv) aller Kriterien je »Problem« *hier* erfolgen. Auf diese Weise spare ich in der vorherigen, eigentlichen »Zustands-

Tabelle: Position und Definition der Labels von CHOICE

Bezeichn.	Zeile	Bedeutung
hauptwahl	107	Rücksprungziel bei falscher Eingabe
ende	388	Ende von CHOICE
start_z	116	Beginn von Teil ‹ Z ›
ueber_z	127	Rücksprungziel bei falscher Eingabe (bei Übernahme von "Problemen" in Teil ‹Z›
eingabe_z	141	Beginn des Eingabeteils für "Probleme" in ‹Z›
weiter_z	146	Sprungziel bei Auslassung der "Problem"-Eingabe
wieder_z	152	Sprungziel nur bei wiederholtem Durchlauf ‹W› ('logische Abfrage' wäre auch mögl. gewesen)
wahl_z	199	Rücksprungziel bei falscher Menü-Eingabe
exit1	156	Ende des zu überspringenden Abschnitts (bei Erstbeginn oder Neustart)
start_e	210	Beginn von Teil ‹ E ›
ueber_e	217	Rücksprungziel bei falscher Eingabe (bei Übernahme von "Problemen" in Teíl ‹E›
eingabe_e	232	Beginn des Eingabeteils in ‹E›
weiter_e	238	Sprungziel bei Auslassung der "Problem"-Eingabe
wahl_e	248	Sprungziel Menü
neu_k	263	Beginn des Eingabeteils für Kriterien
neu_g	289	Beginn des Eingabeteils für Gewichtung
ueber_e_k	266	Rücksprungziel bei falscher Eingabe (bei wiederholter Übernahme von Kriterien
exit2	308	Rücksprungziel bei falscher Eingabe der Gesamtgewichtung
exit3	335	Rücksprungziel bei falscher Eingabe der Zustandsbewertung
exit4	367	Rücksprungziel bei falscher Eingabe des Einflusses
gew	346	Sprungziel am Beginn der Eingabe des Einflusses, bzw. zum Zweck der Zustandseingabe an dieser Stelle, statt in der vorherigen Eingabe

Abb. 6

abfrage« die Routine, die ja später doch kommt, aber mit *der* Fragestellung eben nicht gültig sein kann. Beantwortet man also die Frage nach der Verschiedenheit der Probleme mit <J>, hat man sich für »objektiv« entschieden und gelangt sofort in die letzte Eingabemaske. Im Unterschied zum Ablauf mit vorheriger Zustandsbeschreibung werden natürlich auch keine Zustandssymbole (++, +, usw.) dargestellt, und der Anwender muß sich nun den veränderten Sinn der Eingabe vor Augen halten.

Statt der komplizierten Frage nach Einfluß hat er nur den Zustand zu beurteilen und kann sich dabei aber (sozusagen als Entschädigung) eines umfangreicheren Menüs bedienen. Falls man die allgemein gehaltenen Attribute manchmal nicht so recht auf die Frage (Einfluß oder Zustand) beziehen kann, gelingt dies vielleicht eher mit den Symbolen (+++,++, usw.), deren Bedeutung (in Anlehnung an bekannte Testergebnisse) geläufig sein dürften. Hinter allen Gewichtungen stehen Zahlen, über deren Vorzeichen, Höhe und Zuordnung zu den sinnreicheren Begriffen wie »wichtig« oder »positiv« jeder Anwender nach Studium des Listings selbst verfügen kann. Meine Vorbelegungen sind nur als Vorschlag zu sehen. Aber vor zu extremen Werten sei gewarnt: die maximale Summe aus 15 (Kriterien) mal max. Wichtigkeit mal max. Zustand mal max. Einfluß plus Vorzugskonstante darf den INTEGER-Zahlenbereich von −32768 bis +32767 nicht überschreiten! Bei meiner Voreinstellung ergibt sich eine maximale Punktzahl je »Problem« von: 15 * 3 * 3 * 4 + 10 = 550 (Der »Bonus« ist allerdings nur für einen »Kandidaten« möglich).

Die Auswertung besteht für CHOICE darin, Ihnen die Summen auszurechnen und darzustellen. Auswählen und endgültig entscheiden müssen Sie dann. Deswegen und auch aus Compilerplatzgründen verzichte ich auf eine Darstellung vom Typ: »Der Sieger ist ..... mit xxx Punkten«, um die Bedeutung des Programms nicht zu übertreiben. Außerdem besteht ja die Möglichkeit, daß es mehrere »Sieger nach Punkten« gibt. Was dann? Eine »Sieger-Routine« müßte zwangsläufig alle betreffenden »Kandidaten« auflisten und wäre Ihnen damit gedient? Sie müßten sich dann sowieso entscheiden, warum sollten also nicht auch gleichgroße Zahlen genügen?

Sollten Sie in dem Fall unentschieden sein (bzw. gab es überhaupt ein Unentschieden), so haben Sie die Möglichkeit, das Programm (evtl. aus Verdruß) zu beenden, sich ohne CHOICE/JOYCE oder auch überhaupt nicht zu entscheiden, oder aber Sie beginnen mit neuer Gewichtung bzw. sogar mit der Eingabe neuer Kriterien.

Zum Listing

Nach soviel Theorie bleibt schon gar nicht mehr viel zum Listing zu sagen. Die Flexibilität der Entscheidungen wird durch mehrere Labels unterstützt, die ich im Einzelnen durch Tabelle 1 erläutern möchte. Im Listing sind sie, ebenfalls nach Programmteilen geordnet, im Vereinbarungsteil zu finden. Anhand der Namensgebung dürfte der Umgang auch mit derart vielen Sprungzielen nicht schwer fallen. Die Absicherung gegen Fehlbedienungen geschieht weitgehendst durch die eindeutigen »Case«-Abfragen. So gibt es im Prinzip auch nur an den beiden Stellen »Ärger«, wo auf Zahleneingaben (Anzahl der Zufallszahlen, Vorzugs-Nummer) gewartet wird aber mehr oder weniger absichtlich etwas anderes eingegeben wird. Das Programm »stürzt ab«; aber nun, da Sie dies wissen, werden Sie auf derart überflüssige Versuche verzichten können. Die Fragen im Programm sind jedenfalls eindeutig.

Bedienung des Programms anhand eines Beispiels:

Angenommen, ich könnte mich heute nicht zwischen: Lesen, Ausgehen, Fernsehen, Schwimmen, Programmieren und Faulenzen entscheiden. Nach dem Starten von CHOICE erscheint das Titelbild (Abb. 2). Ich möchte zuerst eine zufällige Entscheidung sehen und wähle <Z>. Als Darstellungszeichen hätte ich gerne das Gleichheitszeichen = . Nacheinander gebe ich die »Probleme« mit <RETURN> ein und erhalte nach der sechsten Eingabe und erneutem <RETURN> einen kleinen Text und bekomme die Frage nach der Anzahl der Zufallszahlen gestellt. Ich gebe 555 ein, <RETURN>. Nach kurzem Warten ist die Auswertung beendet (Abb. 3). Ich

möchte zu <E> wechseln, drücke die entsprechende Taste, und beantworte auch die »Übernahme-Frage« mit der j-Taste. Die »Probleme« werden erneut aufgelistet und nach »Vorzug« gefragt. Ich bevorzuge Nummer 2.

Das Menü erscheint. Unabsichtlich wähle ich schon mal <G>, die Gewichtung an, aber es erscheint das Eingabefeld der Kriterien. Ich gebe möglichst passend zu den »Problemen« die Kriterien: Spaß, Kosten, Zeitbedarf und Wetter ein und befinde mich danach wieder im Menü.

Aber nun klappt es mit der Gewichtung <G> und ich gebe der Reihe nach meine Einschätzungen der Wertigkeit: Spaß <1>, Kosten <3>, Zeitbedarf <3> und Wetter <2> und des Zustandes ein.

So, wie ich momentan gelaunt bin, hätte ich wohl an *jedem* »Problem« »viel« Spaß → Zustand »gut« = <2>. Wenn das nicht der Fall wäre (mir aber »Spaß« »sehr wichtig« war), dann hätte ich die Hilfe von CHOICE kaum gebraucht. Zugegeben, das Kriterium »Kosten« ist leider sehr uneinheitlich für alle »Probleme«.

Im Grunde müßte man (nach meiner Definition) die »objektive« Entscheidung wählen, um dem jeweiligen Zustand gerecht zu werden. Aber ich möchte subjektiv bleiben und auch später den Einfluß bewerten wollen. Deswegen sehe ich es so:

Ich ärgere mich über diesen ungünstig gewählten Begriff (so etwas kann einem öfter passieren), interpretiere ihn als »meine Finanzlage«, und die ist »schlecht« = <4>.

Zum Zeitbedarf: Ich würde für alle gleich viel Zeit haben, und zwar genug = normal = <3>.

Das Wetter ist momentan gut = <2>.

Nun die Sache mit dem Einfluß: Ich halte den Einfluß von »Spaß« im Zustand »gut«, also auf gut Deutsch: von viel Spaß, auf »Lesen« für durchaus »positiv« = <2>. Kosten negativ, bzw. nun »schlechte Finanzlage« ist für »Lesen« ja positiv, da es bei bereits vorhandener Lektüre nichts kosten muß.

»Normaler Zeitbedarf« liegt für »Lesen« wohl auch im Rahmen, ist also »normal« = <4>.

Und schließlich finde ich gutes Wetter »schlecht« für das Lesen, da ich vorhatte, drinnen in der Wohnung zu lesen, ich aber bei solchem Wetter nicht gerne drinnen wäre.

Die Summe dieser Gewichtungen der vier Kriterien erbringt somit 13 Punkte für das »Problem« Lesen. Die höchste Punktzahl bekam nach diversen weiteren Eingaben das »Ausgehen«.

Wenn ich dem VORSCHLAG auch noch zustimmen mag, dann wäre dieses Ergebnis nach »Lage der Dinge«, bzw. meiner Vorgaben, für mich momentan »das Beste« (Abb. 4).

Sie erkennen vielleicht daran, das die überlegte Anwendung von CHOICE schon einige Gedankengänge erfordert.

Ich selber hatte ja schon viel Zeit damit verbracht.

Vielleicht sollten Sie eher mal mit einer »objektiven« Entscheidungsfindung beginnen, indem Sie z.B. Autos vergleichen (z.B. Abb. 5).

Viel Erfolg und viel Spaß!

(Matthias Laue)

```
1 Program CHOICE;  (*********************************************)
2                  (** Matthias Laue, 3300 Braunschweig ** 1987 **)
3 | $U+ |          (*********************************************)
4
5 Label hauptwahl,ende,
6       start_z,ueber_z,eingabe_z,weiter_z,wieder_z,wahl_z,exit1,
7       start_e,ueber_e,eingabe_e,weiter_e,wahl_e,
8       neu_k,neu_g,ueber_e_k,exit2,exit3,exit4,gew;
9
10 Type  wort      =string[15];
11      symb      =string[2];
12      symbol    =array[1..15] of symb;
13      indiz_wort=array[1..15] of wort;
14
15 Var   prob,krit,dummy: indiz_wort;
16      anz     :array[1..10] of integer;
17      a,b     :array[1..15] of integer;
18      zustand:symbol;
19      marker,behalten,weitermachen: boolean;
20      x,y,z,i,j,k,n,m,max,zeile,spalte,nr_vorzug,grenze,wieviel: integer;
21      zeichen,wahl : char;
22
23 Procedure Strich (zeile:integer);
24 Begin
25   for i:=2 to 89 do begin
26     gotoxy(i,zeile);write('*')
27   end
28 end;
29
30 Procedure Rand (zeile:integer);
31 Begin
32   for i:=0 to 1 do begin
33     for j:=0 to 4 do begin
34       gotoxy(1+89*i,zeile+j);write('#')
35     end
36   end;
37 end;
38
39 Procedure Eingabe(grenze,spalte,zeile:integer;Var wieviel:integer;Var dummy:indiz_wort);
40 Begin
41   j:=0;wieviel:=0;
42   gotoxy(spalte,zeile);write('Nr.  ....:....:....:');
43   repeat;
44     if j<>0 then begin
45       gotoxy(spalte,zeile+1+j);write(j:3)
46     end;
47     wieviel:=j;j:=j+1;
48     gotoxy(spalte+5,zeile+1+j);readln(dummy[j]);
49     if (j=grenze)and(dummy[j]<>'') then begin
50       gotoxy(spalte,zeile+1+j);write(j:3);
51       wieviel:=j
52     end;
53     if (j=grenze)and(dummy[j]='') then wieviel:=j-1
54   until (dummy[j]='')or(j=grenze)
55 end;
56
57 Procedure Menue (Var wahl:char);
58 Begin
59   gotoxy(1,30);write('<P>  neue Probleme eingeben,  <K>  neue Kriterien eingeben,  <G>  neue Gewichtung');
60   gotoxy(1,31);write('<N>  neuer Start von  < E >,  <Z>  Wechsel zu < Z >,  CHOICE <B>eenden;  Ihre Wahl ? ');
61   read(kbd,wahl)
62 end;
63
```

```
64 Procedure Loeschen (zeile:integer);
65 Begin
66   for i:= 3 to 88 do begin
67     gotoxy(i,zeile);write(' ')
68   end
69 end;
70
71 Procedure Listen (spalte,zeile,wieviel:integer; dummy:indiz_wort);
72 Begin
73   for i:=1 to wieviel do begin
74     gotoxy(spalte,zeile+i);write('  ',i:3,'  ',dummy[i])
75   end
76 end;
77
78
79 Begin (* Beginn des Hauptprogramms *)
80   clrscr;
81   n:=1;marker:=false;behalten:=false;weitermachen:=false;
82   zeile:=1;Strich(zeile);gotoxy(1,5);
83   writeln('         #######   ####     ####     #######    #####    #######    ##########');
84   writeln('       ##       ##  ##      ##    ##       ##   ###   ##       ##  ##');
85   writeln('      ##          ##      ##  ##          ##  ###  ##            ##');
86   writeln('      ##          ##########  ##          ##  ###  ##            ########');
87   writeln('      ##          ##      ##  ##          ##  ###  ##            ##');
88   writeln('       ##       ##  ##      ##    ##       ##   ###   ##       ##  ##');
89   writeln('         #######   ####     ####     #######    #####    #######    ##########');
90   gotoxy(20,13);writeln('D I E   E N T S C H E I D U N G S H I L F E   M I T');
91   gotoxy(39,16);writeln('J O Y C E');
92   zeile:=18;Strich(zeile);gotoxy(77,18);write('(ML, 1987)');zeile:=30;Strich(zeile);
93   for z:=0 to 5 do begin
94   zeile:=2+(4*z);Rand(zeile)
95   end; zeile:=25;Rand(zeile);
96   for z:=0 to 1 do begin
97     for k:=0 to 1 do begin
98       gotoxy(2+87*z,2+27*k);write('#')
99     end
100  end;
101  gotoxy(8,20);write('Diese Programm soll Ihnen bei spontanen und fundierten Entscheidungen helfen.');
102  gotoxy(32,22);write('Sie haben die Wahl zwischen:');
103  gotoxy(14,24);write('Spontaner Auswahl durch JOYCE mit Hilfe von Zufallszahlen:  < Z > ');
104  gotoxy(42,25);write('o d e r');
105  gotoxy(13,26);write('Auswahl durch eigene und auch von Ihnen bewertete Kriterien:  < E > ');
106  gotoxy(65,29);write('CHOICE beenden:  < B >');
107  hauptwahl:
108  gotoxy(39,28);write('Ihre Wahl: ');read(kbd,wahl);
109  Case wahl of (* Verzweigung: < Z >, < E > oder Ende *)
110    'b','B':goto ende;
111    'e','E':goto start_e;
112    'z','Z':goto start_z
113    else goto hauptwahl
114  end;
115
116  Start_z: clrscr; (* START von  < Z > *)
117    max:=10; (* Vorbelegung bei Erstbeginn < Z > und Neustart < N > *)
118    zeile:=1;Strich(zeile);
119    gotoxy(21,2);writeln('A u s w a h l   durch   Z U F A L L S Z A H L E N');
120    gotoxy(21,3);writeln('-----------------------------------');
121    gotoxy(3,4);writeln('Sie koennen bis zu 10 "Probleme" eingeben. Die Ausgabe erfolgt in Balkendiagramm-Form.
');
122    gotoxy(18,6);writeln('Ob Sie dann den Vorschlag annehmen liegt ganz bei Ihnen !');
123    zeile:=7;Strich(zeile);zeile:=2;Rand(zeile);
124    gotoxy(16,9);write('Welches Darstellungszeichen waehlen Sie ? (zB:  X, =, # ): '); readln(zeichen);
125    if marker then begin
```

```
126       gotoxy(6,11);write('Falls Sie vorher im Teil ‹ E › waren:  Uebernahme der "Probleme" ? ‹ J / N › ');
127       ueber_z: read(kbd,wahl);
128       case wahl of
129         'j','J':begin
130                   gotoxy(6,13);write('Das waren diese "Probleme":');
131                   spalte:=33;zeile:=12;wieviel:=n;Listen(spalte,zeile,wieviel,prob);
132                   goto weiter_z
133                 end;
134         'n','N':begin (* Loeschen der Zeile 11 *)
135                   zeile:=11;Loeschen(zeile);
136                   goto eingabe_z
137                 end;
138         else goto ueber_z
139       end
140     end;
141     eingabe_z:
142     gotoxy(14,11);writeln('E i n g a b e  von maximal 10 "Problemen"  (Ende = <RETURN>): ');
143     grenze:=10;spalte:=34;zeile:=13;
144     Eingabe(grenze,spalte,zeile,wieviel,prob);
145     n:=wieviel;marker:=true;
146     weiter_z:
147     gotoxy(1,27);writeln('Ueber die Anzahl der maximal zu bearbeitenden Zufallszahlen koennen Sie die Aussagekr
aft');
148     writeln('der Darstellung beeinflussen. Bei geringerer Anzahl gehen oft einige "Probleme" leer aus.');
149     zeile:=30;
150   (* Diese Stelle (bis EXIT1) wird bei Erstbeginn ‹ Z › oder Neustart ‹ N › uebersprungen *)
151     goto exit1;
152       wieder_z:
153       clrscr;gotoxy(8,2);writeln('(Fuer nochmaligen Durchlauf mit  g l e i c h e r  Anzahl:  <RETURN> )');
154       gotoxy(17,5);writeln('Im letzten Durchlauf waren es ',max:4,'  Zufallszahlen.');
155       zeile:=9;
156     exit1:
157     gotoxy(8,zeile);write('Wieviele Zufallszahlen sollen durchlaufen werden ?  (maximal: 1000): ');
158     readln(max);if max<1 then max:=1;if max>1000 then max:=1000;
159
160     clrscr;zeile:=1;Strich(zeile);gotoxy(35,2);write('A u s w e r t u n g');
161     gotoxy(35,3);writeln('-------------------');zeile:=5;Strich(zeile);
162     gotoxy(3,7);writeln(' Nr.     "Problem"  ');gotoxy(3,8);writeln('=================');
163     gotoxy(26,7);writeln('1   5   10   15   20   25   30   35   40   45   50     Anzahl');
164     gotoxy(26,8);writeln('....:....:....:....:....:....:....:....:....:....:   ========');
165     writeln;
166     for i:=1 to n do begin
167       writeln('  ',i:3,'  ',prob[i]);writeln
168     end;
169     for z:=0 to 5 do begin
170       zeile:=2+4*z;Rand(zeile)
171     end; zeile:=25;Rand(zeile);zeile:=30;Strich(zeile);gotoxy(34,31);write('Einen Moment bitte ...');
172     for i:=1 to n do anz[i]:=0; (* Vorbelegung *)
173   (* Zufallszahlen erzeugen, Summieren auf jeweiliges Kriterium *)
174     for i:=1 to max do begin
175       (* etwas 'rauhere' Verteilung durch mehrfachen Einsatz von:  M O D  *)
176       z:=random(n+1);
177       if(i mod 2)=0 then z:=0;if(i mod 5)=0 then z:=0;if(i mod 7)=0 then z:=0;
178       y:=1;
179       for x:=1 to n do begin (* und beliebig staerkere Gewichtung *)
180         if(i mod 9)=0 then y:=4;
181         if z=x then anz[x]:=anz[x]+y (* Hier: Summierung *)
182       end;
183     end;
184     x:=0; (* Ausgabe in Balkenform je Kriterium *)
185     for i:=1 to n do begin
186       x:=2*i;if anz[i]>50 then anz[i]:=50;
187       if i<n then begin
188         gotoxy(26,9+x);write('    .    .    .    .    .    .    .    .    .    .') end;
189       for j:=1 to anz[i] do begin
190         gotoxy(25+j,8+x);writeln(zeichen)
191       end;
192       gotoxy(82,8+x);
193       if anz[i]=0 then write('  -')
194       else if anz[i]<10 then write('  ',anz[i])
195       else write((anz[i] div 10):1,'  ',(anz[i] mod 10):1)
196     end;
197     gotoxy(17,4);write('Unentschieden ?  Sie haben doch beliebig viele Versuche !');
198     gotoxy(1,31);write('Ihre Wahl:  (W)iederholung,  (N)eue Eingaben,  Wechsel zu (E),  (B)eenden von CHOIC
E ');
199     wahl_z: read(kbd,wahl);
200     case wahl of
201       'w','W':goto wieder_z;
202       'n','N':begin
203                 marker:=false;goto start_z
204               end;
205       'e','E':goto start_e;
206       'b','B':goto ende
207       else goto wahl_z
208     end;
209
210   Start_e: clrscr; (* START von  ‹ E › *)
211     zeile:=1;Strich(zeile);zeile:=2;Rand(zeile);
212     gotoxy(16,3);write('Auswahl durch  e i g e n e  Kriterien und deren Gewichtung');
213     gotoxy(16,5);write('Sie koennen bis zu 10 "Probleme" und 15 Kriterien eingeben');
214     zeile:=7;Strich(zeile);
215     if marker then begin (* Eingabe der Probleme *)
216       gotoxy(4,10);write('Falls Sie vorher im Teil ‹ Z ›  waren:  Uebernahme der "Probleme" ? ‹ J / N › ');
217       ueber_e: read(kbd,wahl);
218       case wahl of
219         'j','J':begin
220                   gotoxy(4,12);write('Das waren diese "Probleme":');
221                   spalte:=32;zeile:=11;wieviel:=n;Listen(spalte,zeile,wieviel,prob);
222                   goto weiter_e
223                 end;
224         'n','N':begin (* Loeschen der Zeile 10 *)
225                   zeile:=10;Loeschen(zeile);
226                   goto eingabe_e
227                 end;
228         else goto ueber_e
229       end
230     end;
231
232     eingabe_e:
233     gotoxy(14,10);writeln('E i n g a b e  von maximal 10 "Problemen"  (Ende = <RETURN>): ');
234     grenze:=10;spalte:=34;zeile:=12;
235     Eingabe(grenze,spalte,zeile,wieviel,prob);
236     n:=wieviel;marker:=true;
237
238     weiter_e: nr_vorzug:=0;
239     gotoxy(12,26);write('Wollen Sie vorab vielleicht doch schon ein "Problem" bevorzugen ?');
240     gotoxy(19,27);Write('(Sozusagen als "heimliche Vorliebe")  ‹ J / N › ? '); read(kbd,wahl);
241     if (wahl='j')or(wahl='J') then begin
242       repeat;
243         gotoxy(58,28);write('    ');
244         gotoxy(26,28);write('Geben Sie bitte dessen Nr. ein: ');read(nr_vorzug);
245       until (nr_vorzug>0)and(nr_vorzug<=n)
246     end;
247
248     wahl_e: Menue(wahl);
249     case wahl of
250       'n','N':begin
251                 marker:=false;behalten:=false;goto start_e
252               end;
253       'p','P':begin
254                 marker:=false;clrscr;goto eingabe_e
255               end;
256       'k','K':goto neu_k;
257       'g','G':goto neu_g;
258       'z','Z':goto start_z;
259       'b','B':goto ende
260       else goto wahl_e
261     end;
262
263     neu_k: clrscr; (* Kriterien *)
264     if behalten then begin
265       gotoxy(7,4);write('Falls Sie schon mal  h i e r  waren:  Uebernahme der "Kriterien" ? ‹ J / N › ');
266       ueber_e_k: read(kbd,wahl);
267       case wahl of
268         'j','J':begin
269                   gotoxy(4,7);write('Das waren diese Kriterien:');
270                   spalte:=35;zeile:=6;wieviel:=m;Listen(spalte,zeile,wieviel,krit);
271                   goto wahl_e
272                 end;
273         'n','N':begin (* Loeschen der Zeile 4 *)
274                   zeile:=4;Loeschen(zeile)
275                 end;
276         else goto ueber_e_k
277       end
278     end;
279     gotoxy(14,1);write('E i n g a b e  von maximal 15 Kriterien  (Ende = <RETURN>):');
280     gotoxy(12,5);write('Passend fuer jedes dieser Probleme:');
281     for i:=1 to n do begin
282       gotoxy(19,6+i);write(i:3,'  ',prob[i])
283     end;
284     grenze:=15;spalte:=54;zeile:=5;
285     Eingabe(grenze,spalte,zeile,wieviel,krit);
286     m:=wieviel;behalten:=true;weitermachen:=true;
287     goto wahl_e;
288
289     neu_g: clrscr; (* Gewichtung *)
290     if m=0 then weitermachen:=false;
291     if weitermachen then begin
292       for i:=1 to m do begin
293         a[i]:=0;b[i]:=0
294       end;
295       gotoxy(29,1);write('GEWICHTUNG  der  Kriterien');
296       gotoxy(1,3);write('Wie wichtig ist dieses Kriterium  i n s g e s a m t  (im Vergleich aller Kriterien) ?'
);
297       gotoxy(60,10);write(' sehr wichtig     ‹ 1 ›');
298       gotoxy(60,12);write('    wichtig       ‹ 2 ›');
299       gotoxy(60,14);write('weniger wichtig   ‹ 3 ›');
300       gotoxy(60,16);write('   unwichtig      ‹ 4 ›');
301       for i:=1 to m do begin
302         gotoxy(15,6+i);write(krit[i])
303       end;
304       for i:=1 to m do begin
305         gotoxy(1,6+i-1);write('             ');
306         gotoxy(1,6+i);write('Kriterium ==>');
307         gotoxy(60,20);write('Ihre Bewertung:     ');
308         exit2: read(kbd,wahl);
309         case wahl of
310           '1':a[i]:=3;
311           '2':a[i]:=2;
312           '3':a[i]:=1;
313           '4':a[i]:=0
314           else goto exit2
315         end;
316       end;
317     end
318     else goto neu_k;
319     clrscr;
320     gotoxy(4,1);write('Gewichtung: Wie ist der ZUSTAND dieses Kriteriums (fuer alle Probleme) zur Zeit ?');
321     gotoxy(7,3);write('Ist bei allen Kriterien der Zustand je "Problem" verschieden  ‹ J / N › ?');
322     gotoxy(60,10);write('sehr gut       ++  ‹ 1 ›');
323     gotoxy(60,12);write('gut             +  ‹ 2 ›');
324     gotoxy(60,14);write('normal          o  ‹ 3 ›');
325     gotoxy(60,16);write('schlecht        -  ‹ 4 ›');
326     gotoxy(60,18);write('sehr schlecht   --  ‹ 5 ›');
327     for i:=1 to m do begin
328       zustand[i]:='  ';
329       gotoxy(15,6+i);write(krit[i])
330     end;
331     for i:=1 to m do begin
332       gotoxy(1,6+i-1);write('             ');
333       gotoxy(1,6+i);write('Kriterium ==>');
334       gotoxy(60,22);write('Ihre Bewertung :     ');
335       exit3: read(kbd,wahl);
336       case wahl of
337         '1':begin a[i]:=a[i]*3;zustand[i]:='++' end;
338         '2':begin a[i]:=a[i]*2;zustand[i]:=' +' end;
339         '3':begin a[i]:=a[i];zustand[i]:=' o' end;
340         '4':begin a[i]:=a[i]*-2;zustand[i]:=' -' end;
341         '5':begin a[i]:=a[i]*-3;zustand[i]:='--' end;
342         'j','J':goto gew
343         else goto exit3
344       end;
345     end;
346     gew: clrscr;
347     gotoxy(2,1);write('Welchen EINFLUSS (subjektiv) hat "Kriterium <=" in dessen Zustand auf: "==> Problem" ?'
);
348     gotoxy(28,4);write('G e w i c h t u n g  der  Kriterien');
349     gotoxy(4,7);write('Hatten Sie vorher "ZUSTAND" durch <J> uebersprungen, so bewerten Sie diesen jetzt:');
350     gotoxy(9,12);write('Problem     Wertung');gotoxy(30,9);write('Zustand   Kriterium');
351     gotoxy(60,12);write('Der Einfluss (Zustand) ist eher');
352     gotoxy(64,14);write('sehr positiv  +++ ‹ 1 ›');gotoxy(64,15);write('positiv         ++ ‹ 2 ›');
353     gotoxy(64,16);write('noch positiv     + ‹ 3 ›');gotoxy(64,17);write('ausgewogen       o ‹ 4 ›');
354     gotoxy(64,18);write('schon negativ    - ‹ 5 ›');gotoxy(64,19);write('negativ         -- ‹ 6 ›');
355     gotoxy(64,20);write('sehr negativ   --- ‹ 7 ›');gotoxy(64,22);write('nicht vorhanden   ‹ 8 ›');
356     for i:=1 to m do begin
357       gotoxy(32,10+i);write(zustand[i],'  ',krit[i])
358     end; j:=1;max:=1;
359     for i:=1 to n do begin
360       anz[i]:=0;
361       if i=nr_vorzug then anz[i]:=anz[i]+10;
362       gotoxy(1,12+i);write('    ');gotoxy(1,13+i);write('==> ',prob[i]);
363       gotoxy(54,10+m);write('    ');
364       b[j]:=0;
365       for j:=1 to m do begin
366         gotoxy(54,9+j);write('    ');gotoxy(54,10+j);write('<= ');
367         exit4: read(kbd,wahl);
368         case wahl of
369           '1':b[j]:=abs(a[j])*4;
370           '2':b[j]:=abs(a[j])*3;
371           '3':b[j]:=abs(a[j])*2;
372           '4':b[j]:=a[j];
373           '5':b[j]:=abs(a[j])*-2;
374           '6':b[j]:=abs(a[j])*-3;
375           '7':b[j]:=abs(a[j])*-4;
376           '8':b[j]:=abs(a[j])*0
377           else goto exit4
378         end;
379         anz[i]:=anz[i]+b[j]
380       end;
381       gotoxy(22,13+i);write('*',anz[i]:4,' *')
382     end;
383     zeile:=1;Strich(zeile);zeile:=2;Rand(zeile);
384     gotoxy(10,4);write('C H O I C E  schlaegt Ihnen das "Problem" mit der hoechsten Wertung vor');
385     zeile:=7;Strich(zeile);
386     gotoxy(11,28);write('Unentschieden ?  Sie koennen ja ueber ‹ G › die Gewichtung aendern.');
387   goto wahl_e;
388   ende: clrscr;
389 end.
```

KALENDER

Mit diesem Programm machen sich nun alle Joyce-Besitzer unabhängig von den Kalenderherstellern. Und zwar gleich für die nächsten dreihundert Jahre, bzw. noch länger. Es ist doch schon interessant zu wissen, an welchem Wochentag man eventuell seinen hundertsten Geburtstag feiern kann.

KALENDER ermöglicht den Ausdruck eines korrekten Kalenders irgendeines Jahres zwischen 1600 und 2300. Ein Termin vor 1582 wäre nicht sinnvoll, da damals Papst Gregor den derzeit gültigen julianischen Kalender reformierte und dieses Programm nur den Gregorianischen berücksichtigt.

Der Algorithmus zur Berechnung des Kalenders berücksichtigt selbstverständlich die Schaltjahre. Nach dem Gregorianischen Kalender sind alle Jahre, deren beiden Endziffern durch vier teilbar sind, Schaltjahre, allerdings mit Ausnahme der Jahre 1700, 1800, 1900, 2100, 2200, 2300 usw., die nicht durch 400 teilbar sind. So fallen also alle 400 Jahre drei Schaltjahre aus.

Das Programm selbst ist wie folgt aufgebaut: Bis Zeile 110 beinhaltet es das Titelbild und einige Initialisierungen. Es folgt die Abfrage des zu berechnenden Jahres. In Zeile 140 wird das Druckfeld des Kalenders für einen zweiten Kalender zurückgesetzt. Innerhalb der Schleife zwischen den Zeilen 170 und 240 wird die Druckposition des ersten Januars des betreffenden Jahres ausgerechnet. Alle weiteren Tage werden dann einfach »angereiht«.

KALENDER

Januar 2300

Son	Mon	Die	Mit	Don	Fre	Sam
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Februar 2300

Son	Mon	Die	Mit	Don	Fre	Sam
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

März 2300

Son	Mon	Die	Mit	Don	Fre	Sam
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

April 2300

Son	Mon	Die	Mit	Don	Fre	Sam
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Mai 2300

Son	Mon	Die	Mit	Don	Fre	Sam
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Juni 2300

Son	Mon	Die	Mit	Don	Fre	Sam
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Juli 2300

Son	Mon	Die	Mit	Don	Fre	Sam
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

August 2300

Son	Mon	Die	Mit	Don	Fre	Sam
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

September 2300

Son	Mon	Die	Mit	Don	Fre	Sam
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Oktober 2300

Son	Mon	Die	Mit	Don	Fre	Sam
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

November 2300

Son	Mon	Die	Mit	Don	Fre	Sam
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Dezember 2300

Son	Mon	Die	Mit	Don	Fre	Sam
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Abb. 1: falls es Sie interessieren sollte: so sieht der Kalender für das Jahr 2300 aus...

Zwischen Zeile 260 und Zeile 470 wird nun der eigentliche Kalender berechnet. Der DATA-Zeile 940 werden Monatsname und Anzahl der Tage dieses Monats entnommen und je Schaltjahr dem Februar ein Tag hinzugefügt. Der Kalendertext wird jeweils zu einer Zeile in druck−(r,z) geschrieben. Die Variable r steht für das erste oder zweite Halbjahr, z markiert die Zeile. Dimensioniert wurde das Feld mit DIM druck−(2,65), was für jeden möglichen Kalender genügend Platz läßt.

Ab Zeile 500 erhält man dann die Wahl der Ausdruckmöglichkeit. Einerseits Drucker oder Monitor, andererseits ein- oder zweispaltig. Beim Monitor scrollt der Kalender auf dem Bildschirm wie ein Listing durch und kann z.B. mit der f5-Taste bzw. ALT-S angehalten und wieder weiterlaufen gelassen werden. Beim Ausdruck auf dem Drucker paßt ein zweispaltiger Ausdruck auf ein DIN A 4 Blatt. Bei dieser Option bietet das Programm die Möglichkeit, den auszudruckenden Kalender mit einer besonderen Titelzeile zu versehen, die fett ausgedruckt wird. So kann man einen Kommentar oder Gruß jemandem, dem man einen »persönlichen« Kalender schenkt, auf den Ausdruck bringen.

Der jeweilige Ausdruck geschieht dann in den Zeilen 600 bis 850. Die darauffolgenden Programmzeilen fragen ab, ob etwa der gleiche oder ein anderer Kalender berechnet oder nur der berechnete Kalender ausgedruckt werden soll. Damit ein zweiter Programmdurchlauf nichts durcheinander bringt, geschieht in 960-980 eine FlagAbfrage bzw. -Setzung. Die Zeilen 1000 bis 1020 gewährleisten einen optisch korrekten Ausdruck bei der Zwei- Spalten-Wahl dadurch, daß jede Druckzeile auf 35 Zeichen verlängert wird. Ein Tip zum Abtippen: Es werden keine REMARK-Zeilen angesprungen. Sie können also durchaus weggelassen werden. Eventuell ist KALENDER eine Ergänzung zum Programm Termin'87 aus Schneider PC International Heft 7/87 und eine Grundlage für dessen Anpassungen an die folgenden Jahre. Vielleicht schafft jemand eine Verknüpfung der beiden Programme, auch wenn der Autor von Termin'87 einen immerwährenden Kalender nicht nötig fand. (C. Reimann)



LISTING >KALENDER<, REMARK = >REM<.

```
<48> 10 DIM druck$(2,65)'Dimensionierung des Druckfelde
     s
<42> 20 cls$=CHR$(27)+"E"+CHR$(27)+"H":PRINT cls$'Bilds
     chirm löschen
<56> 30 ' Titelbild
<26> 40 a$=CHR$(150)+STRING$(17,154)+CHR$(156):PRINT SP
     C(36)a$
<21> 50 b$=CHR$(149)+" K A L E N D E R "+CHR$(149):PRIN
     T SPC(36)b$
<14> 60 c$=CHR$(147)+STRING$(17,154)+CHR$(153):PRINT SP
     C(36)c$
< 4> 70 PRINT:PRINT:PRINT
<79> 80 PRINT SPC(32)CHR$(150)STRING$(27,154)CHR$(156)
<81> 90 PRINT SPC(32)CHR$(149)" ñ. by. Christoph. Reima
     nn "CHR$(149)
<95> 100 PRINT SPC(32)CHR$(147)STRING$(27,154)CHR$(153)
<47> 110 FOR i=1 TO 15:PRINT:NEXT
<49> 120 PRINT "Eingabe des Jahres (1600 bis 2300) ";
<85> 130 INPUT j
<43> 140 RESTORE:FOR z=65 TO 0 STEP -1:druck$(1,z)="":d
     ruck$(2,z)="":NEXT:PRINT:PRINT"Ich rechne":PRINT:P
     RINT"Bitte ca. 20 Sekunden warten
<85> 150 ' Bestimmung des ersten Tages, Berechnung der
     Druckposition
<84> 160 c=6
<90> 170.. FOR x=1600 TO j
<39> 180.. IF j=x THEN 230
<52> 190.. IF x/4<>INT(x/4) THEN 220
<11> 200.. IF (x-1700)*(x-1800)*(x-1900)*(x-2100)*(x-22
     00)*(x-2300)=0 THEN 220'Kontrolle, ob Schaltjahr
<98> 210.. c=c+2:GOTO 230
<80> 220.. c=c+1
<72> 230.. IF c>6 THEN c=c-7
<80> 240.. NEXT x
<97> 250.. 'Monatsroutinen
<76> 260.. FOR monat=1 TO 12:IF monat<7 THEN r=1 ELSE G
     OSUB 960
<86> 270.. READ monat$
<88> 280.. z=z+1:druck$(r,z)=SPACE$(INT(((35-LEN(monat$
     +" "+STR$(j)))/2)))+monat$+" "+STR$(j):GOSUB 1000
<81> 290.. READ b
<96> 300.. ' Addition eines Schalttages
<48> 310.. IF j/4<>INT(j/4) THEN 350
<57> 320.. IF a$<>"Februar" THEN 350
<67> 330.. b=b+1
<63> 340.. ' Kalender des Monats monat$ im Jahre j
<31> 350.. z=z+1:druck$(r,z)=". Son. Mon. Die. Mit. Don
     . Fre. Sam.. ":GOSUB 1000
<30> 360.. z=z+1:FOR d=1 TO b
<77> 370.... druck$(r,z)=druck$(r,z)+SPACE$(5-LEN(STR$(
     d)))+STR$(d)
<71> 380.... c=c+1
<90> 390.... IF c<7 THEN 420
<69> 400.... druck$(r,z)=SPACE$(35-LEN(druck$(r,z)))+dr
     uck$(r,z):z=z+1
<27> 410.... c=0
<31> 420.... NEXT d:GOSUB 1000
<74> 430.. IF d<20 THEN druck$(r,z)=SPACE$(35-LEN(druck
     $(r,z)))+druck$(r,z):z=z+1:GOSUB 1000:GOTO 450
<65> 440.. GOSUB 1000:z=z+1
<76> 450.. druck$(r,z)=q$:GOSUB 1000
<69> 460.... z=z+1:druck$(r,z)=SPACE$(36)
<24> 470 NEXT monat
< 3> 480 PRINT cls$
<32> 490 PRINT TAB(34);"Ausdruck des Kalenders
<73> 500 PRINT:PRINT TAB(30)"1.. einspaltig auf dem Mon
     itor
<96> 510 PRINT:PRINT TAB(30)"2.. zweispaltig auf dem Mo
     nitor
<76> 520 PRINT:PRINT TAB(30)"3.. einspaltig auf dem Dru
     cker
<75> 530 PRINT:PRINT TAB(30)"4.. zweispaltig auf dem Dr
     ucker
<18> 540 PRINT:PRINT:PRINT TAB(30)"Bitte wählen:":PRINT
     :PRINT:PRINT:PRINT:PRINT
<95> 550 abfr$=INKEY$:IF abfr$="" THEN 550
<19> 560 IF abfr$<"1" OR abfr$>"4" THEN 550
<77> 570 ON ASC(abfr$)-48 GOTO 600,670,720,790
<38> 580 '
<47> 590 ' Ausdruck des Kalenders einspaltig auf dem Mo
     nitor
<26> 600 FOR sp=1 TO 2
<88> 610.. FOR z=0 TO 65
<92> 620.. PRINT:PRINT TAB(22)druck$(sp,z)
<96> 630 NEXT:NEXT
<53> 640 GOTO 870
<33> 650 '
<70> 660 ' Ausdruck des Kalenders zweispaltig auf dem M
     onitor
<85> 670 PRINT TAB(36);a$:PRINT TAB(36);b$:PRINT TAB(36
     );c$
<66> 680 FOR z=0 TO 65:PRINT TAB(8);druck$(1,z);:PRINT
     TAB(47);druck$(2,z):NEXT
<63> 690 GOTO 870
<24> 700 '
<57> 710 ' Ausdruck des Kalenders einspaltig auf dem Dr
     ucker
<31> 720 FOR sp=1 TO 2
<93> 730.. FOR z=0 TO 65
<10> 740.. LPRINT druck$(sp,z)
<36> 750 NEXT:LPRINT:LPRINT:LPRINT:LPRINT:LPRINT:NEXT
<58> 760 GOTO 870
<38> 770 '
<75> 780 ' Ausdruck des Kalenders zweispaltig auf dem D
     rucker
<64> 790 INPUT"Anderer Titel gewünscht? Bitte eine Zeil
     e eingeben:";abfr$
<30> 800 IF LEN(abfr$)>80 THEN 790
<41> 810 IF abfr$<>"" THEN b$=abfr$ ELSE b$="KALENDER"
<37> 820 b$=CHR$(27)+"G"+CHR$(27)+"E"+b$+CHR$(27)+"F"+C
     HR$(27)+"H"
<91> 830 LPRINT TAB(40-LEN(b$)/2);b$:LPRINT
<22> 840 FOR z=0 TO 65:LPRINT druck$(1,z);:LPRINT TAB(4
     0);druck$(2,z):NEXT
<57> 850 GOTO 870
<37> 860 '
<75> 870 PRINT:PRINT:PRINT"Neuer Kalender mit anderer J
     ahreszahl? (J/N)";:INPUT abfr$
<89> 880 abfr$=UPPER$(abfr$):IF abfr$="J" THEN frg=0:GO
     TO 120
<93> 890 IF abfr$<>"N" THEN 870
<85> 900 PRINT"Kalender des gleichen Jahres? (J/N)";:IN
     PUT abfr$
<42> 910 abfr$=UPPER$(abfr$):IF abfr$="J" THEN 480
<28> 920 IF abfr$<>"N" THEN 900
<37> 930 END
<84> 940 DATA Januar,31,Februar,28,März,31,April,30,Mai
     ,31,Juni,30,Juli,31,August,31,September,30,Oktober
     ,31,November,30,Dezember,31
<59> 950 'Abfrage, ob Durchlauf erste Jahreshälfte
<12> 960 IF frg=1 THEN 980
<40> 970 z=-1:r=2:frg=1
<91> 980 RETURN
<81> 990 'Erweiterung von druck$(r,z) auf 35 Zeichen Lä
     nge
<72> 1000 IF LEN(druck$(r,z))>35 THEN 1020
<64> 1010 druck$(r,z)=druck$(r,z)+SPACE$(36-LEN(druck$(
     r,z)))
<84> 1020 RETURN
```

Ein Spiel für bis zu fünf Personen und einen Computer in Anlehnung an ein weltbekanntes Maklerspiel.



Start des Programms

Auf Ihrer Diskette müssen die Dateien SUBURBIA.BAS (21k), FONT__SUB.BAS (2k) und KAR TE.SB (9k) vorhanden sein. Laden Sie das System und geben Sie BASIC SUBURBIA ein. Folgen Sie dann den Anweisungen auf dem Bildschirm. Die Basic-Listings können Sie von BASIC aus eintippen. Die seltsam anmutenden Zeichen sind Joyce Sonderzeichen wie z.B. der gefüllte bzw. der ungefüllte Kreis (siehe Abb.1). Die Datei KARTE.SB erzeugen Sie wie folgt unter LocoScript: Im Menüpunkt ERSTELLEN tippen Sie den Text zu KARTE.SB unter dem Dateinamen TEXT.000 ab, wobei nach jeder Zeile die RETURN-Taste folgen muß. Nach der Eingabe gehen Sie mit EXIT (beenden) ins Hauptmenü, von wo aus Sie jetzt mit »f7« (ASCII-Datei erstellen) Ihren Text umwandeln. Geben Sie der ASCII-Datei den Namen KARTE.SB und speichern Sie *unformatiert* ab. Geben Sie die Sonderzeichen (geschweifte Klammern, Tilde usw.) wie angegeben ein, damit später die Umlaute stimmen. Für die seltsamen Zeichen gilt das oben Gesagte.

Warnung

Während des Programmablaufs werden Teile des Programms, die später nicht mehr benötigt werden, gelöscht, um weiteren Speicherplatz zu gewinnen. Es wird also empfohlen, vor dem Starten des Programms dieses auf Diskette abzuspeichern und ggf. eine Sicherheitskopie anzufertigen!

Das Spiel

Der Computer ist der erste Spieler (JOYCIE). Das sich im Umlauf befindliche Geld setzt sich wie folgt zusammen:

Bankvermögen:	*$500000,–*
Wert der Straße:	*$109900,–*
Wert der Häuser:	*$164750,–*
Gesamt:	*$774650,–*

Jeder Spieler bekommt zu Beginn des Spieles $40000,– Startkapital von der Bank. Außerdem wird der Pott mit $1000,– aus dem Bankvermögen gefüllt.

Im Maximalfall (fünf Spieler plus JOYCIE) hat die Bank also $259000,– Startkapital (bar) = $533650,– Gesamtkapital. Die Würfel werden durch Betätigen einer beliebigen Taste angehalten. Es wird um die gewürfelte Augenzahl beider Würfel im Uhrzeigersinn vorgerückt. Wird ein Pasch gewürfelt, darf der Spieler noch einmal ziehen. Es darf beliebig oft hintereinander ein Pasch gewürfelt werden, ohne daß dem Spieler daraus Nachteile entstehen. Es können mehrere Spieler auf einem Feld stehen. Gestartet wird auf dem Feld Arbeitsamt (Feld 4 – die Felder sind im Uhrzeigersinn von 0 bis 35 durchnumeriert; beginnend mit dem ???-Feld auf der unteren Hälfte des Stadtplans).

Der Pott: In ihm landen alle Steuergelder, Strafen, u.ä. (Parkplatz).

Spezialfelder

Arbeitsamt (4) Überquert ein Spieler dieses Feld in der angegebenen Richtung, so erhält er $5000,– Arbeitslo-



Foto Benedikt

senunterstützung (schließlich arbeiten Sie ja nicht beim Spielen – oder?) aus der Bank.

Erreicht er genau das Feld, so erhält er $10000,–.

Schwarzwaldklinik (13) /

Gehen Sie in die Schwarzwaldklinik (31)

Erreicht ein Spieler Feld (13), so muß er dort verweilen, bis er einen Pasch würfelt oder einer der Würfel eine 6 zeigt. Dann erst darf er weiterziehen; ist er durch einen Pasch entlassen worden, so darf er nochmals würfeln. Erreicht ein Spieler Feld (31), so muß er sich in die Schwarzwaldklinik begeben und darf auf seinem Weg dorthin nicht $5000,– einziehen.

Parkplatz (22)

Landet der Spieler auf Feld (22), so erhält er den Pott, dieser wird wieder mit $1000,– aus der Bank gefüllt.

???-Felder (0,16) / !!!-Felder (10,29)

Auf diesen Feldern können verschiedene Ereignisse auftreten. Den Forderungen auf dem Bildschirm (rechts unten) muß Folge geleistet werden.

Grundsteuer (6) / Vermögenssteuer (25)

Auf diesen Feldern muß der Spieler fünf Prozent Grundsteuer bzw. zehn Prozent Vermögenssteuer zahlen. Als zu versteuernder Betrag versteht sich ausschließlich das Barvermögen des Spielers.

Straßen

Erreicht ein Spieler eine Straße, so kann er sie von der Bank zum aufgeschriebenen Preis kaufen, sofern sie noch nicht im Besitz eines anderen Mitspielers ist.

Bebaubare Straßen:
1, 3, 5, 8, 11 – 12, 14 – 15, 18 – 20, 23 – 24, 27, 30, 33 – 34

Ist der Spieler bereits im Besitz der Straße, so kann er darauf bauen. Er



Abb. 1: Das »Titelbild« von SUBURBIA

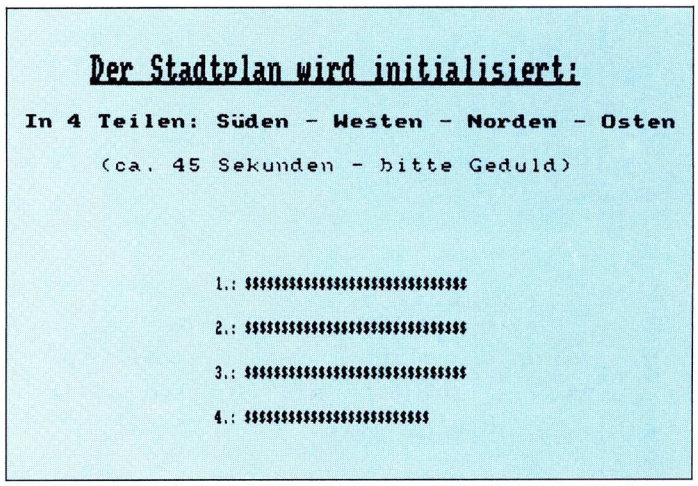


Abb. 2: Joyce initialisiert gerade das letzte Spielplanviertel...

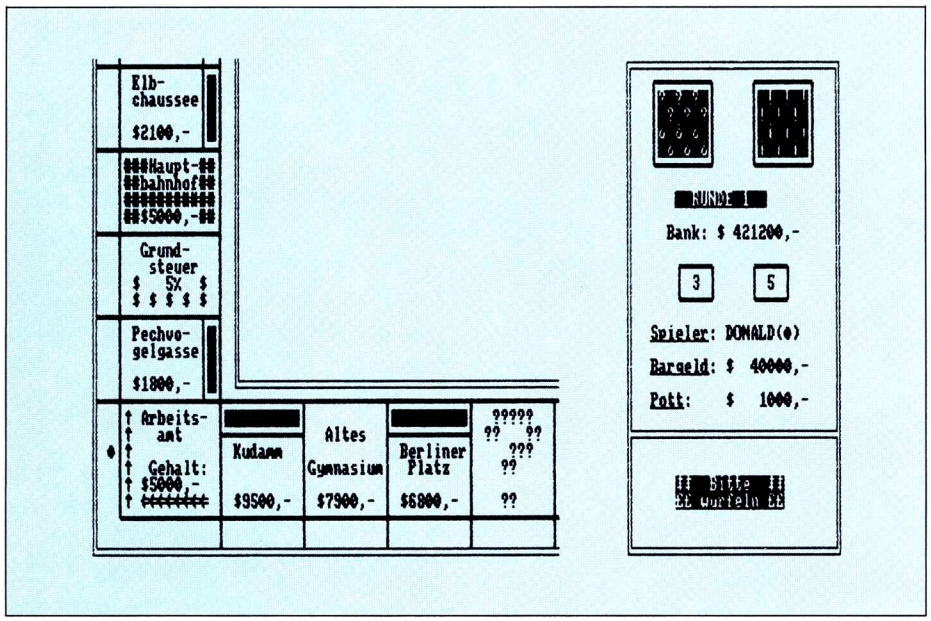


Abb. 3: Joycie hat gerade gezogen! Donald ist an der Reihe

kann bis zu 4 Häuser oder 1 Hotel pro Straße von der Bank kaufen (1 Hotel wird als 5 Häuser gerechnet). Es können auch Häuser dazugekauft werden, wenn noch kein Hotel auf der Straße steht. Ein Haus kostet 50 Prozent des Preises einer Straße. Auch kann der Spieler Häuser an die Bank verkaufen. Ist die Straße schon im Besitz eines anderen Mitspielers, muß der Spieler Miete zahlen: 2 hoch Anzahl der daraufstehenden Häuser mal zehn Prozent des Straßenpreises (d.h. mit jedem neuen Haus verdoppelt sich die Miete!). Die Maximalmiete ist $30400,– (Kudamm (3), 1 Hotel).

Nicht bebaubare Straßen:
2, 7, 9, 17, 21, 26, 28, 32, 35
Auf diesen Straßen können keine Häuser oder Hotels gebaut werden. Die Miete beträgt: gewürfelte Augenzahl mal fünf Prozent des Straßenpreises.

Die Maximalmiete beträgt $4740,– (Altes Gymnasium (2), 12 gewürfelt). Weiteres unter »Bebaubare Straßen«.

Verkaufen

Straßen und Häuser bzw. Hotels können auch wieder an die Bank verkauft werden, jedoch jeweils nur zum halben Preis. Straßen können nur im unbebauten Zustand verkauft werden.

Spielende

Das Spiel ist beendet, sobald die Bank oder einer der Spieler bankrott ist, d.h. dessen Gesamtvermögen (Barvermögen plus Wert der Häuser und Straßen) $<= 0$ ist.

Allgemeines

Es wird empfohlen, auch wirklich nur dann eine Taste zu drücken, wenn dies erforderlich ist! Der Computer speichert nämlich alle Tastendrückereien, und das kann zu unerfreulichen Ergebnissen führen!!

Wenn während des Würfelns EXIT gedrückt wird, so kann der Spielstand übersehen werden. Ein weiterer Tastendruck führt ins Spiel zurück.

(B. Breede)

```
LISTING >SUBURBIA<, REMARK = >REM<.

<17> 10 '$$$$$$$$$$$$$$$$$$$$$$ S U B U R B I A $$$$$$$
     $$$$$$$$$$$$$
<85> 20 '$$$ VERSION 1.01 (C) 6/1987 by B. Breede Softw
     are & Co. $$$$
<75> 30 '$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
     $$$$$$$$$$$$$
< 4> 40 '
<57> 50 CHAIN MERGE"FONT_SUB",60
< 4> 60 GOSUB 60000:DELETE 60000-,70
< 7> 70 '
<13> 80 '********** Codes **********
< 9> 90 '
<45> 100 e$=CHR$(27)
<20> 110 ston$=e$+"l"
<66> 120 stoff$=e$+"0"
<54> 130 con$=e$+"e"
<89> 140 coff$=e$+"f"
< 2> 150 cls$=e$+"E"+e$+"H"
<91> 160 fullscr$=e$+"X"+CHR$(32)+CHR$(32)+CHR$(63)+CHR
     $(122)
<78> 170 DIM einw$(5),stein$(5),bar(5),feld(5),stra$(35
     ),plan$(3,29),Ekarte$(9),Gkarte$(9)
<34> 180 '
<80> 190 '********** $$$ M A I N $$$ **********
<19> 200 '
<66> 210 GOSUB 4560
< 4> 220 GOSUB 890
<35> 230 GOSUB 1210
<49> 240 PRINT cls$:DELETE 890-1580,250
<74> 250 GOSUB 4560
<91> 260 GOSUB 2240
<62> 270 RANDOMIZE stw
<68> 280 runde=1:pott=1000:bank=bank-1000:PRINT cls$:GO
     SUB 1590
<47> 290 PRINT coff$
<69> 300 FOR spieler=0 TO einw
<44> 310 GOSUB 1840
<27> 320 GOSUB 4800
```

```
<76> 330 IF ges(spieler)<=0 OR gbank<=0 GOTO 360
<40> 340 NEXT spieler
<51> 350 runde=runde+1:GOTO 300
<32> 360 '
<29> 370 '********** ENDE **********
<36> 380 '
< 4> 390 PRINT cls$
<53> 400 gbank=bank:FOR I=0 TO einw+1:ges(I)=bar(I):hsr
     (I)=0:stra(I)=0:NEXT
<79> 410 IF sub GOTO 460
<28> 420 x%=17:y%=2:text$=FNf$(58," ▟ ▟ ▟.. E. N. D. E.
     . ▟ ▟ ▟ ")
<33> 430 GOSUB 4530
<11> 440 x%=14:y%=4:text$=FNf$(16,"Dieses Spiel dauerte
     "+STR$(runde)+" Runden.")
<37> 450 GOSUB 4530
<47> 460 PRINT FNloc$(0,6)"SPIELER:.....
     BAR:..... STRASSEN:. HAEUSER:..... GESA
     MTVERMOEGEN:"
<87> 470 PRINT FNloc$(0,7)STRING$(90,154)
<70> 480 x%=1:y%=11:text$=FNf$(16,"BANK")
<45> 490 GOSUB 4530
<87> 500 x%=34-LEN(STR$(bank))*2:text$=FNf$(16,STR$(ban
     k)+",-")
<30> 510 GOSUB 4530
<25> 520 FOR I=0 TO einw
<25> 530 x%=1:y%=13+I*2:text$=FNf$(16,einw$(I))
<36> 540 GOSUB 4530
<81> 550 x%=34-LEN(STR$(bar(I)))*2:text$=FNf$(16,STR$(b
     ar(I))+",-")
<40> 560 GOSUB 4530
<52> 570 NEXT
<41> 580 GOSUB 4800
<47> 590 FOR I=0 TO einw+1
<87> 600 x%=55-LEN(STR$(stra(I)))*2:y%=11+I*2:text$=FNf
     $(16,STR$(stra(I)))
<31> 610 GOSUB 4530
<24> 620 x%=65-LEN(STR$(hsr(I)))*2:text$=FNf$(16,STR$(h
     sr(I)))
```

```
<35> 630 GOSUB 4530
<47> 640 NEXT
<95> 650 x%=84-LEN(STR$(gbank))*2:y%=11:text$=FNf$(16,S
     TR$(gbank)+",-")
<41> 660 GOSUB 4530
<22> 670 gesamt=gbank+pott
<38> 680 FOR I=0 TO einw
<14> 690 x%=84-LEN(STR$(ges(I)))*2:y%=13+I*2:text$=FNf$
     (16,STR$(ges(I))+",-")
<30> 700 GOSUB 4530
<40> 710 gesamt=gesamt+ges(I)
<44> 720 NEXT
< 5> 730 PRINT FNloc$(0,23)STRING$(90,154)
<75> 740 x%=1:y%=26:text$=FNf$(16,"POTT:")
<40> 750 GOSUB 4530
<62> 760 x%=34-LEN(STR$(pott))*2:text$=FNf$(16,STR$(pot
     t)+",-")
<44> 770 GOSUB 4530
<92> 780 x%=45:text$=FNf$(16,"GESAMT")+FNf$(1,"(KONTROL
     LE)")+FNf$(16,":")
<48> 790 GOSUB 4530
< 7> 800 x%=84-LEN(STR$(gesamt))*2:text$=FNf$(16,STR$(g
     esamt)+",-")
<33> 810 GOSUB 4530
<59> 820 IF sub THEN a$=INPUT$(1):RETURN
<84> 830 x%=12:y%=30:text$=FNf$(48,"Herzlichen Dank für
      dieses Spiel!")
<39> 840 GOSUB 4530
<90> 850 a$=INPUT$(1)
<77> 860 PRINT cls$con$ston$:END
<39> 870 '
<32> 880 '$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
<43> 890 '
<84> 900 '********** Titel **********
<28> 910 '
<88> 920 PRINT cls$coff$stoff$
<29> 930 PRINT FNloc$(17,3)FNi$(CHR$(134)+STRING$(53,13
     8)+CHR$(140))
<84> 940 FOR I=4 TO 8:PRINT FNloc$(17,I)FNi$(CHR$(133)+
     STRING$(53,36)+CHR$(133)):NEXT
<25> 950 PRINT FNloc$(17,9)FNi$(CHR$(131)+STRING$(53,13
     8)+CHR$(137))
<95> 960 x%=33:y%=13:text$=FNf$(17,"VERSION 1.01")
<46> 970 GOSUB 4530
<55> 980 x%=10:y%=15:text$=FNf$(16,"ñ 6/1987 by B. Bree
     de Software & Co.")
<50> 990 GOSUB 4530
<13> 1000 x%=19:y%=20:text$=FNf$(48,"▟▟▟")+FNf$(32," Ei
     n Spiel für 5 Personen und 1 Computer ")+FNf$(48,"
     ▟▟▟")
<79> 1010 GOSUB 4530
<39> 1020 x%=12:y%=24:text$=FNf$(48,"▜▜▜")+FNf$(32," In
      Anlehnung an ein weltbekanntes ½ ")+FNf$(48,"Makl
     erspiel")+FNf$(32," ¼ ")+FNf$(48,"▜▜▜")
<85> 1030 GOSUB 4530
<57> 1040 x%=10:y%=29:text$=FNf$(18,STRING$(35,36))
<91> 1050 GOSUB 4530
<34> 1060 RESTORE 1130
<29> 1070 FOR I=5 TO 7
<34> 1080 PRINT FNloc$(20,I);
<81> 1090 FOR J=1 TO 49
<28> 1100 READ a:PRINT CHR$(a);
<33> 1110 NEXT J
<25> 1120 NEXT I
<73> 1130 DATA 32, 32,150,154,154,156, 32, 32,148, 32,
     32,148, 32, 32,158,154,154,156, 32, 32, 32,148, 32
     , 32,148, 32, 32,158,154,154,156, 32, 32,158,154,1
     54,156, 32, 32, 32,158, 32, 32,150,154,154,156, 32
     , 32
<18> 1140 DATA 32, 32,147,154,154,156, 32, 32,149, 32,
     32,149, 32, 32,151,154,154,155,156, 32, 32,149, 32
     , 32,149, 32, 32,151,154,158,153, 32, 32,151,154,1
     54,155,156, 32, 32,149, 32, 32,151,154,154,157, 32
     , 32
<48> 1150 DATA 32, 32,147,154,154,153, 32, 32,147,154,1
     54,153, 32, 32,155,154,154,154,153, 32, 32,147,154
     ,154,153, 32, 32,155, 32,147,152, 32, 32,155,154,1
     54,154,153, 32, 32,155, 32, 32,145, 32, 32,145, 32
     , 32
< 1> 1160 FOR stw=1 TO 8000
<17> 1170 IF INKEY$<>"" GOTO 1190
< 9> 1180 NEXT
< 3> 1190 PRINT con$
<82> 1200 RETURN
<90> 1210 '
<83> 1220 '********** Starteingaben **********
<96> 1230 '
<43> 1240 PRINT cls$:bank=500000!
<80> 1250 x%=3:y%=2:text$=FNf$(48,"W i l l k o m m e n.
     . i n.. ")
<98> 1260 GOSUB 4530
<57> 1270 x%=59:text$=FNf$(50,"S U B U R B I A")
< 5> 1280 GOSUB 4530
<31> 1290 PRINT:PRINT:PRINT STRING$(90,138)
<38> 1300 PRINT:PRINT FNu$("Bitte Geben Sie die Einwohn
     er ein:")
<78> 1310 PRINT:PRINT:PRINT:PRINT TAB(20)"Einwohner 0 :
     .. JOYCIE":einw$(0)="JOYCIE"
<37> 1320 FOR I=1 TO 5
<12> 1330 PRINT:PRINT TAB(20)"Einwohner"I":.. ";:INPUT"
     ",einw$(I)
< 4> 1340 IF I=1 AND einw$(I)="" GOTO 1240
<58> 1350 IF einw$(I)="" THEN einw=I-1:GOTO 1380
< 7> 1360 NEXT
<16> 1370 einw=I-1
<75> 1380 RESTORE 1420
<94> 1390 FOR I=0 TO einw
<51> 1400 READ a:stein$(I)=CHR$(a):einw$(I)=UPPER$(einw
     $(I)):bar(I)=40000!:feld(I)=4:bank=bank-40000!
<93> 1410 NEXT
<94> 1420 DATA 187,188,42,35,36,163
< 4> 1430 PRINT FNloc$(0,7)CHR$(27)+"J"
<81> 1440 FOR I=0 TO einw
<52> 1450 PRINT:PRINT TAB(20)"Einwohner"I":.. "
<12> 1460 x%=38:y%=(I+1)*2+8:text$=FNf$(16,"("+stein$(I
     )+"). "+einw$(I))
< 6> 1470 GOSUB 4530
<15> 1480 NEXT
<65> 1490 x%=11:y%=27:text$=FNf$(17,"Sind diese Eingabe
     n korrekt (J/N)?")
<52> 1500 PRINT coff$
<89> 1510 GOSUB 4530
<14> 1520 a$="":WHILE a$<>"N" AND a$<>"J"
<79> 1530 a$=UPPER$(INKEY$)
<52> 1540 stw=stw+1:IF stw>9999 THEN stw=0
<36> 1550 WEND
< 2> 1560 PRINT con$
<16> 1570 IF a$="N" GOTO 1240
<13> 1580 RETURN
<21> 1590 '
< 6> 1600 '********** Spielfeld malen **********
<98> 1610 '
<37> 1620 PRINT FNwdw$(64,0,90,32)
<97> 1630 PRINT CHR$(134)STRING$(24,138)CHR$(140);
<56> 1640 FOR I=1 TO 21:PRINT CHR$(133)SPC(24)CHR$(133)
     ;:NEXT
<66> 1650 PRINT CHR$(135)STRING$(24,138)CHR$(141);
< 3> 1660 FOR I=1 TO 6:PRINT CHR$(133)SPC(24)CHR$(133);
     :NEXT
<14> 1670 PRINT CHR$(131)STRING$(24,138)CHR$(137);
<90> 1680 PRINT FNwdw$(67,1,87,32)
<23> 1690 PRINT CHR$(150)STRING$(6,154)CHR$(156)SPC(4)C
     HR$(150)STRING$(6,154)CHR$(156)
<17> 1700 PRINT CHR$(149)FNi$("? ? ? ")CHR$(149)SPC(4)C
     HR$(149)FNi$(" ! ! !")CHR$(149)
<11> 1710 PRINT CHR$(149)FNi$(" ? ? ?")CHR$(149)SPC(4)C
     HR$(149)FNi$("! ! ! ")CHR$(149)
<14> 1720 PRINT CHR$(149)FNi$("« « « ")CHR$(149)SPC(4)C
     HR$(149)FNi$(" » » »")CHR$(149)
<14> 1730 PRINT CHR$(149)FNi$(" « « «")CHR$(149)SPC(4)C
     HR$(149)FNi$("» » » ")CHR$(149)
< 3> 1740 PRINT CHR$(147)STRING$(6,154)CHR$(153)SPC(4)C
     HR$(147)STRING$(6,154)CHR$(153)
<84> 1750 PRINT:PRINT:PRINT:PRINT:PRINT
< 7> 1760 PRINT".. "CHR$(150)STRING$(3,154)CHR$(156)"..
     . "CHR$(150)STRING$(3,154)CHR$(156)
<24> 1770 PRINT".. "CHR$(149)" 0 "CHR$(149)"... "CHR$(1
     49)" 0 "CHR$(149)
<30> 1780 PRINT".. "CHR$(147)STRING$(3,154)CHR$(153)"..
     . "CHR$(147)STRING$(3,154)CHR$(153)
<26> 1790 PRINT:PRINT FNu$("Spieler")":"
<83> 1800 PRINT:PRINT FNu$("Bargeld")":"
<27> 1810 PRINT:PRINT FNu$("Pott")":"
<95> 1820 PRINT fullscr$
< 4> 1830 RETURN
<12> 1840 '
<41> 1850 '********** Spielzug **********
<18> 1860 '
<47> 1870 GOSUB 4120
<15> 1880 GOSUB 4710
<21> 1890 GOSUB 1990
< 0> 1900 IF feld(spieler)=13 AND NOT(w1=w2 OR w1=6 OR
     w2=6) THEN FOR I=1 TO 3000:NEXT:RETURN
< 3> 1910 IF w1=w2 THEN nochmal=-1 ELSE nochmal=0
< 3> 1920 a=feld(spieler):feld(spieler)=feld(spieler)+w
     1+w2
<19> 1930 IF feld(spieler)>35 THEN feld(spieler)=feld(s
     pieler)-36
<50> 1940 IF (a>29 OR a<4) AND feld(spieler)>4 THEN bar
     (spieler)=bar(spieler)+5000:bank=bank-5000:GOSUB 4
     710
<43> 1950 GOSUB 4120
<49> 1960 GOSUB 3040
<14> 1970 GOSUB 4710
<16> 1980 IF nochmal GOTO 1840 ELSE FOR I=1 TO 3000:NEX
     T:RETURN
<29> 1990 '
< 5> 2000 '********** Würfel **********
<87> 2010 '
<47> 2020 IF spieler=0 GOTO 2140
<61> 2030 PRINT FNloc$(70,26)FNi$("$$. Bitte. $$")
<35> 2040 PRINT FNloc$(70,27)FNi$("úú würfeln úú")
<86> 2050 GOSUB 2200
<56> 2060 a$="":WHILE a$="":a$=INKEY$
<92> 2070 GOSUB 2200
<36> 2080 WEND
<29> 2090 sub=(ASC(a$)=27)
<79> 2100 IF sub THEN GOSUB 360:PRINT cls$:GOSUB 1590:G
     OSUB 4710:GOSUB 4120:GOTO 1990
<17> 2110 PRINT FNloc$(70,26)SPC(14)
<43> 2120 PRINT FNloc$(70,27)SPC(14)
<90> 2130 RETURN
<66> 2140 FOR K=1 TO 10+INT(RND*20)
<51> 2150 sub=(INKEY$=CHR$(27))
<97> 2160 IF sub THEN GOSUB 360:PRINT cls$:GOSUB 1590:G
     OSUB 4710:GOSUB 4120:GOTO 1990
<94> 2170 GOSUB 2200
<10> 2180 NEXT
< 9> 2190 RETURN
<51> 2200 w1=INT(RND*6)+1:w2=INT(RND*6)+1
<12> 2210 PRINT FNloc$(71,14)w1
```

```
<40> 2220 PRINT FNloc$(80,14)w2
<92> 2230 RETURN
< 1> 2240 '
< 7> 2250 '********** Stadtplan init **********
< 7> 2260 '
<76> 2270 PRINT cls$coff$
<78> 2280 x%=12:y%=4:text$=FNf$(52,"Der Stadtplan wird
     initialisiert:")
< 9> 2290 GOSUB 4530
<61> 2300 x%=3:y%=7:text$=FNf$(16,"In 4 Teilen: Süden -
      Westen - Norden - Osten")
<86> 2310 GOSUB 4530
< 9> 2320 x%=13:y%=10:text$=FNf$(17,"(ca. 45 Sekunden -
      bitte Geduld)")
<92> 2330 GOSUB 4530
<61> 2340 OPEN "I",#1,"KARTE.SB"
<11> 2350 FOR J=0 TO 3
<60> 2360 PRINT FNloc$(28,(J+1)*3+14)USING("#.: ");J+1;
<26> 2370 FOR I=0 TO 29
<85> 2380 PRINT"$";
<32> 2390 plan$(J,I)=""
<13> 2400 IF J=2 AND I>10 THEN plan$(J,I)=SPACE$(38)+CH
     R$(133)+" "
<72> 2410 IF J=3 AND I<19 THEN plan$(J,I)=SPACE$(38)+CH
     R$(133)+" "
<35> 2420 INPUT #1,a$:IF a$="0" THEN 2450
<64> 2430 IF VAL(a$)>255 THEN plan$(J,I)=plan$(J,I)+STR
     ING$(VAL(LEFT$(a$,1)),VAL(MID$(a$,2))) ELSE IF VAL
     (a$)=0 THEN plan$(J,I)=plan$(J,I)+a$ ELSE plan$(J,
     I)=plan$(J,I)+CHR$(VAL(a$))
<98> 2440 GOTO 2420
<64> 2450 IF J=0 AND I<19 THEN plan$(J,I)=plan$(J,I)+SP
     ACE$(38)
<33> 2460 IF J=1 AND I>10 THEN plan$(J,I)=plan$(J,I)+SP
     ACE$(38)
<47> 2470 NEXT I
<92> 2480 PRINT
<64> 2490 NEXT J
<51> 2500 FOR I=0 TO 35:INPUT #1,a$:stra$(I)="90"+a$:NE
     XT
< 6> 2510 FOR I=0 TO 9:INPUT #1,a$:Ekarte$(I)=a$:NEXT
<71> 2520 FOR I=0 TO 9:INPUT #1,a$:Gkarte$(I)=a$:NEXT
<24> 2530 CLOSE #1
<96> 2540 PRINT con$
< 5> 2550 RETURN
<13> 2560 '
<80> 2570 '********** Häuser zeichnen **********
<19> 2580 '
<43> 2590 ON (INT(feld(spieler)/9)+1) GOSUB 2610,2660,2
     710,2770
<91> 2600 RETURN
<75> 2610 a=FNhsr#(8):IF a THEN IF a<5 THEN FOR I=1 TO
     a:PRINT FNhaus$(14,1+I):NEXT ELSE PRINT FNhotel$(1
     4,4)
<27> 2620 a=FNhsr#(5):IF a THEN IF a<5 THEN FOR I=1 TO
     a:PRINT FNhaus$(14,16+I):NEXT ELSE PRINT FNhotel$(
     14,19)
<39> 2630 a=FNhsr#(3):IF a THEN IF a<5 THEN FOR I=1 TO
     a:PRINT FNhaus$(15+I*2,22):NEXT ELSE PRINT FNhotel
     $(20,22)
<55> 2640 a=FNhsr#(1):IF a THEN IF a<5 THEN FOR I=1 TO
     a:PRINT FNhaus$(35+I*2,22):NEXT ELSE PRINT FNhotel
     $(40,22)
< 7> 2650 RETURN
<51> 2660 a=FNhsr#(14):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(15+I*2,9):NEXT ELSE PRINT FNhotel
     $(20,9)
<43> 2670 a=FNhsr#(15):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(25+I*2,9):NEXT ELSE PRINT FNhotel
     $(30,9)
< 7> 2680 a=FNhsr#(12):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(14,10+I):NEXT ELSE PRINT FNhotel$
     (14,13)
<92> 2690 a=FNhsr#(11):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(14,15+I):NEXT ELSE PRINT FNhotel$
     (14,18)
<93> 2700 RETURN
<56> 2710 a=FNhsr#(18):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(I*2,9):NEXT ELSE PRINT FNhotel$(5
     ,9)
<94> 2720 a=FNhsr#(19):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(10+I*2,9):NEXT ELSE PRINT FNhotel
     $(15,9)
<40> 2730 a=FNhsr#(20):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(20+I*2,9):NEXT ELSE PRINT FNhotel
     $(25,9)
<22> 2740 a=FNhsr#(23):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(41,10+I):NEXT ELSE PRINT FNhotel$
     (41,13)
<40> 2750 a=FNhsr#(24):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(41,15+I):NEXT ELSE PRINT FNhotel$
     (41,18)
<12> 2760 RETURN
<71> 2770 a=FNhsr#(27):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(41,1+I):NEXT ELSE PRINT FNhotel$(
     41,4)
<79> 2780 a=FNhsr#(30):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(41,16+I):NEXT ELSE PRINT FNhotel$
     (41,19)
<51> 2790 a=FNhsr#(34):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(10+I*2,22):NEXT ELSE PRINT FNhote
     l$(15,22)
<83> 2800 a=FNhsr#(33):IF a THEN IF a<5 THEN FOR I=1 TO
      a:PRINT FNhaus$(20+I*2,22):NEXT ELSE PRINT FNhote
     l$(25,22)
<98> 2810 RETURN
< 7> 2820 '
<35> 2830 '********** Spielsteine malen **********
<13> 2840 '
<93> 2850 FOR I=0 TO einw
< 8> 2860 IF feld(I)>=INT(feld(spieler)/9)*9 AND feld(I
     )<=INT(feld(spieler)/9)*9+8 THEN GOSUB 2880
<62> 2870 NEXT:RETURN
<47> 2880 ON (INT(feld(spieler)/9)+1) GOSUB 2900,2930,2
     960,3000
<23> 2890 RETURN
<42> 2900 IF feld(I)>3 THEN PRINT FNloc$((I MOD 2)+1,(2
     5-(feld(I)-4)*5)-(I MOD 3))stein$(I)
<81> 2910 IF feld(I)<4 THEN PRINT FNloc$((48-feld(I)*10
     )+I,29)stein$(I)
< 4> 2920 RETURN
<51> 2930 IF feld(I)<14 THEN PRINT FNloc$((I MOD 2)+1,(
     29-(feld(I)-9)*5)-(I MOD 3))stein$(I)
<53> 2940 IF feld(I)>13 THEN PRINT FNloc$(17+(feld(I)-1
     4)*10+I,2)stein$(I)
<13> 2950 RETURN
<98> 2960 IF feld(I)<22 THEN PRINT FNloc$((feld(I)-18)*
     10+I+2,2)stein$(I)
<56> 2970 IF feld(I)=22 THEN PRINT FNloc$(45+I,5)stein$
     (I)
< 9> 2980 IF feld(I)>22 THEN PRINT FNloc$(53+(I MOD 2),
     12+(feld(I)-23)*5+(I MOD 3))stein$(I)
<25> 2990 RETURN
<25> 3000 IF feld(I)<31 THEN PRINT FNloc$(53+(I MOD 2),
     (3+(feld(I)-27)*5)+(I MOD 3))stein$(I)
< 8> 3010 IF feld(I)=31 THEN PRINT FNloc$(53+(I MOD 2),
     27+(I MOD 3))stein$(I)
<27> 3020 IF feld(I)>31 THEN PRINT FNloc$(32-(feld(I)-3
     2)*10+I,29)stein$(I)
<89> 3030 RETURN
<97> 3040 '
<89> 3050 '********** Felder: **********
< 4> 3060 '
<47> 3070 IF feld(spieler)=13 THEN RETURN
<34> 3080 IF feld(spieler)=6 AND bar(spieler)>0 THEN a=
     ROUND(bar(spieler)/20):bar(spieler)=bar(spieler)-a
     :pott=pott+a:RETURN
<19> 3090 IF feld(spieler)=25 AND bar(spieler)>0 THEN a
     =ROUND(bar(spieler)/10):bar(spieler)=bar(spieler)-
     a:pott=pott+a:RETURN
<96> 3100 IF feld(spieler)=4 THEN bar(spieler)=bar(spie
     ler)+10000:bank=bank-10000:RETURN
<80> 3110 IF feld(spieler)=31 THEN feld(spieler)=13:RET
     URN
<58> 3120 IF feld(spieler)=22 THEN bar(spieler)=bar(spi
     eler)+pott:pott=1000:bank=bank-1000:RETURN
< 8> 3130 IF feld(spieler)=0 OR feld(spieler)=16 THEN G
     OSUB 3330:RETURN
<23> 3140 IF feld(spieler)=10 OR feld(spieler)=29 THEN
     GOSUB 3520:RETURN
<96> 3150 IF FNbes#(feld(spieler))=9 THEN GOSUB 3220:RE
     TURN
<24> 3160 DATA 2,7,9,17,21,26,28,32,35
<26> 3170 IF FNbes#(feld(spieler))<>spieler THEN GOSUB
     4200:RETURN
<18> 3180 RESTORE 3160:FOR I=1 TO 9:READ a:IF feld(spie
     ler)=a GOTO 3200 ELSE NEXT
<73> 3190 IF FNhsr#(feld(spieler))<5 THEN GOSUB 3710 EL
     SE GOSUB 3890
<55> 3200 IF FNhsr#(feld(spieler))=0 THEN GOSUB 4370
<87> 3210 RETURN
<95> 3220 '
<49> 3230 '********** Straße kaufen **********
< 2> 3240 '
<57> 3250 IF spieler=0 GOTO 3310
<54> 3260 PRINT fenster$"Möchtest Du diese Straße.. kau
     fen (J/N)? ";
<52> 3270 a$=UPPER$(INPUT$(1)):IF a$<>"J" AND a$<>"N" G
     OTO 3270
<66> 3280 IF a$="N" THEN PRINT cls$fullscr$:RETURN
<88> 3290 MID$(stra$(feld(spieler)),1,1)=RIGHT$(STR$(sp
     ieler),1):bar(spieler)=bar(spieler)-FNprs#(feld(sp
     ieler)):bank=bank+FNprs#(feld(spieler))
<49> 3300 PRINT cls$fullscr$:RETURN
<48> 3310 IF FNprs#(feld(0))<bar(0)/2.5 THEN PRINT fens
     ter$" Ich kaufe diese Straße.":MID$(stra$(feld(spi
     eler)),1,1)="0":bar(0)=bar(0)-FNprs#(feld(0)):bank
     =bank+FNprs#(feld(0)):FOR I=1 TO 5000:NEXT:PRINT c
     ls$fullscr$
<92> 3320 RETURN
< 1> 3330 '
<96> 3340 '********** ???-Karten **********
< 7> 3350 '
<66> 3360 PRINT fenster$;
<65> 3370 a=INT(RND*10)
<77> 3380 PRINT Ekarte$(a)fullscr$
<85> 3390 IF spieler=0 THEN FOR I=1 TO 5000:NEXT ELSE P
     RINT FNloc$(73,29)FNi$("<RETURN>"):a$=INPUT$(1)
<84> 3400 PRINT fenster$cls$fullscr$
< 2> 3410 IF a=0 THEN IF feld(spieler)=0 THEN feld(spie
     ler)=13 ELSE feld(spieler)=11:GOSUB 4120:GOSUB 304
     0:RETURN
< 9> 3420 IF a=1 THEN feld(spieler)=2:GOSUB 4120:GOSUB
     3040:RETURN
<63> 3430 IF a=2 THEN bar(spieler)=bar(spieler)+5000:ba
     nk=bank-5000:RETURN
< 6> 3440 IF a=3 THEN bar(spieler)=bar(spieler)+5000:ba
     nk=bank-5000:feld(spieler)=15:GOSUB 4120:GOSUB 304
     0:RETURN
<48> 3450 IF a=4 THEN feld(spieler)=5:GOSUB 4120:GOSUB
```

```
     3040:RETURN
<40> 3460 IF a=5 THEN bar(spieler)=bar(spieler)-1000:po
     tt=pott+1000:RETURN
<76> 3470 IF a=6 THEN IF feld(spieler)=0 THEN bar(spiel
     er)=bar(spieler)+5000:bank=bank-5000
<52> 3480 IF a=6 THEN feld(spieler)=28:GOSUB 4120:GOSUB
      3040:RETURN
<73> 3490 IF a=7 THEN bar(spieler)=bar(spieler)-200:pot
     t=pott+200:RETURN
<47> 3500 IF a=8 THEN feld(spieler)=4:GOSUB 4120:GOSUB
     3040:RETURN
< 6> 3510 bar(spieler)=bar(spieler)-(einw+1)*1500:FOR I
     =0 TO einw:bar(I)=bar(I)+1500:NEXT:RETURN
< 2> 3520 '
<70> 3530 '********** !!!-Karten **********
< 8> 3540 '
<67> 3550 PRINT fenster$;
<66> 3560 a=INT(RND*10)
<54> 3570 IF spieler=0 AND a=7 THEN a=8
< 6> 3580 PRINT Gkarte$(a)fullscr$
<89> 3590 IF spieler=0 THEN FOR I=1 TO 5000:NEXT ELSE P
     RINT FNloc$(73,29)FNi$("<RETURN>"):a$=INPUT$(1)
<88> 3600 PRINT fenster$cls$fullscr$
<35> 3610 IF a=0 THEN bar(spieler)=bar(spieler)+4000:ba
     nk=bank-4000:RETURN
<43> 3620 IF a=1 THEN FOR I=0 TO einw:bar(I)=bar(I)-10:
     NEXT:bar(spieler)=bar(spieler)+10*(einw+1):RETURN
<22> 3630 IF a=2 THEN bar(spieler)=bar(spieler)-800:pot
     t=pott+800:RETURN
<24> 3640 IF a=3 THEN bar(spieler)=bar(spieler)-3000:po
     tt=pott+3000:RETURN
<43> 3650 IF a=4 THEN bar(spieler)=bar(spieler)+20:bank
     =bank-20:RETURN
<24> 3660 IF a=5 THEN bar(spieler)=bar(spieler)+850:ban
     k=bank-850:RETURN
<49> 3670 IF a=6 THEN feld(spieler)=feld(spieler)+1:GOS
     UB 4120:GOSUB 3040:RETURN
<12> 3680 IF a=7 THEN bar(spieler)=bar(spieler)-500:bar
     (0)=bar(0)+500:RETURN
<94> 3690 IF a=8 THEN bar(spieler)=bar(spieler)-400:pot
     t=pott+400:RETURN
<61> 3700 feld{spieler)=13:GOSUB 4120:GOSUB 3040:RETURN
< 3> 3710 '
<79> 3720 '********** Häuser kaufen **********
< 9> 3730 '
<43> 3740 IF spieler=0 GOTO 3830
<78> 3750 PRINT fenster$" Möchtest Du auf dieser. Straß
     e bauen (J/N)? ";
<84> 3760 a$=UPPER$(INPUT$(1)):IF a$<>"J" AND a$<>"N" G
     OTO 3760
<73> 3770 IF a$="N" THEN PRINT cls$fullscr$:RETURN
<27> 3780 PRINT cls$" Ein Haus kostet "USING("$####_,-"
     );FNprs#(feld(spieler))/2:INPUT" Wieviele Häuser m
     öch-... test Du kaufen? ",a
<96> 3790 IF FNhsr#(feld(spieler))+a>5 THEN PRINT CHR$(
     7):GOTO 3780
<56> 3800 MID$(stra$(feld(spieler)),2,1)=RIGHT$(STR$(FN
     hsr#(feld(spieler))+a),1)
<97> 3810 bar(spieler)=bar(spieler)-a*FNprs#(feld(spiel
     er))/2:bank=bank+a*FNprs#(feld(spieler))/2
<65> 3820 PRINT cls$fullscr$:RETURN
<85> 3830 IF FNprs#(feld(0))/2>bar(0)/2.5 THEN FOR I=1
     TO 3000:NEXT:RETURN
<86> 3840 PRINT fenster$" Ich baue hier.":FOR I=1 TO 10
     00:NEXT
<87> 3850 MID$(stra$(feld(0)),2,1)=RIGHT$(STR$(VAL(MID$
     (stra$(feld(0)),2,1))+1),1)
<54> 3860 bar(0)=bar(0)-FNprs#(feld(0))/2:bank=bank+FNp
     rs#(feld(0))/2
<31> 3870 PRINT cls$fullscr$
<80> 3880 IF FNhsr#(feld(0))<5 GOTO 3830 ELSE FOR I=1 T
     O 3000:NEXT:RETURN
<29> 3890 '
<84> 3900 '********* Häuser verkaufen **********
< 7> 3910 '
<16> 3920 IF spieler=0 GOTO 4050
<72> 3930 PRINT fenster$". Möchtest Du Häuser..... verk
     aufen (J/N)? ";
<80> 3940 a$=UPPER$(INPUT$(1)):IF a$<>"J" AND a$<>"N" G
     OTO 3940
<71> 3950 IF a$="N" THEN PRINT cls$fullscr$:RETURN
<34> 3960 PRINT cls$:INPUT" Wieviele? ",a
<76> 3970 IF a>FNhsr#(feld(spieler)) THEN PRINT CHR$(7)
     :GOTO 3960
<60> 3980 PRINT cls$" Der Verkauf von"a
<10> 3990 PRINT" Häusern bringt Dir"
<71> 4000 PRINT USING("... $#####_,-");a*FNprs#(feld(sp
     ieler))/4
<48> 4010 MID$(stra$(feld(spieler)),2,1)=RIGHT$(STR$(5-
     a),1)
<89> 4020 bar(spieler)=bar(spieler)+a*FNprs#(feld(spiel
     er))/4
< 9> 4030 bank=bank-a*FNprs#(feld(spieler))/4
<62> 4040 FOR I=1 TO 3000:NEXT:PRINT cls$fullscr$:RETUR
     N
<53> 4050 IF bar(0)>1000 OR FNhsr#(feld(0))=0 THEN RETU
     RN
< 2> 4060 PRINT fenster$" Ich verkaufe Häuser."
<24> 4070 MID$(stra$(feld(0)),2,1)=RIGHT$(STR$(VAL(MID$
     (stra$(feld(0)),2,1))-1),1)
< 4> 4080 bar(0)=bar(0)+FNprs#(feld(0))/4:bank=bank-FNp
     rs#(feld(0))/4
< 6> 4090 FOR I=1 TO 3000:NEXT
<96> 4100 PRINT cls$fullscr$
<98> 4110 GOTO 4050
<94> 4120 '
<38> 4130 '********** Plan malen **********
< 1> 4140 '
<93> 4150 PRINT fullscr$
<40> 4160 J=INT(feld(spieler)/9)
<16> 4170 FOR I=0 TO 29:PRINT plan$(J,I):NEXT
<80> 4180 GOSUB 2560:GOSUB 2820
<11> 4190 RETURN
<90> 4200 '
<19> 4210 '********** Miete zahlen **********
<96> 4220 '
< 9> 4230 RESTORE 3160
<13> 4240 FOR I=1 TO 9
<42> 4250 READ a:IF feld(spieler)=a THEN b=(w1+w2)*FNpr
     s#(a)/20:bar(spieler)=bar(spieler)-b:bar(FNbes#(a)
     )=bar(FNbes#(a))+b:GOSUB 4320:RETURN
< 8> 4260 NEXT
<16> 4270 b=FNprs#(feld(spieler))/10*(2^FNhsr#(feld(spi
     eler)))
<60> 4280 bar(spieler)=bar(spieler)-b
<53> 4290 bar(FNbes#(feld(spieler)))=bar(FNbes#(feld(sp
     ieler)))+b
<45> 4300 GOSUB 4320
<90> 4310 RETURN
<89> 4320 PRINT fenster$" Du mußt an "LEFT$(einw$(FNbes
     #(feld(spieler))),11)
<72> 4330 PRINT USING(" $#####_,-");b;:PRINT" Miete zah
     len."
<55> 4340 PRINT fullscr$:IF spieler THEN PRINT FNloc$(7
     3,29)FNi$("<RETURN>")
< 1> 4350 IF spieler=0 THEN FOR I=1 TO 5000:NEXT ELSE a
     $=INPUT$(1)
<98> 4360 PRINT fenster$cls$fullscr$:RETURN
<14> 4370 '
< 0> 4380 '********** Straße verkaufen **********
<20> 4390 '
<54> 4400 IF spieler=0 GOTO 4470
< 2> 4410 PRINT fenster$"Möchtest Du diese Straße verka
     ufen (J/N)? ";
<23> 4420 a$=UPPER$(INPUT$(1)):IF a$<>"J" AND a$<>"N" G
     OTO 4420
<56> 4430 IF a$="N" THEN PRINT cls$fullscr$:RETURN
<47> 4440 PRINT cls$" Der Verkauf dieser Str. bringt Di
     r"USING(" $####_,-");FNprs#(feld(spieler))/2
<74> 4450 GOSUB 4490
<51> 4460 PRINT fullscr$FNloc$(73,29)FNi$("<RETURN>"):a
     $=INPUT$(1):PRINT fenster$cls$fullscr$:RETURN
< 2> 4470 IF bar(0)>1500 THEN RETURN
<97> 4480 PRINT fenster$".. Ich verkaufe diese.... Stra
     ße.":GOSUB 4490:FOR I=1 TO 5000:NEXT:PRINT cls$ful
     lscr$:RETURN
<12> 4490 MID$(stra$(feld(spieler)),1,1)="9"
< 1> 4500 bank=bank-FNprs#(feld(spieler))/2
< 9> 4510 bar(spieler)=bar(spieler)+FNprs#(feld(spieler
     ))/2
<97> 4520 RETURN
< 6> 4530 '********************
<40> 4540 CALL font(x%,y%,text$)
< 7> 4550 RETURN
<15> 4560 '
<21> 4570 '********** FN's **********
<21> 4580 '
<66> 4590 DEF FNi$(x$)=e$+"p"+x$+e$+"q"
<66> 4600 DEF FNu$(x$)=e$+"r"+x$+e$+"u"
<67> 4610 DEF FNf$(x,x$)=e$+CHR$(x)+x$
<46> 4620 DEF FNloc$(x,y)=e$+"Y"+CHR$(32+y)+CHR$(32+x)
<71> 4630 DEF FNwdw$(x,y,a,b)=e$+"X"+CHR$(32+y)+CHR$(32
     +x)+CHR$(32+b-y)+CHR$(32+a-x)
<15> 4640 DEF FNprs#(x)=VAL(RIGHT$(stra$(x),4))
<95> 4650 DEF FNbes#(x)=VAL(LEFT$(stra$(x),1))
< 1> 4660 DEF FNhsr#(x)=VAL(MID$(stra$(x),2,1))
<98> 4670 DEF FNhaus$(x,y)=FNloc$(x,y)+FNi$(e$+CHR$(3))
<88> 4680 DEF FNhotel$(x,y)=FNloc$(x,y)+FNi$(e$+CHR$(8)
     )
<68> 4690 fenster$=FNwdw$(65,24,88,29)
<95> 4700 RETURN
< 4> 4710 '
<57> 4720 '********** Anzeige **********
<10> 4730 '
<54> 4740 PRINT FNloc$(70,9)FNi$(". RUNDE"+STR$(runde)+
     ". ")
<24> 4750 PRINT FNloc$(69,11)"Bank:"USING(" $#######_,-
     ");bank
< 5> 4760 PRINT FNloc$(76,17)SPC(13):PRINT FNloc$(76,17
     )LEFT$(einw$(spieler),10)"("stein$(spieler)")"
<65> 4770 PRINT FNloc$(75,19)SPC(14):PRINT FNloc$(75,19
     )USING(" $#######_,-");bar(spieler)
<69> 4780 PRINT FNloc$(75,21)SPC(14):PRINT FNloc$(75,21
     )USING(" $#######_,-");pott
<23> 4790 RETURN
< 3> 4800 '
<12> 4810 '********** Gesamtvermögen berechnen ********
     **
< 9> 4820 '
<33> 4830 gbank=bank:FOR I=0 TO einw+1:ges(I)=bar(I):hs
     r(I)=0:stra(I)=0:NEXT:hsr(0)=85
<85> 4840 FOR I=1 TO 35
<27> 4850 IF I=4 OR I=6 OR I=10 OR I=13 OR I=16 OR I=22
      OR I=25 OR I=29 OR I=31 GOTO 4930
<84> 4860 a=NOT(I=2 OR I=7 OR I=9 OR I=17 OR I=21 OR I=
     26 OR I=28 OR I=32 OR I=35)
<32> 4870 IF FNbes#(I)<>9 GOTO 4910
<85> 4880 gbank=gbank+FNprs#(I):stra(0)=stra(0)+1
<78> 4890 IF a THEN gbank=gbank+5*FNprs#(I)/2
<94> 4900 GOTO 4930
```

```
< 9> 4910 IF a THEN gbank=gbank+(5-FNhsr#(I))*FNprs#(I)
     /2
<57> 4920 ges(FNbes#(I))=ges(FNbes#(I))+FNprs#(I)+(FNhs
     r#(I)*FNprs#(I)/2):stra(FNbes#(I)+1)=stra(FNbes#(I
     )+1)+1:hsr(FNbes#(I)+1)=hsr(FNbes#(I)+1)+FNhsr#(I)
     :hsr(0)=hsr(0)-FNhsr#(I)
<13> 4930 NEXT
<12> 4940 RETURN
```

```
LISTING >FONT_SUB<, REMARK = >REM<.

< 3> 60000 '
<17> 60010 '********** Schriftarten-Subroutine ********
     **
<11> 60020 '
<30> 60030 font=&HF000
<15> 60040 MEMORY &HEFFF
<47> 60050 RESTORE 60100
< 0> 60060 FOR adr=&HF000 TO &HF1C9
<94> 60070 READ byte$:v=VAL("&H"+byte$):POKE adr,v
<17> 60080 NEXT
<79> 60090 RETURN
< 4> 60100 DATA 6E,1A,67,2D,25,22,BD,F1
<45> 60110 DATA 69,60,7E,B7,C8,4F,06,00
<53> 60120 DATA 32,BC,F1,23,7E,23,66,6F
<66> 60130 DATA 11,E9,F1,ED,B0,01,26,F0
<28> 60140 DATA CD,5A,FC,E9,00,C9,3A,BC
< 4> 60150 DATA F1,47,21,E9,F1,7E,FE,1B
<73> 60160 DATA 20,0B,23,05,C8,7E,32,BB
<81> 60170 DATA F1,23,05,C8,7E,C5,E5,6F
< 3> 60180 DATA 26,00,29,29,29,11,00,B8
<53> 60190 DATA 19,11,C9,F1,01,08,00,ED
<13> 60200 DATA B0,21,01,01,22,BF,F1,3A
<77> 60210 DATA BB,F1,FE,FF,28,29,1F,DC
<38> 60220 DATA BC,F0,1F,DC,CE,F0,1F,DC
<95> 60230 DATA B6,F0,1F,DC,DE,F0,1F,DC
<21> 60240 DATA EC,F0,1F,DC,17,F1,1F,DC
< 7> 60250 DATA 3D,F1,1F,DC,3A,F1,CD,66
<69> 60260 DATA F1,E1,C1,23,10,A7,C9,21
<31> 60270 DATA C9,F1,06,08,ED,5B,BD,F1
<97> 60280 DATA 7A,90,57,14,C5,D5,E5,06
<86> 60290 DATA 08,4E,CB,01,21,C9,F1,38
< 7> 60300 DATA 03,21,C1,F1,CD,92,F1,1C
<72> 60310 DATA 7B,10,EF,E1,D1,C1,23,10
<94> 60320 DATA E2,32,BD,F1,18,CB,21,D0
<43> 60330 DATA F1,36,FF,C9,F5,21,C9,F1
<36> 60340 DATA 06,08,7E,CB,3F,A6,77,23
< 1> 60350 DATA 10,F8,F1,CB,87,C9,F5,21
<44> 60360 DATA C9,F1,06,08,7E,CB,3F,B6
<18> 60370 DATA 77,23,10,F8,F1,C9,F5,21
<17> 60380 DATA C9,F1,06,08,7E,2F,77,23
<61> 60390 DATA 10,FA,F1,C9,F5,3E,02,32
<36> 60400 DATA BF,F1,DD,21,C9,F1,0E,08
<27> 60410 DATA 06,08,DD,7E,00,1F,DD,CB
<61> 60420 DATA 00,1E,DD,CB,10,1E,DD,CB
<14> 60430 DATA 00,2E,DD,CB,10,1E,10,ED
<37> 60440 DATA DD,23,0D,20,E3,F1,C9,3E
<90> 60450 DATA 02,32,C0,F1,11,D0,F1,CD
<91> 60460 DATA 2A,F1,3A,BF,F1,3D,C8,11
<29> 60470 DATA E0,F1,21,08,00,19,06,08
<45> 60480 DATA 1A,1B,77,2B,77,2B,10,F8
<71> 60490 DATA AF,C9,AF,18,02,3E,02,32
< 3> 60500 DATA C0,F1,11,D8,F1,CD,50,F1
<14> 60510 DATA 3A,BF,F1,3D,C8,11,E8,F1
<71> 60520 DATA CD,5C,F1,21,FC,FF,19,01
<12> 60530 DATA 08,00,ED,B8,AF,12,1B,12
<49> 60540 DATA 1B,12,1B,12,1B,C9,21,C9
<51> 60550 DATA F1,ED,4B,BF,F1,ED,5B,BD
<27> 60560 DATA F1,D5,E5,15,05,05,CC,92
<36> 60570 DATA F1,14,CD,92,F1,14,04,04
< 9> 60580 DATA CC,92,F1,E1,11,10,00,19
<43> 60590 DATA D1,1C,0D,20,E4,ED,53,BD
<39> 60600 DATA F1,C9,7B,FE,5A,D0,7A,FE
< 6> 60610 DATA 20,D0,C5,D5,E5,16,00,07
<19> 60620 DATA 17,17,17,6F,7A,CE,B6,67
<74> 60630 DATA 7E,23,66,6F,EB,29,29,19
<80> 60640 DATA 29,EB,E1,01,08,00,ED,B0
<23> 60650 DATA D1,C1,C9,00,00,00,00,00
<49> 60660 DATA 00,00,00,00,00,00,00,00
<92> 60670 DATA 00,00
```

```
130,2154,159,9154,159,154,157,32,133,0,133,2032,149," Elb-    ",149,27,"p",32,27,"q"
149,32,133,0,133,2032,149," chaussee",149,27,"p",32,27,"q",149,32,133,0,133,2032
149,9032,149,27,"p",32,27,"q",149,32,133,0,133,2032,149," $2100,- ",149,27,"p",32,27,"q"
149,32,133,0,130,2154,159,9154,155,154,157,32,133,0,133,2032,149,"###Haupt-##",149,32,133
0,133,2032,149,"##bahnhof##",149,32,133,0,133,2032,149,9035,2035,149,32,133,0,133,2032
149,"##$5000,-##",149,32,133,0,130,2154,159,9154,2154,157,32,133,0,133,2032,149
" Grund-   ",149,32,133,0,133,2032,149,"   steuer  ",149,32,133,0,133,2032,149
" $   5% $ ",149,32,133,0,133,2032,149," $ $ $ $ $ ",149,32,133,0,130,2154,159,9154
158,154,157,32,133,0,133,2032,149," Pechvo- ",149,27,"p",32,27,"q",149,32,133,0
133,2032,149," gelgasse",149,27,"p",32,27,"q",149,32,133,0,133,2032,149,9032,149,27,"p"
32,27,"q",149,32,133,0,133,2032,149," $1800,- ",149,27,"p",32,27,"q",149,32,131,9138,9138
9138,9138,2138,0,130,2154,159,9154,155,154,159,9154,158,9154,158,9154,158,9154,158,0,133
2032,149,"^ Arbeits- ",149,27,"p",9032,27,"q",149,9032,149,27,"p",9032,27,"q",149
" ????? ",149,0,133,2032,149,"^  amt    ",151,9154,157," Altes ",151,9154,157
" ??   ?? ",149,0,133,2032,149,"^ ",9032,149," Kudamm  ",149,9032,149," Berliner",149
"    ???  ",149,0,133,2032,149,"^ Gehalt: ",149,9032,149,"Gymnasium",149," Platz  "
149," ??   ",149,0,133,2032,149,"^ $5000,- ",149,9032,149,9032,149,9032,149,9032
149,0,133,2032,149,"^ ",8253,32,149," $9500,- ",149," $7900,- ",149," $6800,- ",149
"  ??   ",149,0,133,2032,147,9154,2154,159,9154,159,9154,159,9154,159,9154,159,0
133,9032,5032,149,9032,149,9032,149,9032,149,9032,149,0,131,9138,5138,129,9138,129,9138
129,9138,129,9138,129,0,134,9138,5138,132,9138,132,9138,132,9138,132,9138,132,0,133
" nicht zum ",149,9032,149,9032,149,9032,149,9032,149,0,133,"Besuch,sondern",151
9154,159,9154,159,9154,159,9154,159,0,133," voll drin!  ",149," Sesam-  ",149
" Linden- ",149," ?????  ",149," Schwimm-",149,0,133," SCHWARZWALD- ",149
" stra~e  ",149," stra~e  ",149," ??   ?? ",149,"  bad   ",149,0,133
"    KLINIK   ",149,9032,149,9032,149,"   ???  ",149,9032,149,0,133,7032,27,"p",2032,27
"q"5032,149," $2600,- ",149," $2800,- ",149,"   ??    ",149," $3200,- ",149,0,133,5032,27
"p",6032,27,"q",3032,151,9154,159,9154,157,9032,149,9032,151,0,133,7032,27,"p",2032
27,"q",5032,149,27,"p",9032,27,"q",149,27,"p",9032,27,"q",149,"   ??    ",149,9032
149,0,130,2154,158,9154,158,154,159,9154,155,9154,155,9154,155,9154,155,0,133,2032,149
" Reeper- ",149,27,"p",32,27,"q",149,32,134,9138,9138,9138,9138,2138,0,133,2032,149
"  bahn   ",149,27,"p",32,27,"q",149,32,133,0,133,2032,149,9032,149,27,"p",32,27,"q",149
32,133,0,133,2032,149," $2400,- ",149,27,"p",32,27,"q",149,32,133,0,130,2154,159,9154
159,154,157,32,133,0,133,2032,149," Helenen-",149,27,"p",32,27,"q",149,32,133,0,133
2032,149," stra~e  ",149,27,"p",32,27,"q",149,32,133,0,133,2032,149,9032,149,27,"p"
32,27,"q",149,32,133,0,133,2032,149," $2200,- ",149,27,"p",32,27,"q",149,32,133,0,130,2154
159,9154,155,154,157,32,133,0,133,2032,149,32,134,7138,140,32,149,32,133,0,133,2032,149
32,133,"! ! ! !",133,32,149,32,133,0,133,2032,149,32,133," ! ! ! ",133,32,149,32,133,0
133,2032,149,32,131,7138,137,32,149,32,133,0,130,2154,159,9154,2154,157,32,133,0,133,2032
149,"*ATOMKRAFT-",149,32,133,0,133,2032,149,"***WERK****",149,32,133,0,133,2032,149,9042
2042,149,32,133,0,133,2032,149,"**$6000,-**",149,32,133,0,130,2154,159,9154,158,154,157
32,133,0,132,9138,132,9138,132,9138,132,9138,132,9138,5138,140,0,149,9032,149,9032,149
9032,149,9032,149,9032,5032,133,0,159,9154,159,9154,159,9154,159,9154,157," PARKPLATZ   "
133,0,149,"Emmerdale",149,"Die Farm ",149,"Schimmel-",149," Kneipe  ",149,3032,150,6154
156,3032,133,0,149," Farm   ",149,"der Tiere",149,"  hof   ",149,9032,149,3032,149,6032
149,3032,133,0,149,9032,149,9032,149,9032,149,"  RED   ",149,2032,150,155,6154,155,156
2032,133,0,149," $3400,- ",149," $3400,- ",149," $3500,- ",149,"  LION   ",149,2032,149
"╗",4138,"╗",149,2032,133,0,151,9154,159,9154,159,9154,157,9032,149,2032,147,2158,4154
2158,153,2032,133,0,149,27,"p",9032,27,"q",149,27,"p",9032,27,"q",149,27,"p",9032,27,"q"
149," $3600,- ",149,3032,147,153,4032,147,153,3032,133,0,155,9154,155,9154,155,9154,155
9154,159,154,158,9154,158,2154,136,0,9138,9138,9138,9138,2138,140,32,149,27,"p",32,27,"q"
149," Alexan- ",149,2032,133,0,149,27,"p",32,27,"q",149," derplatz",149,2032,133,0,149
27,"p",32,27,"q",149,9032,149,2032,133,0,149,27,"p",32,27,"q",149," $3800,- ",149,2032
133,0,151,154,159,9154,159,2154,136,0,149,27,"p",32,27,"q",149," Goethe- ",149,2032,133
0,149,27,"p",32,27,"q",149,"  Gasse  ",149,2032,133,0,149,27,"p",32,27,"q",149,9032,149
2032,133,0,149,27,"p",32,27,"q",149," $4000,- ",149,2032,133,0,151,154,155,9154,159,2154
136,0,149,"Verm¦gens- ",149,2032,133,0,149," Steuer   ",149,2032,133,0,149," £  10%  £ "
149,2032,133,0,149," £ £ £ £ £ ",149,2032,133,0,151,9154,2154,159,2154,136,0,149
"Bahnhof Zoo",149,2032,133,0,149,9035,2035,149,2032,133,0,149,"##$5000,-##",149,2032
133,0,149,9035,2035,149,2032,133,0,151,154,158,9154,159,2154,136,0,151,154,158,9154,159
2154,136,0,149,27,"p",32,27,"q",149,"Bahnhofs-",149,2032,133,0,149,27,"p",32,27,"q",149
"  Platz  ",149,2032,133,0,149,27,"p",32,27,"q",149,9032,149,2032,133,0,149,27,"p",32,27
"q",149," $4700,- ",149,2032,133,0,151,154,155,9154,159,2154,136,0,149," Kinder-  ",149
2032,133,0,149,"  garten  ",149,2032,133,0,149,9032,2032,149,2032,133,0,149," $3800,-  "
149,2032,133,0,151,9154,2154,159,2154,136,0,149,32,134,7138,140,32,149,2032,133,0,149,32
133,"! ! ! !",133,32,149,2032,133,0,149,32,133," ! ! ! ",133,32,149,2032,133,0,149,32,131
7138,137,32,149,2032,133,0,151,154,158,9154,159,2154,136,0,149,27,"p",32,27,"q",149
" Markt-  ",149,2032,133,0,149,27,"p",32,27,"q",149," platz  ",149,2032,133,0,149,27,"p"
32,27,"q",149,9032,149,2032,133,0,9138,9138,9138,9138,9138,2138,137,32,149,27,"p",32,27,"q"
149," $4600,- ",149,2032,133,0,158,9154,158,9154,158,9154,158,9154,159,154,155,9154,155
2154,136,0,149," Polizei-",149,27,"p",9032,27,"q",149,27,"p",9032,27,"q",149," WASSER- "
149," Gehen Sie in ",133,0,149,"  revier ",151,9154,159,9154,157,"  WERK   ",149
"     die      ",133,0,149," Hill    ",149," Oxford ",149," Penny  ",149," $4500,- "
149,9032,5032,133,0,149,"  Street ",149,"  Road   ",149,"  Lane   ",149,3032,146,158,152
149,2032,149," SCHWARZWALD- ",133,0,149,9032,149,9032,149,9032,149,2032,150,154,155,154
157,2032,149,"   KLINIK!   ",133,0,149," $5000,- ",149," $4300,- ",149," $4000,- ",149
6032,149,2032,149,6032,27,"p",2032,27,"q",6032,133,0,159,9154,159,9154,159,9154,159,9154
157,4032,27,"p",6032,27,"q",4032,133,0,149,9032,149,9032,149,9032,149,9032,149,6032,27
"p",2032,27,"q",6032,133,0,129,9138,129,9138,129,9138,129,9138,129,9138,5138,137,0,0000
6800,7900,9500,0000,1800,0000,5000,2100,6000,0000,2200,2400,0000,2600,2800,0000,3200,3400
3400,3500,3600,0000,3800,4000,0000,5000,4700,3800,0000,4600,0000,4500,4000,4300,5000
" Gehe 5 Felder zur¦ck!"
"  R¦cke vor bis zum      Alten Gymnasium."
"  Deine Aktien sind     gestiegen. Die Bank      zahlt Dir $5000,-"
"  R¦cke vor bis zur       Lindenstra~e."
" Gehe zur¦ck bis zur     Pechvogelgasse."
"Strafe f¦r zu schnelles  Fahren: $1000,-"
" Es ist Zeit f¦r Dich,   in den Kindergarten        zu gehen!"
"    Rasen betreten         verboten!           Strafe: $200,-"
"  R¦cke vor bis zum       Arbeitsamt."
" Du bist Bier-K¦nig     geworden. Zahle jedem    Spieler $1500,-"
" Steuer-R¦ckzahlung:        $4000,-"
" Ziehe zum Namenstag     von allen Spielern        $10,- ein."
" Wir bitten um eine      Spende f¦r's Alte      Gymnasium: $800,-"
"    Zahle an die         SCHWARZWALDKLINIK         $3000,-"
" Du gewannst bei einer   Rassehundeschau den     1.Preis ($20,-)"
" Ziehe Deine Rente ein       $850,-"
"Du erbst einen FIAT 126. Gehe ein Feld weiter."
"  Zahle an Spieler 0     (JOYCIE) eine Spende       von $500,-"
" Zahle Stromrechnung.        $400,-"
"    Gehe in die          SCHWARZWALDKLINIK!"
```

ES GIBT VIELE COMPUTERZEITSCHRIFTEN



– das Konzept von PASCAL ist einmalig.

Erstklassige Programme für alle Rechner, die eine CPU haben!

Nicht nur Pascal – fit auch in Modula-2, C, FORTRAN, PROLOG und BASIC.

Aktuelle News aus allen Bereichen des Computeralltages, Report und Reviews neuester Software.

Buchbesprechungen, Feuilleton und Spiele.

Eine »heiße Hotline« auf vier Kanälen als Leserforum und noch einiges mehr!

Mit PASCAL – keine Angst mehr vor Computersprachen – für Experten, Hobbyisten und Lernende.

PASCAL-International kostet im Abonnement:

Im Inland und Westberlin:	für 1/2 Jahr	40,– DM
	für 1 Jahr	80,– DM
Im europäischen Ausland:	für 1/2 Jahr	50,– DM
	für 1 Jahr	100,– DM
Im außereurop. Ausland:	für 1/2 Jahr	70,– DM
	für 1 Jahr	140,– DM

In diesen Preisen sind Kosten für Porto und Verpackung, sowie die MwSt. enthalten.

DMV GmbH · PASCAL-International
Postfach 250 · 3440 Eschwege · Tel. (0 56 51) 87 02

Turbo-Grafik für den JOYCE

– Grafik mit Turbo Pascal und ohne GSX...

So ziemlich für alle Rechner, auf denen Turbo Pascal läuft, gibt es grafikfähige Versionen dieser leistungsfähigen Programmiersprache – nur nicht für den JOYCE. Aber damit ist Schluß, denn jetzt gibt es eine leistungsfähige Grafikerweiterung, die sogar ohne die speicherplatzfressende GSX-Einbindung auskommt...

Am Anfang stand eine aus England stammende RSX, die unter Mallard Basic das Setzen und Löschen von Bildschirmpunkten erlaubte. Das Ganze war zwar etwas umständlich zu handhaben, aber immer noch besser als umständliches und speicherverschwendendes Operieren mit GSX, vor allem, wenn es nur um ein paar Linien ging. Mit der Zeit kam dann das Turbo Pascal-Fieber auf und die Idee einer Grafikerweiterung nahm Gestalt an. Was lag da näher, als auf Bekanntes zurückzugreifen. Allerdings war das RSX-Konzept nicht unbedingt das Wahre, hier mußte eine Abhilfe ersonnen werden, was schließlich zur Ersetzung der RSX durch External-Routinen führte. Dazu mußte natürlich die RSX erstmal »geknackt« werden – eine harte Nuß, die nach zwei durchhackten Nächten und einigen sehenswerten Systemabstürzen endlich zum ersten Erfolg führte: Pixel konnten schonmal gesetzt und gelöscht werden. Mit der Zeit wurde der maschinenabhängige Teil erweitert, erste Linien-Routinen geschrieben, Zeichensätze verändert und vieles mehr. »With a little help from our friends« – speziell Martin Schlöters Grafikserie »Vom Punkt zur 3. Dimension« aus »PASCAL International« verdanken wir viele Anregungen und »Zitate« – wuchs die JOYCE-Grafik-Erweiterung immer mehr, bald tummelten sich Kreise, Ellipsen und vieles mehr auf dem Schirm und der Drucker wurde mit einer Hardcopy-Routine zum Glühen gebracht. Schließlich war die vorliegende Turbo-Grafik fertig. Sie besteht aus einer Reihe von Include-Files, die, in den Programmtext eingebunden, ein völlig neues Turbo-Gefühl aufkommen lassen. Es wurde ein modularer Aufbau gewählt, der zwar zu mehr Files führte, aber ein selektives Einbinden der benötigten Funktionen erlaubt: nur das File KERNEL.INC muß eingebunden werden, der Rest nur je nach Bedarf. Im folgenden werden die einzelnen Funktionen der Erweiterung nach Include-Files geordnet vorgestellt. Auf umfangreichen theoretischen Ballast wird verzichtet, wissenschaftliche Hintergründe können vor allem in obiger Serie nachgelesen werden. Also ran an die Tastatur...

Als Arbeitsbereich wurde der Bildschirmbereich von 0/0 bis 719/247 (Grafik-Koordinaten) gewählt, was im Textmodus dem kompletten Bildschirm außer der Statuszeile (0/0 bis 89/30 in Textkoordinaten) entspricht. Dies hat zwei praktische Gründe: einerseits bleibt die Statuszeile frei für Meldungen, die nicht in die Bilder gehören, andererseits werden die Pixel der Statuszeile nicht gelöscht, wenn diese Statuszeile eingeschaltet ist. Eine Anpassung an den vollen Bereich des Bildschirms ist zwar möglich, dazu muß aber einiges in den Files geändert werden. Auch sollte Ihr Turbo Pascal an den großen Bildschirm des JOYCE angepaßt werden, da es sonst interessante »Nebeneffekte« geben könnte. Borland liefert Turbo Pascal aus unverständlichen Gründen in einer 80x24-Zeichen-Version aus, während der JOYCE ja 90x32 Zeichen unterstützt. Erreicht wird dies durch einen Durchlauf des Installationsprogramms TINST. Dort sollten die Fragen nach dem Senden von Terminal-Init- und TerminalExit-Sequenzen verneint und die neue Bildschirmgröße angegeben werden.

KERNEL.INC

KERNEL.INC ist das Herz unserer Grafik-Erweiterung und muß auf alle Fälle in die Programme eingebunden werden.

Es enthält neben dem unumgänglichen Maschinenspracheteil (auf den später noch eingegangen wird) die wichtigsten Elementarfunktionen und Teile, die auch von den anderen Include-Files benötigt werden. Die Funktionen im einzelnen:

GraphInit:
Diese Funktion ist die wichtigste der ganzen Erweiterung: sie initialisiert den MCode für die External-Routinen, und ohne die läuft die ganze Erweiterung nicht. Der Aufruf von GraphInit sollte daher der erste Befehl in Ihren Grafik-Programmen sein!!!

Plot(x,y,mode):
Der Plotbefehl ist (zusammen mit der External-Routine GX_Dot) der zweitwichtigste Teil der Erweiterung. Mit ihm ist es möglich, die einzelnen Punkte des Bildschirms (Pixel) anzusprechen. Dazu werden die Koordinaten des Punktes und der Zeichenmodus übergeben. Die Koordinaten sollten im Bereich $0<x<719$ und $0<y<247$ liegen (entspricht dem normalen Bildschirm ohne Statuszeile, Punkt 0/0 ist die linke obere Ecke), der Modus legt fest, ob das Pixel gesetzt (0), gelöscht (1) oder invertiert (2) werden soll, bei Modus 3 wird geprüft, ob ein Pixel gesetzt oder gelöscht ist (wird in der Funktion DotColor verwendet und ist normalerweise uninteressant).

Screen_Off, Screen_On:
Über den Port 248 kann der Bildschirm ein- und ausgeschaltet werden. Dies ist besonders praktisch, wenn Bilder »versteckt« gezeichnet werden sollen. (Man sollte aber nicht vergessen, den Bildschirm vor dem Verlassen des Programmes wieder einzuschalten...)

HideCursor, ShowCursor:
Falls der Cursorblock stören sollte, kann er mit diesen Befehlen ein- und ausgeschaltet werden. Erreicht wird dies ganz einfach durch die Ausgabe der entsprechenden Escape-Sequenzen.

DotColor(x,y):boolean:
Mit dieser Funktion kann der Zustand eines Pixels abgefragt werden, sie liefert den Wert TRUE, wenn das Pixel gesetzt ist. (Wird z.B. in der Hardcopy-Prozedur benötigt.)

Symbol(Code,rp0,..,rp7):
Diese Prozedur ermöglicht das Ändern der Bildschirmmatrix eines Zeichens, indem der ASCII-Code des Zeichens und das neue Muster als 8 Bildzeilen zu 8 Zeichen übergeben werden. Dabei bedeutet '*', daß der entsprechende Matrixpunkt gesetzt wird, ein Leerzeichen, daß er dunkel bleibt.

```
Symbol(255,'********',
           '*      *',
           '*      *',
           '*      *',
           '*      *',
           '*      *',
           '*      *',
           '********')
```

ändert also die Darstellung des Zeichens #255 in die angegebene Form. Diese Routine wurde samt der zugehörigen Umwandlungsroutine GX__Convert (wandelt die Binärzeile in einen Bytewert um) mit freundlicher Genehmigung der Autoren aus der CGX-Grafikerweiterung für den PC 1512 übernommen.

Hex__Symbol(Code,r0,..,r7):
Funktioniert genauso wie Symbol, allerdings werden die Werte direkt als Byte-Werte übergeben. Beide Prozeduren verwenden die External-Routine GX__Symb, die eine Zeichenmatrix in das Matrix-RAM kopiert.
Zu beachten ist noch, daß die Compilerendadresse niemals größer als $F3FF gesetzt wird, da sonst die External-Routinen von Variablen überschrieben werden, aus ähnlichem Grund muß auch der Run-Modus vermieden werden, wenn man nicht einen klassischen Systemabsturz erleben will. (Die Standard-Endadresse ist sowieso kleiner, man braucht im Prinzip also nichts verändern.)

SIMPLE.INC

Dieses Include-File stammt eigentlich noch aus der Anfangsphase der Grafikerweiterung und enthält sehr einfache Grafikbefehle. Wesentlich leistungsfähigere Versionen finden sich in GRAPHLIB.INC, auf Grund ihrer etwas anderen Syntax und dadurch für manche Anwendungen vielleicht besser geeignet, seien sie hier vorgestellt:

S__Horizontal(x1,x2,y,mode):
Zeichnet eine horizontale Linie von x1=y nach x2=y im durch mode spezifizierten Plotmodus.

S__Vertikal(x,y1,y2,mode):
Zeichnet eine vertikale Linie von x/y1 nach x/y2 im durch mode spezifizierten Plotmodus.

S__Box(x,y,xlen,ylen,mode):
Zeichnet einen Kasten im Plotmodus mode ab der Koordinate x/y mit den Maßen xlen in X- und ylen in YRichtung.

GRAPHLIB.INC
– Die Grafikbibliothek –

Mancher von Ihnen wird sich vielleicht beim Lesen gesagt haben:

Public-Domain für CPC, Joyce und C-128

Public Domain-Software für Ihren **Schneider-CPC, Joyce** und **Commodore-128** mit deutschen Programmtexten und einem gedruckten Handbuch - so machen diese preisgünstigen Super-Programme richtig Spaß!

Programm des Monats: Z80-Assembler (Diskette 2)

Profi-Programme für die Assemblerprogrammierung! Rasend schneller Z80-Assembler, leistungsfähiger Linker. Der Debugger stellt den Singlestep am Bildschirm dar. Der Reassembler unterscheidet Daten und Programmcode.

1- **JRT-Pascal** - vollständiges Pascal mit 64K-Strings, Overlays ... *
3- Künstliche Intelligenz - Interpreter für **XLISP** und **E-PROLOG** *
4- C-Compiler **Small-C** - Fließkommazahlen, umfangreiche Bibliotheken *
5- **FORTH-83** - mit Assembler, Decompiler, Screen-Editor ...
6- CP/M-Utilities wie **Dateikompressor, Diskmonitor, UNERA** ...
7- Alle Programme aus dem **Großen CPC-Arbeitsbuch** (nur CPC)
8- Adventure **Colossal Cave** (Programm englisch, Anleitung deutsch) *
9- **CPC-Disk Utilities** - kopiert geschützte Software (nur CPC)
10- **BizBasic** - CPC-Basic-Erweiterung (relative Dateiverwaltung etc.)
11- Basic-Compiler **E-BASIC** - CBASIC-kompatibel, viele Befehle
12- **Turbo Pascal-Programme** - INLINE-Generator, GSX- und ROM-Grafik

* auf dem CPC-464/664 nur mit Speichererweiterung (64K genügen).
Disketten 1-4 und 11 mit WordStar-kompatiblem Editor. C-128: Disk 12 ohne Grafik. 1570/1571-Format. Kein 1541-Format.

Der Preis? Sage und schreibe **nur 30,- Mark** pro Diskette inklusive Porto und Verpackung! Bitte Diskettenformat (3 Zoll, Vortex, 1570/1571) angeben. Lieferung per **Nachnahme** oder **Vorauskasse**, Ausland: nur Vorauskasse.

MARTIN KOTULLA

Grabbestraße 9, 8500 Nürnberg 90, Telefon 09 11/30 33 33

Weitere Bezugsquellen für CPC, Joyce und Commodore-128:

TESCO GmbH, Rüdenhausenerstraße, 8714 Wiesentheid, Telefon 0 93 83 / 12 37
Computer Solutions, Hansastraße 9, 8000 München 40, Telefon 089 / 5 70 25 39
Computer Hard-Software Simon, Espenstraße 79, 4600 Dortmund 1, Telefon 02 31 / 51 13 70
Mükra-Datentechnik, Schöneberger Straße 5, 1000 Berlin 42, Telefon 030 / 7 52 91 50
Soft- und Hardware Ulrike Becker, Fasanenweg 2, 6690 St. Wendel 8, Telefon 06 856 / 504

Weitere Bezugsquellen für Schneider-CPC und Joyce:

Computerstore, Hochstraße 11, 8500 Nürnberg 80, Telefon 0911 / 28 90 28
Techn. Büro Hochholzer, E.-Prunner-Str. 1, 8062 Markt Indersdorf, Tel. 08136 / 1625
Weeske Computer-Elektronik, Potsdamer Ring 10, 7150 Backnang, Telefon 07191 / 1528-29
EDV-Beratung Schult, Von-Beck-Straße 6, 7500 Karlsruhe 1

JOYCE ● CPC

Papierführung:
● zwei verstellbare Seitenführungen = gerader Papiereinzug
● In anthrazit, blau, gelb, grün, orange, rot
DM 29,-
sFr. 29,-/öS 229,-

Bildschirmfilter:
● vermindert störende Reflexionen
● angepaßt an das Design des Monitors
● leichte Montage
DM 49,-
sFr. 49,-/öS 395,-

FD-2:
● 1 MB Drive für 8256
DM 499,-
sFr. 499,-/öS 4 490,-

RAM-Erweiterung:
● Original 257er Chips!
DM 99,-
sFr. 99,-/öS 799,-

Farbband:
● 2 Stück
DM 29,-
sFr. 29,-/öS 229,-

Bildschirmfilter:
● für CTM 640/644
DM 39,-
sFr. 39,-/öS 275,-

● für GT 64/65
DM 29,-
sFr. 29,-/öS 229,-

Joyce-Hard-disc:
● 20 MB
DM 1495,-
sFr. 1 495,-/öS 11 995,-

HÄNDLERANFRAGEN SIND WILLKOMMEN!

Das komplette ABD-Zubehör ist bei den Schneider-Computer-Händlern sowie bei Hertie, Horten, Kaufhof und Ringfoto erhältlich.

Alle unsere Produkte werden zu **Tagespreisen** gehandelt.
Deshalb können die Preise tatsächlich niedriger sein als genannt!

Deutschland:
ABD Electronic
Zettachring 12
7000 Stuttgart 80
Telefon 07 11-7 15 00 37

Schweiz:
Swicom SA
Route de Boujean
CH-2502 Biel-Bienne
Telefon (0 32) 42 27 84
oder 23 18 33

Österreich:
Wagner Electronics
Hauptstraße 171
3001 Mauerbach
Telefon 02 22-97 21 66

»Jetzt kann ich also Punkte an beliebige Bildschirmpositionen setzen, sie löschen, invertieren, ihren Status abfragen ... Was soll's, damit habe ich noch lange keine Grafik!!!« Richtig, aber die kommt jetzt; Die Prozeduren in GRAPHLIB.INC sind zumindest zum Teil an GSX orientiert. Sie bieten vieles, was der Bildschirmtreiber DDSCREEN.PRL ebenfalls kann; manchmal etwas weniger, manchmal etwas langsamer, aber oft vielseitigere und vor allem variablere Funktionen und – ohne den Rattenschwanz an Funktionen, die man für das Programm, was man gerade schreibt, sowieso nicht braucht, die aber Speicherplatz »fressen«, den man wiederum oft sehr nötig hat. Aus GRAPHLIB.INC kann man sich die entsprechenden Teile wenn nötig einzeln in den Quelltext kopieren.

Die Prozeduren und ihre Verwendung

Um eine Linie, einen Kreis oder andere Objekte auf den Bildschirm zu zaubern, müssen zunächst alle Punkte berechnet und einer nach dem anderen gesetzt werden. Dafür sind die Prozeduren LINE() bis CIRCLE() zuständig. LINE() zieht eine Linie zwischen den Bildschirmpunkten, deren Koordinaten im Aufruf übergeben wurden. Die einfachste Art dies zu tun, wäre, mit Hilfe der Geradengleichung f(x) = Steigung * x + y-Achsenabschnitt für die xKoordinaten die Werte für f(x) zu berechnen, sie zu Runden, da der Bildschirm ja nur diskrete Schritte erlaubt, und sie anschließend mit PLOT() zu setzen. Da die Berechnung über REAL-Werte erfolgen müßte, wäre dies aber mit Sicherheit auch die langsamste Art. LINE() rechnet nur mit INTEGER-Variablen. Ausgangspunkt ist auch hier wieder die Geradengleichung – diesmal aber ohne Achsenabschnitt. Vielleicht erinnern Sie sich noch mehr oder weniger dunkel an den Mathematikunterricht?! Die Steigung einer Geraden läßt sich aus den Differenzen der Koordinaten zweier Punkte – in unserem Fall von Start und Endpunkt – berechnen. Umgeformt lautet die Gleichung dann: x * DY = y * DX. Mit DX und DY werden bei jedem Schritt die Abweichungen der gesetzten Punkte von der vorgegebenen Gerade bestimmt und zur Auswahl der nächsten Schrittrichtung benutzt. Mehr Details will ich Ihnen und mir ersparen. BOX() bringt ein Rechteck auf den Bildschirm. Übergeben werden die Ecken links oben und rechts unten und – wie auch bei den anderen Prozeduren – der Modus (setzen, löschen oder invertieren). Um die Box nicht unnötig zu bremsen, wird nicht LINE() zum ziehen der Linien benutzt, sondern die Sonderfälle für horizontale und vertikale Linien.

POINT_SWAP() sorgt dafür, daß die Startkoordinate immer kleiner als die Endkoordinate ist und nicht über den Rand gemalt wird. SQUARE() ist ein Spezialfall von BOX() und zeichnet – wie zu erwarten – ein Quadrat. Als Parameter werden die linke obere Ecke und die Breite in Pixeln übergeben. Wie wird nun das Quadrat zum Quadrat? Hierfür ist die Konstante ASPECT_RATIO zuständig; sie gibt das Verhältnis von Pixelhöhe zu Pixelbreite an. Ihr Wert kann durch Ausmessen von »Boxen« am Bildschirm ermittelt werden – probieren geht auch hier über studieren. Auch ELLIPSE() und CIRCLE() erhalten durch das Einbeziehen der ASPECT_RATIO erst das gewünschte Aussehen. Ebenso wie die Prozedur LINE() vermeidet auch ELLIPSE() weitgehend REALzahlen. Auch hier wird nur die Abweichung betrachtet, die durch jeden Schritt entsteht. Außerdem macht sich die Symmetrie der Figur nutzbringend bemerkbar: Wir müssen nur ein Achtel aller Punkte berechnen, alle andern erhalten wir durch Spiegelungen entsprechende Kombination von Addition und Subtraktion der errechneten Werte von den Koordinaten des Mittelpunktes. Radius_x gibt die Anzahl der Pixel der x-Halbachse an, die y-Halbachse wird so umgerechnet, daß das Größenverhältnis auf dem Bild dem gewünschten Verhältnis von Radius_y zu Radius_x entspricht. CIRCLE() wird als Ellipse mit gleich langen Halbachsen berechnet; übergeben wird dementsprechend außer den Mittelpunktskoordinaten nur noch der Radius.

Soweit die »Grundfunktionen«. Aber es gibt noch mehr; Die Prozedur PLOT_MARKER() ist vor allem zur Darstellung von Meßwerten u.ä. gedacht. Sie setzt auf den Punkt x, y einen »Marker« aus den in MARKER_TYPE deklarierten Elementen. PLOT_MARKER(10, 20, Kreis, 0) zieht also einen Kreis um den Punkt 10,20. Wenn Sie MARKER_HEIGHT mit im Aufruf übergeben, können Sie auch verschiedene Größen erhalten, was mir persönlich allerdings nicht besonders sinnvoll erschien, da die Marker dann zum Teil doch etwas unförmig werden. Als nächstes folgen vier POLY_...-Prozeduren. Wie der Name vermuten läßt, wird hier etwas mehrfach ausgeführt. Die erforderlichen Koordinaten werden als zweidimensionales Array übergeben; POINT_ARRAY(.i,0.) enthält die x-, POINT_ARRAY(.i,1.) die y-Werte. POINT_ARRAY ist auf 250 dimensioniert, was für die meisten Anwendungen völlig ausreichen dürfte. Die Zahl der wirklich benutzten Arrayelemente wird als Parameter mit übergeben. POLY_PLOT() setzt lediglich Punkte auf die Koordinatenpaare des Arrays; POLY_MARKER() die in MARKER_TYPE definierten Marker. POLY_LINE() verbindet die Punkte in der Reihenfolge, in der sie im Array enthalten sind, mit Linien. Alle Prozeduren können vielseitig eingesetzt werden. Etwa um Funktionen oder Meßwerte darzustellen, wie im Beispiel GRAPHIC.PRO demonstriert, oder um Polygone zu zeichnen. Bei POLY_LINE() muß dann das erste Koordinatenpaar noch einmal als letztes im POINT_ARRAY enthalten sein, damit eine geschlossene Figur entsteht. Ebenso verhält es sich bei der vierten POLY_..-Prozedur POLY_FILL(). Sie füllt ein Polygon, dessen Eckpunkte als POINT_ARRAY übergeben wurden, im entsprechenden Modus, d.h. sie füllt den Inhalt »weiß«, löscht oder invertiert alle Punkte.

Allerdings wird es durch die Art der Berechnung notwendig, im Anschluß an das Füllen mit PLOY_LINE() noch die 'Begrenzung' zu zeichnen.

Ihre Anwendung im Demo-Programm DREID (entnommen aus PASCAL INTERNATIONAL 2(2), Januar 1987) zeigt am besten den Effekt, den man so erzielen kann: Hidden-LineDarstellungen ohne komplizierte Umrechnungen und große Arrays. Soweit zum Inhalt von GRAPHLIB.INC.

PATTERNS.INC Erweiterungen für LINE() und POLY_FILL()

Das Modul PATTERNS.INC bietet neben PATTERN_LINE(), einer erweiterten LINE()-Prozedur, eine analoge Erweiterung von POLY_FILL(), PATTERN_FILL(). Wo in der einfachen Version nur der Modus angegeben werden konnte, werden jetzt die Muster LINE_TYPE und FILL_TYPE übergeben: fünf Linienarten und mindestens neun Füllmuster sind in den beiden Prozeduren verfügbar. Wem dies nicht ausreicht oder wem das Vorhandene nicht gefällt, der kann sich seine eigenen Varianten definieren - als * und Leerzeichen in einem String, einfacher geht's wirklich nicht. Um die Füllmuster zu erhalten, werden die Zeichen 248 bis 255 des Bildschirmzeichensatzes benutzt. Zwar gehen diese dabei für den Gebrauch innerhalb des Programms verloren, aber diese Vorgehensweise hat zwei Vorteile, die dies rechtfertigen. Der erste ist, daß auf diese Weise auch die »normalen« Zeichen als Füllmuster verwendet werden können; z.B. das Leerzeichen (# 32) oder die Zeichen zwischen 128 und 159, die die Rahmen vieler JOYCE-Programme aufbauen. Im Demoprogramm ist die einfache, senkrechte Linie als Beispiel aufgenommen worden. Der zweite Vorteil ist, daß man großflächige »Füllungen« auch mit write(chr(# #)) vornehmen kann, wenn die Felder sich an die Zeilen und Spalten des Textbildschirms halten, was um einiges schneller ist, als der Weg über die Grafik.

GRAPHIC.PRO Anregungen für eigene Erweiterungen

Zum Abschluß noch einige Worte über die »Programmgestaltung«. Für viele Anwendungen ist es sinnvoller, wenn man im »eigenen Koordinatensystem« arbeiten kann, ohne vorher alle Daten in Systemkoordinaten umrechnen zu müssen. GRAPHIC.PRO stellt als Beispiel eine Prozedur vor, mit der man an beliebige Bildschirmausschnitte, die durch die VIEW_...- Parameter festgelegt werden, ein Koordinatensystem zeichnen kann, in dem ein POINT_ARRAY dargestellt wird. Die Koordinaten der Achsen können als WIN_...-Parameter angegeben werden. Zur Umrechnung der Benutzerkoordinaten in die Systemkoordinaten dient die Prozedur KOORD_TRANS(). Wenn das ganze Programm in Benutzerkoordinaten arbeiten soll, kann man VIEW_X_MIN, VIEW_X_MAX, usw. als globale Variablen deklarieren und beispielsweise über eine Prozedur SET_VIEWPORT(), ähnlich wie CBASIC sie zur Verfügung stellt, verändern. Weiterhin kann man dann in der PLOT()-Prozedur abfragen, ob diese Grenzen überschritten werden, und nur in solchen Fällen die Punkte setzen, in denen sie innerhalb dieser Grenzen liegen. Weist man zusätzlich jeweils die aktuelle Position des 'Grafikcursors' zwei globalen Variablen - C_X_POS und C_Y_POS zu, so ist der Weg offen zur eigenen Turtlegrafik a la LOGO mit Prozeduren wie DRAW_TO usw. oder zur Abfrage der Cursorposition und damit zum Beispiel zu Menueauswahlen wie sie in DR DRAW verwendet werden.

PRINTAT.INC

PRINTAT.INC enthält einen wesentlich erweiterten Ausgabebefehl für Turbo Pascal. Mit ihm ist es möglich, eine Positionierung der Ausgabe mit verschiedenen Schriftarten sowie verschiedenen Ausgaberichtungen zu verbinden. Die Syntax des neuen Printbefehls ist PRINTAT (x,y,font, heading,text). X und y sind die Koordinaten (bezogen auf Zeichen-Koordinaten, $0<x<90$, $0<y<32$) für den Beginn der Textausgabe und entsprechen denen des Turbo-Standards »gotoxy()«. Font spezifiziert eine von 5 verschiedenen Schriftarten. Im einzelnen sind 5 Schriftarten möglich:

Font=0: normaler Zeichensatz
Font=1: Schmalschrift
Font=2: Fettschrift

COMPUTER-STUDIO

MERCATOR
Experten-Finanzbuchhaltung
frei wählbarer Kontenplan
manuelle u. automat. Steuer
Voll-Kurz-Sammelbuchungen
Primanota – Journal – Kontoauszug
Summen + Saldenliste – Deb./Kred.
permanente Saldenbilanz G+V
OP's-Mahnung-Kunden/Lief. Briefe
für JOYCE 8256 DM 399,–
für JOYCE 8512 plus DM 499,–

Computer-Studio-Offenbach
R. Reiss, Kl. Biergrund 17
6050 Offenbach, Tel. (069) 81 06 13

OFFENBACH

Michael Naujoks

Academy	69,90
Bounder	49,90
Brian Clough's Football Fortunes	49,90
Classic Collection	44,90
Frank Bruno's Boxing	69,90
Jewels of Darkness	79,90
Southern Bell + Heathrow ATC	59,90
Star Glider	79,90
Steve Davis Snooker	49,90
Tau Ceti	69,90
The Pawn	85,90
ACE	59,90



Joystick-Controller	69,–
Sound-Synthesizer + Joystick-Interface	129,–
Echtzeituhren-Modul	129,–
Adapter f. obige Geräte	29,–
256 k Speichererweiterung	109,–

Entwicklung & Vertrieb von Computer Soft- und Hardware
Rottmannstr. 40, 6900 Heidelberg

Hotline: (06221) 46885

Font=3: inverse Schrift (entspricht dem Turbo-Befehl »Lowvideo«)
Font=4: Kursivschrift
(Zur Erzeugung dieser Schriftarten vgl. Serie »Vom Punkt zur 3. Dimension«, Pascal International 8/87 S.34 ff.)

Heading bestimmt die Ausgaberichtung des Textes. Es sind hierbei vier Richtungen möglich, die den Text um jeweils 90 Grad gedreht ausgeben. (Heading 0 = normale Ausgabe, vgl. PTEST.PAS).
Der Text kann maximal 90 Zeichen umfassen (Typ Strg__90, definiert in KERNEL.INC). Er wird Zeichen für Zeichen ausgeben, wobei die Zeichenmatrix für jedes Zeichen aus dem Matrix-RAM ausgelesen wird (GX__Get), je nach Bedarf gedreht und anschließend als Zeichen mit dem ASCII-Wert 0 in die Matrix zurückgeschrieben und normal ausgedruckt wird. Die Matrix für dieses Zeichen ist also geändert, Sie sollten in Ihren Programmen den Befehl »write #27,#0« vermeiden. (Oder das entsprechende Sonderzeichen bei Bedarf neu definieren...) Eine Demonstration des neuen Ausgabebefehls finden Sie in PTEST.PAS.

SPRINTAT.INC

Dieses Include-File verdankt seine Entstehung eigentlich einem Tippfehler bei der Entwicklung von PRINTAT.INC. Es stellt den Befehl SPRINTAT zur Verfügung, der von Syntax und Optionen genau PRINTAT entspricht, nur daß hier die Ausgabe in Spiegelschrift erfolgt...

HARDCOPY.INC

Grundlage dieser Hardcopy-Routine war ein Listing in PASCAL International, welches jedoch zuerst funktionsfähig gemacht und anschließend erweitert wurde.
Mit ihr ist es möglich, einen beliebigen Bildschirmbereich in 4 Größen horizontal oder vertikal auszudrucken. Die Syntax ist Hardcopy (left, top, right, bottom, size, direction). Dabei sind »left« und »top« die Koordinaten des linken oberen, »right« und »bottom« des rechten unteren Pixels, die den Ausgabebereich festlegen. »Size« legt den Vergrößerungsfaktor fest und muß 1,2,4 oder 8 betragen. »Direction« gibt die Ausgaberichtung (»V« oder »H«) an.

Zu beachten ist, daß diese Optionen gewissen Einschränkungen unterliegen. Da der JOYCE-Drucker nur 960 Punkte in einer Zeile drucken kann, ist es z.B. nicht möglich, eine Hardcopy des ganzen Bildschirms horizontal in doppelter Größe auszudrucken, weil dazu eine Auflösung von 1440 Punkten nötig wäre. Man sollte also schon vor dem Aufruf etwas rechnen. Ist eine Ausgabe wegen falscher Parameter nicht möglich, bricht die Routine von allein ab. In vertikaler Richtung sind keine Grenzen gesetzt, allerdings kommt es in diesem Modus zu Verzerrungen der Proportionen.

DISKETTE.INC

Natürlich möchte man die Grafiken nicht nur auf Papier, sondern zur späteren Verwendung eventuell auch auf Diskette speichern. Hierzu dient das Include-File DISKETTE.INC. Es stellt die Prozeduren SaveScreen (Dateiname) und LoadScreen (Dateiname) zur Verfügung. Als Parameter wird der Dateiname über den vordefinierten Typ Strg__14 übergeben, wodurch neben Namen und Extension auch ein Laufwerk angegeben werden kann. Diese Funktionen übertragen den Bildschirminhalt auf Diskette oder umgekehrt und haben nur einen kleinen Nachteil: sie benötigen einen 22 KByte großen Puffer, der natürlich vom Hauptspeicher abgeht (desgleichen auf der Diskette...). Dafür sind sie allerdings sehr schnell: da der Bildschirm nicht Pixel für Pixel sondern zeichenweise abgetastet wird, ist der Vorgang ca. 64 mal so schnell wie ein Vorgehen mit DotColor und Plot (Diskettenzugriffe vernachlässigt). Der Bidaufbau erfolgt nach dem Einlesen etwa genau so schnell wie eine Ausgabe mit dem write-Befehl. (Evtl. kann man auch mit Screen__Off einen versteckten Bildaufbau erreichen.) Anzumerken ist noch, daß keine Fehlerprüfung über IORESULT erfolgt (- ein Zeichen des Vertrauens in die Anwender...).

FONT.INC

Ähnlich wie mit den Bildern verhält es sich mit den Zeichensätzen. Schön, wenn man sie auch auf Diskette hat. Die in diesem File enthaltenen Prozeduren LoadFont (Dateiname) und SaveFont (Dateiname) ermöglichen das Laden und Speichern von kompletten Zeichensätzen. Die Zeichensätze müßen als 2 KByte großes File vorliegen (dies ist auch die benötigte Puffer-größe). Die *.LET-Files des »Mouse Packs« können direkt verwendet werden, bei mit dem »Character Designer« erstellten Zeichensätzen muß erst der Ladervorspann mit LETCOM entfernt werden.

SOUND.INC

In Sachen Musik ist auf dem JOYCE ja relativ wenig los, CD-Qualtät kann der eingebaute Beeper beim besten Willen nicht liefern. Etwas mehr als das kurze Piepsen bei der Ausgabe von »↑G« ist aber dennoch möglich, wenn man den Beeper direkt ansteuert. Zuständig hierfür ist unser Universal-port 248, über den man den Beeper beliebig ein- und ausschalten kann. SOUND.INC stellt hierzu folgende Funktionen zur Verfügung:
Sound__On: Schaltet den Beeper für eine beliebig lange Zeit ein.
Sound__Off: Schaltet den Beeper wieder aus.
Play(Anzahl,Einzeit,Auszeit): Startet einen Beeper-Zyklus, der Anzahlmal durchlaufen wird, wobei zwischen den Ein- und Ausschalten des Beepers die entsprechende übergebene Zeit vergeht. In der vorliegenden Version wird diese Zeit noch durch eine Leerschleife erzeugt, durch Verwendung des Turbo-Befehls Delay() ließe sich hier eventuell eine »Normierung« erreichen.
Musikalische Ehren kann man damit zwar nicht verdienen, einige gute

bzw. abwechlungsreiche Effekte sind aber dennoch möglich. Zu erwähnen ist noch, daß SOUND.INC das einzige Include-File dieser Reihe ist, welches ohne KERNEL.INC auskommt...

Technisches

Das Listing GRAFINC.PRN zeigt den Assemblercode der External-Routinen GX_Plot, GX_Symb GX_Get und GX_ScrAcc. Sie machen alle vier von der XBIOS-Funktion #35 (Einblenden des Screenenvironments) Gebrauch und leisten folgendes:

GX_Plot: Berechnet und bearbeitet Pixelwerte über das Roller-RAM. Lage im Speicher ab $F400, per absolute Variablen übergebene Parameter sind X-Koordinate ($F410/11), YKoordinate ($F412) und Plotmodus ($F413). Optionaler Einsprung (z.B. für Inline-Routinen) bei $F407 mit X im DE- Register, Y und Modus in HL. Bei der Statusabfrage (Modus 3) wird der Zustand in die Speicherstelle des Modus zurückgeschrieben (Wert< >0 = Pixel gesetzt).

GX_Symb: Kopiert 8 Bytes aus einer Tabelle ab $F477 in das Matrix-RAM. Die Zieladresse muß aus dem Zeichencode errechnet und als absolute Variable bei $F472/73 übergeben werden. Einsprung nach Übergabe der Werte bei $F462, kein optionaler Einsprung möglich.

GX_Get: Kopiert 8 Bytes aus dem Matrix-RAM in die Tabelle ab $F477. Die Quelladresse muß aus dem Zeichencode berechnet und als absolute Variable bei $F489/99 übergeben werden. Einsprung nach Übergabe der Werte bei $F47F, kein optionaler Einsprung möglich.

GX_ScrAcc: Kopiert 8 Byte, die einem Standard-Zeichen entsprechen, aus dem VideoRAM in die Tabelle ab $F477 oder umgekehrt. Die Koordinaten werden als Text-Koordinaten absolut übergeben: 0<x<89 bei $F4A0, 0<y<30 bei $F49E. In $F4B9 wird die Zugriffsrichtung übergeben: steht hier 0, wird vom Video-RAM in die Matrix geschrieben, andernfalls wird die Matrix in den Bildschirm kopiert. Diese Routine wird zur Zeit nur im File DISKETTE.INC verwendet, andere Möglichkeiten wären eine Verwendung in PRINTAT.INC oder zur Darstellung von Windows und Sprites (bereits in Planung...). Der Einsprung erfolgt nach Übergabe der Werte X, Y und AccessMode (ggf. auch der Matrix) bei $F494.

Mit diesen Informationen ist es möglich, die External-Routinen auch direkt anzusprechen. Dies dürfte besonders für versiertere Programmierer interessant sein, da sich durch die Verwendung von Inline-Code noch eine wesentliche Steigerung der Geschwindigkeit erreichen ließe. Anregungen und Verbesserungsvorschläge aus der Leserschaft sind jederzeit willkommen...

Demo-Programme

Zu dieser Grafikerweiterung existieren einige Demos (teilweise noch aus der Entwicklungsphase), die hier jedoch nicht alle abgedruckt werden können. Auf der DATABOX zum Sonderheft sind sie jedoch komplett als Quelltexte enthalten.

Es sind im einzelnen:

DEMO1.*: Verwendung von Elementarbefehlen

DEMO.*: komplexere Dinge

PTEST.*: Möglichkeiten des PRINTAT-Befehls

Ein besonderes 'Zuckerl' ist das Programm DREID.*. Es stellt Funktionen als dreidimensionale Netz-Grafiken dar und wurde in PASCAL International 2/87 für den ATARI ST veröffentlicht. Hierzu steht ein eigenes Include-File namens DREID.INC für die Grafikroutinen zur Verfügung. Das Programm läuft jetzt auch auf dem JOYCE und ist nur rund 10 mal langsamer, was angesichts der Tatsache, daß es sich um reines Turbo Pascal handelt, recht beachtlich ist. Es sind 3 Hardcopy-Arten möglich, die nach dem Bildaufbau über <FORM> (=normal), <TAB> (=doppelt hoch) und <EXIT> (= fett + doppelt hoch) ausgewählt werden. <RETURN> ermöglicht die Eingabe von neuen Winkeln, wird für Beta 0 eingegeben, so endet das Programm.

Ausblick

Es dürfte klar sein, daß die vorliegende Version der Grafik-Erweiterung noch nicht unbedingt der Weisheit letzter Schluß ist, aber für den Anfang ist sie bestimmt brauchbar. Daher also viel Vergnügen mit dem neuen Turbo Pascal. Neue Teile dürften bald folgen, wir sind auch auf die Resonanz bei der Leserschaft gespannt. Also bitte keine Zurückhaltung in Sachen Anregungen, Verbesserungen und Anwendungen... (Besonders interessieren würde uns ein Tip, wie man etwas mehr Farbe auf den JOYCE bekommt, denn das ist uns beim besten Willen nicht gelungen...) Was im Augenblick noch in unseren Köpfen herumspukt, wären Erweiterungen in Richtung Sprites und Windows mit Wiederherstellung des Bildschirms sowie eine generelle Geschwindigkeitssteigerung und eine Möglichkeit zur Verwendung mit dem »Mouse Pack« erstellter Bilder. Für diese Sonderheft soll es jedoch genug an Grafik sein, wir wünschen »Auf Wiederlesen« im nächsten Sonderheft oder im Stamm-Magazin...

(M. Anton/E. Wirth)

LITERATUR

Martin Walser, Andreas Stiller, Plotter selbstgebaut, Teil 3: Treiber Magazin für Computer Technik, November 1984, S. 56 – 59

Frank Schmidt, Grafik mit dem Drucker, Erstellung und Ausgabe hochauflösender Grafik im Bitmustermodus Magazin für Computer Technik, Juni 1984, S. 44 – 48

Roland Waclawek, Kreisgenerator, »Ganz schnelle« Kreise in Theorie und Z80-Praxis Magazin für ComputerTechnik, September 1986, S. 122 – 126

Martin Schlöter, Vom Punkt zur dritten Dimension (Serie;) PASCAL INTERNATIONAL ab Ausgabe 2(4), Januar 1987

Ron Worston, Armstrad PCW 8256 Plot RSX Personal Computer World 9(7), July 1986, pp. 190 – 192

```
ä ------------------------ KERNEL.INC ------------------------
  Include-File fuer Grafikbefehle auf dem JOYCE unter Turbo Pascal.
  Hauptmodul, enthaelt MCode und elementare Routinen. Muß auf alle
  Fälle ins Hauptprogramm eingebunden werden.
  Es sind enthalten:

     procedure GraphInit       -> Initialisierung des MCode
     procedure Plot(x,y,m)     -> Pixel set/reset/invert
     procedure Screen_On       -> Bildschirm einschalten
     procedure Screen_Off      -> Bildschirm ausschalten
     procedure ShowCursor      -> Cursor ein
     procedure HideCursor      -> Cursor aus
     function  DotColor()      -> TRUE=Pixel gesetzt
     procedure Symbol(...)     -> Zeichensatz aendern
     procedure Hex_Symbol(...) ->      "         "

  Variablen & Prozeduren mit dem Namen GX_ sind zu vermeiden, um
  nicht mit Interna der Grafikerweiterungen zu kollidieren...

  Zum Compilieren die Endadresse auf $F3FF setzen, um nicht den
  MCode zu ueberschreiben!!!
------------------------------------------------------------ ü


type GX_MCode  = array[0..202] of byte;
     GX_String = string[8];
     Strg_14   = string[14];
     Strg_90   = string[90];

var  GX_MProg : GX_MCode   absolute $F400;

const GX_Data:GX_MCode =
  ($ED,$5B,$10,$F4,$2A,$12,$F4,$01,$14,$F4,$CD,$5A,$FC,$E9,$00,$C9,
   $00,$00,$00,$00,$3E,$03,$A2,$57,$E5,$26,$00,$29,$01,$00,$B6,$09,
   $4E,$23,$46,$79,$E6,$F8,$6F,$60,$29,$19,$7D,$E6,$F8,$6F,$79,$E6,
   $07,$B5,$6F,$7B,$E6,$07,$3C,$47,$AF,$37,$1F,$10,$FD,$C1,$4F,$78,
   $FE,$00,$20,$04,$79,$B6,$77,$C9,$FE,$01,$20,$05,$79,$2F,$A6,$77,
   $C9,$FE,$02,$20,$04,$79,$AE,$77,$C9,$FE,$03,$C0,$79,$A6,$32,$13,
   $F4,$C9,$01,$6B,$F4,$CD,$5A,$FC,$E9,$00,$C9,$21,$77,$F4,$01,$08,
   $00,$11,$00,$00,$ED,$B0,$C9,$00,$00,$00,$00,$00,$00,$00,$00,$01,
   $88,$F4,$CD,$5A,$FC,$E9,$00,$C9,$21,$00,$00,$01,$08,$00,$11,$77,
   $F4,$ED,$B0,$C9,$01,$9D,$F4,$CD,$5A,$FC,$E9,$00,$C9,$1E,$00,$3E,
   $00,$16,$00,$07,$17,$17,$17,$6F,$7A,$CE,$B6,$67,$7E,$23,$66,$6F,
   $EB,$29,$29,$19,$29,$01,$08,$00,$3E,$00,$FE,$00,$20,$06,$11,$77,
   $F4,$ED,$B0,$C9,$EB,$21,$77,$F4,$ED,$B0,$C9);


procedure GX_Dot; external $F400;

procedure GX_Symb; external $F462;

procedure GX_Get; external $F47F;

procedure GX_ScrAcc; external $F494;

procedure GraphInit;
begin
  GX_MProg:=GX_Data;
end;

procedure Plot (x:integer; y,m:byte);
var  GX_X    : integer absolute $F410;
     GX_Y    : byte    absolute $F412;
     GX_M    : byte    absolute $F413;
begin
  GX_X:=x;
  GX_Y:=y;
  GX_M:=m;
  GX_Dot;
end;


function DotColor (x:integer; y:byte):boolean;
begin
  Plot(x,y,3);
  DotColor:=mem[$F413] <> 0;
end;


procedure Screen_On;
begin
  port[248]:=7;
end;

procedure Screen_Off;
begin
  port[248]:=8;
end;

procedure HideCursor;
begin
 write(#27,'f');
end;

procedure ShowCursor;
begin
  write(#27,'e');
end;


function GX_Convert(var rowpic:GX_String):byte;
var  i,j : byte;
begin
  if length(rowpic)<8 then
    for i:=succ(length(rowpic)) to 8 do rowpic:=rowpic+' ';
  j:=0;
  for i:=1 to 8 do begin
    j:=j shl 1;
    if rowpic[i]='*' then j:=succ(j)
  end;
  GX_Convert:=j
end;


procedure Symbol(Code:byte; rp0,rp1,rp2,rp3,rp4,rp5,rp6,rp7:GX_String);
var  matrix : array[0. 7] of byte absolute $F477;
     CStart : integer absolute $F472;
begin
  matrix[0]:=GX_Convert(rp0);
  matrix[1]:=GX_Convert(rp1);
  matrix[2]:=GX_Convert(rp2);
  matrix[3]:=GX_Convert(rp3);
  matrix[4]:=GX_Convert(rp4);
  matrix[5]:=GX_Convert(rp5);
  matrix[6]:=GX_Convert(rp6);
  matrix[7]:=GX_Convert(rp7);
  CStart:=Code*8+$B800;
  GX_Symb;
end;


procedure Hex_Symbol(Code,r0,r1,r2,r3,r4,r5,r6,r7:byte);
var  matrix : array[0..7] of byte absolute $F477;
     CStart : integer absolute $F472;
begin
  matrix[0]:=r0;
  matrix[1]:=r1;
  matrix[2]:=r2;
  matrix[3]:=r3;
  matrix[4]:=r4;
  matrix[5]:=r5;
  matrix[6]:=r6;
  matrix[7]:=r7;
  CStart:=Code*8+$B800;
  GX_Symb;
end;


ä ------------------ Ende von KERNEL.INC ------------------ ü
```

```
{ ------------------------ SIMPLE.INC ------------------------
  Include-File fuer einfachste Grafikbefehle auf dem JOYCE unter
  Turbo Pascal. Stellt folgende Prozeduren zur Verfuegung:

     procedure S_Horizontal() -> Horizontale Linie
     procedure S_Vertikal()   -> Vertikale Linie
     procedure S_Box()        -> Rechteck zeichnen

  Es wird das Include-File KERNEL.INC benoetigt...

------------------------------------------------------------ }


procedure S_Horizontal (x1,x2:integer; y,m:byte);
var i : integer;
begin
  for i:=x1 to x2 do Plot(i,y,m);
end;

procedure S_Vertikal (x:integer; y1,y2,m:byte);
var i : byte;
begin
  for i:=y1 to y2 do Plot(x,i,m);
end;

procedure S_Box (x:integer; y:byte; xlen:integer; ylen,m:byte);
begin
  S_Horizontal(x,x+xlen,y,m);
  S_Vertikal(x,y,y+ylen,m);
  S_Horizontal(x,x+xlen,y+ylen,m);
  S_Vertikal(x+xlen,y,y+ylen,m);
end;

{ ------------------ Ende von SIMPLE.INC ------------------ }
```

```
(*****************************************************************)
(*                                                               *)
(*       Grafikbibliotheksmodul fuer Joyce                       *)
(*          ( setzt KERNEL.INC voraus )                          *)
(*    Prozeduren:                                                *)
(*      point_swap (VAR a, b : INTEGER);                         *)
(*      line (x1, y1, x2, y2 : INTEGER; modus : BYTE);           *)
(*      box  (x_links, y_oben, x_rechts, y_unten : INTEGER;      *)
(*                    modus : BYTE);                             *)
(*      square (x_oben, y_oben, breite : INTEGER; modus : BYTE); *)
(*      ellipse (x_center, y_center, radius_x, radius_y : INTEGER;*)
(*                    modus : BYTE);                             *)
(*      circle (x_center, y_center, radius: INTEGER; modus : BYTE);*)
(*      plot_marker (x, y : INTEGER;  marker : marker_type;      *)
(*                    modus: BYTE );                             *)
```

```
(*       poly_plot (number_of_points : BYTE; koord : point_array;       *)
(*                                  modus : BYTE);                      *)
(*       poly_marker (number_of_points : BYTE; koord : point_array;     *)
(*                          marker : marker_type;  modus : BYTE);       *)
(*       poly_line (number_of_points : BYTE; koord : point_array;       *)
(*                                  modus : BYTE);                      *)
(*       poly_fill (number_of_points : BYTE; koord : point_array;       *)
(*                                  modus : BYTE);                      *)
(*                                                                      *)
(************************************************************************)

(*        Grafikkonstanten f|r JOYCE              *)

CONST  aspect_ratio = 0.45;

(*        Typendeklarationen fuer Grafik          *)

TYPE   point_array = ARRAY(.0..250,0..1.) of INTEGER;
       marker_type = (kreis, quadrat, filled_quadr, dreieck, kreuz, plus, stern);


(***********          Prozeduren          ************)

procedure point_swap(VAR a, b : INTEGER);

(* vertauscht den Inhalt der Variablen a und b *)

BEGIN
   a:=(a xor b);
   b:=(b xor a);
   a:=(b xor a);
END;



procedure line(x1, y1, x2, y2 : INTEGER; modus : BYTE);

(* zieht Linie zwischen den Punkten x1/y1 und x2/y2
   im entsprechenden Modus                          *)

VAR  x, dx, dy, diff, i1 : INTEGER;
     y, z : BYTE;

BEGIN
  z:=1;
  dx := abs (x1 - x2);
  dy := abs (y1 - y2);
  IF x1<x2 THEN BEGIN
     x:=x1; y:=y1;
     IF y1>y2 THEN z:=-1;
                END

  ELSE BEGIN
   x:=x2; y:=y2;
   IF y2>y1 THEN z:=-1;
END;
plot(x,y,modus);
IF (dx+dy)=0 THEN exit;
IF dx>dy THEN
BEGIN
 (*    horizontale Linie  *)
 IF dy=0 THEN
    FOR i1:=1 TO dx DO
    BEGIN
      x:=succ(x); plot(x, y, modus);
    END
 ELSE
 (*   Steigung < 1        *)
 BEGIN
 diff:= dx shr 1;
   FOR i1:=1 TO dx DO BEGIN
            diff:=diff+dy;
    IF diff>=dx THEN
            BEGIN
            diff:=diff-dx; y:=y+z;
            END;
            x:=succ(x);
    plot (x, y, modus);
    END;
   END;
 END

ELSE
BEGIN
(*   vertikale Linie   *)
IF dx=0 THEN
   FOR i1:=1 TO dy DO
   BEGIN
     y:=y+z; plot(x, y, modus);
   END
ELSE
(*  Steigung >=1   *)
BEGIN
diff:= dy shr 1;
   FOR i1:=1 TO dy DO BEGIN
            diff:=diff+dx;
    IF diff>=dy THEN
            BEGIN
            diff:=diff-dy; x:=succ(x);
            END;
            y:=y+z;
    plot (x, y, modus);
  END;
 END;
END;
END;
```

```
procedure box(x_links, y_oben, x_rechts, y_unten : INTEGER; modus : BYTE);

(* Rechteck mit linker, oberer und rechter, unterer Ecke
   Seiten parallel zu den Bildschirmachsen
   im entsprechenden Modus                          *)

VAR  i : integer;

 BEGIN
 IF x_links > x_rechts THEN point_swap(x_links,x_rechts);
 IF y_oben  > y_unten  THEN point_swap(y_oben,y_unten);

 FOR i:=x_links TO x_rechts DO
     BEGIN
       plot(i,y_oben,modus);
       plot(i,y_unten,modus);
     END;

 FOR i:=y_oben TO y_unten DO
     BEGIN
       plot(x_links,i,modus);
       plot(x_rechts,i,modus);
     END;

 END;



 procedure square(x_oben, y_oben, breite : INTEGER; modus : BYTE);

 (* Quadrat mit linker, oberer Ecke und Breite in Pixeln
    mit Hilfe der Konstanten aspect_ratio aus p box()
    auf gleiche Seitenlaengen umgerechnet           *)

 VAR y_unten, x_unten : INTEGER;

 BEGIN

 x_unten:=breite + x_oben;
 y_unten:=round(breite*aspect_ratio + y_oben);

 box(x_oben,y_oben, x_unten,y_unten, modus);

 END;



 procedure ellipse(x_center, y_center, radius_x, radius_y : INTEGER; modus : BYTE);

 (* Ellipse mit Mittelpunkt und x- bzw. y-Halbachse im entsprechenden Modus
    mit Hilfe der Konstanten aspect_ratio wird das Achsenverhaeltnis gewahrt
    Achsen parallel zu den Bildschirmachsen                                 *)

 VAR  Diff, AspRX, AspRY, x, y : INTEGER;
      ax_ratio                 : REAL;

 BEGIN
   x:=0; y:=radius_x; Diff:=pred(radius_x);
   ax_ratio:=aspect_ratio*radius_y/radius_x;
   REPEAT
    IF Diff < 0 THEN
    BEGIN
     y:=pred(y); Diff:=Diff+y+y;
    END;
     AspRX:=round(ax_ratio*x);
     AspRY:=round(ax_ratio*y);
     plot(x_center+x,y_center+AspRY,modus);
     plot(x_center-x,y_center+AspRY,modus);
     plot(x_center+x,y_center-AspRY,modus);
     plot(x_center-x,y_center-AspRY,modus);
     plot(x_center+y,y_center+AspRX,modus);
     plot(x_center-y,y_center+AspRX,modus);
     plot(x_center+y,y_center-AspRX,modus);
     plot(x_center-y,y_center-AspRX,modus);
     Diff:=Diff-x-x-1;
     x:=succ(x);
   UNTIL (x > y);
 END;

 procedure circle(x_center, y_center, radius: INTEGER; modus : BYTE);

 (* Kreis mit Mittelpunkt und Radius, berechnet als Ellipse mit gleich
    langen Halbachsen im entsprechenden Modus                          *)

 BEGIN
   ellipse(x_center, y_center, radius, radius, modus);
 END;


 procedure plot_marker(x, y : INTEGER;  marker : marker_type; modus: BYTE );

 (* setzt einen 'Marker' auf den Punkt mit den Koordinaten x und y
    der Marker muss in Marker_Type (Kreis, Quadrat, gefuelltes Quadr.,
    Dreieck, Kreuz, Plus, Stern) enthalten sein
    im jeweiligen Modus; die Groesse wird als Konstante marker_height
    festgelegt, bei Kreis, Kreuz und Stern wurden Korrekturen eingefuehrt,
    um das Aussehen zu verbessern                                       *)

 CONST marker_height = 4;
 VAR   hoehe, i      : BYTE;

 BEGIN
   hoehe:=Round(aspect_ratio*marker_height);
   CASE marker of
             kreis :  circle(x, y, marker_height+1, modus);
           quadrat :  square(x-marker_height, y-hoehe, marker_height shl 1, modus);
       filled_quadr :  FOR i:=y-hoehe TO y+hoehe DO
                         line(x-marker_height, i, x+marker_height, i, modus);
            dreieck :  BEGIN
                         line(x-marker_height, y-hoehe, x+marker_height, y-hoehe, modus);
                         line(x-marker_height, y-hoehe, x, y+hoehe, modus);
```

```
                    line(x+marker_height, y-hoehe, x, y+hoehe, modus);
                  END;
          kreuz :  BEGIN
                    line(x-marker_height-1, y-hoehe,
                         x+marker_height, y+hoehe, modus);
                    line(x-marker_height-1, y+hoehe,
                         x+marker_height, y-hoehe, modus);
                  END;
           plus :  BEGIN
                    line(x-marker_height, y, x+marker_height, y, modus);
                    line(x, y+hoehe, x, y-hoehe, modus);
                  END;
          stern :  BEGIN
                    line(x-marker_height-1, y-hoehe,
                         x+marker_height, y+hoehe, modus);
                    line(x-marker_height-1, y+hoehe,
                         x+marker_height, y-hoehe, modus);
                    line(x-marker_height-1, y, x+marker_height, y, modus);
                    line(x, y+hoehe, x, y-hoehe, modus);
                  END;

        END;
END;


procedure poly_plot(number_of_points : BYTE; koord : point_array; modus : BYTE);

(* plottet ein zweidimensionales array mit 'number_of_points' x- und y-Werten
     im entsprechenden Modus                                              *)

        VAR i : BYTE;

        BEGIN
        FOR i:=0 TO pred(number_of_points) DO
            plot(koord(.i,0.),koord(.i,1.),modus);
        END;


        procedure poly_marker(number_of_points : BYTE; koord : point_array;
                              marker : marker_type;  modus : BYTE);

        (* setzt auf die im zweidimensionalen array koord  x/y-Werte
           (Anzahl : number_of_points) die bezeichneten Marker
           im entsprechenden Modus                                        *)

        VAR i : BYTE;

        BEGIN
        FOR i:=0 TO pred(number_of_points) DO
            plot_marker(koord(.i,0.),koord(.i,1.), marker, modus);
        END;



        procedure poly_line(number_of_points : BYTE; koord : point_array; modus : BYTE);

        (* verbindet die im zweidimensionalen array koord enthaltenen x/y-Werte
           (Anzahl : number_of_points)
           in aufsteigender Reihenfolge mit Linien im entsprechenden Modus      *)

        VAR i : BYTE;
            x_start, y_start :INTEGER;

        BEGIN
        x_start:=koord(.0,0.) ; y_start:=koord(.0,1.);

        FOR i:=1 TO pred(number_of_points) DO
            BEGIN
            line(x_start,y_start,koord(.i,0.),koord(.i,1.),modus);
            x_start:=koord(.i,0.) ; y_start:=koord(.i,1.);
            END;
        END;



        procedure poly_fill(number_of_points : BYTE; koord : point_array; modus : BYTE);

        (* fuellt die Flaeche, deren Eckpunkte im zweidimensionalen array koord
           uebergeben werden   (Anzahl : number_of_points), im entsprechenden Modus  *)

        VAR  Status         : ARRAY (.0..247.) of BYTE;
             Fill_Array     : ARRAY (.0..247,0..1.) of INTEGER;
             y_start,
             y_min, y_max   : BYTE;
             i, i1, x_start : INTEGER;


        procedure find_x(x1, y1, x2, y2 : INTEGER);

        VAR  x, dx, dy, dz, i1 : INTEGER;
             y, z : BYTE;

        BEGIN
          z:=1;
          dx := abs (x1 - x2);
          dy := abs (y1 - y2);
          IF x1<x2 THEN BEGIN
            x:=x1; y:=y1;
            IF y1>y2 THEN z:=-1;
                     END
          ELSE BEGIN
            x:=x2; y:=y2;
            IF y2>y1 THEN z:=-1;
          END;
          IF Status(.y.)<2 THEN
            IF x<>Fill_array(.y,0.)  THEN
                  BEGIN
                    Fill_array(.y,Status(.y.).):=x;
                    Status(.y.):=succ(Status(.y.));
                  END;
          IF (dx+dy)=0 THEN exit;

          IF dx>dy THEN
          BEGIN
           (*   horizontale Linie  *)
           IF dy=0 THEN
              FOR i1:=1 TO dx DO
              BEGIN
                x:=succ(x);
              END
           ELSE
           (*  Steigung < 1        *)
           BEGIN
           dz:= dx shr 1;
             FOR i1:=1 TO dx DO BEGIN
                       dz:=dz+dy;
              IF dz>=dx THEN
                       BEGIN
                       dz:=dz-dx; y:=y+z;
                       IF Status(.y.)<2 THEN
                         IF succ(x)<>Fill_array(.y,0.)  THEN
                          BEGIN
                          Fill_array(.y,Status(.y.).):=succ(x);
                          Status(.y.):=succ(Status(.y.));
                          END;
                       END;
                       x:=succ(x);
             END;
            END;
           END

          ELSE
          BEGIN
          (*   vertikale Linie   *)
          IF dx=0 THEN
             FOR i1:=1 TO dy DO
             BEGIN
               y:=y+z;
               IF Status(.y.)<2 THEN
                IF x<>Fill_array(.y,0.)  THEN
                          BEGIN
                          Fill_array(.y,Status(.y.).):=x;
                          Status(.y.):=succ(Status(.y.));
                          END;
             END
          ELSE
          (*  Steigung >=1   *)
          BEGIN
          dz:= dy shr 1;
             FOR i1:=1 TO dy DO BEGIN
                       dz:=dz+dx;
            IF dz>=dy THEN
                      BEGIN
                      dz:=dz-dy; x:=succ(x);
                      END;
                      y:=y+z;
                      IF Status(.y.)<2 THEN
                       IF x<>Fill_array(.y,0.)  THEN
                         BEGIN
                         Fill_array(.y,Status(.y.).):=x;
                         Status(.y.):=succ(Status(.y.));
                         END;
           END;
          END;
         END;
        END;



        BEGIN
        (*       Max/Min  fuer y-Werte             *)
        y_min:=koord(.0,1.); y_max:=koord(.0,1.);
        FOR i:=0 TO pred(number_of_points) DO
            BEGIN
            IF Koord(.i,1.)<y_min THEN y_min:=koord(.i,1.);
            IF koord(.i,1.)>y_max THEN y_max:=koord(.i,1.);
            END;

        (*       arrays auf Null setzen       *)
        FOR i:=0 TO 247 DO
           BEGIN
            Status(.i.):=0;
            Fill_array(.i,0.):=0;
            Fill_array(.i,1.):=0;
           END;

        (*       Grenzen ermitteln                *)
        x_start:=koord(.0,0.) ; y_start:=koord(.0,1.);

        FOR i:=1 TO pred(number_of_points) DO
            BEGIN
            find_x(x_start,y_start,koord(.i,0.),koord(.i,1.));
            x_start:=koord(.i,0.) ; y_start:=koord(.i,1.);
            END;



        (*      Fuellen              *)
        FOR i:=succ(y_min) TO pred(y_max) DO
            BEGIN
            IF Status(.i.)=2 THEN
            BEGIN
              IF Fill_array(.i,0.)>Fill_array(.i,1.) THEN
                                   point_swap(Fill_array(.i,0.),Fill_array(.i,1.));
              FOR i1:=Fill_array(.i,0.) TO Fill_array(.i,1.) DO plot(i1,i,modus);
            END;
          END;
        END;



        (*             Ende  von GRAPHLIB.INC              *)
```

```
(*********************************************************)
(*                                                       *)
(*  Erweiterte Prozeduren: Line_Pattern()                *)
(*                         Fill_Pattern()                *)
(*  Benutzen die vordefinierten Muster zum Zeichnen      *)
(*  der Linie (Line_Style als Konstanten), zum Fuellen   *)
(*  die undefinierten Zeichen (Fill_Pattern als BYTE).   *)
(*  Fuellmuster muessen mit PatternInit initialisiert    *)
(*  werden.                                              *)
(*  Beide Prozeduren benoetigen das Include-File         *)
(*  KERNEL.INC und Teile von GRAPHLIB.INC                *)
(*                                                       *)
(*********************************************************)

(*  Linien mit verschiedenen Mustern  *)

CONST      Solid =  '****************';
         Dashed =  '****    ****    ';
       L_Dashed =  '********        ';
         Dotted =  '**  **  **  **  ';
   Dashed_Dotted =  '***   **   ***';

TYPE  line_style = String(.16.);


procedure pattern_line(x1, y1, x2, y2 : INTEGER; pattern : line_style; modus : BYTE);
(* zieht Linie zwischen den Punkten x1/y1 und x2/y2
   im entsprechenden Modus mit in Pattern angegebenem Line_Style                *)

VAR  x, dx, dy, diff,
     i1, Raster       : INTEGER;
     y, z             : BYTE;
     m0,m1            : GX_String;

BEGIN
m0:=copy(pattern,1,8);
m1:=copy(pattern,9,8);
Raster:=GX_Convert(m1)+(GX_Convert(m0) shl 8);
  z:=1;
  dx := abs (x1 - x2);
  dy := abs (y1 - y2);
  IF x1<x2 THEN BEGIN
     x:=x1; y:=y1;
     IF y1>y2 THEN z:=-1;
              END
    ELSE BEGIN
     x:=x2; y:=y2;
     IF y2>y1 THEN z:=-1;
  END;
  plot(x,y,modus);
  IF (dx+dy)=0 THEN exit;
  IF dx>dy THEN
  BEGIN
   (*    horizontale Linie  *)
   IF dy=0 THEN
      FOR i1:=1 TO dx DO
      BEGIN
        x:=succ(x);
        IF (Raster and (1 shl (15-(i1 Mod 16))))<>0 THEN plot(x, y, modus);
      END
   ELSE
   (*   Steigung < 1        *)
   BEGIN
   diff:= dx shr 1;
     FOR i1:=1 TO dx DO BEGIN
               diff:=diff+dy;

      IF diff>=dx THEN
               BEGIN
               diff:=diff-dx; y:=y+z;
               END;
               x:=succ(x);
       IF (Raster and (1 shl (15-(i1 Mod 16))))<>0 THEN plot(x, y, modus);
      END;
     END;
  END

  ELSE
  BEGIN
  (*   vertikale Linie   *)
  IF dx=0 THEN
     FOR i1:=1 TO dy DO
     BEGIN
       y:=y+z;
       IF (Raster and (1 shl (15-(i1 Mod 16))))<>0 THEN plot(x, y, modus);
     END
  ELSE
  (*  Steigung >=1   *)
  BEGIN
  diff:= dy shr 1;
     FOR i1:=1 TO dy DO BEGIN
               diff:=diff+dx;
      IF diff>=dy THEN
               BEGIN
               diff:=diff-dy; x:=succ(x);
               END;
               y:=y+z;
      IF (Raster and (1 shl (15-(i1 Mod 16))))<>0 THEN plot(x, y, modus);
     END;
  END;
 END;
END;
(* Ende von procedure pattern_line() *)


(* Fuellen mit vordefiniertem Muster *)

procedure pattern_fill(number_of_points : BYTE; koord : point_array; fill_pattern : BYTE);

VAR  Status     : ARRAY (.0..247.) of BYTE;
     Fill_Array : ARRAY (.0..247,0..1.) of INTEGER;
     y_start,
     y_min,
     y_max      : BYTE;
     i, i1,
     x_start    : INTEGER;
     matrix     : ARRAY (.0..7.) of BYTE absolute $F477;
     CStart     : INTEGER absolute $F489;
     ZStart     : INTEGER absolute $F472;

  procedure find_x(x1, y1, x2, y2 : INTEGER);

  VAR  x, dx, dy, dz, i1 : INTEGER;
       y, z : BYTE;

  BEGIN
    z:=1;
    dx := abs (x1 - x2);
    dy := abs (y1 - y2);
    IF x1<x2 THEN BEGIN
       x:=x1; y:=y1;
       IF y1>y2 THEN z:=-1;
                END



      ELSE BEGIN
       x:=x2; y:=y2;
       IF y2>y1 THEN z:=-1;
    END;
    IF Status(.y.)<2 THEN
      IF x<>Fill_array(.y,0.)  THEN
            BEGIN
              Fill_array(.y,Status(.y.).):=x;
              Status(.y.):=succ(Status(.y.));
            END;
    IF (dx+dy)=0 THEN exit;
    IF dx>dy THEN
    BEGIN
     (*    horizontale Linie  *)
     IF dy=0 THEN
        FOR i1:=1 TO dx DO
        BEGIN
          x:=succ(x);
        END
     ELSE
     (*   Steigung < 1        *)
     BEGIN
     dz:= dx shr 1;
       FOR i1:=1 TO dx DO BEGIN
                 dz:=dz+dy;
        IF dz>=dx THEN
                 BEGIN
                 dz:=dz-dx; y:=y+z;
                 IF Status(.y.)<2 THEN
                   IF succ(x)<>Fill_array(.y,0.)  THEN
                    BEGIN
                    Fill_array(.y,Status(.y.).):=succ(x);
                    Status(.y.):=succ(Status(.y.));
                    END;
                 END;
                 x:=succ(x);
        END;
       END;
     END

     ELSE
     BEGIN
     (*   vertikale Linie   *)
     IF dx=0 THEN
        FOR i1:=1 TO dy DO
        BEGIN
          y:=y+z;
          IF Status(.y.)<2 THEN
           IF x<>Fill_array(.y,0.)  THEN
                    BEGIN
                    Fill_array(.y,Status(.y.).):=x;
                    Status(.y.):=succ(Status(.y.));
                    END;
       END
     ELSE
     (*  Steigung >=1   *)
     BEGIN
     dz:= dy shr 1;
        FOR i1:=1 TO dy DO BEGIN
                 dz:=dz+dx;
        IF dz>=dy THEN
                 BEGIN
                 dz:=dz-dy; x:=succ(x);
                 END;
                 y:=y+z;
                 IF Status(.y.)<2 THEN
                 IF x<>Fill_array(.y,0.)  THEN
                   BEGIN
                   Fill_array(.y,Status(.y.).):=x;
                   Status(.y.):=succ(Status(.y.));
                   END;
       END;
     END;
   END;
  END;

BEGIN
(*       Max/Min  fuer y-Werte              *)
y_min:=koord(.0,1.); y_max:=koord(.0,1.);
FOR i:=0 TO pred(number_of_points) DO
    BEGIN
    IF Koord(.i,1.)<y_min THEN y_min:=koord(.i,1.);
    IF koord(.i,1.)>y_max THEN y_max:=koord(.i,1.);
    END;
```

```
(*       arrays auf Null setzen       *)
FOR i:=0 TO 247 DO
   BEGIN
    Status(.i.):=0;
    Fill_array(.i,0.):=0;
    Fill_array(.i,1.):=0;
   END;

(*       Grenzen ermitteln                *)
x_start:=koord(.0,0.) ; y_start:=koord(.0,1.);

FOR i:=1 TO pred(number_of_points) DO
    BEGIN
    find_x(x_start,y_start,koord(.i,0.),koord(.i,1.));
    x_start:=koord(.i,0.) ; y_start:=koord(.i,1.);
    END;


(*       Muster holen             *)
    CStart:=(fill_pattern*8)+$B800;
    GX_Get;


(*      Fuellen mit Muster     *)
FOR i:=succ(y_min) TO pred(y_max) DO
    BEGIN
    IF Status(.i.)=2 THEN
    BEGIN
      IF Fill_array(.i,0.)>Fill_array(.i,1.) THEN
                          point_swap(Fill_array(.i,0.),Fill_array(.i,1.));
      FOR i1:=Fill_array(.i,0.) TO Fill_array(.i,1.) DO
          IF (matrix(.(i MOD 8).) and (1 shl (7- (i1 MOD 8)))) <> 0
             THEN plot(i1,i,0)
             ELSE plot(i1,i,1);
    END;
  END;
END;
(* Ende von procedure pattern_fill() *)


procedure FillPatternInit;

(* legt von 248 bis 255 neue Zeichen im Zeichensatz ab als Fuellmuster *)
BEGIN
  Symbol (248, '**  **  ',
               '**  **  ',
               '  **  **',
               '  **  **',
               '**  **  ',
               '**  **  ',
               '  **  **',
               '  **  **');
  Symbol (249, '*       ',
               ' *      ',
               '  *     ',
               '   *    ',
               '    *   ',
               '     *  ',
               '      * ',
               '       *');
  Symbol (250, '       *',
               '      * ',
               '     *  ',
               '    *   ',
               '   *    ',
               '  *     ',
               ' *      ',
               '*       ');

  Symbol (251, '*     *',
               ' *   * ',
               '  * *  ',
               '   *   ',
               '   *   ',
               '  * *  ',
               ' *   * ',
               '*     *');
  Symbol (252, '* * * * ',
               '        ',
               '* * * * ',
               '        ',
               '* * * * ',
               '        ',
               '* * * * ',
               '        ');
  Symbol (253, '* * * * ',
               ' * * * *',
               '* * * * ',
               ' * * * *',
               '* * * * ',
               ' * * * *',
               '* * * * ',
               ' * * * *');
  Symbol (254, '********',
               ' * * * *',
               '********',
               ' * * * *',
               '********',
               ' * * * *',
               '********',
               ' * * * *');
  Symbol (255, '********',
               '********',
               '********',
               '********',
               '********',
               '********',
               '********',
               '********');
END;

(*******************     Ende von PATTERNS.INC     ************************)
```



```
(*********************************************************************)
(*                                                                   *)
(*       Erweiterte Grafikprozedur  plot_point_array()               *)
(*       ( benoetigt KERNEL.INC und GRAPHLIB.INC )                   *)
(*       Uebergeben wird ein Array mit Punkten und deren Anzahl      *)
(*       Die view_ -Parameter legen den Bildschirmausschnitt fest,   *)
(*       auf den das Koordinatensystem gezeichnet wird (0..719,0..247)  *)
(*       Die win_ -Parameter legen die Benutzerkoordinaten der Achsen   *)
(*       fest, x_Teil und y_Teil die Anzahl der Teilstriche der Achsen. *)
(*                                                                   *)
(*********************************************************************)


procedure plot_point_array(Feld : point_array; Anz_Pkte, view_x_min,
                           view_x_max, view_y_min, view_y_max, win_x_min, win_x_max,
                           win_y_min, win_y_max : integer; x_Teil, y_Teil, modus : byte);

VAR  Diff_x, Diff_y,
     x_Start, y_Start, x, y,
     Zaehler, Breite, i       : INTEGER;
     Pxf, Pyf                 : REAL;


procedure koord_trans(VAR x_koord, y_koord : integer);

BEGIN
  x_koord:=round((x_koord - win_x_min)*((view_x_max - view_x_min)/(win_x_max - win_x_min)))
                    + view_x_min;
  y_koord:=view_y_max - ROUND((y_koord - win_y_min)*((view_y_max - view_y_min)/
                     (win_y_max - win_y_min)));
END;

BEGIN
(*   Koordinatensystem zeichnen *)
(*   x-Achse  *)
line(view_x_min, view_y_max+5, view_x_max, view_y_max+5, modus);
Breite:=Round((view_x_max - view_x_min)/x_Teil);
Zaehler:=0;
REPEAT
  Line(view_x_min+Zaehler, view_y_max+5, view_x_min+Zaehler, view_y_max+9, modus);
  zaehler:=Zaehler+Breite;
UNTIL Zaehler>(view_x_max - view_x_min);

(*   y-Achse  *)
line(view_x_min-9, view_y_min,  view_x_min-9, view_y_max, modus);
Breite:=Round((view_y_max - view_y_min)/y_Teil);
Zaehler:=0;
REPEAT
  Line(view_x_min-9, view_y_max-Zaehler, view_x_min-16, view_y_max-Zaehler, modus);
  zaehler:=Zaehler+Breite;
UNTIL Zaehler>(view_y_max - view_y_min);

(*   Array zeichnen   *)
  x_start:=Feld(.0,0.);
  y_start:=Feld(.0,1.);
  koord_trans(x_start,y_start);
  FOR i:=1 TO pred(Anz_Pkte) DO
      BEGIN
        x:=Feld(.i,0.); y:=Feld(.i,1.);
        koord_trans(x,y);
        line(x_start, y_start, x, y, modus);
        x_start:=x; y_start:=y;
      END;
END;

(*****************     Ende von GRAPH.PRO      *************************)
```



```
   (* ---------------- PRINTAT.INC ---------------------- *)
   (* Include-File fuer PRINTAT(x,y,font,heading,text)   *)
   (* Benoetigt Include-File KERNEL.INC...               *)


   procedure printat (x, y, font, heading: integer; text: Strg_90);

   var    l       : integer;
          k, i, j : byte;
          Matrix  : array(.0..7.) of byte absolute $F477;
          Raster  : array(.0..7.) of byte;
          CStart  : integer absolute $F489;
```

```
        ZStart  : integer absolute $F472;
        Zeichen : char;


    procedure Get_Matrix;
    (* uebertraegt array Matrix in Raster und setzt Matrix auf Null *)
    BEGIN
      for k:=0 to 7 do
      BEGIN
        Raster(.k.):=Matrix(.k.);
        Matrix(.k.):=0;
      END;
    END;


    BEGIN          (*  Hauptteil  *)
      for l:=1 to length(text) do   (* Text Zeichenweise umwandeln und Schreiben *)
      BEGIN                         (* Zeichen holen *)
        Zeichen:=text(.l.);
        CStart:=(ord(Zeichen)*8)+$B800;
        GX_Get;


      case font of

        1 : for i:=0 to 7 do     (* schmal *)
            BEGIN
             Raster(.i.):=Matrix(.i.) shr 1;
             Matrix(.i.):=Matrix(.i.) and Raster(.i.);
            END;

        2 : for i:=0 to 7 do     (* fett *)
            BEGIN
             Raster(.i.):=Matrix(.i.) shr 1;
             Matrix(.i.):=Matrix(.i.) or Raster(.i.);
            END;

        3 : for i:=0 to 7 do Matrix(.i.):=not Matrix(.i.);  (* invers *)

        4 : BEGIN                           (* kursiv *)
            Matrix(.0.):=Matrix(.0.) shr 2;
            Matrix(.1.):=Matrix(.1.) shr 1;
            Matrix(.2.):=Matrix(.2.) shr 1;
            Matrix(.5.):=Matrix(.5.) shl 1;
            Matrix(.6.):=Matrix(.6.) shl 1;
            Matrix(.7.):=Matrix(.7.) shl 2;
           END;
      END;



      case heading of               (* Raster drehen und auf Matrix *)
                                    (* zurueckschreiben             *)

        1 : BEGIN
            Get_Matrix;         (* Zeilen von oben nach unten   *)
            for i:=0 to 7 do    (* und von hinten nach vorne    *)
                for j:=0 to 7 do
                if (Raster(.j.) and (1 shl (7-i)))<>0 then
                                    Matrix(.i.):=Matrix(.i.) or (1 shl j);
            END;

        2 : BEGIN
            Get_Matrix;         (* Zeilen von unten nach oben   *)
            for i:=0 to 7 do    (* und von hinten nach vorne    *)
                for j:=0 to 7 do
                if (Raster(.7-i.) and (1 shl (7-j)))<>0 then
                                    Matrix(.i.):=Matrix(.i.) or (1 shl j);
            END;

        3 : BEGIN
            Get_Matrix;         (* Zeilen von unten nach oben   *)
            for i:=0 to 7 do    (* und von vorne nach hinten    *)
                for j:=0 to 7 do
                if (Raster(.7-j.) and (1 shl i))<>0 then
                                    Matrix(.i.):=Matrix(.i.) or (1 shl j);
            END;
      END;

    ZStart:=$B800;
    GX_Symb;
    gotoxy(x,y); write(#27,#0);
      case heading of
           0 : x:=succ(x);
           1 : y:=succ(y);
           2 : x:=pred(x);
           3 : y:=pred(y);
      END;
     END;
    END;


    (* ------------------- Ende von PRINTAT.INC ------------------- *)
```

```
    (* ------------------- SPRINTAT.INC ------------------- *)
    (* Include-File fuer SPRINTAT(x,y,font,heading,text)   *)
    (* schreibt in Spiegelschrift                          *)
    (* Benoetigt das Include-File KERNEL.INC...            *)


    procedure sprintat (x, y, font, heading: integer; text: Strg_90);

    var   l      : integer;
          k, i, j : byte;
          Matrix  : array(.0..7.) of byte absolute $F477;
          Raster  : array(.0..7.) of byte;
          CStart  : integer absolute $F489;
          ZStart  : integer absolute $F472;
          Zeichen : char;

    procedure Get_Matrix;

    BEGIN
      for k:=0 to 7 do
      BEGIN
        Raster(.k.):=Matrix(.k.);
        Matrix(.k.):=0;
      END;
    END;

    BEGIN
      for l:=1 to length(text) do
      BEGIN
        Zeichen:=text(.l.);
        CStart:=(ord(Zeichen)*8)+$B800;
        GX_Get;


      case font of

        1 : for i:=0 to 7 do     (* schmal *)
            BEGIN
             Raster(.i.):=Matrix(.i.) shr 1;
             Matrix(.i.):=Matrix(.i.) and Raster(.i.);
            END;

        2 : for i:=0 to 7 do     (* fett *)
            BEGIN
             Raster(.i.):=Matrix(.i.) shr 1;
             Matrix(.i.):=Matrix(.i.) or Raster(.i.);
            END;

        3 : for i:=0 to 7 do Matrix(.i.):=not Matrix(.i.);  (* invers *)

        4 : BEGIN                           (* kursiv *)
            Matrix(.0.):=Matrix(.0.) shr 2;
            Matrix(.1.):=Matrix(.1.) shr 1;
            Matrix(.2.):=Matrix(.2.) shr 1;
            Matrix(.5.):=Matrix(.5.) shl 1;
            Matrix(.6.):=Matrix(.6.) shl 1;
            Matrix(.7.):=Matrix(.7.) shl 2;
           END;
      END;


      case heading of

          0 : BEGIN
              Get_Matrix;
              for i:=0 to 7 do
              for j:=0 to 7 do
              if (Raster(.i.) and (1 shl (7-j)))<>0 then
                                  Matrix(.i.):=Matrix(.i.) or (1 shl j);
              END;

          1 : BEGIN
              Get_Matrix;
              for i:=0 to 7 do
                  for j:=0 to 7 do
                  if (Raster(.j.) and (1 shl i))<>0 then
                                  Matrix(.i.):=Matrix(.i.) or (1 shl j);
              END;

          2 : BEGIN
              Get_Matrix;
              for i:=0 to 7 do
                  for j:=0 to 7 do
                  if (Raster(.7-i.) and (1 shl j))<>0 then
                                  Matrix(.i.):=Matrix(.i.) or (1 shl j);
              END;

          3 : BEGIN
              Get_Matrix;
              for i:=0 to 7 do
                  for j:=0 to 7 do
                  if (Raster(.7-j.) and (1 shl (7-i)))<>0 then
                                  Matrix(.i.):=Matrix(.i.) or (1 shl j);
              END;
      END;

    ZStart:=$B800;
    GX_Symb;

      case heading of
           0 : BEGIN
               gotoxy(x+(length(Text)-pred(l)),y); write(#27,#0);
               END;
           1 : BEGIN
               gotoxy(x,y+(length(Text)-pred(l))); write(#27,#0)
               END;
           2 : BEGIN
               gotoxy(x-(length(Text)-pred(l)),y); write(#27,#0)
               END;
           3 : BEGIN
```

```
            gotoxy(x,y-(length(Text)-pred(l))); write(#27,#0)
          END;
  END;
 END;
END;

(* ---------------------- Ende von SPRINTAT.INC ---------------------- *)
```

```
| ---------------------- HARDCOPY.INC ----------------------
  Include-File fuer Hardcopies auf dem JOYCE unter Turbo Pascal.
  Ermoeglicht die Ausgabe von Bildschirmbereichen in vier Groesen
  und zwei Richtungen auf dem JOYCE-Drucker.

  Benoetigt das Include-File KERNEL.INC...

------------------------------------------------------------------ |


procedure Hardcopy (left, top, right, bottom, size : INTEGER; direction : CHAR);

const  breite = 719;  (* Bildschirmbreite *)
       hoehe  = 248;  (* Bildschirmhoehe  *)
       ptrres = 960;  (* Breite einer Druckzeile, doppelte Aufloesung *)

zweih1  : array (.0..7.) of INTEGER = (128,64,32,16,8,4,2,1);
zweih2  : array (.0..3.) of INTEGER = (192,48,12,3);
zweih4  : array (.0..1.) of INTEGER = (240,15);
zweih8  : array (.0..0.) of INTEGER = (255);

var   bitnumber, i, j, decrement, y, n, spalte  : INTEGER;


function params_ok: BOOLEAN;

BEGIN
  if (left < 0) or (right > breite) or (left > right)
  or (top < 0) or (bottom > hoehe) or (top > bottom)
  or not(size in (.1, 2, 4, 8.))
  or (bitnumber > ptrres) then
     params_ok:= false else params_ok:=true;
END;

BEGIN
if not (direction in (.'v','V'.))
       then direction := 'h';
case direction of

'H','h' :
BEGIN
  bitnumber := size * (right - left +1);
  if params_ok then
  BEGIN
    y:=top;
    decrement := 8 Div size;
    WRITE (Lst, chr(27), chr(65), chr(8));       (* Grafikzeilenabstand  8/72 *)
    while y <=bottom do
    BEGIN
      WRITE (Lst, chr(27), chr(76), chr(Lo(bitnumber)), chr(hi(bitnumber)));
            (* Anzahl der folgenden Grafikdaten, doppelte Aufloesung *)
      for i:= left to right do
      BEGIN
        spalte := 0;
        for n := y to pred(y+decrement) do
            if n <= bottom then
               if DotColor(i, n) then
                  case size of
                    1: spalte := spalte or zweih1(.n-y.);
                    2: spalte := spalte or zweih2(.n-y.);
                    4: spalte := spalte or zweih4(.n-y.);
                    8: spalte := spalte or zweih8(.n-y.);
                  END;
            for j := 1 to size do
              WRITE(Lst, chr(spalte));          (* bits senden *)
            END;
            WRITELN(Lst);
            y := y + decrement;
        END;
        WRITE(Lst, chr(27), chr(50));           (* normaler Zeilenabstand 1/6 *)
      END;
    END;


    'v','V':
    BEGIN
      bitnumber := 2 * size * (bottom - top +1);
      if params_ok then
      BEGIN
        y:=left;
        decrement := 8 Div  size;
        WRITE (Lst, chr(27), chr(65), chr(8));       (* Grafikzeilenabstand  8/72 *)
        while y <=right do
        BEGIN
          WRITE (Lst, chr(27), chr(76), chr(Lo(bitnumber)), chr(hi(bitnumber)));
                (* Anzahl der folgenden Grafikdaten, doppelte Aufloesung *)
          for i:= bottom downto top do
          BEGIN
```

```
          spalte := 0;
          for n := y to pred(y+decrement) do
              if n <= right then
                 if DotColor(n,i) then
                    case size of
                      1: spalte := spalte or zweih1(.n-y.);
                      2: spalte := spalte or zweih2(.n-y.);
                      4: spalte := spalte or zweih4(.n-y.);
                      8: spalte := spalte or zweih8(.n-y.);
                    END;
              for j := 1 to 2 * size do
                WRITE(Lst, chr(spalte));          (* bits senden *)
            END;
            WRITELN(Lst);
            y := y + decrement;
          END;
          WRITE(Lst, chr(27), chr(50));           (* normaler Zeilenabstand 1/6 *)
        END;
      END; (* hgrcopy*)

end;
end;

| ---------------------- Ende von HARDCOPY.INC ---------------------- |
```

```
| ---------------------- DISKETTE.INC ----------------------

  Include-File fuer Diskettenoperationen.
  Stellt folgende Funktionen zur Verfuegung:

    procedure SaveScreen() -> Bild speichern
    procedure LoadScreen() -> Bild laden

  Benoetigt Include-File KERNEL.INC...

  ------------------------------------------------------------ |


var GX_Buffer : array[0..22319] of byte;


procedure SaveScreen (Dateiname: Strg_14);
var
 i, j    : integer;
 Row     : byte absolute $F4A0;
 Col     : byte absolute $F49E;
 AccMode : byte absolute $F4B9;
 Matrix  : array[0..7] of byte absolute $F477;
 Datei   : file;
begin
  j:=0;
  AccMode:=0;
  for Row:=0 to 30 do
   begin
    for Col:=0 to 89 do
      begin
        GX_ScrAcc;
        for i:=0 to 7 do
          begin
            GX_Buffer[j]:=Matrix[i];
            j:=succ(j);
          end;
      end;
   end;
  assign(Datei,Dateiname);
  rewrite(Datei);
  blockwrite(Datei,GX_Buffer,175);
  close(Datei);
end;


procedure LoadScreen (Dateiname: Strg_14);
var
 i, j    : integer;
 Row     : byte absolute $F4A0;
 Col     : byte absolute $F49E;
 AccMode : byte absolute $F4B9;
 Matrix  : array[0..7] of byte absolute $F477;
 Datei   : file;
begin
  j:=0;
  AccMode:=1;
  assign(Datei,Dateiname);
  reset(Datei);
  blockread(Datei,GX_Buffer,175);
  close(Datei);
  for Row:=0 to 30 do
   begin
    for Col:=0 to 89 do
      begin
        for i:=0 to 7 do
```

```
          begin
            Matrix[i]:=GX_Buffer[j];
            j:=succ(j);
          end;
        GX_ScrAcc;
      end;
    end;
end;

{ ---------- Ende von DISKETTE.INC ---------- }
```

```
{ ---------- FONT.INC ----------

  Include-File zur Bearbeitung von Zeichensatzen.
  Stellt folgende Funktionen zur Verfuegung:

    procedure LoadFont() -> Zeichensatz laden
    procedure SaveFont() -> Zeichensatz speichern

  Benoetigt das Include-File KERNEL.INC...

  ---------------------------------------------- }

var GX_CharMat : array[0..2047] of byte;

procedure LoadFont (Dateiname: Strg_14);
var
 Datei   : file;
 Zeichen : byte;
 Start   : integer;
begin
  assign (Datei,Dateiname);
  reset(Datei);
  blockread(Datei,GX_CharMat,16);
  close(Datei);
  for Zeichen :=0 to 255 do
    begin
      Start:=Zeichen*8;
      Hex_Symbol (Zeichen,
                  GX_CharMat[Start+0],
                  GX_CharMat[Start+1],
                  GX_CharMat[Start+2],
                  GX_CharMat[Start+3],
                  GX_CharMat[Start+4],
                  GX_CharMat[Start+5],
                  GX_CharMat[Start+6],
                  GX_CharMat[Start+7]);
    end;
end;


procedure SaveFont (Dateiname: Strg_14);
var
 Datei   : file;
 Zeichen : byte;
 i, j    : integer;
 Start   : integer absolute $F489;
 Matrix  : array[0..7] of byte absolute $F477;
begin
  j:=0;
  for Zeichen:=0 to 255 do
    begin
      Start:=Zeichen*8+$B800;
      GX_Get;
      for i:=0 to 7 do
        begin
          GX_CharMat[j]:=Matrix[i];
          j:=succ(j);
        end;
    end;
  assign (Datei,Dateiname);
  rewrite (Datei);
  blockwrite(Datei,GX_CharMat,16);
  close (Datei);




end;

{ ---------- Ende von FONT.INC ---------- }
```

```
{ ---------- SOUND.INC ----------
  Include-File fuer Sound-befehle auf dem JOYCE unter Turbo Pascal.
  Stellt (bislang) folgende Prozeduren zur Verfuegung:

    procedure Sound_On     -> Beeper an
    procedure Sound_Off    -> Beeper aus
    procedure Play(...)    -> Beeperzyklus Anzahl/Ein-/Auspause

  Kann eigenstaendig oder mit der Grafikerweiterung verwendet
  werden, letzterenfalls sind die Einschraenkungen zu beachten.

  ------------------------------------------------------------ }


procedure Sound_On;
begin
  port[248]:=11;
end;

procedure Sound_Off;
begin
  port[248]:=12;
end;

procedure Play(Num,On,Off:integer);
var i,j : integer;
begin
  for i:=1 to Num do
  begin
    Sound_On;
    for j:=1 to On do;
    Sound_Off;
    for j:=1 to Off do;
  end;
end;

{ ---------- Ende von SOUND.INC ---------- }
```

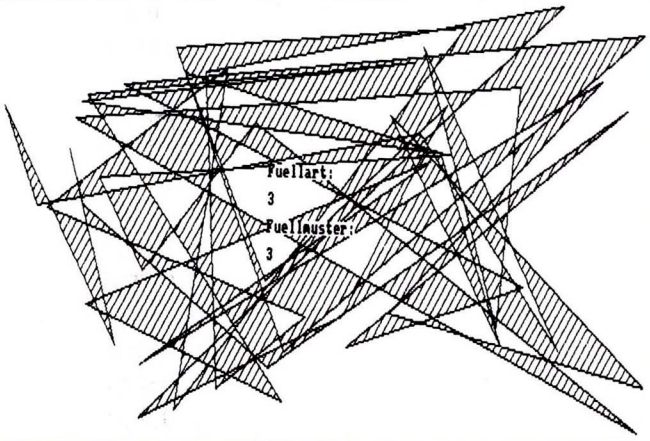
```
MACRO-80 3.44  09-Dec-81      PAGE  1

                        ;************************
                        ;  PLOT / SYMBOL - MCode
                        ;  Turbo Pascal Version
                        ;  1987 by M.A. & E. W.
                        ;  Vers. 10/8/87
                        ;    (incl. DotColor)
                        ;    (incl. Scr_Access)
                        ;************************
                        ;
                        .Z80
0000'                   ASEG
                        ORG 100h
                        ;
F400                    START   EQU 0F400h
FC5A                    XBIOS   EQU 0FC5Ah
00E9                    RUN_SCR EQU 00E9h
                        ;
                        .PHASE START
                        ;
F400                    _PLOT:
                        ;
F400  ED 5B F410        LD DE,(_X)
F404  2A F412           LD HL,(_YM)
                        ;
F407  01 F414           LD   BC,PLOT
F40A  CD FC5A           CALL XBIOS
F40D  00E9              DW   RUN_SCR
F40F  C9                RET
                        ;
                        ;
F410                    _X:  DS 2
F412                    _YM: DS 2
                        ;
                        ;
F414                    PLOT:
F414  3E 03               LD   A,03
F416  A2                  AND  D
F417  57                  LD   D,A
F418  E5                  PUSH HL
F419  26 00               LD   H,00
F41B  29                  ADD  HL,HL
F41C  01 B600             LD   BC,0B600h
F41F  09                  ADD  HL,BC
F420  4E                  LD   C,(HL)
F421  23                  INC  HL
F422  46                  LD   B,(HL)
F423  79                  LD   A,C
F424  E6 F8               AND  0F8h
F426  6F                  LD   L,A
F427  60                  LD   H,B
F428  29                  ADD  HL,HL
F429  19                  ADD  HL,DE
F42A  7D                  LD   A,L
F42B  E6 F8               AND  0F8h
F42D  6F                  LD   L,A
F42E  79                  LD   A,C
```

```
     MACRO-80 3.44   09-Dec-81      PAGE    1-1


F42F   E6 07                AND  07
F431   B5                   OR   L
F432   6F                   LD   L,A
F433   7B                   LD   A,E
F434   E6 07                AND  07
F436   3C                   INC  A
F437   47                   LD   B,A
F438   AF                   XOR  A
F439   37                   SCF
F43A                      LOOP:
F43A   1F                   RRA
F43B   10 FD                DJNZ LOOP
F43D   C1                   POP  BC
F43E   4F                   LD   C,A
F43F   78                   LD   A,B
F440   FE 00                CP   00
F442   20 04                JR   NZ,CLEAR
F444   79                   LD   A,C
F445   B6                   OR   (HL)
F446   77                   LD   (HL),A
F447   C9                   RET
F448                      CLEAR:
F448   FE 01                CP   01
F44A   20 05                JR   NZ,REVERSE
F44C   79                   LD   A,C
F44D   2F                   CPL
F44E   A6                   AND  (HL)
F44F   77                   LD   (HL),A
F450   C9                   RET
F451                      REVERSE:
F451   FE 02                CP   02
F453   20 04                JR   NZ,STATUS
F455   79                   LD   A,C
F456   AE                   XOR  (HL)
F457   77                   LD   (HL),A
F458   C9                   RET
F459                      STATUS:
F459   FE 03                CP   03
F45B   C0                   RET  NZ
F45C   79                   LD   A,C
F45D   A6                   AND  (HL)
F45E   32 F413              LD   (_YM+1),A
F461   C9                   RET
                          ;
                          ;
F462                      _SYMB:
                          ;
F462   01 F46B              LD   BC,SYMBOL
F465   CD FC5A              CALL XBIOS
F468   00E9                 DW   RUN_SCR
F46A   C9                   RET
                          ;
F46B                      SYMBOL:
F46B   21 F477              LD HL,_TABLE
F46E   01 0008              LD BC,0008H
F471   11 0000              LD DE,0000H

     MACRO-80 3.44   09-Dec-81      PAGE    1-2


F474   ED B0                LDIR
F476   C9                   RET
                          ;
F477                      _TABLE: DS 8
                          ;
                          ;
F47F                      _GET:
                          ;
F47F   01 F488              LD   BC,GETMAT
F482   CD FC5A              CALL XBIOS
F485   00E9                 DW   RUN_SCR
F487   C9                   RET
                          ;
F488                      GETMAT:
F488   21 0000              LD HL,0000H
F48B   01 0008              LD BC,0008H
F48E   11 F477              LD DE,_TABLE
F491   ED B0                LDIR
F493   C9                   RET
                          ;
                          ;
F494                      _SCRACC:
F494   01 F49D              LD BC,ACCESS
F497   CD FC5A              CALL XBIOS
F49A   00E9                 DW RUN_SCR
F49C   C9                   RET
                          ;
F49D                      ACCESS:
F49D   1E 00                LD E,0 ; (<- Column)
F49F   3E 00                LD A,0 ; (<- Row)
F4A1   16 00                LD D,0
F4A3   07                   RLCA
F4A4   17                   RLA
F4A5   17                   RLA
F4A6   17                   RLA
F4A7   6F                   LD L,A
F4A8   7A                   LD A,D
F4A9   CE B6                ADC A,0B6H
F4AB   67                   LD H,A
F4AC   7E                   LD A,(HL)
F4AD   23                   INC HL
F4AE   66                   LD H,(HL)
F4AF   6F                   LD L,A
F4B0   EB                   EX DE,HL
F4B1   29                   ADD HL,HL
F4B2   29                   ADD HL,HL
F4B3   19                   ADD HL,DE
F4B4   29                   ADD HL,HL
F4B5   01 0008              LD BC,0008H
F4B8   3E 00                LD A,0 ; (<- Access Mode)
F4BA   FE 00                CP 0   ; (   = READ?)
F4BC   20 06                JR NZ,WRITE
F4BE   11 F477              LD DE,_TABLE
F4C1   ED B0                LDIR
F4C3   C9                   RET
F4C4                      WRITE:

     MACRO-80 3.44   09-Dec-81      PAGE    1-3


F4C4   EB                   EX DE,HL
F4C5   21 F477              LD HL,_TABLE
F4C8   ED B0                LDIR
F4CA   C9                   RET
                          ;
                          .DEPHASE
                          END

     MACRO-80 3.44   09-Dec-81      PAGE    S


Macros:

Symbols:
F47F   _GET          F400   _PLOT         F494   _SCRACC
F462   _SYMB         F477   _TABLE        F410   _X
F412   _YM           F49D   ACCESS        F448   CLEAR
F488   GETMAT        F43A   LOOP          F414   PLOT
F451   REVERSE       00E9   RUN_SCR       F400   START
F459   STATUS        F46B   SYMBOL        F4C4   WRITE
FC5A   XBIOS

No Fatal error(s)
```

```
program graphik_demo;

(*$I KERNEL.INC*)
(*$I GRAPHLIB.INC*)
(*$I PATTERNS.INC*)
(*$I GRAPHIC.PRO*)


VAR ch : CHAR;


(***********************************************)
procedure linedemo;

BEGIN
 pattern_line(20,40,700,40,solid,0);
 pattern_line(20,80,700,80,dashed,0);
 pattern_line(20,120,700,120,dotted,0);
 pattern_line(20,160,700,160,dashed_dotted,0);
 pattern_line(20,200,700,200,l_dashed,0);

 pattern_line(160,5,160,240,solid,0);
 pattern_line(260,5,260,240,dashed,0);
 pattern_line(360,5,360,240,dotted,0);
 pattern_line(460,5,460,240,dashed_dotted,0);
 pattern_line(560,5,560,240,l_dashed,0);
END;

(***********************************************)
procedure xy_grafik;

VAR    i : INTEGER;
   Sinus : Point_array;

BEGIN
  FOR i:=0 TO 200 DO
    BEGIN
      Sinus(.i,0.):=i;
      Sinus(.i,1.):=Round(350*(sin(i))/SQRT(i+1));
    END;
  plot_point_array(Sinus,201,100,700,100,220,0,200,-200,200,10,10,0);
END;

(***********************************************)
procedure Lissajous;

VAR
      xy_per     : REAL;
      x_per, y_per,
      i          : INTEGER;
      Mode       : CHAR;

procedure draw;

Var winkel,
    x1,y1,x2,y2 : real;
    i : INTEGER;
```

```
  BEGIN
    winkel:=0; x2:=700; y2:=240;
    FOR i:=0 TO ROUND(60*xy_per) DO
    BEGIN
      CASE mode OF
    'N' : BEGIN
            x1:=x2; y1:=y2;
          END;
    'P' : BEGIN
            x1:=350; y1:=120;
          END;
    'L' : BEGIN
            x1:=(700/4)*cos(winkel)+360; y1:=120;
          END;
    'K' : BEGIN
            x1:=(700/4)*cos(winkel)+360;
            y1:=(240/4)*sin(winkel)+120;
          END;
      END;
     x2:=350*cos(winkel/x_per) + 350;
     y2:=120*sin(winkel/y_per) + 120;
     line(round(x1),round(y1),round(x2),round(y2),0);
     winkel:=winkel+(PI/30);
 END;
END;

BEGIN
gotoxy(30,10); WRITE(' X-Periode (1..10)? '); READ(x_per);
gotoxy(30,12); WRITE(' Y-Periode (1..10)? '); READ(y_per);
gotoxy(0,14); WRITE(' Ausf}hrung N(ormal), P(unkt), L(inie), K(reis)? '); READ(kbd,Mode);
Mode:=Upcase(Mode); WRITE(Mode);
i:=11;
  REPEAT
    i:=pred(i);
  UNTIL (INT(x_per/i)=x_per/i) and (INT(y_per/i)=y_per/i) ;
xy_per:=x_per*y_per/i;
ClrScr;
draw;
END;

(************************************************)
procedure MarkerDemo;

VAR  x,y : INTEGER;
     Mark: Marker_Type;

BEGIN
Mark:=Kreis;
  REPEAT
    x:=5+random(710);
    y:=3+random(242);
    plot_marker(x,y,Mark,0);
    IF Mark=Stern THEN Mark:=Kreis ELSE Mark:=succ(Mark);
  UNTIL keypressed;
END;

(************************************************)
procedure Musterdemo;


VAR    Viereckp : point_array;
           k,i : INTEGER;

procedure weiter;
BEGIN
FOR k:=0 TO 5 DO
        BEGIN
        Viereckp(.k,0.):=Viereckp(.k,0.)-20;
        Viereckp(.k,1.):=Viereckp(.k,1.)+20;
        END;
END;


BEGIN
        FillPatternInit;
        Viereckp(.0,0.):= 400;
        Viereckp(.1,0.):= 400;
        Viereckp(.2,0.):= 600;
        Viereckp(.3,0.):= 600;
        Viereckp(.4,0.):= 400;
        Viereckp(.0,1.):= 10;
        Viereckp(.1,1.):= 50;
        Viereckp(.2,1.):= 50;
        Viereckp(.3,1.):= 10;
        Viereckp(.4,1.):= 10;

        FOR i:=248 TO 255 DO
            BEGIN
              pattern_fill(5,viereckp,i);
              poly_line(5,viereckp,0);
              weiter;
            END;
         pattern_fill(5,viereckp,32);
         poly_line(5,viereckp,0);
         weiter;
         pattern_fill(5,viereckp,149);
         poly_line(5,viereckp,0);
        END;

        (************************************************)
        procedure Figurdemo;

        VAR  dreieckp : point_array ;

        BEGIN

        dreieckp(.0,0.):=110; dreieckp(.0,1.):=140;
        dreieckp(.1,0.):=250; dreieckp(.1,1.):=90;
        dreieckp(.2,0.):=40; dreieckp(.2,1.):=90;
        dreieckp(.3,0.):=110; dreieckp(.3,1.):=140;
```

```
        line(0,0,719,0,0);
        line(719,247,719,0,0);
        line(0,247,719,247,0);
        line(0,0,0,247,0);

        box(30,30,700,200,0);
        box(50,50,100,100,0);

        circle(200,100,50,0);
        circle(300,80,50,0);
        circle(50,200,30,0);
        circle(360,120,100,0);
        circle(600,70,50,0);
        circle(360,120,250,0);

        square(100,100,50,0);
        square(150,10,200,0);

        ellipse(200,150,50,100,0);
        ellipse(360,125,300,10,0);

        poly_fill(4,dreieckp,1);
        poly_line(4,dreieckp,0);

        END;


        (* Hauptprogramm *)


BEGIN
GraphInit;
ClrScr; HideCursor;

(* Lissajous-Figuren *)
WRITELN(' Lissajous-Figuren ');
WRITELN('  garantiert ohne GSX!!! ');
Lissajous;
GotoXY(2,31); WRITE('Weiter mit beliebiger Taste!');
READ(kbd,ch);
ClrScr;

(* Linienarten: Line_Muster() *)
GotoXY(10,10); WRITE('Es gibt 5 Linienmuster.');
GotoXY(12,12); WRITE('... man kann sich aber auch seine eigenen Muster definieren!');
Linedemo;
GotoXY(5,31); WRITE('Weiter mit beliebiger Taste!');
READ(kbd,ch);
ClrScr;

(* Box(),Square(),Ellipse(),Circle(),
   Poly_Line() und Poly_Fill()          *)
GotoXY(10,5); WRITE('Von Rechtecken bis zur Fl{chenf}llung ist einiges dabei');
Figurdemo;
GotoXY(5,30); WRITE('Weiter mit beliebiger Taste!');
READ(kbd,ch);
ClrScr;

(* Fuellmuster: Pattern_Fill *)
WRITELN;
WRITELN('F}llmuster k|nnen beliebig ');
WRITELN('im Zeichensatz eingef}gt werden');
WRITELN('oder die Grafikzeichen k|nnen verwendet werden');
Musterdemo;
GotoXY(60,28); WRITE('Weiter mit beliebiger Taste!');
READ(kbd,ch);
ClrScr;

(* Plot_Marker mit versch. Typen *)
GotoXY(10,15); WRITE('7 Markertypen sind bisher eingebaut');
GotoXY(20,20); WRITE('Ende mit beliebiger Taste!');
MarkerDemo;

ClrScr;

(* Beispiel fuer Darstellung von Point_arrays *)
GotoXY(10,5); WRITE(' Die Darstellung von Funktionen oder Me~werten ');
GotoXY(10,7); WRITE(' in Koordinatensystemen wird durch Festlegung  ');
GotoXY(10,9); WRITE(' von VIEWPORT und WINDOW-Koordinaten erleichtert.');
xy_grafik;
GotoXY(10,31); WRITE(' Ende des DEMOPROGRAMMS! Dr|cke RETURN');
READ(kbd,ch);

ShowCursor; ClrScr;
END.
```





Wer JOYCE mit dem Original-B-Laufwerk ausrüsten möchte, muß bis zu 700,– DM dafür ausgeben. Zwar bietet das Zweitlaufwerk eine Diskettenkapazität von 706 KB formatiert, mit einem preiswerten einseitigen 40-Spur-Laufwerk kann man jedoch das Preis-/Leistungsverhältnis deutlich zugunsten der Brieftasche verschieben...

Gemeint sind unter anderem die 3Zoll-Diskettenlaufwerke der Fa. Hitachi und BASF, die im Fachhandel oder im Elektronikversand (z.B. Fa. Völkner) zur Zeit für etwa 100 bis 150,– DM angeboten werden.

JOYCE: Zweit-Floppy für 100 DM?

Instruktionen und Informationen zum Selbstbau...

An CPC-Rechnern arbeiteten diese Laufwerke einwandfrei; entsprechende Anschlußvorschläge und Bauanleitungen wurden ja auch in dieser Zeitschrift schon veröffentlicht. Doch der Wunsch vieler JOYCE-Besitzer, so – ohne ein großes Loch im Geldbeutel – an ein Zweitlaufwerk zu gelangen, ließ sich bisher nicht realisieren. Ein Leserbriefschreiber wußte es sogar definitiv: »Ein 40spuriges Laufwerk ist am JOYCE nicht lauffähig!«. Hier irrt der Schreiber, und dies wahrscheinlich zu seiner eigenen Freude. Lassen wir also die Katze aus dem Sack: Der Betrieb mit einem einseitigen 40-Spur-Diskettenlaufwerk ist möglich! Mit einer Einschränkung, die sich jedoch systembedingt ergeben muß: Die Disketten können im Zweitlaufwerk nur gelesen, nicht jedoch beschrieben werden. So mancher JOYCE-Besitzer, manche Joyce-Besitzerin wird nun denken: Na ja, dann war das wohl doch nichts. Weit gefehlt – auch mit dieser Einschränkung lohnt sich der Einbau eines solchen Laufwerks in jedem Falle. Außerdem lassen sich normale CF2-Disketten im 80spurigen Original-Laufwerk (Preis DM 695.-) ebenfalls nur lesen. Jeder, der einmal mit zwei Diskettenstationen gearbeitet hat, möchte darauf nicht mehr verzichten. So auch der Autor, der seinen JOYCE mit dieser »Billig-Floppy« ausgerüstet hat und sich nun ein Arbeiten ohne das Zweitlaufwerk nicht mehr vorstellen kann. Für den Hobbybastler mit ein wenig Elektronikerfahrung dürfte der Um- und Ein-

```
B:SHOW A:[DRIVE]

       A: Drive Characteristics
   1,400: 128 Byte Record Capacity
     175: Kilobyte Drive  Capacity
      64: 32 Byte  Directory Entries
      64: Checked  Directory Entries
     128: Records / Directory Entry
       8: Records / Block
      36: Sectors / Track
       1: Reserved  Tracks
     512: Bytes / Physical Record
```

Abb. 1: Mit SHOW aufgelistete Parameter des Laufwerks A

Byte	Bedeutung
0+1	SPT Sektoren pro Spur
2	BSH Blockverschiebungsfaktor
3	BLM Blockmaske
4	EXM Extensionsmaske
5+6	DSM maximale Blockanzahl -1
7+8	DRM Anzahl der möglichen Directorys -1
9	ALO Reservierte Blöcke für Directory-Einträge
10	AL1 Allocation 1
11+12	CKS Prüfvektor, Diskettenwechsel-Erkennung
13+14	OFF Anzahl der reservierten Sytemspuren
15	PSH Physikalischer Sektor-Verschiebe-Faktor
16	PSM Physikalische Sektor-Verschiebe-Maske
17	Diskettenabtastung, 0 = einseitig
18	Anzahl der Spuren pro Seite
19	Anzahl der Records pro Spur
20	Nummer des ersten Records
21+22	Recordgröße
23	Gap Length (lesen/schreiben)
24	Gap Length (Format)
25	Bitmuster für Multi-Track-Verfahren
26	Freeze Flag, autom. Diskformat-Erkennung

Abb. 2: Struktur des erweiterten Disk-Parameter-Blocks XDPB des PCW 8256

bau eines solchen Floppy-Laufwerks keine großen Probleme bereiten. Vor der Praxis jedoch – so ist es üblich – zunächst ein wenig Theorie.

CP/M PLUS

Sieht man von der Textverarbeitung LOCOSCRIPT ab, so arbeitet der JOYCE mit dem Betriebssystem CP/M PLUS. Beim Booten des Rechners (Starten mit der System-Diskette) wird CP/M Plus in einem reservierten Bereich des Speichers abgelegt. Alle Programme oder Programmiersprachen, die nun in den Rechner geladen werden, greifen während der Programmausführung auf dieses Betriebssystem zu. Sie laufen – wie man sagt – unter CP/M. So handelt es sich bei dem Mallard-BASIC des JOYCE um ein reines CP/M-BASIC, das nach entsprechender Installation auch auf anderen CP/M-Computern lauffähig ist. Im Gegensatz dazu besitzen andere Rechner, wie z.B. der C64 oder die CPC-Rechner ein fest implementiertes BASIC, das auf den jeweiligen Rechner zugeschnitten ist, und nur auf diesem Rechner arbeitet. Der Vorteil eines genormten Betriebssystems liegt somit auf der Hand: Softwareprogramme können unabhängig vom Rechnertyp entwickelt und betrieben werden.

Das BIOS

Da die Hardware-Struktur der Rechner jedoch sehr unterschiedlich sein kann, muß ein Bindeglied zwischen den gerätespezifischen Eigenschaften und den Anforderungen der Software vorhanden sein. Diese Aufgabe übernimmt im wesentlichen das sogenannte BIOS (Basic-InputOutput-System). Vereinfacht kann man sagen, daß im BIOS-Speicherbereich alle wichtigen Hardware-Daten des Rechners abgelegt sind. Auf diese Daten greifen dann die beiden anderen Systemteile, das BDOS (Basic-Disk-Operating-System) und der CCP (Console-Command-Prozessor) während des Programmablaufs immer wieder zu.

Die Kenndaten und Programmroutinen (Sprungtabellen) für Bildschirm, Tastatur, Drucker, Schnittstelle und auch für die Diskettenlaufwerke sind also im BIOS-Bereich abgelegt.

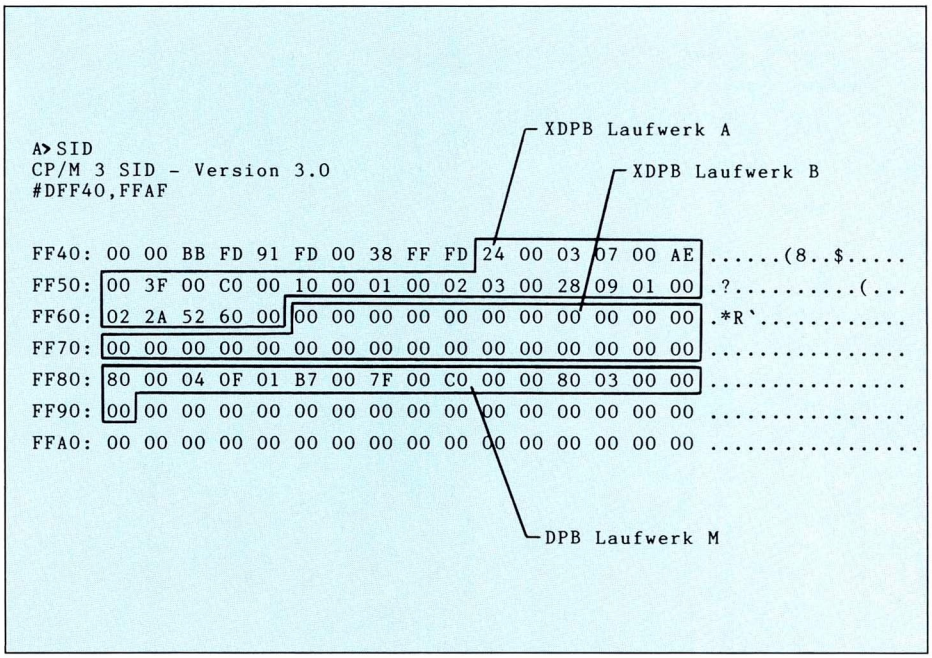


Abb. 3: Speicherbereich FF40 bis FFAF mit den Disk-Parameter-Blöcken für die drei Laufwerke

Zur Lösung unseres Problems interessiert uns natürlich besonders, wo und wie die Daten für die Diskettenlaufwerke abgespeichert sind. Im CP/M festgelegt ist zunächst einmal der Name dieses Bereiches. Er lautet: Disk-Parameter-Block, oder abgekürzt DPB.

Der Vollständigkeit halber sei noch erwähnt, daß weitere Laufwerkdaten auch noch in dem erweiterten Parameter-Block XDPB und dem sogenannten Disk-ParameterHeader DPH abgelegt sind.

Die Disk-Parameter

Doch zurück zum DPB. Es gibt zwei Möglichkeiten, sich seine Daten anzuschauen. Die erste wird manchem JOYCE-Besitzer, mancher JOYCE-Besitzerin bekannt sein. Das Dienstprogramm SHOW.COM auf der Systemdiskette listet nach dem Befehl SHOW A:ÄDRIVE* die Kenndaten des Laufwerks A auf den Bildschirm (siehe Bild 1). Kurz die Bedeutung der einzelnen Positionen. Nach der Angabe des Laufwerks folgt die Größe der Speicherkapazität. Ausgedrückt wird diese in Records (kleinste Speichermenge, bestehend aus 128 Bytes). Das Laufwerk A

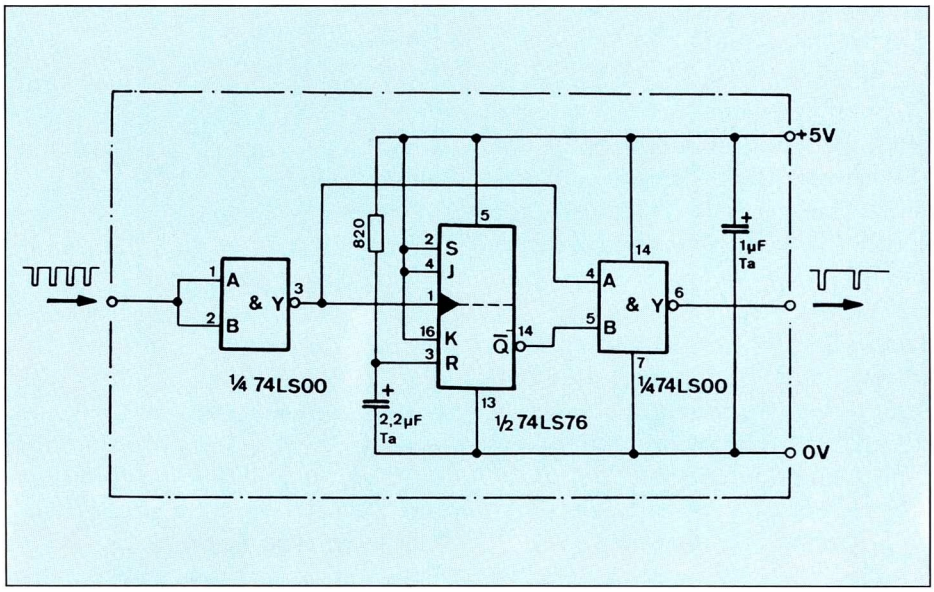


Abb. 4: Untersetzer-Schaltung für das Step-Signal; 40-Spur-Laufwerk an 80-Spur-Treiber.

JOYCE-Stecker 26 polig	SHUGART-Stecker 34 polig	Bedeutung
	2	frei
	4	In Use
	6	Drive Select 3
2	8	Index
4	10	Drive Select 0
6	12	Drive Select 1
frei	14	Drive Select 2
8	16	Motor On
10	18	Direction Select
12	20	Step
14	22	Write Data
16	24	Write Gate
18	26	Track 0
20	28	Write Protected
22	30	Read Data
24	32	Side 1 Select
26	34	Ready
1	9	0 Volt (Masse)
3	11	"
5	13	"
.	.	"
.	.	"
.	.	"
23	33	0 Volt (Masse)
25		frei

Abb. 5: Verbindung JOYCE – Floppylaufwerk mit Shugartanschluß

besitzt danach ein Kapazität von: 1400 * 128 = 179200 Bytes.
In Kilobytes umgerechnet also: 179200 = 1024 = 175 kBytes. Daß die Rechnung stimmt, bestätigt uns die 3. Zeile. Die nächsten beiden Informationen geben Auskunft über das Directory. Für jeden Directory-Eintrag sind 32 Bytes reserviert, 64 Einträge können vorgenommen werden. Die Angabe »Checked» besagt, daß alle 64 Directorys mit Schreibschutz (RO) versehen werden können. Es folgt die Anzahl der pro Directory maximal verwaltbaren Records, hier also 128.
Zeile 7 gibt die Aufteilung der einzelnen Blöcke an. Diese haben beim JOYCE eine Größe von 1 kByte und bestehen folgerichtig aus 8 Records von je 128 Bytes. Die Angabe 36 Records/Track besagt dann, daß jede Spur der Diskette in 36 Records aufgeteilt ist.
Die vorletzte Angabe bedeutet, daß dieses Laufwerk eine Systemspur (Reserved Track) besitzt, die nicht zum Speichern von Daten zur Verfügung steht. Schließlich gibt die unterste Zeile dann noch die Größe eines physikalischen Records an, der normalerweise als Sektor bezeichnet wird.
Soweit die Kennwerte des Laufwerks A, aufgelistet vom Dienstprogramm SHOW.COM. Dies ist jedoch nicht das einzige Laufwerk. Auch der RAM-Speicher wird vom Betriebssystem wie eine richtige Disk behandelt. So möchte ich jedem Joyce Besitzer/jeder JOYCE-Besitzerin empfehlen, sich ruhig einmal mit SHOW M:ÄDRIVE* die Eigenschaften des Laufwerks M anzusehen und diese mit den Daten des Laufwerks A zu vergleichen.

Der SID

Bei der Realisierung unseres Wunsches, ein zweites 40spuriges Laufwerk zu installieren, kommen wir jedoch mit der Anzeige der Kenndaten alleine nicht weiter. Unser Ziel ist ja die Eingabe entsprechender Daten oder deren Veränderung. Hier hilft uns ein anderes Dienstprogramm weiter: der SID (Symbolic Instruction Debugger). Mit dem typisch amerikanischen Ausdruck »Debugger«, auf deutsch: Wanzenentferner, werden in der Computersprache die Fehlersuchprogramme bezeichnet. Mit SID lassen sich unter anderem Speicherinhalte und Dateien untersuchen und gegebenenfalls verändern. Für unser Problem also genau das richtige Werkzeug (oder Tool, wie man in dieser Branche sagt). Jetzt fehlen nur noch die richtigen Speicheradressen der DPBs. Diese zu finden ist nicht ganz so einfach und nur über ein Maschinenprogramm möglich (für den Kenner: BDOS-Funktion 31). Das Ergebnis sei hier verraten. Beim JOYCE PCW8256 lauten sie:

Laufwerk A: FF4A bis FF64 Laufwerk B: FF65 bis FF7F Laufwerk M: FF80 bis FF91

Wie oben bereits angedeutet, besitzen die Laufwerke A und B jeweils einen erweiterten DiskParameter-Block, einen XDPB also. Dieser besitzt zusätzlich zu den 16 Bytes des normalen DPB weitere 10 Bytes mit ergänzenden Angaben. Die Bedeutung der einzelnen Bytes ist in Bild 2 aufgeführt. Es soll hier jedoch auf eine genaue Erläuterung aller Kennwerte verzichtet werden. Dies ist genug Stoff für einen gesonderten Artikel.
Um sich den Inhalt der XDPBs anzuschauen ruft man im CP/M Modus das Programm SID.COM auf. Es befindet sich auf der zweiten Diskette unter der Rubrik Dienstprogramme. Nach Erscheinen des Ziffernkreuzes × ist der Debugger bereit. Durch Eingabe von: DFF40,FFAF wird der gesuchte Bereich aufgelistet (Bild 3). Dort sind also die Hardware-Daten der Laufwerke gespeichert und mit Bleistift und Papier, einem Taschenrechner und einer Hexadezimal-Tabelle lassen sich dort auch die mit SHOW angezeigten Daten wiederfinden.
Auf eine Hex-Tabelle kann man verzichten, da SID mit dem Befehl H eine hexadezimal=dezimal Umrechnung durchführt und neben der Hex- und Dezimalzahl auch noch das entsprechende ASCII-Zeichen ausgibt. Möchte man z.B. die Anzahl der Spuren pro Seite bestimmen, so findet man unter der Adresse FC5C (Byte 18) den Hexadezimalwert 28. Nach Eingabe von H28 liefert SID das Ergebnis: 0028 ×40 '(' Es stimmt also: 40 Spuren.
Noch ein Hinweis, bevor Sie nun mit Feuereifer ans Umrechnen gehen. Bei Zweibyte-Befehlen ist das erste Byte das niedrigwertere. Beispiel: Die Anzahl der Records pro Spur (Byte 0 und 1, bzw. Adresse FF4A und FF4B) beträgt demnach nicht 2400hex, sondern 0024hex, bzw. 36 dezimal.
So weit, so gut. Der Computerfreak weiß schon, was jetzt zu tun ist. In den

freien Speicherbereich für das Laufwerk B braucht man nur noch mit dem Befehl S die gleichen Informationen, wie die des 40spurigen Laufwerks A einzugeben, und schon müßte alles funktionieren. Aber wie so oft in der Computerbranche: theoretisch ja, praktisch nein; Sämtliche Versuche, durch Ändern der DPB-Daten ein Zweitlaufwerk anzupassen, schlugen bisher fehl. Der Grund für dieses Fehlverhalten ist dabei recht schnell gefunden. Wie bei den meisten Rechnern hat man auch bei diesem die ursprüngliche Flexibilität des CP/M-Systems beschränkt. Die JOYCE-Konstrukteure sind davon ausgegangen, daß maximal zwei Laufwerke betrieben werden können, und daß als Zweitlaufwerk nur eine 80-Spur-Floppy angeschlossen wird. Dies führte zur Vereinfachung des Betriebssystems und zur Einsparung von Speicherbedarf.

Die Hardware-Lösung

Der Weg, per Software eine Änderung der Laufwerkdaten zu erreichen, hat sich demnach als Sackgasse erwiesen. Doch die Lösung ist relativ einfach. Wenn der Rechner ein 80-Spur-Laufwerk erwartet, warum bieten wir ihm dann nicht ein solches an? Ein Umbau auf 80 Spuren also? Nein, das wäre mit einer derart preiswerten Floppy sicher nicht zu realisieren. Der Trick: Wir simulieren ein solches Laufwerk nur. Zur Hilfe kommt uns dabei die Konzeption des JOYCE, auch 40spurige CF2-Disketten im Zweitlaufwerk lesen zu können. In diesem Falle sendet der Floppy-Controller immer zwei Step-Signale zur Weiterschaltung des Lesekopfes. Das Original-Laufwerk überspringt so eine achtziger Spur und greift auf die übernächste Spur zu. Dieses Verfahren erlaubt das Lesen beider Formate. Nicht aber das Schreiben, weil der achtziger Magnetkopf zwangsläufig nur die halbe Breite der vierziger Spur besitzt.

Benötigt wird also eine Schaltung, die aus den zwei Impulsen, die der Controller sendet, nur einen an das Laufwerk weitergibt. Eine solche ist in Bild 4 dargestellt. Sie wird vor den Step-Eingang des Laufwerks geschaltet und täuscht so dem Rechner die 80 Spuren vor. Tatsächlich aber legt der Lesekopf nur maximal 40 Spuren zurück.

JOYCE-Stecker 4 polig	Floppy-Stecker 4 polig	Spannung
1	1	+ 12 Volt
2	2	0 Volt (Masse)
3	3	0 Volt (Masse)
4	4	+ 5 Volt

Abb. 6: Anschlußbelegung der Spannungsversorgung

Die Funktion der Schaltung ist schnell erklärt. Ein JK-Flip-Flop (74LS76) ist als 2:1 Untersetzer geschaltet und gibt so über ein NAND-Gatter (74LS00) nur jeden zweiten Impuls zum Ausgang weiter. Die Kombination 820 Ohm / 2,2 uF sorgt für einen definierten Einschaltzustand des Flip-Flops. Da die Signale des Laufwerks mit einem 5 Volt Pegel arbeiten, wurden ICs der 74LS-Serie eingesetzt. Der Aufbau und die Verdrahtung der beiden ICs auf eine kleine Lochrasterplatte dürfte im allgemeinen keinerlei Probleme bringen. Aus diesem Grund wurde auf die Erstellung eines Layouts verzichtet.

Der Umbau in Etappen

Bevor nun mit heißem Lötkolben und Seitenschneider dem Rechner zu Leibe gerückt wird, noch einige Tips über die richtige Vorgehensweise. Vorsichtig zu Werke zu gehen versteht sich von selbst, zumal schon bei Öffnen des Rechnergehäuses eine evt. noch vorhandene Garantie erlischt.

Doch nun ans Werk. Nach Lösen der 6 Befestigungsschrauben ist die Rückwand des Bildschirmgehäuses leicht zu entfernen. Für den Anschluß des Zweitlaufwerkes hat die Firma Schneider/Amstrad alles vorbereitet. Flachbandkabel für die Datensignale und die Spannungszuführung sind bereits vorhanden und könnten sofort aufgesteckt werden, wenn, ja wenn die Stecker passen würden. Auch hier Sonderstecker in Schneider-Norm. Üblicherweise sind die Diskettenlaufwerke mit einem 34poligen Direkt-Stecker versehen, so auch die bereits erwähnten Hitachiund BASF-Geräte. Schneider Laufwerke besitzen dagegen eine 26-polige Stiftleiste.

Gott sei Dank hat man sich bei den elektrischen Signalen an die Norm gehalten: Shugart-Bus, so die offizielle Bezeichnung. In Bild 5 sind die einzelnen Datenleitungen mit der jeweiligen Anschlußbelegung aufgelistet. Dabei fällt auf, daß Amstrad=Schneider konsequent nur zwei Laufwerke ansprechen will und so auf die Leitungen Drive-Select-0 und Drive-Select-3 verzichtet. Dies führt zu einer Verschiebung der Anschlußbelegung. Die Buchsen für die Spannungsversorgung (Bild 6) besitzen zwar die gleiche Numerierung, doch auch hier ist die Bauform unterschiedlich. Entsprechende Buchsen und Stecker sind aber im Fachhandel erhältlich.

Wie geht man nun am besten vor? Viele Wege führen nach Rom, auch im Rechnergeschäft. Nehmen wir den sicheren Weg, d.h. den in Etappen.

1. Etappe: Herstellen der beiden Adapterkabel. Eines für die Spannungsversorgung (kein Problem) und eines für die Daten. Man kann selbstverständlich die Buchse des vorhandenen Flachbandkabels abschneiden und die erforderliche 34-polige Shugart-Buchse anlöten, bzw. aufpressen. Wer jedoch einen derart »schneidenden« Eingriff nicht riskieren möchte oder daran denkt, später vielleicht doch noch einmal ein 1 MByte Laufwerk einzubauen, der sollte ein Adapterkabel anfertigen. Passende Stecker und Buchsen sind im Elektronik-Fachhandel zu erhalten. Benötigt werden eine Shugart-Buchse, passend auf das einzubauende Laufwerk, und eine 26polige Steckerleiste, zweireihig im 1/10 Zoll Raster. Nach der Belegungs-Tabelle in Bild 5 werden nun die Anschlüsse hergestellt. Zweckmäßig verwendet man ein Flachbandkabel, bei dem dann jede zweite Ader mit Masse verbunden wird

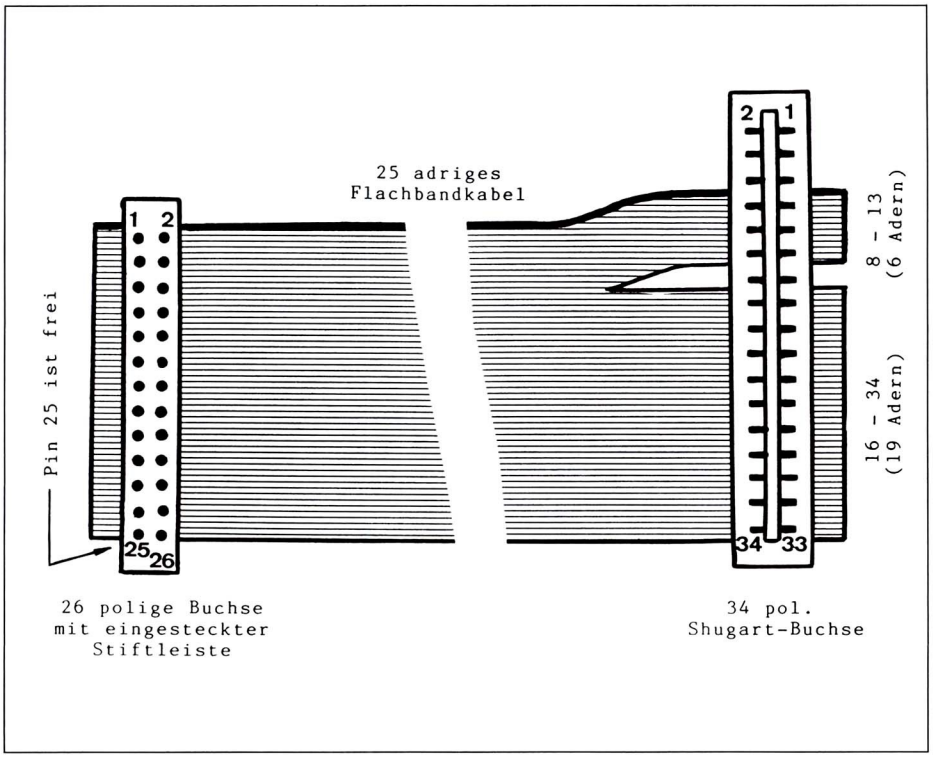


Abb. 7: Vorschlag für ein Verbindungskabel JOYCE-Buchsenleiste mit Zweitlaufwerk

(alle ungeraden Steckerpunkte). Soll diese Verbindung profimäßig und mit Einpress-Steckern (Fachausdruck: Schneidklemmverbinder) hergestellt werden, so kann Bild 7 als Vorlage dienen. Hierbei muß wegen der Verschiebung zwischen den beiden Steckerbelegungen das Flachbandkabel – wie in der Zeichnung angegeben geteilt werden. Leider sind keine 26poligen Steckerleisten in Einpresstechnik im Handel erhältlich. So muß man sich mit Einpress-Buchsen, in die man dann als Verbinder eine zweiseitige Stiftleiste einsteckt, behelfen. Auf diese Ausführung - also Stifte in der 26-poligen Buchsenleiste bezieht sich die Darstellung und die Numerierung in Bild 7.

2. Etappe: Prüfung der Adapterkabel auf richtige Belegung. Jede Ader sollte einzeln auf Durchgang und Verbindung zu den Nachbaradern durchgemessen werden. Dies ist besonders wichtig, weil dadurch eine mögliche Schädigung von Laufwerk oder Rechner vermieden wird.

3. Etappe: Entfernen der Leitungsabschlußwiderstände am Laufwerk. Diese sind in einem Widerstands-IC zusammengefaßt, das in der Nähe des Anschlußsteckers zu finden ist. Der Wert beträgt bei den oben angesprochenen Laufwerken 150 Ohm, und ist für die Treiberbausteine eine zu hohe Belastung. Da im Laufwerk A bereits Abschlußwiderstände vorhanden sind, genügt es – bei relativ kurzem Adapterkabel – wenn das Widerstands-IC aus dem Sockel entfernt wird. Auch der Widerstand für die Laufwerk-Anwahl ist mit 150 Ohm zu niedrig und sollte auf etwa 1 kOhm erhöht werden. Es handelt sich hierbei jedoch um einen Einzelwiderstand, der sich in der Nähe der Brücken für die Laufwerk-Anwahl befindet.

4. Etappe: Bei ausgeschaltetem(;) Rechner die Kabelverbindungen herstellen und das Metallgehäuse des Laufwerks mit Masse verbinden. Die Brücke für die Laufwerk-Anwahl (Drive-Select) auf Position 2 stecken. Achtung, gleich fällt die erste Vorentscheidung; Rechner einschalten, mit der Systemdiskette booten. Auf dem Bildschirm muß nun die neue Meldung erscheinen:
»v 1.21, 61K TPA, 2 Laufwerke ...«.
Das gewohnte »Laufwerk ist A:« in der rechten unteren Ecke wird dann ebenfalls verschwunden sein. Falls dies nicht der Fall ist, so sollten die Verbindungskabel noch einmal überprüft und die Spannungszuführung nachgemessen werden. Da die Umrüstung auf 80 Spuren noch nicht erfolgt ist, darf das Laufwerk B auch noch nicht per Befehl angesprochen werden. Dies führt zu der Fehlermeldung ERROR ON B:. Deshalb: Rechner zunächst wieder ausschalten.

5. Etappe: Einbau der Untersetzerschaltung. Günstig ist es, wenn die kleine Platine direkt auf das Laufwerk montiert wird. Die Spannungsversorgung kann in der Nähe des Steckers 2 abgegriffen werden (5Volt = Pkt.4; 0 Volt = Pkt.2=3).

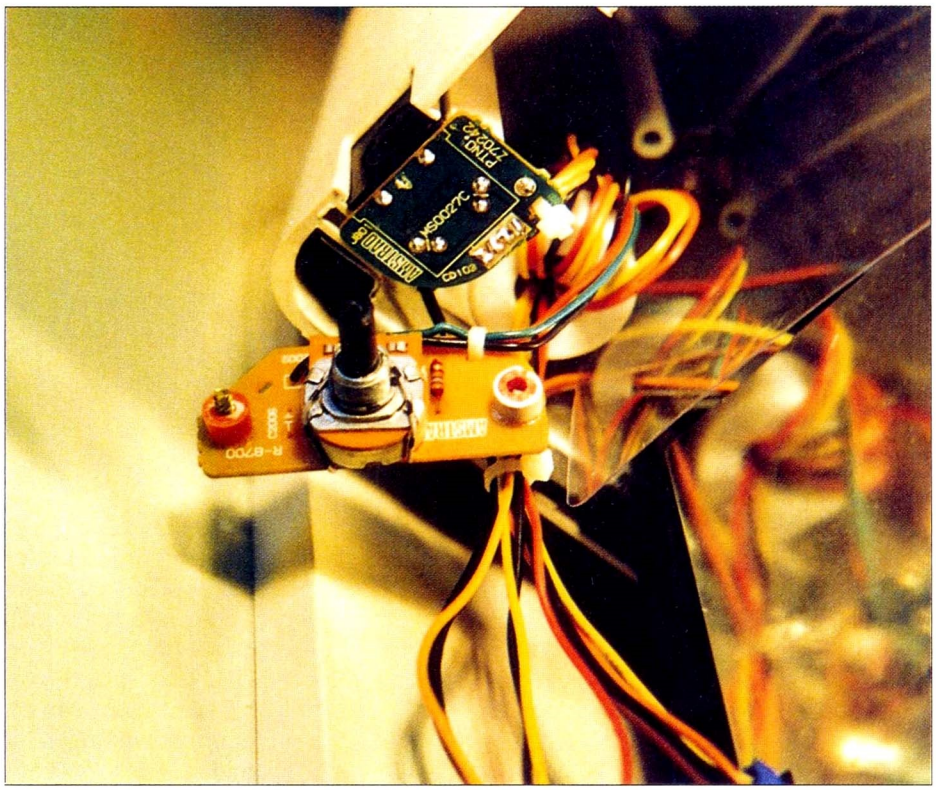
Die Leiterbahn, die das Step-Signal führt, wird mit einem Messer unterbrochen und die Untersetzerschaltung dazwischengeschaltet. Vom Stecker 1 Pkt.20 muß das Signal auf den Eingang der Untersetzerplatine führen, der Ausgang auf die weiterführende Leiterbahn gelötet werden.

Das Ziel ist erreicht; Ihr Joyce ist nun mit einem Zweitlaufwerk ausgerüstet. Und wenn ein erstes DIR B: das gewünschte Ergebnis auf den Bildschirm bringt, dann kommt Freude auf.

Noch ein paar Tips

Zum Abschluß noch ein paar Tips zum Einbau des Laufwerks in das JOYCE-Gehäuse. Auch wenn die angebotenen Laufwerke etwas breiter als das Schneider-Original sind, so ist mit etwas Geschick, einer Feile und der nötigen Vorsicht der Einbau in den dafür vorgesehenen Ausbruch des Bildschirmgehäuses möglich. Ob jedoch im Rechner oder außerhalb stationiert, in jedem Fall sollte das Laufwerk mit einer Metall-Abschirmung umgeben sein. Die Magnetköpfe sind sehr störempfindlich und reagieren schon auf die normale Strahlung des Rechners. Eine ausreichende Abschirmung erreicht man bereits mit einem U-förmig gebogenen Stück Weißblech. Auf der Innenseite mit Folie beklebt (um Kurzschlüsse zu verhindern), wird dieses Blech auf der Elektronikseite des Laufwerks befestigt und mit den 0 Volt der Spannungsversorgung – besser noch mit einer gesonderten Masseleitung mit dem gekennzeichneten Massepunkt im Rechner – verbunden. Daß sich die Mühe des Einbauens gelohnt hat, werden Sie schon bald feststellen. Spätestens aber bei Benutzung des Textsystems. Denn auch hier haben sie Lesezugriff zu Laufwerk B. Die Zeiten des mühevollen Zwischenspeichern nach Laufwerk M sind nun vorbei, Kopien von B nach A direkt möglich. Befürchtete Programmabstürze wegen versuchten Schreibens auf die BDiskette treten nicht auf. Das Betriebssystem erkennt die »Einseitigkeit« des Laufwerks B und gibt entsprechende Fehlermeldungen aus.

(W. Otternberg)



JOYCE-Bildschirmhelligkeit variabler einstellbar

Erste Voraussetzung für längeres und trotzdem ermüdungsarmes Arbeiten am Rechner ist ein nicht nur gut entspiegelter, sondern auch in der Helligkeit richtig eingestellter Bildschirm. Der JOYCE-Benutzer wird mit dieser Tatsache spätestens dann unangenehm konfrontiert, wenn er versucht bei geringer Umgebungshelligkeit mit seinem Bildschirm zu arbeiten, und seine Augen dagegen schmerzhaften Protest einlegen. Der in der JOYCE eingebaute Helligkeitsregler läßt nämlich nur eine Einstellung in einem bestimmten Bereich zu, die in vielen Fällen erheblich zu hell ist.

Durch einen einfachen Austausch zweier billiger Bauteile läßt sich dieser Mangel jedoch unschwer beheben. Die folgende kleine Umbauanleitung soll dies zeigen und zur Nachahmung ermuntern - wobei sich einige weitere praktische Nutzeffekte dann gleich mit einstellen.

Die Helligkeit des Bildschirms hängt bei der JOYCE wie bei den meisten anderen Systemen auch von der Einstellung eines einfachen Drehreglers (Potentiometer) ab, der sich erfreulicherweise von allen anderen Baugruppen getrennt in einer eigenen Ecke des Gehäuses befindet und daher verhältnismäßig leicht zugänglich ist. Elektrotechnisch gesehen bildet dieses Potentiometer, auch kurz Poti genannt, zusammen mit einem (ebenfalls dort befindlichen) Vorwiderstand einen sogenannten Spannungsteiler, der mit einer konstanten niedrigen Gleichspannung gespeist wird und eine über die Drehstellung einstellbare Ausgangsspannung abgibt. Die Höhe dieser Spannung bestimmt die Bildschirmhelligkeit.

Die Flexibilität der Helligkeitsregelung hängt damit entscheidend davon ab, in welchem Bereich sich die Ausgangsspannung des Spannungsteilers variieren läßt, was wiederum von den elektrischen Werten von Poti und Vorwiderstand abhängt. Die mangelhafte Einstellbarkeit im Falle der JOYCE rührt daher, daß diese Werte vom Hersteller grob falsch gewählt wurden; aus unerfindlichen Gründen (wollte man etwa schnelleren Bildschirmverschleiß bewirken ?) wurde der Vorwiderstand viel zu groß gewählt und in der Schaltung so angeordnet, daß er den größten (dunkleren) Teil des an sich möglichen Einstellbereichs einfach abblockt. Er verhindert, daß die vom Poti "geteilte" Spannung einen bestimmten (recht hohen) Wert unterschreitet - und damit eben auch eine bestimmte Bildschirmhelligkeit.

Die Lösung des Problems liegt, wie man schon sieht, in einem Ausbau oder Austausch des Potis oder des (billigeren) Vorwiderstands (wobei die Kosten nicht die überragende Rolle spielen, da auch das Poti nur einige DM kostet). Obwohl es theoretisch ausreichen könnte, den Vorwiderstand einfach kurzzuschließen (zu überbrücken) und damit das "untere" Ende des Potis auf Masse zu legen (entspricht "ganz dunkel"), wird in dieser Anleitung sicherheitshalber das Poti komplett mit dem Widerstand ausgetauscht (mit gleichem Effekt); dadurch wird das Risiko umgangen, die Spannungsversorgung des Potis zu überlasten, da sich der Gesamtwiderstand des Spannungsteilers dabei immerhin auf knapp die Hälfte reduzieren würde. Daß dies tatsächlich problematisch wäre, ist zwar in diesem Fall sehr unwahrscheinlich; da sich der Autor mangels eines kompletten Schaltplans aber nicht vom Gegenteil überzeugen konnte, gehen wir lieber gleich auf Nummer sicher.

Dazu den Spannungsteiler bei gleichbleibendem Gesamtwiderstand nachzuberechnen, sei dem Leser im einzelnen erspart, da es hier vor allem um den "Umbau" selbst gehen soll. Bei einem "alten" Widerstandswert von 5k-8f- für das Poti und 5.6 k-8f- für den Widerstand ergibt sich ein neuer Wert für das Poti von 10 k-8f- (linear) und etwa 1,8 k-8f- für den Vorwiderstand. Letzterer soll außerdem elektrisch am anderen Ende des Potis zu liegen kommen und den Bildschirm nun vor übermäßig heller Einstellung schützen, bei der auf Dauer die Bildröhre einbrennen könnte. (Durch einfaches Vertauschen der Anschlüsse des Teilers wird der Erfolg leider nicht erreicht: es wird zu dunkel.)

Nach diesen theoretischen Erklärungen wird es nun praktisch. Wer etwas



Bild 1: Das Öffnen der Gehäuserückwand



Bild 2: Hier ist ein langer Schraubendreher nötig

handwerkliches Geschick aufbringt (oder die Arbeit in einer Vertragswerkstatt vornehmen läßt, die dazu bereit ist) wird immerhin mit einem vierfachen Vorteil belohnt:

- Die Helligkeit des Bildschirms läßt sich nun jeder Umgebungshelligkeit richtig anpassen, vor allem auch der geringen. Das schont neben den Augen auch die Bildröhre, die sich bekanntlich umso schneller abnutzt, je heller sie betrieben wird.

- Der Bildschirm läßt sich, wenn man JOYCE z.B. in längeren Arbeitspausen stehenläßt, auch ganz dunkel drehen, ohne den Rechner ausschalten zu müssen. Das erspart das lästige Neubooten, ohne jedoch den Bildschirm ständig unnütz zu belasten. Auch vor allzu neugierigen Blicken zufälliger Besucher auf den Bildschirm kann man sich damit wirksam schützen.

- In nun gemäßigter Helligkeit ist (unter CP/M) endlich auch eine vernünftige und augenfreundliche Positivdarstellung (dunkle Schrift auf hellem Hintergrund) möglich, da das dabei bisher sehr störende Flimmern deutlich mit der Helligkeit abnimmt.

- Die Bildröhre selbst ist besser gegen das Einbrennen von Zeichen geschützt, da die Helligkeitsskala jetzt am oberen Ende etwas begrenzt ist. Ständig mit maximaler Helligkeit arbeiten sollte man aber trotzdem nicht, sie sollte besonderen Bedarfsfällen vorbehalten sein.

Bevor Sie loslegen, sollten Sie am besten die folgende Anleitung einmal ganz durchlesen und sich das nötige Werkzeug zurechtlegen. An Werkzeug braucht man zwei Kreuzschlitz-Schraubendreher (Philips Nr.2 und Nr.1),möglichst eine flache, vorne abgewinkelte Pinzette, einen Lötkolben von ca. 30 Watt mit Feinlötspitze und ein Universalmesser (Tapetenmesser). Das Material – Poti und Widerstand sowie etwas Lötzinn – bekommt man für ein paar Mark im Elektronikshop oder auch in einer Radio- und Fernsehwerkstatt. Sicherheitshalber soll noch einmal darauf hingewiesen werden, daß durch den Eingriff wenn man ihn selbst vornimmt natürlich die Herstellergarantie verlorengeht; Wer seinen Händler die Arbeit vornehmen läßt, sollte über diesen Punkt verhandeln.

Als allererstes ist JOYCE auszuschalten und der Stecker aus der Steckdose zu ziehen. Danach sollte man noch etwa eine Stunde warten, bis sich die Spannung an der Bildröhre abgebaut hat (mit der man den Kontakt allerdings auch so ohne Akrobatik vermeiden kann). Bis dahin dürfen Sie schon einmal alle Stecker und Erweiterungseinheiten von der Rückwand entfernen und in der richtigen Reihenfolge die sechs Schrauben lösen, die den hinteren Gehäuseteil mit dem vorderen und mit dem JOYCE-Innenleben verbinden. Die Schrauben sind inzwischen in einer kleinen Schachtel gut aufgehoben, wo sie nicht verlorengehen können.

Falls eine Erweiterung an der JOYCE-Rückseite angeschraubt ist (z.B. die SIO), müssen deren Halteschrauben zuerst gelöst werden. Dabei ist darauf zu achten, daß man wirklich die Halteschrauben erwischt (sie sitzen ggf. ganz oben und unten am Gehäuserand der Erweiterung) und nicht die Schrauben, die die Erweiterung in sich zusammenhalten.

Am JOYCE-Gehäuse müssen als erste unbedingt die beiden kurzen Gewindeschrauben gelöst werden, die das äußere Gehäuse des Rechners mit dem inneren verbinden; sie sind durch Pfeile gekennzeichnet (Bild 1). Dann löst man die vier Schrauben am oberen und unteren Rand der Gehäuserückwand, wobei man zuerst alle vier Schrauben etwas lockern sollte, um Spannungen im Gehäuse zu vermeiden. Diese Schrauben liegen besonders an der Unterkante recht tief im Gehäuse; damit man sie erreicht, ohne den Schraubendreher zu verwinkeln, sind im JOYCE-Gehäuse an dieser Stelle Aussparungen eingearbeitet (Bild 2). Das Herausdrehen gelingt ohne Probleme, wenn man den Schraubendreher gerade und mit deutlichem Druck auf die Schraube ansetzt und gleichmäßig dreht, bis jede Schraube beim Drehen ein leises Knacken von sich gibt, indem sie in den letzten Gewindegang zurückrutscht. Nun kann man das Gehäuse mitsamt den Schrauben nach hinten abziehen. Dies sollte man mit Gefühl tun, da die Schrauben dabei manchmal etwas haken.

Nun suchen wir uns in der Ecke unter der eingebauten Floppy (wer zwei Floppies hat, muß leider die untere ausbauen) die Halteschrauben für die Platine, auf der das Helligkeitspoti sitzt. Mit dem langen Schraubendreher kann sie losgeschraubt werden; dabei sollte man mit der anderen Hand und der Pinzette die Schraube zu fassen bekommen, damit man sie nicht durch Hochhalten und Schütteln der JOYCE (möglich, aber nicht zu empfehlen) aus dem Gehäuse holen muß. Der Platz für diese Arbeit ist ausreichend und läßt sich noch etwas erweitern, indem man die Platine mit der Tastaturbuchse nach hinten aus ihrer Halterung zieht. Nachdem die Platine glücklich losgeschraubt und die Schrauben verwahrt sind, klemmt noch das Anschlußkabel

des Potis unter einer Plexiglasfolie auf der Bildröhrenseite, die man vorsichtig hochbiegt, um das Kabel nach vorn zu führen und so genügend freie Länge zu gewinnen. Ein Herausziehen des Kabels (aus der Steckverbindung auf der Monitorplatine) kann damit erübrigt werden.

Nun kommt die eigentliche Änderung. Zunächst notiere man sich die Anschlußfarben und -Punkte der drei Verbindungsdrähte (schwarz, grün, rot) an der Platine und löte sie ab. Beim Wiederanlöten werden später die Farben rot und schwarz vertauscht; Von der Platine entfernt man nun den 5,6 k Vorwiderstand (Kennzeichnung: Grün-Blau-Rot-...) und lötet dafür den 1,8 k Widerstand ein. Das Poti wird ebenfalls ersetzt; das Ersatzteil muß eine Plastikachse haben, die mit dem Tapetenmesser für den Drehknopf passend gemacht wird, da die handelsüblichen Potis etwas größer sind als das Originalteil. Die Achse ist auf die richtige Länge (nicht zu knapp) abzuschneiden und an einer Seite für den Drehknopf abzuflachen; eine probeweise Montage wegen der Länge empfiehlt sich. Das neue Poti wird dann so auf die Oberseite der Platine gelötet: Mit einer Pinzette oder Silberdrahtzange biegt man die Anschlüsse passend zu den Bohrungen der Platine nach unten, wobei man auf das Potiinnere keine Kräfte ausüben sollte; dann wird es möglichst bündig auf die Platine gesetzt und auf der Lötseite festgelötet. Um die mechanische Stabilität zu erhöhen, legt man noch einen Draht um den Gewindeanschluß, den man an der breiten Masseführung der Platine anlöten kann (großes Bild).

Auf die Platine klebt man auf der Bestückungsseite zwei kurze (ca. 4mm) Abstandshalter für die Schrauben zur Anpassung der Bauhöhe des Potis. Diese halten auch die Schrauben beim Wiedereinsetzen der Platine und erleichtern deren Plazierung. Nun lötet man die Anschlüsse vom Rechner wieder an (rot und schwarz vertauschen;). Bevor man sie endgültig wieder einbaut und den Rechner evtl. probeweise einschaltet, sollte man die Platine auf Kurzschlüsse und Unterbrechungen sorgfältig überprüfen.

Das Einsetzen der Schrauben (mit der Platine) kann man sich erleichtern, indem man etwas (nicht zu kräftigen;) zähen Kleber in die Abstandshalter gibt; dann rutschen die Schrauben nicht heraus. Der Zusammenbau des Gehäuses und die Montage aller losgeschraubten Teile erfolgt in umgekehrter Reihenfolge des Auseinanderbaus.

Nach dem Einschalten ergibt sich noch eine Überraschung: Der Drehsinn der Helligkeitseinstellung ist jetzt vertauscht (Linksanschlag = dunkel). Bei der Gewöhnung daran hilft ein kleines Etikett mit der Markierung:

(− <---> +)

auf der Vorderseite des Gehäuses. Nun wünschen wir Ihnen viel Spaß beim Bauen und im Einsatz Ihrer verbesserten JOYCE.

(Christian Frederking)

PC 1512/1640

HARD-DISC-KIT:
- HD mit Controller und Kabel
- Stahlblechgehäuse mit Lüfter und Blende in Schneider-Beige!

20 MB: DM 999,- / sFr. 899,-/öS 6995,-
30 MB: DM 1199,- / sFr. 1049,-/öS 8395,-

HD-CARD:
- Speziell für PC
- Steckerkompatibel
- komplett montiert auf Alurahmen

20 MB: DM 899,- / sFr. 899,-/öS 1049,-
30 MB: DM 999,- / sFr. 999,-/öS 6995,-

PROFI-BOX:
- Externe HD 20 MB mit Streamer 20 MB in einem zusätzlichen PC-Gehäuse

DM 2498,- / sFr. 2498,-/öS 19950,-

FD-3:
- Original Laufwerk mit Blende in Schneider-Beige!

DM 398,- / sFr. 359,-/öS 2995,-

RAM-Erweiterung: DM 79,- / sFr. 89,-/öS 695,-

Bildschirmfilter:
- s/w und color

DM 59,- / sFr. 59,-/öS 475,-

HÄNDLERANFRAGEN SIND WILLKOMMEN!

Das komplette ABD-Zubehör ist bei den Schneider-Computer-Händlern sowie bei Hertie, Horten, Kaufhof und Ringfoto erhältlich.

Alle unsere Produkte werden zu **Tagespreisen** gehandelt. Deshalb können die Preise tatsächlich niedriger sein als genannt!

Deutschland:
ABD Electronic
Zettachring 12
7000 Stuttgart 80
Telefon 07 11-7 15 00 37

Schweiz:
Swicom SA
Route de Boujean
CH-2502 Biel-Bienne
Telefon (0 32) 42 27 84
oder 23 18 33

Österreich:
Wagner Electronics
Hauptstraße 171
3001 Mauerbach
Telefon 02 22-97 21 66

Von den Entwicklern des BUSINESS-STAR:

- COMAC-KASSE Einnahmen/Überschußrechnung ... DM 128.00
- COMAC-LV Leistungsverzeichniserstellung ... DM 398.00
- COMAC D.M.S. (erfolgreiche Direktwerbung) ... DM 128.00
- COMAC-Holzlager für Einschlagunternehmen ... DM 298.00
- COMAC-Getreideabrechnung (Rhld./Schweiz) für Landwirte ... DM 398.00
- BUSINESS-STAR, Faktura, Lager, Kunden, Mahnwesen, Statistik ... DM 288.00
- BUSINESS-STAR PLUS Auftragsverwaltung ... DM 188.00
- Individuallösungen nach Ihren Wünschen ... a. Anfrage

Angebot des Monats

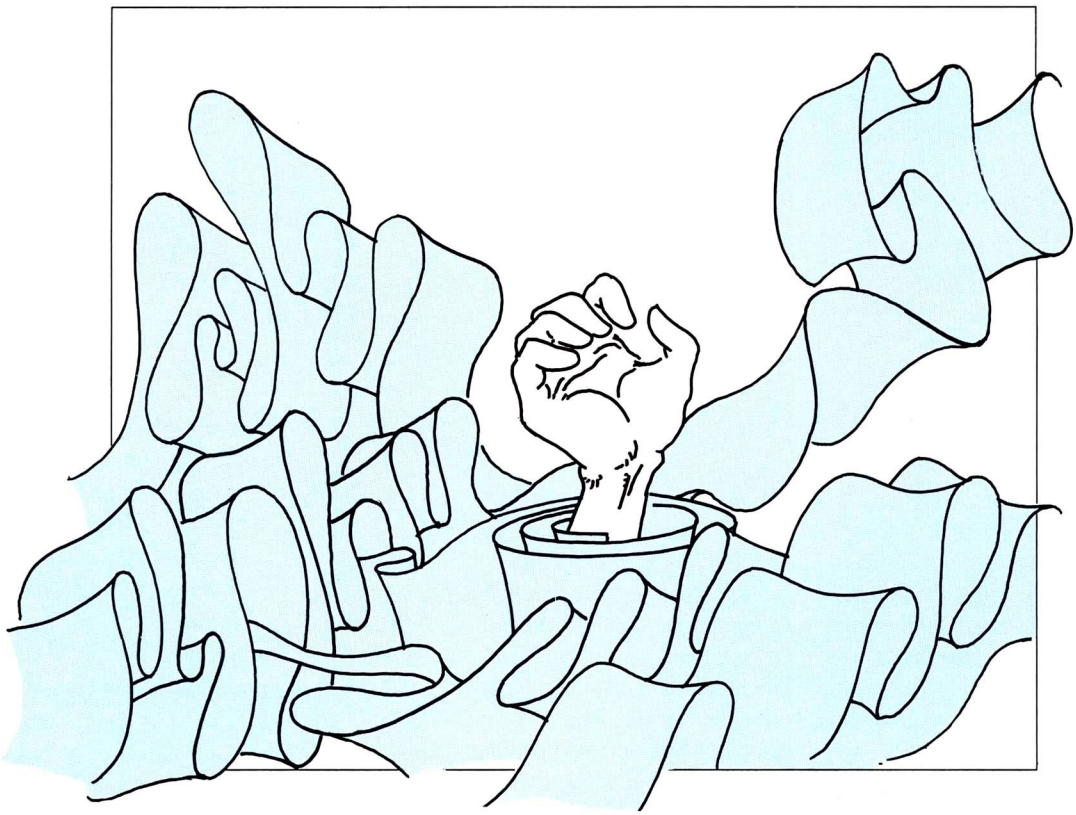
- 3" Marken-Disketten, 10 Stück ... DM 69.00
- Marken-Druckerfarbband für JOYCE ... DM 18.95
- 3" Endlos-Haftetiketten, 100 Stück ... DM 9.95

Verlangen Sie ausführlichen Gesamtprospekt für JOYCE und PC vom:

BFS Büro für Software-Entwicklung

Sonnenstr. 43, 5270 Gummersbach, Tel.: 02261/65434.

Besuchen Sie das Büro für Software-Entwicklung auf der SYSTEMS in München (19.-23. Okt.), Halle 1, Stand 1B2 (am Haupteingang)



Übersichtliche Basic-Listings erstellen...

Der Joyce-LISTER

Aufmerksamen Lesern der CPC International wird dieser Programmtitel bekannt vorkommen. Im Heft 1/86 wurde ein solches Programm für die Schneider CPC's vorgestellt. Es ermöglicht die Ausgabe von übersichtlichen BASIC-Listings, sofern sie als ASCII-File vorliegen. Was den Usern der CPC's recht ist, sollte auch den JOYCERN möglich sein. Daher hier nun die neue Version, die auf dem JOYCE lauffähig ist.

Das Programm erstellt ein Listing von BASIC-Programmen, die mit dem Befehl 'SAVE "Programm",A' abgespeichert wurden, mit folgenden Merkmalen:

1. Nur ein Befehl pro Druckzeile
2. Strukturierung durch automatisches Einrücken bei den Befehlen:
   - *WHILE = WEND*
   - *FOR = NEXT*
   - *IF = THEN*
3. Verzweigungen im Programmablauf werden angezeigt durch:
   - `-----` *bei bedingtem GOTO*
   - `======` *bei unbedingtem GOTO*
   - `--R--` *bei bedingtem RETURN*
   - `==R==` *bei unbedingtem RETURN*
   - `--U--` *bei GOSUB Befehlen*
4. REM-Befehle werden immer an den Zeilenanfang gesetzt.

Anmerkung zu 1.
Das Programm sucht jede Zeile nach Doppelpunkten ab und trennt die Zeile nach Bedarf auf. Doppelpunkte zwischen zwei Anführungszeichen werden nicht beachtet (Zeilen 400 470).

Anmerkung zu 2.
Nach jedem der in 2. genannten Befehle rückt die erste Druckposition der folgenden Zeilen um n Stellen nach rechts, und zwar so lange, bis der entsprechende Folgebefehl erkannt wird. Man sollte jedoch schon beim Programmieren darauf achten, daß FOR und zugehöriges NEXT nicht in einer Zeile stehen.

In Zeile 260 wird die Ausgabe des Listings mit einem POKE-Befehl auf den Drucker umgeleitet und in Zeile 1100 wieder auf den Bildschirm umgeleitet. Daher erfolgt der Ausdruck in Fettdruck. Der Befehl 'PRINT cls$' in Zeile 270 wird ebenfalls auf den Drucker umgeleitet und führt dort zur Sequenz ESC E + ESC H, was für den JOYCE-Matrix-Drucker bedeutet »Fettdruck« an =Doppeldruck aus.
Durch den Befehl 'PRINT window$', der eigentlich nur für die Bildschirmausgabe interessant ist, erscheint auf der Druckerausgabe die Zeichenkombination '$$80'. Im Vergleich zur wesentlich umständlicheren Ansprache des Druckers mit LPRINT-Befehlen ist dies jedoch wohl zu verschmerzen.
Zeile 75 als »Fehlerbehandlungsroutine« mußte eingefügt werden, damit sich das Programm auch selbst listet. Ohne ein RESUME NEXT (in Zeile 1160) brach das Listing regelmäßig in Zeile 960 ab. Das Anführungszeichen hinter dem ersten ELSE in dieser Zeile wird nicht erkannt. Vielleicht weiß da jemand Rat – so wie vorgeschlagen läuft das Programm jedoch ebenfalls gut.

Zeile 1130 fragt ab, ob noch ein Programm gelistet werden soll. Achten Sie bitte darauf, daß nur BASIC-Programme, die als ASCII-Files vorliegen, gelistet werden können. Dafür besteht jedoch die Möglichkeit, das Laufwerk und die Extension frei zu wählen. Speichern Sie also Ihr Programm mit dem Befehl 'SAVE "M:myprog.asc",A' und rufen dann LISTER auf mit dem Befehl 'RUN "lister'. Auf die Frage, welches Programm Sie listen möchten, geben Sie nun ein:

M:myprog.asc <RETURN>

Anregung:
Zur Erhöhung des Bedienungskomforts kann u.U. noch ein DIR-Befehl für das bevorzugt benutzte Arbeitslaufwerk mit eingebunden werden.

Das Programm teilt nach der Abfrage (D)rucker oder (B)ildschirm das Monitorbild in eine Statuszeile, die das gerade zu listende Programm angibt und ein Ausgabefenster für das Listing (dafür die Definition von FNsc$Ö in Zeile 50 und flsc$Ö in Zeile 60 sowie window$Ö in Zeile 70)

```
 1000 '
 1010          IF LEFT$(zeile$(i),4)="GOTO"
               THEN  IF anf=7
                     THEN
PRINT TAB(anf);zeile$(i):PRINT TAB(7);STRING$(72,"="):GOTO 1080
                     ELSE  PRINT TAB(anf);zeile$(i);" ";STRING$(79-anf-LEN(zei
                           le$(i)),"-");:GOTO 1080
 1020 '
 1030 ' *** RETURN ***
 1040 '
 1050          IF LEFT$(zeile$(i),6)="RETURN"
               THEN  IF anf=7
                     THEN
PRINT TAB(anf);zeile$(i): PRINT TAB(7);: FOR u=7 TO 76 STEP 4: PRINT "==R";:NEXT: GOTO 1080                    ELSE  PR
INT TAB(anf);zeile$(i);" ";:FOR u=1 TO (79-anf-LE
                           N(zeile$(i)))/4: PRINT "--R";:NEXT:GOTO 1080
 1060          IF LEN(zeile$(i))+anf>80
               THEN  PRINT TAB(anf);LEFT$(zeile$(i),80-anf);:zeile$(i)=MID$(ze
                     ile$(i),80-anf+1):GOTO 1060
 1070          PRINT TAB(anf);zeile$(i);
 1080       NEXT i
 1090 WEND : CLOSE : PRINT : PRINT
 1100 IF x=8
      THEN  POKE 8793,239
 1110 ' ******* Ende der Hauptschleife ********
 1120 '
 1130
PRINT flsc$;: INPUT "Noch ein Programm listen ( j / n ) ";a$ :IF UPPER$(LEFT$(a$,1))="J" THEN RUN
ELSE  PRINT cls$ : END
 1140 IF VAL(zeile$(i))<>0
      THEN
g$=STR$(VAL(zeile$(i))):g$=RIGHT$(g$,LEN(g$)-1):zeile$(i)=MID$(zeile$(i),LEN(g$)+1):PRINT TAB(anf);g$;" ";STRING$(79-anf
-LEN(g$),"-");:GOTO 570
      ELSE  GOTO 610
 1150 ' *** Fehlerbehandlungsroutine ***
 1160 RESUME NEXT
 1170 RETURN
      --R--R--R--R--R--R--R--R--R--R--R--R--R--R--R--R--R
```

Abb. 1: So sieht ein mit LISTER bearbeitetes Programm aus...

Damit steht jetzt geordneten BASIC-Listings auch auf dem JOYCE nichts mehr im Wege. Das notwendige und sinnvolle Strukturieren, schon bei der Programmierung, nimmt Ihnen jedoch auch dieses LISTER Programm nicht ab.

(Ulrich Schmidt)

BEKANNTMACHUNG

Bei unserem
allseits bekannten und beliebten Telefon-Service,
dem »Heißen Draht«,
können Sie Ihre Fragen und Anregungen



an die Redaktion von
PC Schneider International
richten.
Auf Ihre Anrufe freuen sich:
Michael Ebbrecht
(Hardware, Joyce),
Claus Daschner (CPC) und
Heinrich Stiller
(Spiele/Adventures).

17.00 – 20.00 Uhr

HEISSER DRAHT

Tel. (0 56 51) 87 02

TOPANGEBOTE, TOPANGEBOTE



3" -Disketten, 10er Pack

Maxell CF 2 ohne Klarsichtbox + Label ..... DM 64.50
PEGASYS CF 2 DD mit Klarsichtbox + Label ..... DM 85.00
PANASONIC CF 2 mit Klarsichtbox + Label ..... DM 69.00

5,25"-Disketten, 10er Pack

PEGASYS 5,25"-DS/DD-Disketten
neutral, mit Schreibetiketten, Schreibschutz und Envelope, 100 x fehlerfrei ..... DM **9,90**

Zubehör

PEGASYS-Disketten-box FG50L für ca. 50 Stück 3"- oder 3,5"-Disketten, mit Schloß und Ersatz-schlüssel, tragbar DM **15.90**

PEGASYS-Monitorständer schwenk-, dreh- und kippbar DM **35.00**

Der Versand erfolgt per Nachnahme zuzüglich Versandkosten. **Händleranfragen erwünscht!!!**

Göddeker Computer und Zubehör GmbH
Höftestraße 32, D-4400 Münster 24, Telefon 02 51 / 61 98 81 (8.30-18.00 Uhr)

JOYCE QUALITÄTSSOFTWARE
Bedienungsfreundliche Software durch Windows und Menueführung

GYROS. Adressverwaltung mit Menueführung u. Windows DM 148.-
Adressen eingeben, ändern, loschen, suchen, blättern, Etiketten-, versch. Listen-, WordStar-Dateien sortiert.

OPTI, Aufpreis, Hebel u. Kapitaleinsatz bei Optionsscheinen DM 148.-
CARDBASE, Vereinsverwaltung für Mitglieder u. Beiträge DM 148.-
CARAT-KASSE PLUS, Einnahmen/Ausgaben überschußrechnung DM 168.-
WINDOW-LAGER, Artikel, Lieferanten, Bewegungsjournal.- DM 148.-
CARAT-PROVISION, für Handelsvertreter DM 148.-

CARAT-AUFTRAG mit Lagerverwaltung, Einkauf, Bestellwesen DM 398.-
- Auftragsabwick., OP, Mahnwesen, verschiedene Listen usw.

706 KByte Pub.-Domain-Software(randvoll) für JOYCE plus DM 59.-
Superpreise bei Restposten für Schneider CPC Computer !!!
- WordStar (incl.Mailmerge) für Joyce DM 139.- CPC6128 DM 99.-
- CBASIC-Compiler structuriertes Basic für Joyce + CPC DM 139.-
MicroMarket-Worms
Joh.- Westermann - Platz 1, D-4780 Lippstadt, Tel. 02941 / 5 92 90

```
LISTING >LISTER  <, REMARK = >'<.

< 1> 10 ' ORDNUNG in BASIC-Listings
< 2> 20 '
<93> 30 esc$=CHR$(27)
<87> 40 cls$=esc$+"E"+esc$+"H"
<23> 50 DEF FNsc$(y,x,h,b)=esc$+"X"+CHR$(32+y)+CHR$(32+
     x)+CHR$(31+h)+CHR$(31+b)
<55> 60 flsc$=FNsc$(0,0,32,90). ' Fullscreen
< 1> 70 window$=FNsc$(4,4,25,80)' Window-Funktion
<28> 75 ON ERROR GOTO 1150
< 8> 80 '
< 9> 90 ' LISTER.BAS Vers. 1.2 für JOYCE
<18> 100 ' (C) 1985 by N. Muskatewitz
<20> 110 ' (C) 1987 by U. Schmidt
<22> 120 '
<24> 130 ' ****** Programmauswahl (muss in ASCII abgesp
     eichert sein) ***
< 2> 140 PRINT cls$:PRINT "Ordnen eines BASIC - Listing
     s:":PRINT "Das zu ordnende Listing muß als ASCII-D
     atei vorliegen!":PRINT: PRINT
<85> 150 LINE INPUT "Geben Sie den Programmnamen (mit E
     xtension) ein ==> ",name$
< 2> 160 PRINT cls$;"PROGRAMM:";SPC(10);name$: PRINT ST
     RING$(80,"*"):PRINT
<18> 170 datei%=1 : OPEN "I",datei%,name$
< 8> 180 anf=7
<36> 190 '
<51> 200 DIM zeile$(30)
<96> 210 INPUT "Listing auf (D)rucker oder auf (B)ildsc
     hirm ";a$
<89> 211 PRINT cls$
<23> 220 '
<25> 230 ' ****** Drucker initialisieren ********
<27> 240 '
<35> 250 IF UPPER$(LEFT$(a$,1))="D" THEN x=8
< 6> 260 IF x=8 THEN POKE 8793,234
< 5> 270 PRINT cls$;"PROGRAMM:";SPC(10);name$: PRINT ST
     RING$(80,"*"):PRINT
<22> 280 PRINT window$;
<37> 290 '
<20> 300 ' **** Beginn der Hauptschleife *******
<22> 310 '
<11> 320 WHILE NOT EOF(datei%)
<51> 330 anf=wh*6+fo*4+7
<92> 340 zeile$=""
<29> 350 LINE INPUT #datei%,zeile$
< 4> 360 ERASE zeile$ : DIM zeile$(30)
<44> 370 zn$=DEC$(VAL(zeile$),"#####")
<22> 380 za=INSTR(1,zeile$," "):zeile$=MID$(zeile$,za+1
     )
<14> 390 l=LEN(zeile$):a=1:b=1:c=1:d=1:m=1:m1=1
<65> 400 IF INSTR(1,zeile$,":")=0 THEN zeile$(1)=zeile$
     : GOTO 510
<20> 410 WHILE (a<1) AND (b<>0)
< 6> 420 m=INSTR(zeile$,CHR$(34)): IF m=0 THEN m1=0 : G
     OTO 430
< 3> 430 b=INSTR(d,zeile$,":"): IF m=0 THEN 450
<72> 440 IF b<m OR b>m1 THEN 450 ELSE d=m1+1 : GOTO 430
< 8> 450 IF b<>0 THEN zeile$(c)=MID$(zeile$,a,b-a+1):c=
     c+1:a=b+1:d=a
<94> 460 WEND
<76> 470 IF a>1 THEN zeile$(c)=MID$(zeile$,a,l-a+1) ELS
     E IF a=1 AND m1<>0 THEN zeile$(1)=zeile$: GOTO 510
<37> 480 '
<39> 490 ' **** Ausdruck Zeilennummer *******
<22> 500 '
<36> 510 PRINT:PRINT zn$;
<26> 520 '
<28> 530 ' **** Bearbeiten der einzelnen Zeilen *******
<30> 540 '
<59> 550 zeile$(0)=STRING$(80,CHR$(34))
<16> 560 FOR i=1 TO c
<14> 570 IF zeile$(i)="" THEN 1080 ELSE IF LEN(zeile$(i
     -1))<80 THEN PRINT
<38> 580 '
<40> 590 ' **** Suche nach Anführungszeichen ********
<23> 600 '
<25> 610 m=INSTR(zeile$(i),CHR$(34)):IF m=0 THEN m1=0:
     GOTO 630
<22> 620 m1=INSTR(m+1,zeile$(i),CHR$(34))
<10> 630 IF LEFT$(zeile$(i),1)=" " THEN zeile$(i)=MID$(
     zeile$(i),2): GOTO 630
<31> 640 '
<33> 650 ' *** REM ***
<35> 660 '
<64> 670 IF LEFT$(zeile$(i),3)="REM" OR LEFT$(zeile$(i)
     ,1)="'" THEN PRINT TAB(7);zeile$(i);:GOTO 1080
<39> 680 '
<41> 690 ' *** FOR ***
<24> 700 '
<72> 710 IF LEFT$(zeile$(i),3)="FOR" THEN PRINT TAB(anf
     );zeile$(i);:fo=fo+1:anf=anf+4:GOTO 1080
<28> 720 '
<30> 730 ' *** IF ***
<32> 740 '
<27> 750 IF LEFT$(zeile$(i),2)="IF" THEN j=INSTR(zeile$
     (i),"THEN"):IF m=0 OR j<m OR j>m1 THEN PRINT TAB(a
     nf);LEFT$(zeile$(i),j-1);:zeile$(i)=MID$(zeile$(i)
     ,j+5):PRINT TAB(anf);"THEN";:anf=anf+6: GOTO 1140
<36> 760 '
<38> 770 ' *** WHILE ***
<40> 780 '
<15> 790 IF LEFT$(zeile$(i),5)="WHILE" THEN wh=wh+1: PR
     INT TAB(anf);zeile$(i);:anf=anf+6:GOTO 1080
<25> 800 '
<27> 810 ' *** WEND ***
<29> 820 '
<43> 830 IF LEFT$(zeile$(i),4)="WEND" THEN wh=wh-1:anf=
     anf-6: PRINT TAB(anf);zeile$(i);:GOTO 1080
<33> 840 '
<35> 850 ' *** NEXT ***
<37> 860 '
< 3> 870 IF LEFT$(zeile$(i),4)="NEXT" THEN fo=fo-1:anf=
     anf-4:PRINT TAB(anf);zeile$(i);:GOTO 1080
<41> 880 '
<43> 890 ' *** GOSUB ***
<26> 900 '
<71> 910 IF LEFT$(zeile$(i),5)="GOSUB" THEN PRINT TAB(a
     nf);zeile$(i);" ": FOR u=1 TO (79-anf-LEN(zeile$(i
     )))/4: PRINT "---U";:NEXT:GOTO 1080
<30> 920 '
<32> 930 ' *** ELSE ***
<34> 940 '
<41> 950 j=INSTR(zeile$(i),"ELSE"): IF j=0 OR (m<>0 AND
      j>m AND j<m1) THEN 1010
<51> 960 IF INSTR(LEFT$(zeile$(i),j-1),"GOTO") THEN PRI
     NT TAB(anf);LEFT$(zeile$(i),j-1);STRING$(79-anf-j,
     "-");:anf=anf-6: PRINT TAB(anf);"ELSE";:anf=anf+6:
     zeile$(i)=MID$(zeile$(i),j+5):GOTO 1140
<49> 970 PRINT TAB(anf);LEFT$(zeile$(i),j-1):anf=anf-6:
     PRINT TAB(anf);"ELSE";:anf=anf+6:zeile$(i)=MID$(ze
     ile$(i),j+5):GOTO 1140
<42> 980 '
<44> 990 ' *** GOTO ***
<83> 1000 '
<25> 1010 IF LEFT$(zeile$(i),4)="GOTO" THEN IF anf=7 TH
     EN PRINT TAB(anf);zeile$(i):PRINT TAB(7);STRING$(7
     2,"="):GOTO 1080 ELSE PRINT TAB(anf);zeile$(i);" "
     ;STRING$(79-anf-LEN(zeile$(i)),"-");:GOTO 1080
<89> 1020 '
<92> 1030 ' *** RETURN ***
<95> 1040 '
<62> 1050 IF LEFT$(zeile$(i),6)="RETURN" THEN IF anf=7
     THEN PRINT TAB(anf);zeile$(i): PRINT TAB(7);: FOR
     u=7 TO 76 STEP 4: PRINT "===R";:NEXT: GOTO 1080 EL
     SE PRINT TAB(anf);zeile$(i);" ";:FOR u=1 TO (79-an
     f-LEN(zeile$(i)))/4: PRINT "---R";:NEXT:GOTO 1080
<90> 1060 IF LEN(zeile$(i))+anf>80 THEN PRINT TAB(anf);
     LEFT$(zeile$(i),80-anf);:zeile$(i)=MID$(zeile$(i),
     80-anf+1):GOTO 1060
<72> 1070 PRINT TAB(anf);zeile$(i);
<96> 1080 NEXT i
< 3> 1090 WEND : CLOSE : PRINT : PRINT
<48> 1100 IF x=8 THEN POKE 8793,239
<88> 1110 ' ******* Ende der Hauptschleife ********
<91> 1120 '
<66> 1130 PRINT flsc$;: INPUT "Noch ein Programm listen
      ( j / n ) ";a$ :IF UPPER$(LEFT$(a$,1))="J" THEN R
     UN ELSE PRINT cls$ : END
<90> 1140 IF VAL(zeile$(i))<>0 THEN g$=STR$(VAL(zeile$(
     i))):g$=RIGHT$(g$,LEN(g$)-1):zeile$(i)=MID$(zeile$
     (i),LEN(g$)+1):PRINT TAB(anf);g$;" ";STRING$(79-an
     f-LEN(g$),"-");:GOTO 570 ELSE GOTO 610
< 1> 1150 ' *** Fehlerbehandlungsroutine ***
<19> 1160 RESUME NEXT
< 2> 1170 RETURN
```

DMV präsentiert

Joyce-Programmsammlung Vol. II

***Hochwertige Software** zu **Niedrigpreisen** finden Joyce-Anwender jetzt im Rahmen einer Programmsammlung in der Angebotspalette des DMV-Verlages.*

Jede Ausgabe aus dieser Reihe enthält eines oder zwei Programme, die aus verschiedenen Anwendungsgebieten kommen. Diese Serie erscheint in unregelmäßiger Reihenfolge und wird als komplettes Programmpaket mit 3" - Diskette und Bedienungsanleitung ausgeliefert.

SUPERdat

Eine universelle Dateiverwaltung für PCW 8256/8512 zur Erstellung eigener Dateien. Alle zugehörigen Programme sind in Mallard-Basic geschrieben und verwenden dessen JETSAM-Funktionen zur relativen Verwaltung der Datensätze auf Diskette.

Hinweis: Dies Programm arbeitet nicht mit Peripheriegeräten, die den Basicspeicher der PCWs verringern.

Leistungsumfang:

***MASKE** ist das Vorprogramm, mit dem Sie die Feldnamen- und -Längen sowie die Länge des Suchbegriffs voreinstellen können. Für jede gewünschte Dateiart (Adressverwaltung, Videoarchiv usw.) können Sie so auf separaten Datendisketten eine eigene Maske anlegen.*

***SUPERdat** ist das Hauptprogramm, welches die Daten der gewünschten Datei verwaltet. Neben der Eingabe von Daten in die Maske sind mehrere Sucharten, so z.B. auch Jokersuchen möglich. Jede Datei kann max. acht Felder enthalten, wovon jedes max. 40 Zeichen enthalten darf. Die Gesamtlänge eines Datensatzes darf 255 Zeichen betragen. Alle Eingaben in eine Maske können vor oder nach der Speicherung korrigiert oder verändert werden. Das Druckmenü bietet die Auswahl der auszugebenden Felder (alle, obere, untere oder bestimmte Felder) und eine Schnell-(Übersichts-)druckfunktion. Die meistverwendeten Druckerbefehle können in der Parameteroption voreingestellt werden.*

***SUPERtex** Dieses Programm stellt eine Rundschreib- (Mailmerge-)funktion für SUPERdat zur Verfügung. In einen in Laufwerk M: befindlichen ASCII-Text (z.B. mit RPED erstellt) werden automatisch vom Anwender vorausgewählte beliebige Einträge aus beliebigen SUPERdat-Dateien an beliebiger Stelle eingefügt; dieser Text wird ausgedruckt und die nächsten ausgewählten Einträge werden in den Text eingefügt. Weiterhin stellt SUPERtex auch eine Schnittstelle zu LocoScript dar; so können 30 beliebige Datensätze in eine für LocoScript lesbare Datei umgewandelt werden.*

***SUPERcal** Der Taschenrechner zu SUPERdat. Dieser bietet neben den Grundrechenarten auch Winkelfunktionen, quadratische- und Prozentfunktionen. Eine Klammerebene und Memory-Funktionen vervollständigen das Leistungsangebot dieses Programms. SUPERcal kann sowohl von der Dateiverwaltung als auch von SUPERtex aus aufgerufen werden; das Ergebnis der Kalkulation kann dem aufrufenden Programm übergeben werden.*



VOL.2 für PCW 8256/8512 incl. 3"-Disk und Bedienungsanleitung
Unverbindliche Preisempfehlung: 49,- DM

Bitte Coupon auf Seite 115 benutzen.

DMV-Verlag, Fuldaer Str. 6, Postfach 250, 3440 Eschwege



Balken-Menue – ganz einfach

Sie kennen diese Menues von vielen professionellen Anwendungen (z.B. auch Locoscript): die zur Wahl stehenden Möglichkeiten werden auf dem Bildschirm untereinander angezeigt, man bewegt mit den Cursor-Tasten einen Leuchtbalken auf die gewünschte Möglichkeit und drückt RETURN (bzw. ENTER). Der Weg zur Erstellung eigener Balken-Menues soll hier beschrieben werden...

Diese Menue-Art hebt sich von der Auswahl durch Eintippen bestimmter Kennbuchstaben oder -ziffern, wie sie von Hobby-Programmierern meistens gebraucht wird, wohltuend ab:

– es müssen keine Eingabezeichen auf dem Bildschirm »erspäht« und auf der Tastatur gesucht werden.

– die Auswahl wird besonders übersichtlich und gut lesbar dargeboten,

– eine Fehlwahl durch Vertippen ist ausgeschlossen.

Insgesamt also eine sehr benutzerfreundliche Menue-Führung. Sie können sie in BASIC auf Ihrem JOYCE ganz leicht selbst installieren.

Das »Geheimnis«

des Balken-Menues ist die Möglichkeit Ihres JOYCE, auf Kommando an einer ganz bestimmten Bildschirmstelle Zeichen erscheinen zu lassen, wobei die Angaben für Zeile und Spalte in Variablen untergebracht werden können. Diese Variablen können von einem Programm leicht beeinflußt werden, z.B. kann das Programm den Wert der Zeilen-Variable um 1 erhöhen, wenn die Taste »Pfeil-runter« betätigt wird; folgt jetzt ein PRINT-Befehl, der die erwähnte Bildschirmansteuerung verwendet, so erscheint der auszugebende Text eben eine Zeile tiefer.

Die Variable für die Zeile kann nun aber auch gleichzeitig als Index für ein STRING-Feld (=$) eingesetzt werden. Das macht man sich beim Balken-Menue zunutze: die einzelnen Auswahlmöglichkeiten werden in STRING-Feldern abgelegt, die als Index diejenige Variable haben, die in der Bildschirmansteuerung die Zeile bestimmt. Das klingt so abstrakt formuliert komplizierter als es ist: Hier ein Beispiel:

In das Feld
wahlmöglichkeit$*(z)* mit Werten für z von 1 bis 3 kommen die Texte der Menuezeilen 1 bis 3; z bestimmt zugleich die Zeile am Bildschirm. Wird jetzt z.B. z=2 gesetzt und

wahlmöglichkeit$*(z)* mit Bildschirmansteuerung »geprintet«, dann erscheint eben in Zeile 2 der Text der Menuezeile 2.

Nun etwas mehr zu dem schon mehrfach ein wenig geheimnisvoll »Bildschirmansteuerung« genannten Befehl. Sie finden ihn – und manches andere Interessante – in Ihrem Handbuch Band I, CP/M-Teil, Anhang 3. Der Befehl ist etwas länglich und so unbequem, so daß wir ihn in unserem folgenden Beispielprogramm »balkmenu.bas« (Listing 1) nur ein einziges Mal verwenden, nämlich dazu, mit seiner

Hilfe die Benutzerfunktion FN at$ zu definieren (siehe Zeile 40 in Listing 1). Damit sind wir das Befehlsungetüm los und können künftig einfach schreiben »print FNat$;....« und das Folgende wird schön in der Zeile des jeweiligen Wertes von z auf den Bildschirm gebracht.

(Wenn Sie die Definition in Zeile 40 aufmerksam betrachtet haben, werden Sie bemerkt haben, daß wir unter der Hand auch schon die Spalte festgelegt haben, und zwar mit der Variablen »links«. Die braucht sich übrigens beim Balken-Menue nicht zu verändern, weil alle Menuezeilen ja senkrecht untereinander erscheinen sollen.)

Der »Balken«

Soweit die »theoretischen Grundlagen«. Aber wo bleibt denn nun der »Balken« ? Hier ist er:

Der Balken besteht aus dem Text der jeweiligen Wahlmöglichkeit, der hinten auf eine bestimmte Länge, die Balkenbreite, mit Leerzeichen aufgefüllt (Zeile 70 in Listing 1) und am Bildschirm invers dargestellt wird. Auch die Befehle für die inverse Anzeige sind recht unhandlich, wir machen sie deshalb einfach am Programmbeginn zu Strings mit den einleuchtenden Namen ion$ und ioff$ für invers-ein und -aus (Zeile 30). Der Rest ist einfach:

Soll die Wahlmöglichkeit Nr. 3 auf dem Leuchtbalken erscheinen, so genügt der Befehl

```
z=3 : PRINT ion$;FN
at$;wahl$(z);
```

Die letzten Feinheiten

An wesentlichen Zutaten unseres Programms brauchen wir jetzt eigentlich nur noch eine Eingabe-Kontrolle. Wie sich das für ein richtiges Balken-Menue gehört, soll es ja nur auf die Tasten Pfeil-rauf, Pfeil-runter und RETURN hören, alle anderen sollen ihm nur ein müdes Piepen entlocken. Das läßt sich leicht mit zwei Zeilen wie folgt erreichen:

```
110 taste$=INKEY$:IF taste$ = "" then 10
20 IF taste$ < > CHR$(30) AND taste$ < > CHR$(31) AND taste$ < > CHR$(13) THEN print piep$: GOTO 10
```

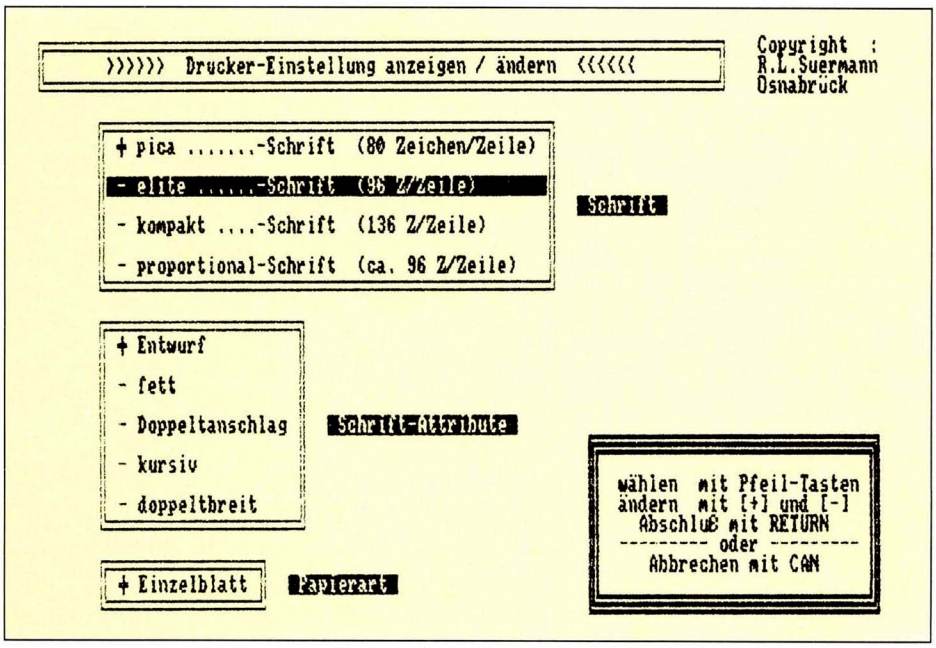


Abb. 1: das Auswahlmenue der Druckersteuerung

Was jetzt noch fehlt, nämlich die einzelnen Menue-Zeilen nach dem Signal der jeweils gedrückten Pfeil-Tasten abwechselnd invers und normal darzustellen, bringt keine besondere Schwierigkeit. Damit ist Ihnen das Prinzip des Balken-Menues auch schon klar geworden. Studieren Sie jetzt bitte noch das Listing 1. Das kleine Beispielprogramm ist auf leichteste Verständlichkeit hin formuliert und besonders ausführlich erläutert (Beim Abtippen können Sie natürlich alle Bemerkungen weglassen und auch kürzere Variablennamen verwenden.).

Wenn Sie das Programm laufen lassen, werden Sie sicher eines gleich bemerken: schön ist unser Balken-Menue ja gerade nicht. Das stimmt, uns geht es aber auch – wie man so nett sagt – »ums Prinzip«. Ist das erst begriffen, so kann die »Schönheit« leicht folgen (z.B können Sie in der Definition von FN at$ in Zeile 40 z ersetzen durch »oben+z«; durch Belegen von »links« und »oben« in Zeile 50 mit beliebigen Werten können Sie das Balken-Menue beliebig auf dem Bildschirm positionieren). Sie können auch Leerzeilen zwischen die einzelnen Menue-Zeilen einschieben, einen hübschen Rahmen herumsetzen usw.. Hier sind Ihrer Gestaltungsfreude keine Grenzen gesetzt (Anregungen hierzu können Sie dem in Listing 3 abgedruckten Programm »dr__steu.bas« entnehmen.)

Auch die Art und Weise, wie ein Balken-Menue in ein Programm eingebaut wird und den weiteren Ablauf steuert, kann sehr verschieden gestaltet werden. Prinzipiell gut geeignet ist die Verzweigung mit ON... GOTO, wie wir sie z.B. in Zeile 240 eingesetzt haben.

Der nächste Schritt Das Optionen-Menue

Mit dem Balken-Menue können wir immer nur eine von mehreren Möglichkeiten auswählen. Oft möchte man aber in einem Menue mehrere Möglichkeiten – »Optionen« gleichzeitig bejahen. Dafür gibt es das sogenannte Options-Menue. Sie kennen das z.B. vom (f4) Schriftmenue in Locoscript. Ein Optionenmenue können wir leicht aus unserem Balken-Menue entwickeln:

Vor jede Wahlmöglichkeit setzen wir einfach eine weitere Feldvariable namens »option$(z)« (wie Sie sehen, nehmen wir als Index natürlich wieder unsere bewährte Zeilen-Variable z). Diese nimmt unsere Merkmale für »Option gewählt« bzw. »Option nicht gewählt« auf, wobei die einfachsten Wahl-Merkmale das Plus- und das Minuszeichen sind. Eine gerade »erleuchtete« Menuezeile erhalten wir dann mit dem Befehl:

```
print
FNat$;ion$;option$(z);wahl$(z)
```

Bei z.B. z=2 und bejahter Option Nr. 2 würde dann in inverser Darstellung gezeigt:

"+ zweitens"

und bei verneinter Option

"- zweitens"

Das »Ein- und Ausschalten«

einer Option bietet uns keine besonderen Schwierigkeiten. Als weitere zulässige Tasten lassen wir die + und die − Tasten zu, die sich neben der Leertaste befinden. Mit diesen Tasten soll aus »+« ein »−« werden und umgekehrt. Das wars schon. Die weiteren Einzelheiten können Sie leicht dem Listing 2 (»menuplus.bas«) entnehmen. Dort sind die Änderungen angegeben, die wir in unserem Balken-Menue-Programm (»balkmenu.bas«) vornehmen müssen, um daraus ein Optionen-Menue zu machen. Auch hier geht es erst einmal nur ums Prinzip (zur Schönheit: siehe oben).

So ein Optionen-Menue kann man natürlich noch sehr viel leistungsfähiger gestalten. Nicht schlecht ist es z.B., wenn beim »Umschalten« einer Option nicht nur ein »+« zu einem »−« wird, sondern sich der Text der Wahlmöglichkeit selbst ändert. Sie kennen das z.B. vom »Entwurf«/»Korrespondenz« im Drucker-Menue, das bei Betätigung von PTR auf der untersten Zeile des Bildschirms erscheint. Eine andere Verbesserung wäre es, durch Bejahen einer Option bestimmte Andere automatisch zu verneinen, wie Sie es z.B. in dem schon erwähnten Schrift-Menue von Locoscript bei der Zeichen/Zoll-Auswahl kennen. Schließlich wäre es nicht schlecht, wenn das Optionen-Menue Eingabefehler abfangen würde. Das läßt sich alles − vergleichsweise einfach − machen, wie Sie sogleich sehen werden.

Die praktische Nutzanwendung komfortable Menue-Druckersteuerung

Der Drucker des JOYCE ist ja bekanntermaßen sehr leistungsfähig. Um so ärgerlicher ist für den BASIC-Anwender die Mühseligkeit der Druckersteuerung. Was liegt also näher, als unsere Kenntnisse von Balken- und Optionen-Menues auf diesem Feld nutzbringend einzusetzen?

Einen Lösungsvorschlag sehen Sie in Listing 3 mit dem Programm »dr__steu.bas«. Dieses läßt sich separat aufrufen und betätigen; da es nach Abmagerung nur ca 5 KB groß ist, eignet es sich aber auch als Unterprogramm in anderen Programmen.

Das Listing ist wieder ausführlich erläutert. Da die Druckersteuerung aber wirklich komfortabel und auch optisch ansprechend sein soll, ist die Sache doch ein wenig komplexer als unsere bisherigen Schlichtbeispiele. Deshalb sollen hier vorab einige Einzelheiten näher beschrieben werden.

− Es gibt drei wesensmäßig verschiedene Einstellungsvorgänge, nämlich die für die Schrift, für die Schriftattribute und für die Papierart. Dies wird im Programm dadurch hervorgehoben, daß die drei Einstellungsbereiche in separaten Blöcken mit unterschiedlicher Balkenbreite dargestellt sind, trotzdem aber alle durch die Pfeilrauf/runter-Tasten erreicht werden.

− Der »Schönheit«, aber auch der Übersichtlichkeit wegen verwendet das Programm mehrfach Umrahmungen (»Kästchen«). Hierzu wird ein Unterprogramm (Zeile 1010-1040) eingesetzt, das Sie auch aus dem Programm herauslösen und in anderen Zusammenhängen einsetzen können. Die Übergabe der Rahmenmaße (Zeile und Spalte für linke obere Ecke sowie Breite und Höhe werden bequem in einem String übergeben:

```
rahmen$ = "zz,ss,bb,hh").
```

− Ein weiteres »Schönheitsopfer« ist das Ersetzen des + Zeichens durch das von CHR$(159) gelieferte Kreuz, das dicker und damit besser sichtbar ist.

− Man kann nicht in mehreren Schriftarten gleichzeitig drucken. Das berücksichtigt das Programm und schaltet bei Wahl einer Schriftart die bisher gewählte aus (Zeile 520530).

− Man kann auch nicht ohne Schriftart drucken. Sollte man versehentlich alle Schriftarten abgewählt haben, macht das Programm hierauf lautstark und unübersehbar aufmerksam und springt zurück (Zeile 730).

− Bei den Schriftattributen Korrespondenz/Entwurf und den Papierarten Einzelblatt/Endlos wird stets die ganze gewählte Bezeichnung angezeigt. Bei Drücken von + ändert sich also etwa der Schriftzug »Korrespondenz« in »Entwurf«; vorsichtshalber hat die Minus-Taste die gleiche Wirkung. Technisch wird das im Prinzip dadurch gelöst, daß in diesen Fällen die Variable option$(z) nicht nur ein Plus- oder Minus-Zeichen enthält, sondern den ganzen Text der Wahlmöglichkeit, während wahl−(z) hier leer läuft (Zeilen 1080-1090).

− Die Optionen sind auf die Werte voreingestellt, die Ihr JOYCE beim Einschalten liefert (Zeile 160). Nach einer Änderung bleibt die neue Einstellung bis zum Ausschalten des Gerätes oder einer erneuten Änderung erhalten (die Einstellung übersteht sogar einen Drucker-Reset!). Das wird durch Abspeichern der gewählten Druckerparameter in einer internen Datei auf Laufwerk M: erreicht (Zeile 980), aus der − falls vorhanden − das Programm später die Voreinstellung einliest (Zeile 170).

− Der Auswahlvorgang wird mit RETURN abgeschlossen. Daraufhin wird der Steuercode zusammengestellt und an den Drucker abgeschickt (Zeilen 780 − 940). Daneben ist es möglich, das Programm mit der CAN-Taste einfach zu verlassen. Das ist sinnvoll, wenn die gegenwärtige Druckereinstellung nur angeschaut, aber nicht verändert wurde.

Natürlich ist auch »dr__steu.bas« noch verbesserungs- und vor allem erweiterungsfähig. Es gibt noch zahlreiche Parameter, die der komfortablen Einstellung harren. Dem BASIC-Freund eröffnet sich also noch ein weites Feld.

(R.L.Suermann)

```
LISTING >BALKMENU<, REMARK = >'<.

<12> 10. ' balkmenu.bas.... Beispiel-Programm für ein..
     ..................... Balken-Menue
<13> 20. '
<49> 30. can$=CHR$(27)+"e":caus$=CHR$(27)+"f":piep$=CHR
     $(7):ion$=CHR$(27)+"p":ioff$=CHR$(27)+"q":cls$=CHR
     $(27)+"E"+CHR$(27)+"H":' Cursor an/aus, Signalton,
     ......................... invers an/aus, Bildschirm
     ......................... frei machen
<62> 40. DEF FN at$=CHR$(13)+CHR$(27)+"Y"+CHR$(32+z)+CH
     R$(32+links):' Definition der Benutzerfunktion für
     ............... die Bildschirmansteuerung
<81> 50 links=30:zeilenzahl=3:balkenbreite=11:'.......
     . linke Begrenzung, Anzahl der Menüzeilen, Breite.
     . des Balkens
< 4> 60. wahl$(1)=" erstens":wahl$(2)=" zweitens":wahl$
     (3)=" drittens":'.................................
     Inhalt der Menüzeilen
<83> 70. FOR x=1 TO zeilenzahl:wahl$(x)=wahl$(x)+STRING
     $(balkenbreite-LEN(wahl$(x))," "):NEXT:'.........
     wahl$s mit Leerzeichen auf Balkenbreite auffüllen
<96> 80 PRINT cls$;caus$:FOR z=1 TO zeilenzahl:PRINT FN
      at$;wahl$(z):NEXT:'.............................
     alle Menuezeilen anzeigen
<64> 90 z=1:GOSUB 290 :'..............................
     Starteinstellung: 1. Menuezeile soll invers darge-
     stellt werden
<18> 100 '
<72> 110 taste$=INKEY$:IF taste$="" THEN 110
<18> 120 IF taste$<> CHR$(30) AND taste$ <> CHR$(31) AN
     D taste$ <> CHR$(13) THEN PRINT piep$:GOTO 110 :'
     nur Pfeil rauf,runter und RETURN werden akzeptiert
<24> 130 '
< 5> 140 IF taste$=CHR$(30) THEN 150 ELSE 180:'.......
     bei Taste Pfeil-runter
<76> 150 IF z=zeilenzahl THEN PRINT piep$:GOTO 110:'..
     Balken darf nicht tiefer rutschen als letzte Menue
     zeile
< 0> 160. GOSUB 300:z=z+1:GOSUB 290:'.................
     jetzige Zeile normal anzeigen; zur nächst tieferen
     Zeile, diese invers anzeigen
<32> 170 '
<47> 180 IF taste$=CHR$(31) THEN 190 ELSE 220:'.......
     bei Taste Pfeil-rauf
<54> 190 IF z=1 THEN PRINT piep$:GOTO 110:'...........
     Balken darf nicht höher rutschen als erste Zeile
<68> 200 GOSUB 300:z=z-1:GOSUB 290:'..................
     jetzige Zeile normal anzeigen; zur nächsthöheren.
     Zeile, diese invers anzeigen
<21> 210 '
<16> 220 IF taste$=CHR$(13) THEN PRINT can$:GOTO 240:'
     bei RETURN Cursor an und weiter zur Befehlsauswahl
<43> 230 GOTO 110:'....... zurück für nächste Eingabe
<81> 240 ON z GOTO 250,260,270:'......................
     weiter je nachdem, bei welcher inversen Zeile....
     RETURN gedrückt wurde
<23> 250 PRINT:PRINT "Gewählt wurde";wahl$(1):GOTO 280
<64> 260 PRINT:PRINT "Gewählt wurde";wahl$(2):GOTO 280
<84> 270 PRINT:PRINT "Gewählt wurde";wahl$(3)
<40> 280 END
<95> 290 PRINT FN at$;ion$;wahl$(z);ioff$:RETURN:'....
     Routine: inverse Anzeige der momentanen Menuezeile
<31> 300 PRINT FN at$;ioff$;wahl$(z):RETURN:'.........
     Routine: nicht-inverse Anzeige
```

```
LISTING >MENUPLUS<, REMARK = >'<.

<65> 10. '. menuplus.bas... erweitert. balkmenu.bas....
     ..................... zum Optionen-Menue.........
     ..........................................
<11> 15 ' in balkmenu.bas mit dem Befehl..............
     ...... > merge "menuplus.bas" <..................
     .... einbinden oder in balkmenu.bas die..........
     .... folgenden Zeilen einfügen bzw. ändern.......
     ..................................................
     ...
<15> 17 ' danach speichern mit........................
     ....... > save "option.bas" <.....................
     ...........
<89> 65. FOR x= 1 TO zeilenzahl: option$(x)="-":NEXT:'
     Eingabe der Option$-Zeichen, zunächst alle "-"
<89> 80 PRINT cls$;caus$:FOR z=1 TO zeilenzahl:PRINT FN
      at$;" ";option$(z);wahl$(z):NEXT:'..............
     alle Menuezeilen mit jew. option$ anzeigen
<23> 120 IF taste$<>CHR$(30) AND taste$<>CHR$(31) AND t
     aste$<>CHR$(13) AND taste$<>CHR$(22) AND taste$<>C
     HR$(28) THEN PRINT piep$:GOTO 110:'..............
     nur Pfeil rauf/runter, RETURN, + und - werden....
     akzeptiert
<77> 180 IF taste$=CHR$(31) THEN 190 ELSE 213
<60> 213 IF taste$=CHR$(22) THEN option$(z)="+":GOSUB 2
     90:'..............................................
     .. bei + Taste option$(z) ändern zu + und Zeile...
     .. invers. anzeigen
<50> 216 IF taste$=CHR$(28) THEN option$(z)="-":GOSUB 2
     90:'..............................................
     .. bei - Taste option$(z) ändern zu - und Zeile...
     .. invers anzeigen
<48> 219 '
<32> 240 PRINT:PRINT:PRINT:PRINT:PRINT"Folgende Optione
     n wurden gewählt:":PRINT:FOR x= 1 TO zeilenzahl:PR
     INT wahl$(x);option$(x):NEXT:'...................
     ... Anzeige des "Wahlergebnisses"
<29> 250 '
<31> 260 '
<33> 270 '
<11> 290 PRINT FN at$;ion$;" ";option$(z);wahl$(z);ioff
     $:RETURN:'........................................
     Routine: inverse Anzeige der momentanen Menuezeile
     mit jeweiliger Option
<58> 300 PRINT FN at$;ioff$;" ";option$(z);wahl$(z):RET
     URN:'.............................................
     Routine: nicht-inverse Anzeige der momentanen Menü
     zeile mit jeweiliger Option
```

```
LISTING >DR_STEU<, REMARK = >'<.

< 1> 10 ' dr_steu.bas.................................
     .... Drucker-Steuerung mit Options-Balken-Menue
< 2> 20 '
< 3> 30 ' Definitionen
< 4> 40 '
<58> 50 esc$=CHR$(27):cls$=esc$+"H"+esc$+"E":can$=esc$+
     "e":caus$=esc$+"f":piep$=CHR$(7):ion$=CHR$(27)+"p"
     :ioff$=CHR$(27)+"q":'.... ESCAPE, Bildschirm frei,
      Cursor an/aus, Signalton, invers an/aus
<54> 60 DEF FN at$=CHR$(13)+esc$+"Y"+CHR$(32+oben+zeile
     (z))+CHR$(32+links):DEF FNpkt$(aa,bb,zeichenkette$
     )=esc$+"Y"+CHR$(32+aa)+CHR$(32+bb)+zeichenkette$:'
     Benutzerfunktionen der Bildschirmansteuerung für.
     Menuezeilen und sonstiges
< 7> 70 '
< 8> 80 ' Menue-Angaben :
< 9> 90 '
<33> 100 links=13:oben=5:zeilenzahl=10:'. linke + obere
      Grenze und Anzahl der Menüzeilen
< 1> 110. zeile(1)=1:zeile(2)=3:zeile(3)=5:zeile(4)=7:z
     eile(5)=11:zeile(6)=13:zeile(7)=15:zeile(8)=17:zei
     le(9)=19:zeile(10)=23 :'.........................
      Bildschirm-Zeilen der Menüzeilen ab "oben"
<57> 120 FOR x=1 TO 4:balkenbreite(x)=43:NEXT:FOR x=5 T
     O 9:balkenbreite(x)=18:NEXT:balkenbreite(10)=14:'.
      Breite des Leuchtbalkens
<24> 130 '
<26> 140 ' Voreinstellung
<28> 150 '
<21> 160 IF FIND$("m:drparam")="" THEN FOR x=1 TO zeile
     nzahl:option$(x)="-":NEXT: option$(1)=CHR$(159):op
     tion$(5)=CHR$(159)+" Entwurf..... ":option$(10)=CH
     R$(159)+" Einzelblatt ":GOTO 190:ELSE 170:'......
     Voreinstellung, wenn noch keine gespeichert
<44> 170 OPEN "I",1,"m:drparam":FOR i=1 TO 10:INPUT #1,
     option$(i):NEXT:CLOSE:'..........................
     Lesen gespeicherter Voreinstellung
<34> 180 '
<41> 190 wahl$(1)=" pica .......-Schrift. (80 Zeichen/Z
```

```
     eile)"
<47> 200 wahl$(2)=" elite ......-Schrift. (96 Z/Zeile)"
< 2> 210 wahl$(3)=" kompakt ....-Schrift. (136 Z/Zeile)
     "
<33> 220 wahl$(4)=" proportional-Schrift. (ca. 96 Z/Zei
     le)"
<39> 230 wahl$(5)=""
<27> 240 wahl$(6)=" fett"
<77> 250 wahl$(7)=" Doppeltanschlag"
<46> 260 wahl$(8)=" kursiv"
<95> 270 wahl$(9)=" doppeltbreit
<58> 280 wahl$(10)=""
<37> 290 '
<20> 300 ' Bildschirm-Aufbau
<22> 310 '
<40> 320 FOR x=1 TO zeilenzahl:wahl$(x)=wahl$(x)+STRING
     $(balkenbreite(x)-LEN(wahl$(x))-LEN(option$(x)),"
     "):NEXT:'.....
     auf Balkenbreite mit Leerzeichen auffüllen
<57> 330 PRINT cls$:PRINT:PRINT TAB(14)">>>>>>. Drucker
     -Einstellung anzeigen / ändern. <<<<<<":PRINT:PRIN
     T:PRINT:'... Überschrift
<28> 340 rahmen$="01,06,68,02":GOSUB 1020:rahmen$="05,1
     2,45,08":GOSUB 1020:rahmen$="15,12,20,10":GOSUB 10
     20:rahmen$="27,12,16,02":GOSUB 1020:rahmen$="21,61
     ,28,08":PRINT ion$;:GOSUB 1020:PRINT ioff$;:'.....
     .... Umrahmungen zeigen
<55> 350 PRINT ion$;FNpkt$(9,60," Schrift "); FNpkt$(20
     ,35," Schrift-Attribute ");FNpkt$(28,31," Papierar
     t ");ioff$;FNpkt$(1,78,"Copyright. :");FNpkt$(2,78
     ,"R.L.Suermann");FNpkt$(3,78,"Osnabrück");:'.....
     Beschriftung
<20> 360 PRINT FNpkt$(23,64,"wählen. mit Pfeil-Tasten")
     ;FNpkt$(24,64,"ändern. mit ↑+↓ und ↑-↓");FNpkt$(25
     ,64,". Abschluß mit RETURN");FNpkt$(26,64,"-------
     -- oder ---------");FNpkt$(27,64,".. Abbrechen mit
      CAN"):'........ Hinweis-Kästchen
< 4> 370 PRINT caus$:FOR z=1 TO zeilenzahl:GOSUB 720:NE
     XT:' alle Menuezeilen anzeigen
<52> 380 z=1:GOSUB 710 :'. Zeile 1 invers zeigen
<38> 390 '
<21> 400 ' Zeichen-Eingabe
<23> 410 '
<28> 420 taste$=INKEY$:IF taste$="" THEN 420
<95> 430 IF taste$<> CHR$(30) AND taste$ <> CHR$(31) AN
     D taste$ <> CHR$(13) AND taste$<>CHR$(22) AND tast
     e$<>CHR$(28) AND taste$ <> CHR$(8) THEN PRINT piep
     $: GOTO 420:'. nur Pfeil-rauf,Pfeil-runter, +, -,
     RETURN und CAN werden akzeptiert
<29> 440 '
<38> 450 IF taste$=CHR$(30) THEN 460 ELSE 480:' Taste.
     Pfeil-runter
<55> 460 IF z=zeilenzahl THEN PRINT piep$:GOTO 420:ELSE
      GOSUB 720:z=z+1:GOSUB 710
<35> 470 '
<19> 480 IF taste$=CHR$(31) THEN 490 ELSE 510:' Taste.
     Pfeil-rauf
<39> 490 IF z=1 THEN PRINT piep$:GOTO 420: ELSE GOSUB 7
     20:z=z-1:GOSUB 710
<22> 500 '
< 0> 510 IF taste$=CHR$(22) THEN 520 ELSE 580:' + Taste
<77> 520 IF z<5 THEN 530 ELSE 540:'.....
     nur eine Schrift-Option gleichzeitig möglich
<91> 530 zmerk=z:FOR z=1 TO 4: option$(z)="-":GOSUB 720
     :NEXT:z=zmerk:GOTO 560
< 7> 540 IF z=10 THEN 1080 ELSE 550:'.....
      Spezialroutine bei Papierart
<47> 550 IF z=5 THEN 1090 ELSE 560:'.....
      Spezialroutine bei Korrespondenz/Entwurf
<53> 560 option$(z)=CHR$(159):GOSUB 710:'auf + schalten
<36> 570 '
<76> 580 IF taste$=CHR$(28) THEN 590 ELSE 630:'.......
     - Taste
<55> 590 IF z=5 THEN 1090 ELSE 600:'. Spezialroutine
<71> 600 IF z=10 THEN 1080 ELSE 610:' Spezialroutine
<67> 610 option$(z)="-":GOSUB 710:'... umschalten auf -

<27> 620 '
<81> 630 IF taste$=CHR$(8) THEN 990:'.. bei CAN Abbruch
<31> 640 '
< 3> 650 IF taste$=CHR$(13) THEN 730:'.....
     bei RETURN aus Eingabeteil heraus
<35> 660 '
<34> 670 GOTO 420:'.... zurück für nächste Eingabe
<39> 680 '
<41> 690 ' Routinen Zeilen invers/normal zeigen,.....
     ..... Warnung, wenn keine Schrift gewählt
<24> 700 '
<23> 710 PRINT FN at$;ion$;" ";option$(z);wahl$(z);ioff
     $;:RETURN:'..... Routine: inverse Zeile zeigen
<43> 720 PRINT FN at$;ioff$;" ";option$(z);wahl$(z);:RE
     TURN :'..... Routine: nicht-inverse Zeile zeigen
<19> 730 IF option$(1)+option$(2)+option$(3)+option$(4)
     ="----" THEN 740 ELSE 780:'.....
     Warnung und zurück, wenn keine Schrift gewählt
<98> 740 FOR i= 1 TO 5: PRINT ion$FNpkt$(11,65," keine
     Schrift gewählt !");ioff$;STRING$(5,piep$);:FOR y=
     1 TO 50:NEXT y:PRINT FNpkt$(11,65,STRING$(27," "))
     ;:FOR y=1 TO 30:NEXT y:NEXT i:PRINT caus$:GOSUB 72
     0:GOTO 380
<34> 750 '
<36> 760 ' Druckersteuercode aufbauen :
<38> 770 '
<10> 780 drucker$="":'..... drucker$ zunächst leer
<17> 790 IF option$(1) <> "-" THEN drucker$=esc$+"M"+es
     c$+"P":'..... pica-Schrift
< 5> 800 IF option$(2)<>"-" THEN drucker$=esc$+"M" :'..
     ..... elite-Schrift
< 8> 810 IF option$(3) <> "-" THEN drucker$=esc$+CHR$(1
     5):'..... kompakt-Schrift
<83> 820 IF option$(4)<>"-" THEN drucker$=esc$+"p1":'..
     ..... proportional-Schrift
<31> 830 '
<86> 840 IF option$(5) <> CHR$(159)+" Entwurf..... " TH
     EN drucker$=drucker$+esc$+"m1" ELSE drucker$=druck
     er$+esc$+"m0":'... Korrespondenz/Entwurfs-Qualität
<55> 850 IF option$(6) <> "-" THEN drucker$=drucker$+es
     c$+"E" ELSE drucker$=drucker$+esc$+"F":'... fett
<96> 860 IF option$(7) <> "-" THEN drucker$=drucker$+es
     c$+"G" ELSE drucker$=drucker$+esc$+"H":'.....
     ..... Doppeltanschlag
<72> 870 IF option$(8) <> "-" THEN drucker$=drucker$+es
     c$+"4" ELSE drucker$=drucker$+esc$+"5":'.. kursiv
<85> 880 IF option$(9) <> "-" THEN drucker$=drucker$+es
     c$+"W1" ELSE drucker$=drucker$+esc$+"W0":'.....
     ..... doppelte Breite
<43> 890 '
<12> 900 IF option$(10)<>CHR$(159)+" Einzelblatt " THEN
      drucker$=drucker$+esc$+"C"+CHR$(72)+esc$+"c"+esc$
     +"N"+CHR$(4)+esc$+"9" ELSE drucker$=drucker$+esc$+
     "C"+CHR$(70)+esc$+"$"+esc$+"N"+CHR$(3)+esc$+"8":'.
     ..... Endlospapier / Einzelblatt
<28> 910 '
<57> 920 drucker$=drucker$+esc$+"d":'.....
     ... Neueinstellung wird Defaults-Einstellung
<32> 930 '
<44> 940 LPRINT drucker$;:'. Steuercode abschicken
<36> 950 '
<40> 960 'Druckparameter speichern und Programmabschluß
<40> 970 '
<19> 980 OPEN "O",1,"m:drparam":FOR i=1 TO zeilenzahl:P
     RINT #1,option$(i):NEXT:CLOSE:'.....
     Druckparameter auf M: in Datei "drparam" speichern
<90> 990 PRINT can$;cls$;:ERASE wahl$,option$,balkenbre
     ite,zeile:END:'.....
     Bildschirm fei, Cursor an, Felder löschen, Ende
<83> 1000 '
<86> 1010 ' Routine für Umrahmungen.....
     ..... (benötigt die Rahmen-Maße Zeile+Spalte für..
     ..... linke obere Ecke, Breite, Höhe in der.....
     ..... Variablen rahmen$="zz,ss,bb,hh" )
<82> 1020. FOR i=1 TO 10 STEP 3:r(i)=VAL(MID$(rahmen$,i
     ,2)):NEXT
<21> 1030 PRINT FNpkt$(r(1),r(4),CHR$(134));:FOR i=1 TO
      r(7)-1:PRINT CHR$(138);:NEXT:PRINT CHR$(140): FOR
      i=1 TO r(10)-1:PRINT FNpkt$((r(1)+i),r(4),CHR$(13
     3));FNpkt$(r(1)+i,r(4)+r(7),CHR$(133)):NEXT
<66> 1040 PRINT FNpkt$(r(1)+r(10),r(4),CHR$(131));:FOR
     i=1 TO r(7)-1:PRINT CHR$(138);:NEXT:PRINT CHR$(137
     );:RETURN
<98> 1050 '
< 2> 1060 ' Spezialroutinen
< 5> 1070 '
< 5> 1080 IF option$(10)=CHR$(159)+" Einzelblatt " THEN
      option$(10)=CHR$(159)+" Endlospapier":GOSUB 710:G
     OTO 420 ELSE option$(10)=CHR$(159)+" Einzelblatt "
```

```
     :GOSUB 710:GOTO 420:' Spezialroutine bei Papierart
     :stets "Einzelblatt" oder "Endlospapier" zeigen
<98> 1090 IF option$(5)=CHR$(159)+" Entwurf..... " THEN
      option$(5)=CHR$(159)+" Korrespondenz":GOSUB 710:G
     OTO 420:ELSE option$(5)=CHR$(159)+" Entwurf..... "
     :GOSUB 710:GOTO 420:'....... Spezialroutine: stets
      "Entwurf" oder "Korrespondenz" zeigen
```

```
LISTING >OPTIONS<, REMARK = >'<.

<36> 5 ' options.bas. (=balkmenu.bas+menuplus.bas)
<38> 7 '
<65> 10. '. menuplus.bas... erweitert. balkmenu.bas....
     ..................... zum Optionen-Menue.........
     ..........................................
<11> 15 ' in balkmenu.bas mit dem Befehl..............
     ...... > merge "menuplus.bas" <...................
     .... einbinden oder in balkmenu.bas die...........
     .... folgenden Zeilen einfügen bzw. ändern........
     ..............................................
     ...
<15> 17 ' danach speichern mit.......................
     ....... > save "option.bas" <.....................
     ...........
<13> 20. '
<49> 30. can$=CHR$(27)+"e":caus$=CHR$(27)+"f":piep$=CHR
     $(7):ion$=CHR$(27)+"p":ioff$=CHR$(27)+"q":cls$=CHR
     $(27)+"E"+CHR$(27)+"H":' Cursor an/aus, Signalton,
     ..................... invers an/aus, Bildschirm
     ......................... frei machen
<62> 40. DEF FN at$=CHR$(13)+CHR$(27)+"Y"+CHR$(32+z)+CH
     R$(32+links):' Definition der Benutzerfunktion für
     .............. die Bildschirmansteuerung
<81> 50 links=30:zeilenzahl=3:balkenbreite=11:'........
     . linke Begrenzung, Anzahl der Menüzeilen, Breite.
     . des Balkens
< 4> 60. wahl$(1)=" erstens":wahl$(2)=" zweitens":wahl$
     (3)=" drittens":'.................................
     Inhalt der Menüzeilen
<89> 65. FOR x= 1 TO zeilenzahl: option$(x)="-":NEXT:'
     Eingabe der Option$-Zeichen, zunächst alle "-"
<83> 70. FOR x=1 TO zeilenzahl:wahl$(x)=wahl$(x)+STRING
     $(balkenbreite-LEN(wahl$(x))," "):NEXT:'.........
     wahl$s mit Leerzeichen auf Balkenbreite auffüllen
<89> 80 PRINT cls$;caus$:FOR z=1 TO zeilenzahl:PRINT FN
      at$;" ";option$(z);wahl$(z):NEXT:'..............
     alle Menuezeilen mit jew. option$ anzeigen
<64> 90 z=1:GOSUB 290 :'..............................
     Starteinstellung: 1. Menuezeile soll invers darge-
     stellt werden
```

```
<18> 100 '
<72> 110 taste$=INKEY$:IF taste$="" THEN 110
<23> 120 IF taste$<>CHR$(30) AND taste$<>CHR$(31) AND t
     aste$<>CHR$(13) AND taste$<>CHR$(22) AND taste$<>C
     HR$(28) THEN PRINT piep$:GOTO 110:'..............
     nur Pfeil rauf/runter, RETURN, + und - werden....
     akzeptiert
<24> 130 '
< 5> 140 IF taste$=CHR$(30) THEN 150 ELSE 180:'.......
     bei Taste Pfeil-runter
<76> 150 IF z=zeilenzahl THEN PRINT piep$:GOTO 110:'..
     Balken darf nicht tiefer rutschen als letzte Menue
     zeile
< 0> 160. GOSUB 300:z=z+1:GOSUB 290:'.................
     jetzige Zeile normal anzeigen; zur nächst tieferen
     Zeile, diese invers anzeigen
<32> 170 '
<77> 180 IF taste$=CHR$(31) THEN 190 ELSE 213
<54> 190 IF z=1 THEN PRINT piep$:GOTO 110:'...........
     Balken darf nicht höher rutschen als erste Zeile
<68> 200 GOSUB 300:z=z-1:GOSUB 290:'..................
     jetzige Zeile normal anzeigen; zur nächsthöheren.
     Zeile, diese invers anzeigen
<21> 210 '
<60> 213 IF taste$=CHR$(22) THEN option$(z)="+":GOSUB 2
     90:'..............................................
     .. bei + Taste option$(z) ändern zu + und Zeile...
     .. invers. anzeigen
<50> 216 IF taste$=CHR$(28) THEN option$(z)="-":GOSUB 2
     90:'..............................................
     .. bei - Taste option$(z) ändern zu - und Zeile...
     .. invers anzeigen
<48> 219 '
<16> 220 IF taste$=CHR$(13) THEN PRINT can$:GOTO 240:'
     bei RETURN Cursor an und weiter zur Befehlsauswahl
<43> 230 GOTO 110:'....... zurück für nächste Eingabe
<32> 240 PRINT:PRINT:PRINT:PRINT:PRINT"Folgende Optione
     n wurden gewählt:":PRINT:FOR x= 1 TO zeilenzahl:PR
     INT wahl$(x);option$(x):NEXT:'...................
     ... Anzeige des "Wahlergebnisses"
<29> 250 '
<31> 260 '
<33> 270 '
<40> 280 END
<11> 290 PRINT FN at$;ion$;" ";option$(z);wahl$(z);ioff
     $:RETURN:'........................................
     Routine: inverse Anzeige der momentanen Menuezeile
     mit jeweiliger Option
<58> 300 PRINT FN at$;ioff$;" ";option$(z);wahl$(z):RET
     URN:'.............................................
     Routine: nicht-inverse Anzeige der momentanen Menü
     zeile mit jeweiliger Option
```


Der JOYCE ist noch lange nicht tot!!

Noch nie konnten Sie für Ihren JOYCE so viele gute Programme und Zubehörartikel bekommen! Hier eine kleine aktuelle Auswahl aus unserem großen Angebot:

Spiele:

Batman (3-D-Action-Adventure)	47.90	Bounder (schnelles Arcadegame)	39.90
Head over Heels (3-D-Action-Adventure)	39.90	Tomahawk (3-D Hubschrauber-Simul.)	59.90
Leader Board (3-D Golf)	59.90	Frank Bruno's Boxing (Boxkampf)	56.90
Cyrus II 3-D Chess (Schach)	47.90	Distractions (3 Super-Spiele)	59.90
Strike Force Harrier (Flugsimul.)	59.90	Steve Davis Snooker (Billard)	45.90
Southern Belle + Air Traffic Control (Dampfloksimul.+Flughafentowersimulation)	49.90	JOYCESTICK-Set (Adapter,Joystick, Flugsimulator ACE) NUR *** EINFÜHRUNGSPREIS ***	99.90

Anwendungen:

Fleet Street Editor plus (Desk Top Publishing System)					199.00
Tasword 8000 (die schnelle, einfache deutsche Textverarbeitung!)					139.90
DR Draw / DR Graph	je	199.00	dBASE II / Multiplan	je	199.00
Public Domain Software	je Disk	29.90			

Gerne senden wir Ihnen unseren kompletten JOYCE-Katalog.



SUNSHINE-Software, Auf dem Schilde 14,
5270 Gummersbach, Tel. (02261) 75752
*** In Luxemburg unt. Tel. (00352) 818212 ***

Zahlungsweise: Vorkasse (Scheck) oder Nachnahme (zzgl. Nachnahmegebühren). Bei einem Auftragswert über DM 100,– liefern wir bis auf NN-Gebühren frei Haus, darunter berechnen wir eine Versandkostenpauschale von DM 5,–.

3D-Funktionenplot

Mathematische Grafik mit GSX

Nachdem in vorangegangenen Zeitschriften der PC-International das GSX-Problem für den Schneider Joyce sehr ausführlich behandelt wurde, steht nun hier ein weiteres Programm zur Verfügung, welches die Draw-Line-Funktion des GSX nutzt und sicherlich für einige Joyce-User von Interesse oder gar Nutzen ist.

Dieses 3-D-Plot-Programm berechnet die Linien eines Netzes, das auf eine von zwei Variablen abhängige mathematische Funktion projiziert wurde und stellt sie auf dem Monitor dar. Dabei werden nach Möglichkeit die gesamten Abmessungen des Monitors genutzt. Sämtliche mathematischen Funktionen des Mallard-Basic sind nutzbar und können verknüpft werden, wobei sich lange Funktionen natürlich auf die Rechengeschwindigkeit auswirken.

Das Menü des Hauptprogramms bietet außer dem Programmende noch die Möglichkeiten »Funktion editieren« und »Funktion plotten«. Bei »Funktion editieren« wird mittels des Edit-Befehls aus dem Programm heraus die Zeile editiert, die die zu plottende Funktion enthält. Eine Rückkehr ins Programm ist nur über »run« und <ENTER> möglich. Die Funktion muß im Programm abhängig sein von u und v, die x und y entsprechen. Beispiel: z=sin(sqr(u*u↑v*v)). Ist man dann ins Programm zurückgekehrt, wählt man zweckmäßig »Funktion plotten«. Zur Bestimmung des Intervalls, über dem die Funktion geplottet werden soll, wird man nun nach dem jeweils größten und kleinsten x und y gefragt. Hier gibt man sinnvollerweise die Werte ein, bei denen man eine interessante Struktur vermutet. Als Anzahl der Schritte gibt man zunächst relative kleine Zahlen, so etwa 10 an, damit man einerseits sehen kann, wie die Funktion aussieht, andererseits die Rechenzeit geringgehalten wird. Hat man dann die richtigen Werte für das Intervall gefunden, können auch höhere Werte für die Maschenzahl des Netzes eingegeben werden. Vorsicht bei zu hohen Werten! Da es zu einem Speicherüberlauf kommen kann und das Programm daraufhin stoppt, nicht zu hohe Werte eingeben. Der Wert der Schrittanzahl entspricht der Maschenzahl in x-Richtung. In yRichtung beträgt die Maschenzahl wegen der optischen Verzerrung nur das 0.7-fache der x-Richtung. Danach fragt das Programm nach einer z-Streckung. Sie gibt an, um wieviel die Funktionswerte »nach oben«, also in z-Richtung überhöht werden sollten. Bei sinusförmigen Funktionen wird dadurch quasi die Amplitude angegeben. Empfehlenswert sind im allgemeinen Werte für z um etwa 3000. Ein Ausprobieren bringt auf jeden Fall genauere Angaben. Für die Vergrößerung ist meistens der Wert 1 sinnvoll. Werte kleiner als 1 verkleinern das Bild. Da im Programm aus Gründen des beschränkten Speicherplatzes zu umfangreiche Fehlerabfragen weggelassen wurden, die auch zu Lasten der Auflösung gingen, sollte man sich bereits vor dem Plotten die Werte genau überlegen, um die Rechenzeit nicht sinnlos zu vergeuden. Bei komplizierten Funktionen mit hoher Auflösung können Rechenzeiten von mehreren Minuten entstehen. Interessant ist sicherlich, daß das Programm berechnet, welche Flächen man überhaupt sehen kann. Die unsichtbaren Flächen werden garnicht erst gezeichnet, da sie nur verwirren würden. Da im Programm keine REM-Zeilen angesprungen werden, können sie beim Tippen auch weggelassen werden. Die im Listing in der Zeile 130 auftretenden kursiven Buchstaben P und T sollten eigentlich »[« und »]« heißen. Wer eine dauerhafte Ausgabe der Grafik auf Papier haben möchte, kann sich selbstverständlich mittels <EXTRA> + <PTR> eine Hardcopy ausdrucken lassen. Auf eine GSX-mäßige Ausgabe wurde in diesem Programm verzichtet, da dies zu Lasten des für die Auflösung nötigen Speicherplatzes ginge. Viel Spaß beim Probieren aller möglicher Funktionen!

(Christoph Reimann)

Anm. d. Red.: Zum Betrieb dieses Programms benötigen Sie eine GSX-vorbereitete Arbeitsdiskette. Diese erstellen Sie wie folgt:

1) Kopieren Sie BASIC.COM (Systemdisk S.2) und GENGRAF.COM (Systemdisk S.4) mit dem Dienstprogramm PIP.COM (Systemdisk S.2) auf eine formatierte Arbeitsdiskette.

2) Geben Sie ein: GENGRAF BASIC.COM, nach ca. 30 sec. erscheint wieder A

3) Löschen Sie GENGRAF.COM durch ERA von dieser Diskette

4) Kopieren Sie von S.4 der Systemdisks die Dateien - ASSIGN.SYS - DDFXHR8.PRL - DDFXLR7.PRL - DDSCREEN.PRL - GSX.SYS auf die Arbeitsdiskette.

5.) Kopieren Sie 3DPLOT.BAS auf die Arbeitsdiskette.

Nach Erscheinen von A>geben Sie ein: BASIC 3DPLOT, und das Programm startet.

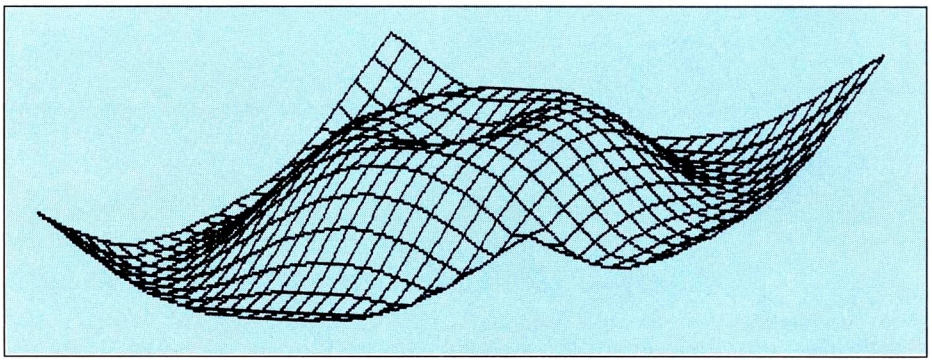


*Abb. 1: Der »Hut« entsteht durch die Formel z=sin(sqr(u*u+v*v)), die Parameter sind: Xmin/Ymin=-5; Xmax/Ymax=5; 30 Schritte; zStreckung = 3000; Zoom = 1*

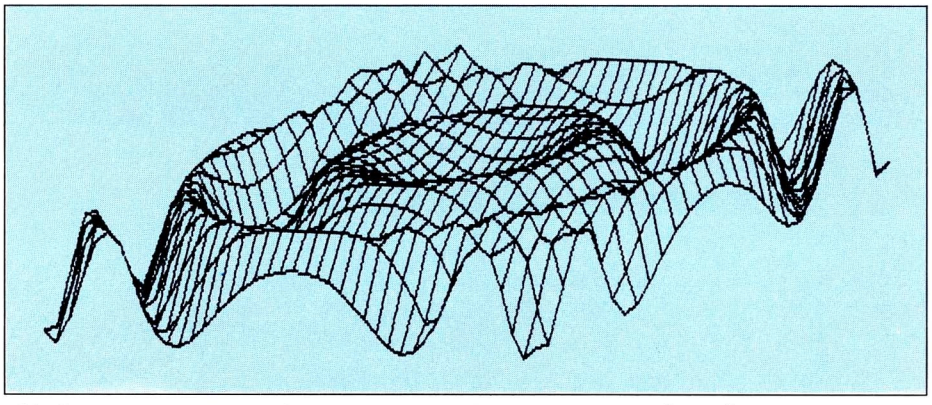


Abb. 2: Dies wundersame Gebilde erwächst aus der Formel $z=\sin(u^2 + v^2)$, Parameter: Xmin, Ymin=-3;Xmax, Ymax=3; 40 Schritte; z-Streckung = 3000; Zoom = 1

```
LISTING >3DPLOT  <, REMARK = >REM<.

<11> 10 '*******************************************
     ... *.......... 3D-Funktionenplotter.......... *
     ... *......... (C) by Christoph Reimann......... *
     ... *......................... 5100 Aachen...... *.
     .. *.......................................... *.
     ...
<68> 20 '*******************************************
<72> 30 c$=CHR$(27)+"e":co$=CHR$(27)+"f":cls$=CHR$(27)+
     "E"+CHR$(27)+"H" 'Funktionsdefinitionen
<31> 40 GOSUB 500'Initialisierung GSX
<89> 50 '*** Hauptprogramm mit Menü
<58> 60 PRINT cls$:PRINT TAB(35);"3D-Funktionenplotter"
     :PRINT:PRINT:PRINT.............................
     ..............................................
     ..............................................
     ..............................................
     ....
< 0>
<55> 70 PRINT:PRINT:PRINT:PRINT TAB(32);"Funktion editi
     eren.....1":PRINT:PRINT TAB(32);"Funktion plott
     en.....2"
<59> 80 PRINT:PRINT TAB(32);"Programmende.............
     3"
<80> 90 a$=INKEY$:IF a$<"1" OR a$>"3" THEN 90
<36> 100 w=ASC(a$)-48:ON w GOSUB 130,200,430
<62> 110 GOTO 60
<60> 120 '*** Funktionseditor
<65> 130 PRINT c$;cls$:PRINT TAB(27);"Nach dem Editiere
     n ↵ENTER↳-Taste drücken. Neustart des Programms mi
     t 'run'":PRINT
<63> 140 EDIT 170
<77> 150 RETURN
<68> 160 '*** zu plottende Funktion
<79> 170 z=SIN(SQR(u*u+v*v))
<83> 180 RETURN
<69> 190 '*** Eingabeparameter
<96> 200 PRINT cls$;c$
<55> 210 INPUT "x-min: ",kx:PRINT:INPUT"x-max: ",gx:PRI
     NT:INPUT"y-min: ",ky:PRINT:INPUT"y-max: ",gy:PRINT
     :INPUT"Anzahl Schritte (10-60): ",sx:PRINT:INPUT"z
     -Streckung (1000-20000): ",m:PRINT:INPUT"Vergrößer
     ung (1=Normal): ",g
<66> 220 PRINT co$:sy=0.7*sx:DIM x(sx,sy),y(sx,sy)
<31> 230 dx=(gx-kx)/sx:dy=(gy-ky)/sy
<87> 240 s=15000/sx*g
< 0> 250 PRINT:PRINT"Funktionswerte werden berechnet"
<83> 260 FOR i=sx TO 0 STEP -1:FOR j=sy TO 0 STEP-1
<23> 270 u=kx+i*dx:v=ky+j*dy:GOSUB 170:z=m*z
<17> 280 x(i,j)=32722!-INT(41000!/(g+2)+(i-j)*s*1.2):y(
     i,j)=INT(8786/g+j*s-z):NEXT j,i
<19> 290 PRINT:PRINT"Verdeckte Flächen werden berechnet
     "
<78> 300 FOR i=sx TO 1 STEP -1:FOR j=sy TO 1 STEP -1
<68> 310 IF y(i-1,j-1)>y(i,j) THEN y(i-1,j-1)=y(i,j)
<42> 320 NEXT j,i:PRINT cls$
<95> 330 FOR i=0 TO sx-1 :FOR j=0 TO sy-1
<78> 340 '*** Zeilen 330 - 370: Berechnung der Grafik
<12> 350 IF i*dx<32722! THEN x1=x(i,j):y1=y(i,j):x2=x(i
     +1,j):y2=y(i+1,j):GOSUB 450
<68> 360 IF j*dy<32636! THEN x1=x(i,j):y1=y(i,j):x2=x(i
     ,j+1):y2=y(i,j+1):GOSUB 450
<42> 370 NEXT j:x1=x(i,sy):y1=y(i,sy):x2=x(i+1,sy):y2=y
     (i+1,sy):GOSUB 450
< 4> 380 NEXT i
<83> 390 FOR j=0 TO sy-1:x2=x(sx,j):y2=y(sx,j):x1=x(sx,
     j+1):y1=y(sx,j+1):GOSUB 450
<37> 400 NEXT
<69> 410 WHILE INKEY$="" :WEND
< 9> 420 RUN
<85> 430 PRINT c$;cls$:END
<13> 440 '*** Draw-Line
<45> 450 contrl%(1)=6:contrl%(2)=2
<58> 460 ptsin%(1)=32722-x2:ptsin%(2)=32636-y2:ptsin%(3
     )=32722-x1:ptsin%(4)=32636-y1
<69> 470 GOSUB 670
<86> 480 RETURN
<57> 490 '*** Init GSX
<89> 500 DIM pb%(5),contrl%(10),intin%(255),ptsin%(100)
     ,intout%(45),ptsout%(100)
<85> 510 gdos%=&HBDF0:MEMORY gdos%-1
< 6> 520 POKE gdos%,14:POKE gdos%+1,115
< 2> 530 POKE gdos%+2,195:POKE gdos%+3,5:POKE gdos%+4,0
<65> 540 device%=1:GOSUB 570
<81> 550 RETURN
<88> 560 '*** Open Workstation
<34> 570 GOSUB 640
<38> 580 contrl%(1)=1:contrl%(2)=2:contrl%(4)=10
<30> 590 intin%(1)=device%
<82> 600 RESTORE 630
<34> 610 FOR i=2 TO 10:READ intin%(i):NEXT
<76> 620 GOSUB 670:RETURN
<41> 630 DATA 1,1,1,1,1,1,2,6,1
<22> 640 contrl%(1)=2:contrl%(2)=0
<82> 650 GOSUB 670:RETURN
<79> 660 '*** GDOS-Aufruf
<37> 670 pb%(1)=UNT(VARPTR(contrl%(1)))
<11> 680 pb%(2)=UNT(VARPTR(intin%(1)))
< 5> 690 pb%(3)=UNT(VARPTR(ptsin%(1)))
<94> 700 pb%(4)=UNT(VARPTR(intout%(1)))
<88> 710 pb%(5)=UNT(VARPTR(ptsout%(1)))
<67> 720 CALL gdos%(gdos%,pb%(1))
<79> 730 RETURN
```



Labelprt.bas

Das Programm ermöglicht das einfache Erstellen von Diskettenaufklebern mit getrennten Seiten A und B, sowie dem Namen der Diskette.

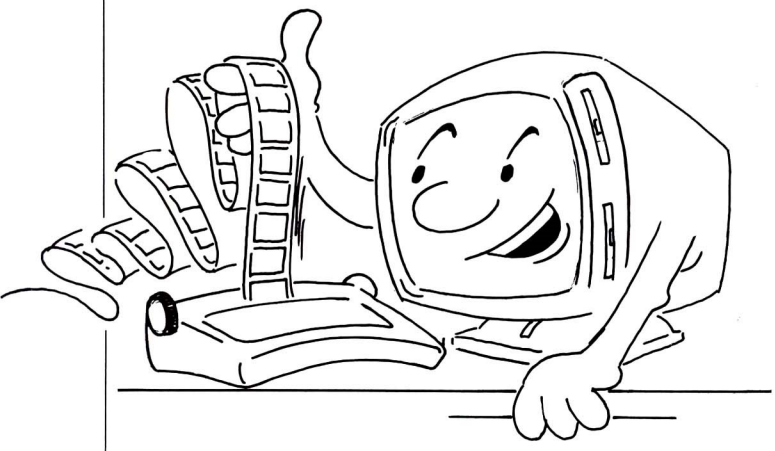
Da aus Platzgründen auch in der hochkomprimierten Schrift nur maximal 36 Dateien pro Diskettenseite ausgedruckt werden können, ist der Einsatz des Programmes nur für Laufwerk A: sinnvoll, hierfür reichen 36 Einträge fast immer aus. Für Laufwerk B: mit seiner vierfachen Kapazität sollte Labelprt sinnvollerweise nicht benutzt werden; dies ist jedoch kein Beinbruch, da B: zumeist ohnehin als Daten Only-Laufwerk genutzt wird.

Eine Programmbeschreibung erübrigt sich eigentlich; dennoch kurz zum Ablauf: Nach dem Laden mit run "labelprt" die gewünschte Diskette einlegen, den Diskettennamen eingeben und Taste drücken. Nach Aufforderung die Diskette umdrehen und Taste drückenfertig ist der Aufkleber.

(J. Brandt)

```
LISTER   .BAS   BALKMENU.BAS   DR_STEU .BAS
MENUPLUS.BAS    OPTIONS .BAS   BSP2    .PAS
PASCOM   .BAS   PRIMHEX .PAS   3DPLOT  .BAS
ADRESSEN.BAS    ADRESSEN.RND   ADRESSEN.IND
ARCHIV   .BAS   ARCHIV1 .RND   ARCHIV1 .IND
VIDEO1   .RND   VIDEO1  .IND   BIBL1   .RND
BIBL1    .IND   TON1    .RND   TON1    .IND
CURSOR   .BAS   NOTEN   .BAS   NACHTRAG.BAS
AUSWAHL  .BAS   KLLISTE .BAS   HILFSPRO.BAS
DATEN    .BAS   LISTEAUS.BAS   ZEUGNIS .BAS
FUNKTION.LOG    DEMO1   .PIC   FUN2EICH.LOG
DEMO2    .PIC

        Joyce Sonderheft 2
```

Abb. 1



```
LISTING >LABELPRT<, REMARK = >REM<.

<89> 10 REM labelprt.bas
<90> 20 REM Ein Programm zum Erstellen von Diskettenauf
     klebern (A:) mit Namen und Inhalt
<40> 30 CLEAR
<80> 40 :
<93> 50 REM micro-Schrift
<82> 60 :
<76> 70 LPRINT CHR$(27);"S";CHR$(0). :..... REM S
     uperscript an
<85> 80 LPRINT CHR$(27);"SI";CHR$(15) :..... REM K
     ompakt ein
<12> 90 LPRINT CHR$(27);"3";CHR$(20). :..... REM Z
     eilenabstand xx/256
<11> 100 PRINT CHR$(27);"E";CHR$(27);"H"
<16> 110 :
< 2> 120 REM programmstart
<20> 130 :
<63> 140 PRINT
<22> 150 PRINT :PRINT "Bitte legen Sie die Diskette mit
      Seite A in Laufwerk A:"
<67> 160 PRINT
<88> 170 INPUT "Welche Namen hat die Diskette Seite A/B
      (max. 44 Zeichen)";titel$
<71> 180 PRINT
<49> 190 PRINT :PRINT "Nun bitte die Etiketten wie folg
     t justieren :"
<44> 200 PRINT :PRINT "Oberkante Etikett = Oberkante Fa
     rbband, linke Kante Etikett = Druckkopf Start (RES
     ET)"
< 9> 210 PRINT :PRINT "Nun eine Taste drücken für Druck
     beginn "
<59> 220 WHILE INKEY$="":WEND
<18> 230 PRINT CHR$(27);"E";CHR$(27);"H"
<37> 240 WIDTH. LPRINT 45
<25> 250 :
<11> 260 REM Drucken
<29> 270 :
<11> 280 FOR i = 1 TO 36
<44> 290 a$=FIND$("*.*",i):IF a$="" THEN 320
<16> 300 LPRINT a$;". ";
<38> 310 NEXT
<79> 320 i=-i+1
<34> 330 j=INT(i/3)
<11> 340 j=-j
<12> 350 IF j >=12 THEN PRINT "Es gehen nur 36. Zeilen
     auf das Label"
<44> 360 FOR k= j TO 14
<88> 370 LPRINT
<52> 380 NEXT
<73> 390 LPRINT "..... ";titel$
<14> 400 PRINT CHR$(27);"E";CHR$(27);"H"
<24> 410 PRINT "Bitte wechseln Sie die Diskettenseite u
     nd drücken Sie danach eine Taste"
<61> 420 WHILE INKEY$="":WEND
<81> 430 LPRINT
<83> 440 LPRINT
<64> 450 FOR i=1 TO 36
<60> 460 a$=FIND$("*.*",i): IF a$="" THEN 490
<31> 470 LPRINT a$;". ";
<53> 480 NEXT
<94> 490 i=-i+1
<30> 500 j=INT(i/3)
< 7> 510 j=-j
< 8> 520 IF j >=12 THEN PRINT "Es gehen nur 36. Zeilen
     auf das Label"
<65> 530 PRINT
<26> 540 :
<12> 550 REM Drucker wieder auf Normalschrift setzen
<30> 560 :
<14> 570 LPRINT CHR$(27);"DC2";CHR$(18)
<47> 580 LPRINT CHR$(27);"T";
<93> 590 LPRINT CHR$(27);"2";
<16> 600 PRINT CHR$(27);"E";CHR$(27);"H"
```

Meine Meinung zu diesem Sonderheft:

Hier haben Sie Gelegenheit, uns Ihre Meinung zu diesem Sonderheft mitzuteilen. Wir hoffen auf eine rege Teilnahme, nur so können wir unsere Fehler erkennen, Ihre Anregungen verarbeiten und es das nächste Mal besser machen, denn – Nobody's perfect !

Füllen Sie die freigelassenen Felder für Ihre Bemerkungen aus und senden Sie das Blatt (oder eine Fotokopie, falls Sie Ihr Sonderheft nicht zerpflücken möchten) an o.g. Anschrift. Wir danken schon jetzt für Ihre aktive Unterstützung!

An den
DMV Verlag
Stichwort: Joyce-Sonderheft Nr. 2
Postfach 250
3440 Eschwege

Das finde ich gut: _____

_____

_____

Das finde ich nicht gut: _____

_____

_____

Ich wünsche mir: _____

_____

_____

_____

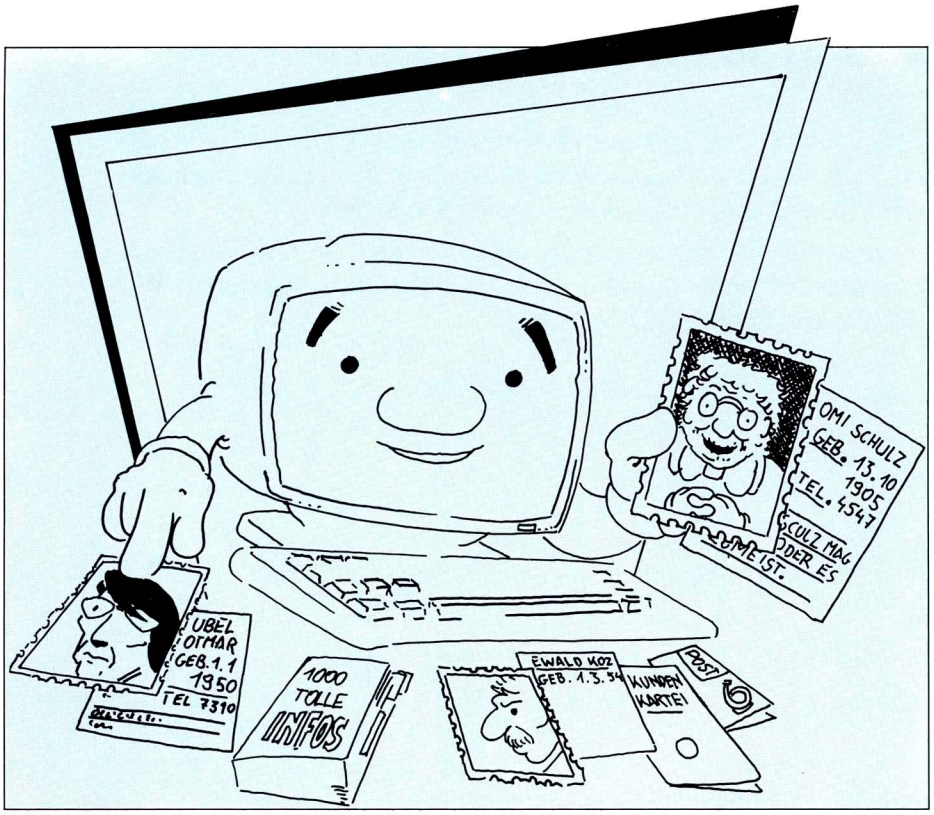
Name und Anschrift: _____

_____

_____

Computerkonfiguration: _____

Ihre persönlichen Daten werden nur zu internen Verwaltungszwecken elektronisch gespeichert.



Quick-Register

Das programmlose Programm

Wann (zum Kuckuck!) hat Tante Frieda Geburtstag?
Welche Telefon-Nummer hat die Autowerkstatt?
Wie lautet die Kunden-Nr. der Fa. Meier?
Welche Nummer hat die Krankenversicherung?
Welche Briefmarke muß auf einen Brief mit 200 g ?

Solche und ähnliche Informationen braucht man immerzu und ..... dann geht die Sucherei los!

Lassen Sie doch Ihren JOYCE (CPC 6128) Ihr Merk-Helfer sein! Wenn Sie jetzt denken: »Aha, jetzt kommt (schon wieder) ein Datenbank-Programm oder so...«, haben Sie sich getäuscht. Solche Programme – vom professionellen Datenbank-System bis zu BASIC-Notiz-Programmen – sind oft sehr leistungsstark. Sie haben aber alle einen – gewaltigen Nachteil: Sie müssen vor ihrer Benutzung eigens geladen werden. Ein wirklich schnelles Nachschauen in der elektronischen Datei ist damit praktisch unmöglich. Deshalb läßt man es in der Praxis denn auch oft genug sein und wendet sich aufseufzend doch wieder dem papierenen Zettelstapel zu. Und das phantastische Datenbank-Programm ruht jedenfalls für diese Anwendung – in Frieden in der Diskettenbox.

Es geht aber auch anders. Unser Computer liefert uns nämlich – viel einfacher und viel schneller – die gewünschte Hilfe ohne Programm;

Wie das? Nun, alles was Sie brauchen ist Ihr CP/M-Betriebssystem, und das haben Sie ja sowieso. Was Sie im »Quick-Register« gespeichert haben, können Sie jederzeit ohne irgendwelche Ladevorgänge oder andere Vorarbeiten sofort und gezielt abrufen, wenn Sie den CP/M-Prompt »>« auf dem Bildschirm haben.

Sofern Ihre Gedächtnishilfe nicht gewaltige Ausmaße annimmt, kann sie übrigens bequem auf Ihrer gewöhnlichen Startdiskette (Kopie von Seite 2 der mitgelieferten Systemdisketten) untergebracht werden; Sie brauchen also im Normalfall nicht einmal die Diskette zu wechseln!

Dinosaurier raus!

Nun ist aber die erwähnte mitgelieferte Diskettenseite randvoll gepackt. Das hindert uns aber nicht weiter, denn bei näherem Betrachten zeigt sich, daß das märchenhaft altmodische und unpraktische ED.COM – der Dinosaurier unter den Texteditoren -, das Sie garantiert noch nie ernsthaft gebraucht haben noch je brauchen werden, satte 10 K belegt. Wenn wir dazu noch ein bißchen unnützen »Kleinkram« entfernen, nämlich KEYS.WP, PROFILE.GER, SET24X80 und LANGUAGE.COM haben wir immerhin schon 14 K Platz! Also vernichten wir die genannten Programme der Reihe nach mit ERA ..... (natürlich nur auf der Kopie (!) der Systemdiskette!!).

Quick - Register rein!

Um Ihre Gedächtnisstütze auf Diskette zu bringen, empfehle ich Ihnen dringend den Texteditor RPED.BAS. Auch der ist ja schon auf unserer Diskette und läßt sich von dort bequem mit SUBMIT RPED aufrufen. Die Arbeit mit diesem Hilfsprogramm geht ganz flott und wie von Locoscript gewohnt; einige besondere Tastenwirkungen sind in der Kopfleiste des RPED-Bildschirms erläutert, sonst ist alles »normal«. Nach dem Aufruf des Editors wählen Sie mit f3 die Funktion, die sich merkwürdigerweise »neuen Bildschirminhalt erstellen« nennt, tatsächlich erstellen Sie natürlich keinen »Bildschirminhalt«, sondern eine Datei, und zwar unter dem Namen, den Sie in das sodann auftauchende Käst-

chen schreiben. Wir nennen unsere Datei »quickreg«; eine Extension (hinter dem Punkt) ist nicht nötig.

Wie das zu speichernde Register im einzelnen aufgebaut ist, sehen Sie an der Beispielsversion in Bild 1 bis 5. Sie haben im Prinzip hier völlig freie Hand. Beachten sollten Sie nur folgende Punkte:

- Die einzelnen inhaltlich unterschiedlichen Teile Ihres Verzeichnisses sollten durch Steuerzeichen voneinander getrennt werden, damit Sie nachher gezielt einen Teil aufrufen können und nicht immer das gesamte Verzeichnis durchgehen müssen. Diese Steuerzeichen sind #KENNBUCHSTABE (z.B. #A) am Beginn eines Abschnittes und # # an seinem Ende.

- Ganz oben steht eingerahmt von && die »Überschrift« von quickreg mit Hinweis auf den Aufruf der Hilfsseite (siehe Bild 1).

- Danach folgt eben diese Hilfsseite, die mit #Q beginnt (und wie alle Teile des Verzeichnisses) mit # # abschließt. Hier tragen Sie den Inhalt und die Kennbuchstaben Ihres Verzeichnisses ein, siehe Bild 1.

- Es ist ohne weiteres möglich, Unterteile der einzelnen Verzeichnisteile gezielt aufzurufen. Wie das gemacht wird, sehen Sie am Beispiel des Teilverzeichnisses »Adressenkartei« in Bild 2. Dieses Unterverzeichnis kann buchstabenweise aufgerufen werden, z.B. alle Namen mit dem Anfangsbuchstaben D. Zu diesem Zweck muß lediglich jeder Buchstabenbereich mit #ABUCHSTABE (z.B.: #AD) eingeleitet und mit # # abgeschlossen werden; die gesamte Adressenkartei wird mit # #AAA beendet.

- Wie sich eine noch »feinere« Unterteilung erreichen läßt, sehen Sie im Beispiel des Verzeichnisteils »Geburtstagsliste« (Bild 3). Hier soll nur eine Zeile angezeigt werden, nämlich der gesuchte Geburtstag. Zu

```
quickreg        Register per Betriebs-System      C,: R,L: Suermann  Osnabrück
&&
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>   quickreg   <<<<<<<<<<<<<<<<<<<<<<<<<<<<   Hilfe ---> q
&&
#Q
   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ++                                                                            ++
   ++  Hilfe zum                    Bildschirm-Anhalten / -weiter --> PTR / EXIT ++
   ++  quick - Register              Programmlauf abbrechen ---------> STOP       ++
   ++  ----------------                                                          ++
   ++                                Diese Hilfe-Anzeige zeigen -----> q         ++
   ++                                                                            ++
   ++                                                                            ++
   ++       Inhalt:                    Aufruf:                       Beispiel:    ++
   ++ ------------------------------------------------------------------------- ++
   ++  Adressen-Verzeichnis       q aBUCHSTABE                        q aC        ++
   ++           (gesamt: )        q a aaa                             q a aaa     ++
   ++                                                                            ++
   ++  Geburtstagsliste           q g%NAME                            q g%Silke   ++
   ++        (gesamt: )           q g ggg                             q g ggg     ++
   ++                                                                            ++
   ++  Post-Tarife                q p                                 q p         ++
   ++                                                                            ++
   ++  JOYCE-Bildschirm-                                                         ++
   ++  Steuer-kommandos           q j                                 q j         ++
   ++                                                                            ++
   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
##
```

Abb. 1: Das Menue des Quickregisters

```
#AA-------------------- Adressen - Verzeichnis --------------------------------
      ------ A ------
Abel, Johannes,,,,,,,,,,0228/475432,,,,,,,,5300 Bonn, Münsterplatz 1
Adam, Manfred,,,,,,,,,,,0447/2345,,,,,,,,,2444 Altenau, Peterstr, 236a
Atomanski, Rainer Dr, ,,,0334/2444,,,,,,,,,5575 Neirieden, OttilenstraBe 1
          immer mit Doktortitel ansprechen !!
Axgor, Eva,,,,,,,,,,,,,,,4556,,,,,,,,,,,,,,,2311 Wiedenbach, Siemensstra~e 13
Azurro, Pietra,,,,,,,,,,,08867/6543,,,,,,,,,8764 Oberpfarrkirchhofen, Albertus-
##                                                  Magnus-Platz 12
#AB    ------ B ------
Bandel, Eberhard,,,,,,,,,0998/65413 (oder 65414),,,,,1000 Berlin, Paulstr, 12
Bexter, Fritz,,,,,,,,,,,,kein Telefon !, ,,,8889 Praffenhofen, Sepplstr, 145
Bishop, Jack,,,,,,,,,,,,,0665/99899,,,,,,,,,5221 Kaldenkirchen, Am Wingert 13
     spricht nur Englisch (notfalls Freundin Gisela um Jbersetzung bitten !)
Buse, Fred,,,,,,,,,,,,,,,0221/454542,,,,,,,,5000 Köln, Jerichostr, 16
Butzemann, Richard,,,,,,,0331/4599,,,,,,,,,,7110 Liederbuch, Hauptstr, 3
##
#AC   ------ C ------
Clausen, Ludwig Rainer,,0667/23231,,,,,,,,,4433 Geltters, Rainstraße 17
##
#AD   ------ D ------
Dammer, Nils,,,,,,,,,,,,0887/22346,,,,,,,,,6668 Butzebach, Pilsener Str, 1
Demisch, Klaus,,,,,,,,,,0221/23487,,,,,,,,,5000 Köln, Paul Str, 88
Domani, Archangelo,,,,,,0228/476532,,,,,,,,5300 Bonn, Siegbert-Platz 34a
         ist Portugiese, kein Italiener !!
Dustmann, Xaver,,,,,,,,,07765/12345,,,,,,,,6600 Frindelnbach, Exterstr, 22
Duttmander, Peter,,,,,,,0554/3456,,,,,,,,,,5000 Köln, Wagnerplatz 33
##
#AE   ------ E ------
Eidam,Ansgar,,,,,,,,,,,,0228/43432,,,,,,,,,5300 Bonn, Rheinuferstr, 11
Eger,Helmut,,,,,,,,,,,,,05454/69998,,,,,,,,7711 Obereg, Rufus-Pl, 99a
Ersting, Lilli,,,,,,,,,,0441/220101,,,,,,,,2900 Oldenburg,
                                                  Nepomukstr, 45c
##
#AZ   ------ Z ------
Zirnowitsch, Pjotr,,,,,,00221/65434322,,,,,,Moskau/UDSSR, Roter Platz 1a
##
##AAA  -----> hier Ende Adressen-Verzeichnis
```

Abb. 2: Das Quickadressenverzeichnis

diesem Zweck wird jedem Namen #G* vorangestellt (z.B.: #G* RAINER) und jede Zeile mit dem Abschlußzeichen # # versehen.

Ans Ende der gesamten Geburtstagsliste kommt # #GGG.

Das Prinzip ist Ihnen jetzt sicherlich klar geworden. Das System ist natürlich völlig flexibel und kann von Ihnen ganz auf Ihre Wünsche angepaßt werden.

```
#G
-- ------------------------- Geburtstagsliste ------------------------------

 +++++++++++++ Januar ++++++++++++++++++++++++

#G%ADAM,.....,02,01,1966 ##

#G%MEIER, OTTO,.............,17,01,1908  -- an 80, Geburtsag denken !! ##

#%Wolf, PETER DR,...........,29,01,1944 ##


 +++++++++++++ Februar ++++++++++++++++++++++++

#G%MINA, TANTE,.............,08,02,1933  -- keine Blumen schenken ! ##

#G%CLAUS,.....,12,02,1955 ##

#G%WÖLFLING, Geschäftsführer Fa, Wölfling & Co,,,,12,02,???? ##

#G%RAINER, (Waldi),.........,13,02,1975 ##


 +++++++ März ++++++++++++++++++++++++++

#G%SILKE,.....,02,03,1974 ##

#G%UPHOFF, MICHAEL,.........,11,03,1956 ##


 ++++++++++++ April ++++++++++++++++++++++

##GGG ---> hier Ende Geburtstagsliste
```

Abb. 3: Ebenfalls möglich: der Geburtstagskalender

Zwei Mini-Dateien.....

Damit Ihre elektronische Merkhilfe mit den gewünschten Teilen auf den Bildschirm kommt, brauchen Sie jetzt nur noch zwei Dinge:

1. Eine Mini-Datei namens Q.SUB
Wir erstellen diesen Zweizeiler am einfachsten wieder mit RPED:

↑ÄE↑ÄH
PIP
CON:=quickregÄS&&↑ZQ&&↑ZÜ,
quickregÄS#$1↑ZQ# #$2↑ZÜ

(Wie das im einzelnen funktioniert können Sie im Handbuch unter PIP nachlesen!)

2. Den Befehl an den Computer, auch SUB-Dateien wie unser Q.SUB schon beim bloßen Aufruf des Namens (bei uns also: q) auszuführen.

Das geht bei eingelegter Seite 2 der Systemdisketten mit dem Kommando

SETDEF ÄORDER=(SUB,COM)

Sie können das jetzt per Tastatur eingeben. Für das künftige Arbeiten mit quickreg ist es aber viel komfortabler, Sie nehmen diese Befehlszeile in eine Datei mit dem Namen PROFILE.SUB auf (schnell erstellt mit RPED).

Dann wird künftig der Befehl beim Start des Betriebssystems automatisch gegeben.

....mit großer Wirkung

Wie läuft das »programmlose« Register-Programm nun in der Praxis?

Zum Beispiel so:

Eingabe:

q	Es erscheint die Hilfsseite
q aB	Es erscheinen alle Namen mit B aus der Adressenliste.

```
#P

================================== Post-Gebühren ====================================

                  Briefe;   Drucksachen;   Büchersdg;   Briefdrucks,;   Warensendung;
------------------------------------------------------------------------------------
Standard,.............0,80 ...........0,50 ....... -  ...........0,70 ...........0,50

bis 50 g,.............1,30 ...........0,80 ....... -  ...........1,10 ...........0,80
über 50 bis 100 g,....1,90 ...........1,10 .......0,50 ...........1,60 ...........1,10

über 100 bis 250 g,...2,50 ...........1,40 .......0,70 ...........2,10 ...........1,40
über 250 bis 500 g,...3,10 ...........2,00 .......1,00 ...........2,50 ...........2,00
über 500 bis 1000 g,..3,70 ........... - .......1,80 ........... - ........... -

Päckchen;.............3,00
Einschreibegeb;.......2,00  (eigenhändig; + 1,50   /  mit Rückschein; + 1,50)

Zahlkarte; bis 10,- DM,.....0,90
          darüber,.........1,50   (Eilzuschlag; + 3,00     telegrafisch; + 10,00)

====================================================================================
##   ---> hier Ende Post-Gebühren
```

Abb. 4: Postgebühren auf einen Blick

q a aaa Das gesamte Anschriftenverzeichnis wird angezeigt.

q p Es erscheinen die Post-Tarife

q g*Silke Es erscheint der Geburtstag von Silke

Dabei beträgt die Zeit bis zur gewünschten Anzeige durchschnittlich nur rund 10 Sekunden (und wenn Sie vom Laufwerk M aus arbeiten sogar nur rund 3 Sekunden)!!

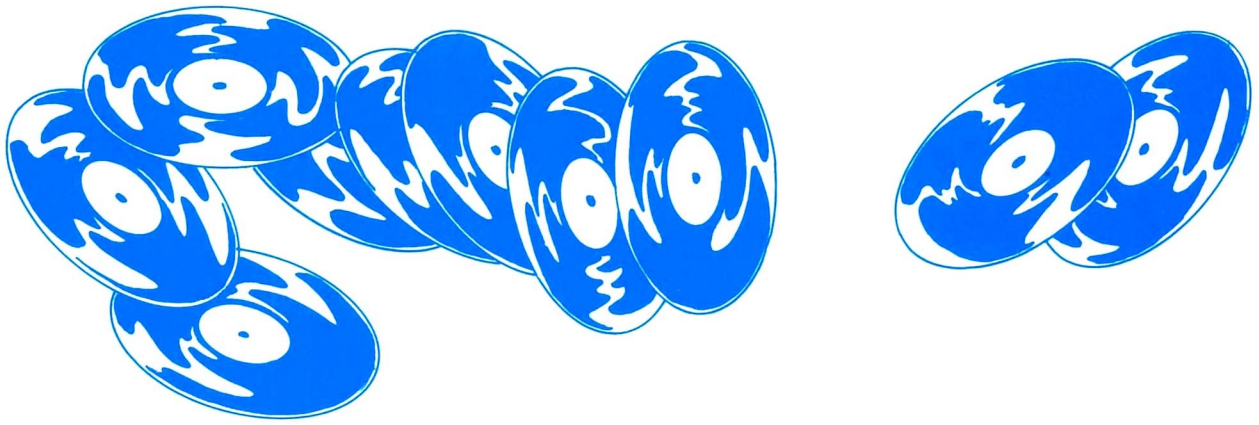
Ihr »quickreg« können Sie mit RPED komfortabel ergänzen, ändern etc... Dabei ist es ein großer Vorteil, daß an beliebiger Stelle beliebig lange Eintragungen einfach eingefügt werden können (bei den erwähnten Datenbank-Programmen müssten Sie hierfür erst neue Sätze anfügen und/oder neue Felder definieren usw.).

Und nun wünsche ich Ihnen viel Spaß mit Ihrem individuellen schnellen »Gedächtnis«.

(R.L. Suermann)

```
#J
============================== JOYCE - Bildschirmsteuercodes ========================

ESCAPE = ↑Ä  (27)  (1BH)  'ESC'

Status-Zeile;       -ein; ESC 1   -aus; ESC 0
invers;             -an;  ESC p   -aus; ESC q
unterstreichen;     -ein; ESC r   -aus; ESC u

Cursor; eine Zeile -höher;.......ESC A  (mit scroll; ESC I)  -tiefer; ESC B
        eine Spalte nach -rechts; ESC C   -links; ESC D
        nach links oben;.........ESC H
        an best. Position;.......ESC Y 32+zeile 32+spalte
        unsichtbar machen;.......ESC f
        sichtbar machen;.........ESC e
        -position speichern;.....ESC j
        an gespeicherte Position; ESC k

Löschen; ges. Darstellungsfeld;..... ESC E
         ab Cursor bis zum Seitenende:..ESC J
         ab Seitenanfang bis Cursor:....ESC d
         ab Zeilenanfang bis Cursor:....ESC o
         Cursorzeile verschwindet;.....ESC M  (nur loschen; ESC l)
         Zeichen unter dem Cursor;.....ESC N

Leer-Zeile einfügen; ESC L

Window; ESC X 32+oberezeile 32+linkespalte 31+zeilenzahl 31+spaltenbreite
-------------------------------------------------------------------------------
## --> hier Ende Bildschirmcodes
```

Abb.5: ...oder die Codes zur Bildschirmsteuerung des Joyce...



ANWENDERPAKET
für den Handwerker

HANDWERK1

für Joyce 8512
ANGEBOT – RECHNUNG – KALKULATION
AUFMASS – LEISTUNGSBESCHREIBUNG
ZEITVORGABE – NACHKALKULATION
ausführliches Handbuch

DM 998.00

DEMO-DISKETTE inkl. Kun. Bedienungsanleitung DM 35,–

KMS-SOFT KRAUTWALD
Julius-Leber-Str. 28, 2000 Hamburg 50

Anwendungsprogramme
für JOYCE PCW

ADRESCOMP - Adressendatei 58.- DM
COMFORM - Überweisungsdruck 48.- DM
DATENREM - Dateiverwaltung 68.- DM
ETATGRAF - Haushaltsbuch 58.- DM
FIBUKING - Buchführung 136.- DM
LAGDAT - Lagerdatei 68.- DM
PROFIREM - Faktu/Lager/Kunden 136.- DM
VOKABI - Vokabeltrainer 58.- DM

Fordern Sie unsere aktuelle Info J2 an

VAN DER ZALM-SOFTWARE

Elfriede van der Zalm, Software-Entwicklung & Vertrieb
Schieferstätte 1, 2949 Wangerland 3, Tel. 04461/5524

Lernen mit Zufallsaufgaben

Problembeschreibung: Erstellt man ein Basic-Programm, das nach zufälliger Auswahl Vokabeln (Rechenaufgaben) abfragen soll, so bewirkt der Zufallsgenerator, daß immer neue, zufällig erscheinende Aufgaben gestellt werden. Das funktioniert jedoch nur, solange das Programm geladen ist. Wurde der Rechner zwischenzeitlich ausgeschaltet oder lief ein anderes Programm, tritt ein unerwünschter Effekt auf: Bei jedem Neustart des Programms werden die Aufgaben in der gleichen Reihenfolge gestellt!

Das kleine Programm 'RANDOM-M' beseitigt dieses Problem (Listing 1). Die Zeilen 40/50 u. 70/80 veranlassen den Computer zu zählen, solange kein Tastendruck erfolgt. Sobald die Leertaste gedrückt wird (Zeile 60), wird die Zählschleife verlassen und der gerade anliegende Zahlenwert an den Zufallsgenerator in Zeile 90 übergeben. Damit ist die Aufgabenreihenfolge praktisch immer unterschiedlich, denn der Anfangswert des Zufallsgenerators ist jetzt abhängig von der Reaktionszeit bis zum Drücken der Leertaste. Sie müssen jetzt nur noch diese Programmzeilen mit Hilfe von 'MERGE' in Ihr bestehendes Programm einbinden.

Wer sich dabei nicht so sicher fühlt, kann den ganzen Vorgang am kleinen und leicht verständlichen Beispielprogramm "1×11" nachvollziehen:

Listing 2 ist die ursprüngliche Fassung des Lernprogramms zum Großen 1×1 mit den oben angegebenen Problemen.

1. Schritt: Listings 1 u. 2 abtippen und mit save "RANDOM-M bzw. save "1×11 abspeichern

2. Schritt: Zielprogramm auflisten und Ort für die Einbindung suchen (bei "1×11": zw. die Zeilen 120 u. 130!).

3. Schritt: 'RANDOM-M' kopieren, damit die Routine auch für eine spätere Verwendung noch zur Verfügung steht.

4. Schritt: 'RANDOM'-Kopie laden und Zeilen umnumerieren, damit beim Einbinden mit 'MERGE' keine Zeilen überschrieben werden! (hier: 121129)

5. Schritt: In das Zielprogramm (hier: '1×11') mit 'MERGE' einbinden und abspeichern!!! 3)

Jetzt kann die verbesserte Programmversion gestartet werden.

(Harald Ackermann)

Anmerkungen (für Einsteiger):

1) Kopiervorgang:

a) CP/M Diskette einlegen und SYSTEM eintippen; Return-Taste drücken. (A> erscheint)

b) PIP eintippen; Return-Taste drücken (erscheint)*

c) Diskette mit den Basic-Programmen einlegen.

*d) Hinter das * folgendes eintippen: RANDOM.BAS=RANDOM-M.BAS Return-Taste drücken. (* erscheint) Dabei wurde eine Kopie erzeugt und unter dem neuen Namen RANDOM abgelegt. Mit dieser Datei wird weitergearbeitet.*

e) Return-Taste drücken, um 'PIP' zu verlassen (A> erscheint).

f) CP/M-Diskette einlegen und das Wort BASIC eintippen; Return-Taste drücken (OK erscheint).

g) Diskette mit den Basic-Programmen wieder einlegen. Fertig! (Weiter mit Schritt 4)

2) Umnumerierung:

a) Kopie laden durch eintippen: LOAD "RANDOM Return-Taste drücken (OK erscheint)

b) Eintippen: RENUM 121,,1 Return-Taste drücken (OK erscheint)

c) Zur Kontrolle: LIST eintippen, Return-Taste drücken (Listing erscheint mit den Zeilen-Nr. 121129).

3) Einbinden:

a) Eintippen: MERGE "1×11 Return-Taste drücken (OK erscheint)

b) Eintippen: Save "1×11 Return-Taste drücken (OK erscheint). Beide Teile sind zusammengefügt und die neue Fassung ist gespeichert.

```
LISTING >RANDOM-M<, REMARK = >REM<.

<94> 10 REM*** Zufallszahl in Abhängigkeit von der Zeit
      bis zum Tastendruck ***
<78> 20 :
<66> 30 PRINT:PRINT:PRINT TAB(30)"Drücken Sie bitte die
      Leertaste!":PRINT:PRINT
< 1> 40 WHILE INKEY$=""
<51> 50.. FOR i=1 TO 10000
<48> 60.... IF INKEY$=" " THEN GOTO 90
<64> 70.. NEXT i
< 4> 80 WEND
<20> 90 RANDOMIZE(i): REM**Ergibt auch bei Neustart and
     ere Zufallszahl!**
```

```
LISTING >1X11    <, REMARK = >REM<.

<19> 10 REM********** Training Großes 1 X 1 **********
<78> 20 :
<35> 30 REM*****Bildschirm löschen*****
<80> 40 :
<61> 50 PRINT CHR$(27)"E"CHR$(27)"H"
<82> 60 :
<38> 70 REM***** Überschrift *****
<84> 80 :
<24> 90. PRINT TAB(25)"*********************************
     **********"
<51> 100 PRINT TAB (25)"*.. Trainingsaufgaben zum Große
     n 1 X 1.. *"
```

```
<24> 110 PRINT TAB(25)"*********************************
     **********"
<18> 120 :
<73> 130 REM***** Stellen der Aufgaben und Prüfung der
     Lösungen *****
<22> 140 :
<45> 150 PRINT:PRINT TAB(36) "Wieviele Aufgaben";:INPUT
      N
<44> 160 PRINT
<58> 170 R=0
<76> 180 FOR i=1 TO N
<69> 190.. Y=INT(RND(1)*10)+1
<96> 200.. Z=INT(RND(1)*10)+11
<10> 210.. PRINT TAB(40)Y"*"Z"=";
<68> 220.. INPUT E
<60> 230.. IF E=Y*Z THEN PRINT TAB(36)"Richtig!":R=R+1:
     ELSE PRINT TAB(32)"Falsch! ";Y"*"Z"="Y*Z
<41> 240 PRINT
<96> 250 NEXT i
<27> 260 :
<38> 270 REM***** Auswertung *****
<31> 280 :
<63> 290 P=R/N*100
<16> 300 PRINT TAB(25)P" % der Aufgaben wurden richtig
     gelöst!":PRINT:PRINT
<18> 310 :
<98> 320 REM***** Programmausstieg oder weiterer Durchl
     auf *****
<22> 330 :
<83> 340 PRINT TAB(32) "Noch eine Aufgabe (J/N)";: INPU
     T A$
< 2> 350 IF A$="J" OR a$="j" THEN GOTO 150 ELSE PRINT:P
     RINT:PRINT "Hat's Spaß gemacht?":PRINT:
<46> 360 PRINT "Auf Wiedersehen!"
<39> 370 END
```

Nützlich-originelle Datei-Namen unter LocoScript 1.21

Im Handbuch des JOYCE heißt es im Teil 1 der Ausgabe 1985 auf Seite 311: "Es können allerdings nicht alle Schriftzeichen zur Bildung von Dateinamen verwendet werden, nur Buchstaben von A bis Z sowie die Ziffern von 0 bis 9 eignen sich dafür." Das wäre nun für ein Textverarbeitungssystem, mit dem man womöglich ein Buch schreiben möchte, nicht gerade berauschend.

Glücklicherweise stimmt das Handbuch nicht. In Wirklichkeit akzeptiert LocoScript noch eine Reihe weiterer Zeichen in Dateinamen. Es sind dies:

- " **Anführungszeichen**
- { **Geschweifte Klammer auf (SHIFT)(ALT)(Ä)**
- $ **Dollarzeichen**
- } **Geschweifte Klammer zu (SHIFT)(ALT)(Ü)**
- % **Prozentzeichen**
- @ **kommerzielles à kaufmännisches zu-Zeichen at-sign "Klammeraffe" (EXTRA)(3)**
- ` **Accent grave (SHIFT)(↑) oder (EXTRA)(8)**
- ≈ **Tilde <EXTRA><9>**
- ' **Apostroph**
- ↑ **Potenzzeichen**
- # **Nummernzeichen**
- _ **Unterstrich**

Diese Zeichen dürfen sowohl als Dateiname als auch in der dreistelligen Extension hinter dem Punkt verwendet werden.

Der mit <EXTRA><PTR> – übrigens nach dem empfehlenswerten Hardware-Hardcopy-Umbau wie im JOYCE Sonderheft 1/87, Seite 66 ff. beschrieben, erstellte Diskverwaltungs-Ausdruck zeigt ein paar Beispiele.

Mit den Anführungszeichen " läßt sich ein stets zuerst zu lesender Hinweistext als "LIES_ZU.RST oder "READ_ME.1ST stets zuerst innerhalb der Gruppe plazieren.

Das liegt daran, daß das Anführungszeichen von den verwendbaren Zeichen für Dateinamen den niedrigsten ASCII-Wert hat, nämlich 34. Danach sortiert Locoscript die Dateien alphabetisch innerhalb der Gruppen.

Mit dem Anführungszeichen " kann man also eine Textdatei immer als erste in der Gruppe plazieren, mit der Tilde ~ ans Ende schieben. Das erste Kapitel eines Buches BUCH_KAP._1_ kann z.B. durch einen Nachtrag zu ergänzen sein, der vorsichtshalber noch nicht eingefügt wird, sondern als BUCH_KAP._1~ – sozusagen als Insert – innerhalb der Gruppe hinter dem entsprechenden Kapitel einsortiert wird.

Mit dem "Klammeraffen" @, der in Süddeutschland auch "Schneckennudel" genannt wird, können Dateien beispielsweise zwischen Zahlen und Buchstaben eingereiht werden:

Die Ziffern 1...9 entsprechen ASCII 49...57, die großen Buchstaben A...Z entsprechen ASCII 65-90. Der "Klammeraffe" hat uner Locoscript den Wert 64 (wie das Paragraphenzeichen § unter CP=MPlus, BASIC oder dBASEII).

Mit den immerhin genau ein Dutzend zusätzlich erlaubten Zeichen für Dateinamen sind eine Reihe von Anwendungen möglich. Das BASIC-Programm ASCIWERT.BAS gibt Ihnen die ASCII-Werte aller JOYCE-Zeichen aus. Bei der Tilde und dem Klammeraffen macht LocoScript 1.21 eine Ausnahme gegenüber reinen ASCII-Dateien. Das stört aber den Nutzen der zusätzlich möglichen Textnamen überhaupt nicht. Übrigens: Die Zeichen dürfen auch bei Diskettennamen und Gruppennamen verwendet werden.

(G.Ott)

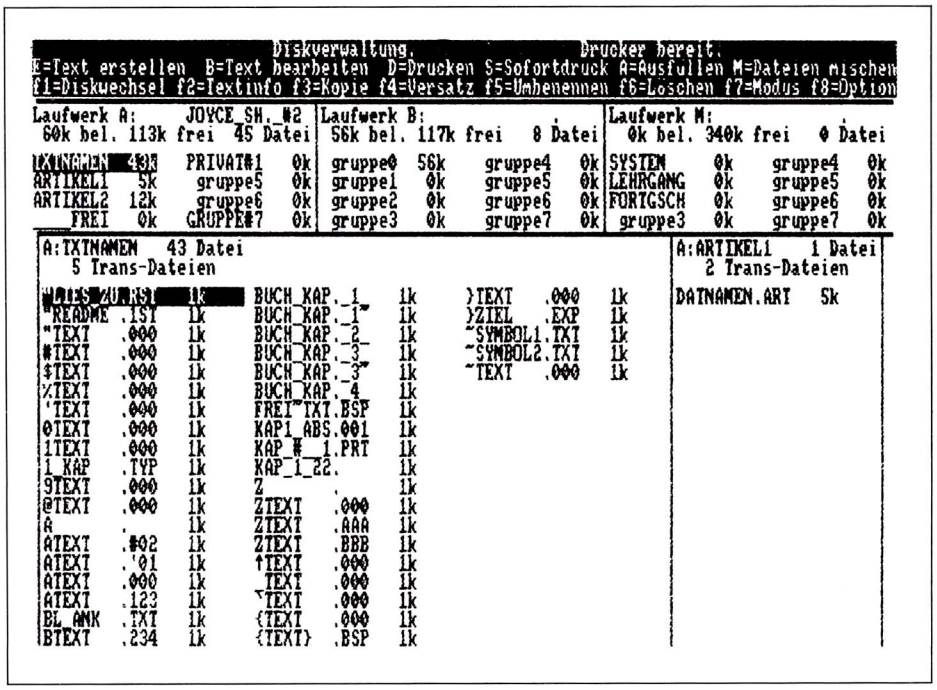


Abb. 1: ein Directory mit Hilfe von Sonderzeichen sortiert...

```
LISTING >ASCIWERT<, REMARK = >REM<.

<89> 10 REM asciwert
<67> 20.... FOR i=33 TO 255
<96> 30..... PRINT i;" = ";CHR$(i);
<91> 40.... NEXT
<32> 50 END
```

Abb. 2

Spaltensatz unter LocoScript problemlos realisiert

Die Möglichkeiten eines Programms oder Betriebssystems sind, wie die Erfahrung zeigt, keineswegs in den dazugehörigen Handbüchern vollständig erläutert oder dokumentiert. So ist zum Beispiel die Erstellung von LOCOSCRIPT-Texten im Spaltensatz (Zeitschriftensatz) im JOYCE-Handbuch mit keinem Wort erwähnt, obwohl der Spaltensatz für JOYCE eigentlich überhaupt kein Problem darstellt.

Alle Funktionen, die für den Spaltensatz erforderlich sind, bietet LOCOSCRIPT von Haus aus an, nämlich: – frei wählbares Layout (Textbreite), – mehrere Layouts in einem Text, – automatischen Zeilenumbruch und – Blocksatz. Es kommt also nur noch auf die gelungene Kombination dieser Möglichkeiten an!

Arbeitsanleitung

Dieser Text enthält die Arbeitsanleitung für die Erstellung von LOCOSCRIPT-Texten im Spaltensatz (Zeitschriftensatz). Dies wird am Beispiel für einen zweispaltigen Text erklärt; es ist aber prinzipiell auch jede höhere Zahl von Spalten denkbar. Bei Wahl einer kleineren Schriftbreite oder voller Ausnutzung der möglichen Breite der Druckzeilen wird auch drei- oder gar vierspaltiger Satz interessant. Als Erstes müssen die erforderlichen Layouts erstellt werden (vgl. Abb. 1-3). Hierbei empfiehlt es sich – wie stets beim Arbeiten mit LocoScript – als Schriftgröße des Basislayouts die im Text am häufigsten verwendete Schrift einzustellen. Im vorliegenden Beispiel ist dies die Proportionalschrift.

Jetzt kommt die Steuerung des spaltenweisen Umbruchs an die Reihe: Hierzu müssen die Codes und Zeichen
[(+LayouT2)(+ZlAbst0)]
auf den Schirm gebracht und mit
[COPY/AUSBL 2]
auf die Taste »2« und
[(+LayouT1)]
mit
[COPY/AUSBL 1]

auf die Taste »1« gelegt werden. (Die Leerstellen vor und nach den Codes - keine Hard-Trenns! – sind wichtig!) Diese Kombinationen werden nun am Zeilenende jeweils statt des »RETURN« durch [EINBL 1] bzw. [EINBL 2] an die Zeilenenden gesetzt, und zwar an bzw. kurz vor der Stelle, an der LOCOSCRIPT den automatischen Zeilenumbruch vornimmt. Dabei bedeutet EINBL 1: Es folgt Spalte 1 und EINBL 2: Es folgt Spalte 2. Der Blocksatz (bündig) bleibt so in beiden Spalten erhalten!

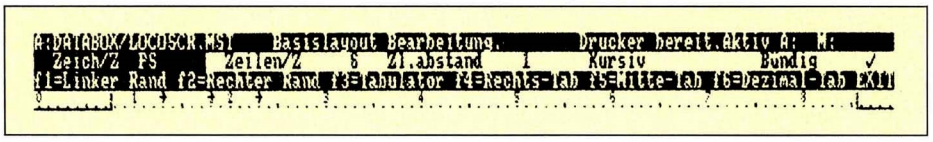


Abb. 1: Das Basislayout...

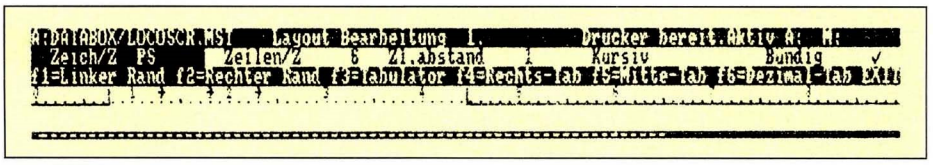


Abb. 2: Layout1, die linke Spalte...

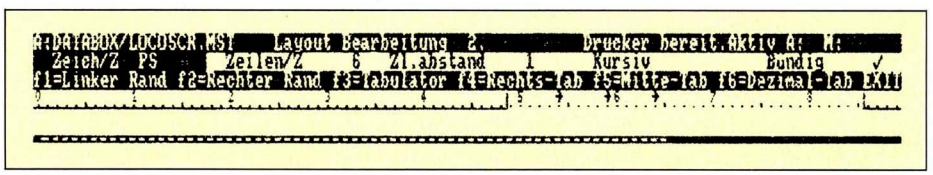


Abb. 3: Layout2, die rechte Spalte...

Es ist aber erforderlich, die auf gleicher Höhe erscheinenden Zeilen beider Spalten jeweils nacheinander – also von Spalte zu Spalte wechselnd – einzutippen.

Jetzt ist alles richtig vorbereitet und es kann losgehen! Allerdings wird das »Bäumchenwechseldich« schwierig, wenn man nicht vorher bereits den Text im Entwurf spaltenweise aufgeteilt hat. Um dies zu umschiffen, kann man folgende »Zeilenersatzfunktion« benutzen: Die Taste »0« wird
(durch [COPY/AUSBL 0])
mit folgendem Text belegt:
[(+LayouT2)(+ZlAbst0)
0000000000000000000000000000000
0 (+LayouT1)].

(Die Leerstellen vor und nach den Codes – keine Hard-Trenns! – sind wichtig!) Nach Aufruf des (+LayouT1) für die erste Spalte kann der Text dieser Spalte nun durchgehend bis zum Seitenende eingegeben werden, wobei an jedes Zeilenende [EINBL 0] zu setzen ist (Dies gilt nicht, wenn die zweite Spalte auf entsprechender Höhe leer bleiben soll. Dann ist ohne jedes RETURN einfach weiterzu schreiben.). Damit wird die jeweils entsprechende Zeile der zweiten Spalte zunächst in voller Länge mit Nullen gefüllt, was den Wagenrücklauf zur nächsten Zeile der ersten Spalte bewirkt.

Wird eine Zeile nicht voll beschrieben, sondern endet dort ein Absatz, so kann ein RETURN gesetzt werden. Hierbei muß man – je nachdem, in welcher Spalte man sich befindet – aufpassen, ob der Zeilenvorschub (+ZlAbst0) oder (+ZlAbst1) stimmt.

Soll nun, nachdem in der ersten Spalte das Seitenende erreicht ist, die zweite Spalte beschrieben werden, so geht

man an den Seitenanfang zurück und benutzt jetzt jeweils die Suchfunktion von LOCOSCRIPT, um den jeweiligen Zeilenanfang zu finden.

Zweckmäßig ist es, nach »0000« zu suchen. Ist der nächste Zeilenanfang erreicht, löscht man mit »DEL->« sämtliche Nullen aus und schreibt den gewünschten Text in die Zeile, bis sie voll ist, d.h. man trennt oder unterbricht, kurz bevor LOCOSCRIPT durch automatischen Zeilenumbruch eine neue Zeile anfängt. Hat man zuviel eingetippt, kann man es mit <-DEL« wieder löschen.

Die Abb.4 und 5 zeigen die Bildschirmwiedergabe eines Mustertextes einmal mit und einmal ohne Anzeige von Codes und Linealen.

Es sei noch einmal darauf hingewiesen, daß innerhalb des Textes, auch beim Layoutwechsel, keine [RETURNS] gesetzt werden dürfen, es sei denn, daß ein Absatz zu Ende ist; nur so bleibt der Blocksatz (bündig) in beiden Spalten erhalten!

Und noch ein Trick, um das unerwartete Hin- und Herschieben kurzer Wörter zu vermeiden: Die Leerstellen zwischen dem ersten und zweiten Wort einer Zeile einerseits und zwischen dem vorletzten und dem letzten Wort andererseits sollten als »Hard-Space« ausgeführt werden (Bei langen Wörtern überflüssig!).

(M.Meyer)

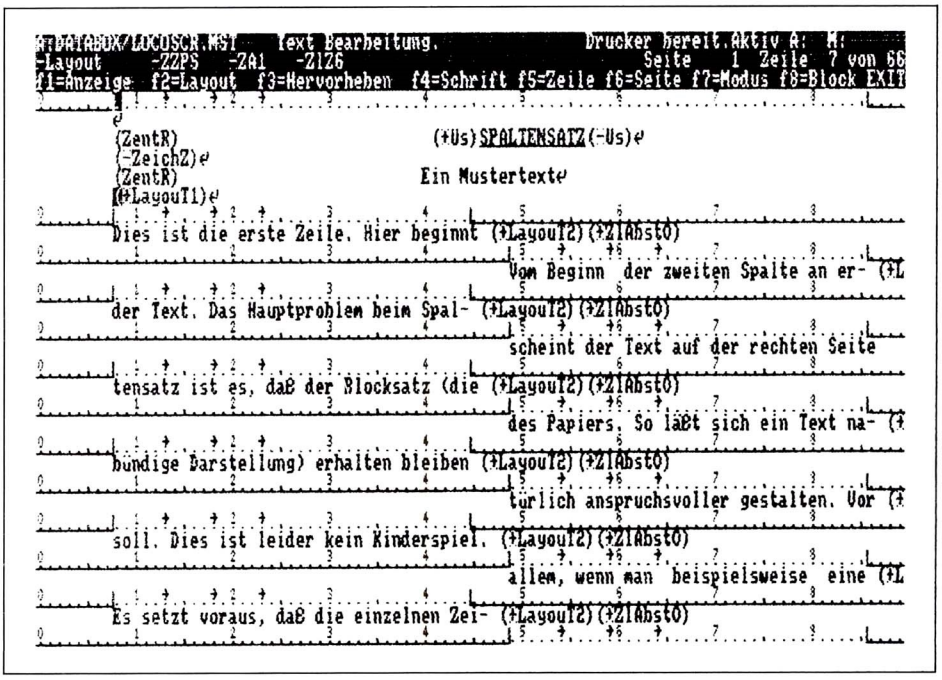


Abb.4: Ein Mustertext mit...

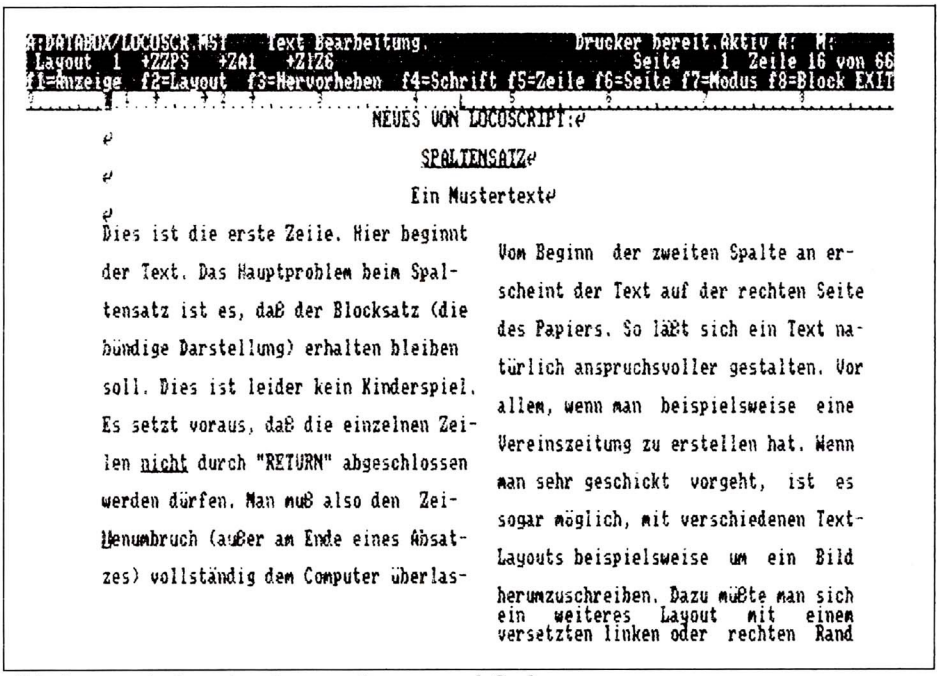


Abb.5: ...und ohne Anzeige von Layout und Codes

C COMPILER

MI-C für CP/M, CP/M 86, MS-DOS

vereint hohen Bedienungskomfort mit hervorragender Leistung

- Vollständige Version mit 13stelliger BCD-Arithmetik für Gleitkommazahlen
- Erzeugt kurze und schnelle Programme, die auch in ein ROM gebracht werden können.
- Ausgabe in Z80-, 8080-, 8086-Assemblercode
- Kompatibel zu M80/L80 (MASM) von Microsoft
- Fehlerverfolgung mittels Trace möglich
- Umfangreiche Bibliothek inkl. math. Funktionen
- für MS-DOS/CP/M86. 4 Speichermodelle
- 8087 Math. Prozessor Unterstützung enthalten
- AMD 9511 Unterstützung erhältlich
- Unix-kompatibel
- Deutsche oder englische Version lieferbar
- 8"-/5.25"-/3.5"-/3"-Disk+dt. Handbuch

MI-C für CP/M	445,– DM
MI-C für CP/M86, MS-DOS	575,– DM
MI-C Crosscompiler (Ziel Z80/8080)	745,– DM
MI-C Crossassembler+Linker	645,– DM
MI-C Crosspaket (Ziel 8051)	1495,– DM

Herbert Rose EDV, Bogenstraße 32, 4390 Gladbeck, Telefon (0 20 43) 2 49 12, 4 35 97
Vertrieb in Österreich:
Dr. Willibald Kraml, Microcomputer-Software, Degengasse 27/16, A-1030 Wien

Effekta *bietet:*

Wirtschaftliche Computerprogramme für die Arztpraxis

auf

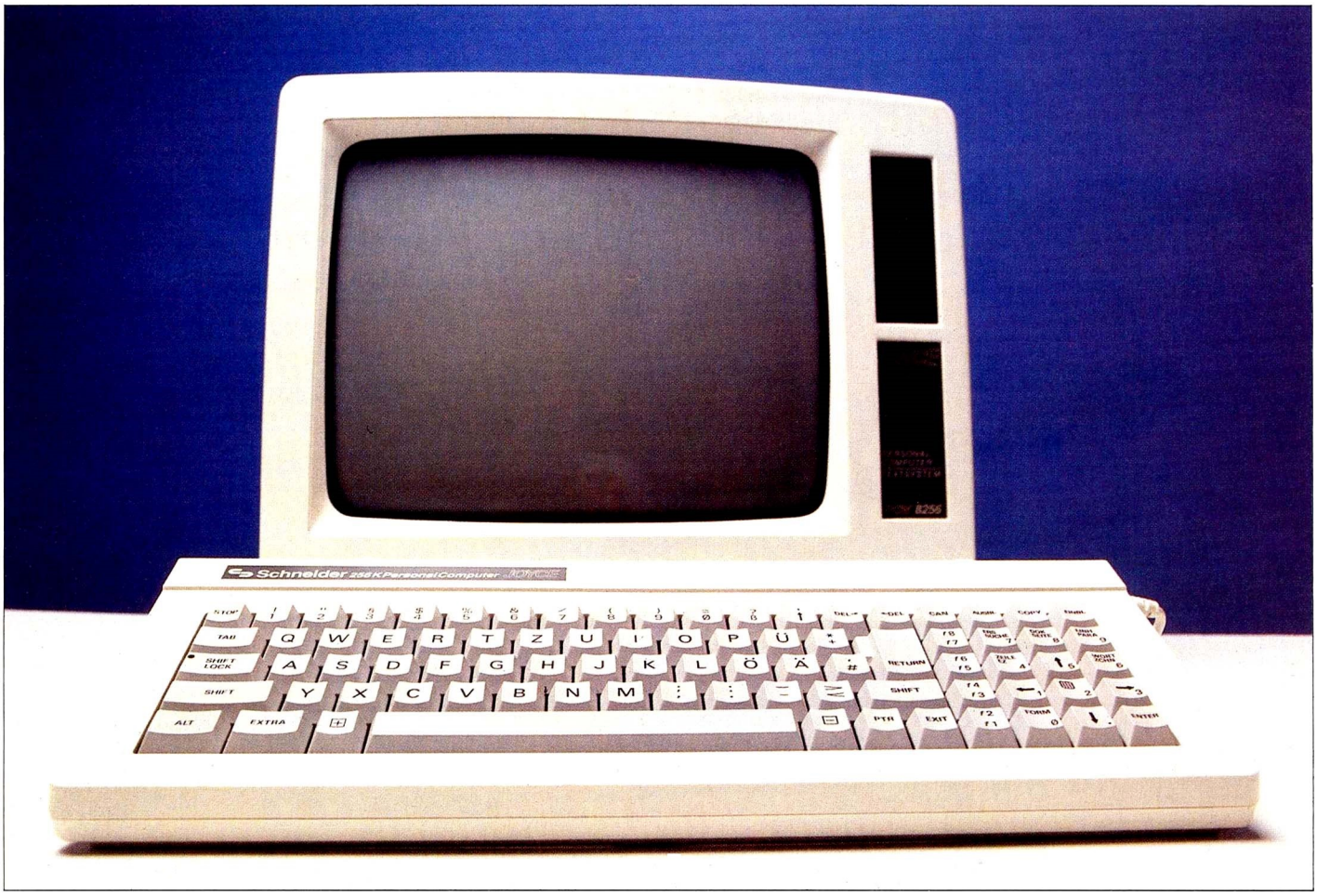
Schneider CPC, Joyce, PC und IBM-Kompatiblen mit

★ ★ **persönlicher Kundenberatung** ★ ★

und

★ ★ **individueller Anpassung** ★ ★

Effekta, Am Wiggert 9c, 4500 Osnabrück, Tel. 05 41/44 24 16



F
K
E
Y
S

komfortable Funktionstastenbelegung unter BASIC

Das Belegen der Funktionstasten ist unter CP/M nicht sehr komfortabel. Deshalb entstand dieses Programm. Es ist nötig, daß Sie sich vor dem Starten des Programms das CP/M-Dienstprogramm SETKEYS.COM und den Basic-Interpreter BASIC.COM auf die Arbeitsdiskette oder in die Ram-Disk kopieren.

Nach dem Start durch BASIC F-TASTEN können Sie durch Auswahl des entsprechenden Menüpunktes eine neue Definitionsdatei erstellen oder eine Alte editieren.

Wenn Sie die Funktionstasten mit Kommandos belegen und Sie wollen, daß diese mit Benutzung der entsprechenden Funktionstaste ausgeführt werden, müssen Sie im Editor hinter die Kommandos ein ↑M (Darstellung des Control-Codes für RETURN) setzen. Wenn Sie die Darstellung des RETURN-Codes weglassen, wird die Belegung der F-Taste nur auf den Bildschirm ausgegeben. Erst wenn Sie danach die RETURN-Taste drücken, wird die Belegung abgeschickt.

Es sei noch darauf hingewiesen, daß die Taste [AUSBL] im Editor eine ganze Zeile löscht, während Sie mit [EINBL] Zeichen einfügen können. Die [CAN]-Taste bewirkt im Editor und bei allen anderen Programmpunkten einen Rücksprung.

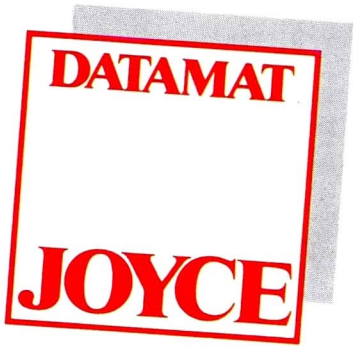
Nachdem Sie einige oder alle Funktionstasten belegt haben, können Sie den Editor durch [EXIT] verlassen. Danach schreibt das Programm die Definitionsdatei auf Diskette.

Nun können Sie diese Datei durch SETKEYS <name>.KEY aktivieren. Damit sind die Funktionstasten nach Ihren Wünschen belegt.

(A. Schmidt)

JOYCE Komplett

Richtig, der kleine Profi-Rechner von Schneider wird von Hause aus komplett mit einem Textverarbeitungsprogramm geliefert. Das heißt aber noch lange nicht, daß damit die Fähigkeiten dieses preiswerten Computers erschöpft sind. Was der Joyce kann und wie Sie das Beste aus ihm herausholen, liegt ganz allein an Ihnen.



Einerseits können Sie Ihrem Joyce mit fertigen Programmen neue Fähigkeiten beibringen. So wird der Joyce beispielsweise durch den Einsatz von DATAMAT Joyce zu einem kinderleicht zu bedienenden Datenverarbeitungssystem mit vielfachen Anwendungsmöglichkeiten. Sie können dann mit dem Joyce sowohl Ihre private Plattensammlung archivieren als auch die Übersicht über die Lagerbestände Ihrer Firma behalten. Erstaunlich ist auch der Preis von nur DM 99,–.

Wenn Sie allerdings lieber ein wenig Zeit investieren möchten, um den Joyce ganz individuell auf Ihre Anwendungsbereiche zuzuschneiden, könnte sach- und fachkundige Hilfe nicht schaden. Ganz gleich, ob Sie nun Einsteiger sind, fortgeschrittener Anwender, der Tips & Tricks für den Joyce sucht oder eine kompakte, übersichtliche Befehlserläuterung für Ihre tägliche Arbeit brauchen – DATA BECKER kann Ihnen Pfadfinder an die Hand geben, die dafür sorgen, daß Sie auch in Zukunft viel Freude mit Ihrem Joyce haben:



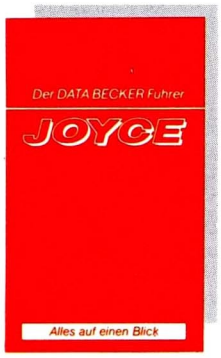
Dieses Buch hilft Ihnen, den Joyce von der Pike auf kennenzulernen, seine kleinen Eigenheiten zu verstehen und auszunutzen. Vom Auspacken und Anschließen über eine umfangreiche, ausführliche Einführung in die mitgelieferte Textverarbeitung Locoscript bis hin zu Schnellkursen für Programmiersprachen und Informationen zum Betriebssystem CP/M-Plus ist alles drin. Eben ein DATA BECKER Einsteiger-Buch.

JOYCE für Einsteiger
248 Seiten, DM 29,–



Wenn Sie den Einstieg in die Joyce-Welt geschafft haben, kommt unweigerlich der Punkt, an dem Sie mehr über den Joyce und sein Innenleben erfahren möchten. Seien es die CPM-Dienstprogramme oder Einführungen in die wichtigsten Standardprogramme – dieses Buch macht endgültig Schluß mit allen Problemen, die es bei der Anwendung und Programmierung des Joyce gibt.

Das große JOYCE-Buch
Hardcover
424 Seiten, DM 59,–



Jetzt sind Sie Joyce-Profi. Aber auch ein Profi kann unmöglich alle Befehle und Speicheraufteilungen jederzeit im Kopf haben. Sie zur Hand zu haben, ist dagegen kein Problem, wenn dieses Buch seinen Platz neben dem Computer eingenommen hat. Schnell und gezielt finden Sie die Lösung für Ihre Probleme – kurz und knapp, denn der DATA BECKER Führer ist das Nachschlagewerk für Profis und kein Lehrbuch.

Der DATA BECKER Führer zum JOYCE
180 Seiten, DM 29,80

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name
Straße
Ort

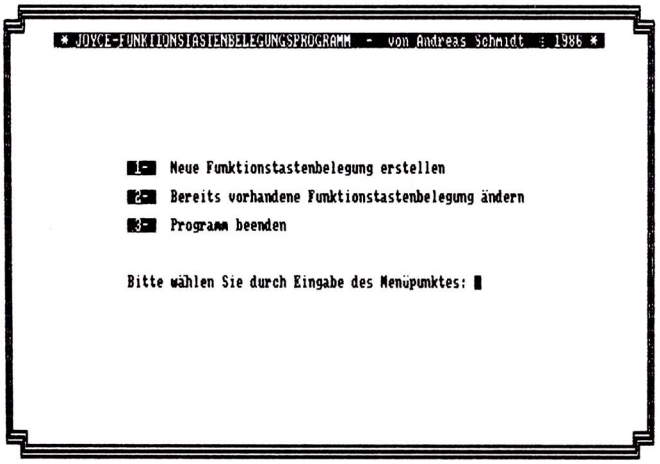


Abb. 1: Das Hauptmenue von FKEYS

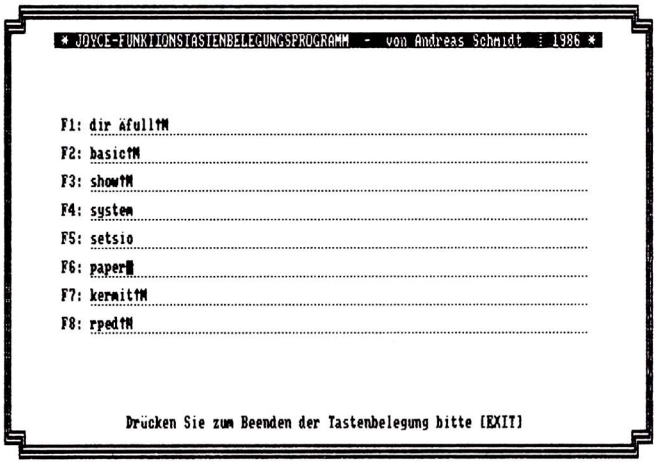


Abb. 2: Der Editor; so könnte eine Tastaturbelegung aussehen...

F K E Y S

```
LISTING >FKEYS    <, REMARK = >REM<.

<62> 10 ' Funktionstastenbelegungsprg.
< 2> 20 '
< 9> 30 ' Programm zur Erstellung von Funktionstasten-D
     efinitionsdateien
<50> 40 ' Die Definitionsdateien werden mit der Extensi
     on .KEY abgespeichert und
<77> 50 ' können im Betriebssystem durch SETKEYS <name>
     .KEY aktiviert werden.
< 6> 60 '
<73> 70 ' Andreas Schmidt
<44> 80 ' 3203 Sarstedt
< 9> 90 '
<18> 100 '
<44> 110 OPTION RUN:esc$=CHR$(27):clr$=esc$+"H"+esc$+"E
     ":rv$=esc$+"p":ro$=esc$+"q":pos$=esc$+"j":at$=esc$
     +"Y":bell$=CHR$(7):bs$=CHR$(8)
<66> 120 DEF FN yx$(y0,x0)=at$+CHR$(32+y0)+CHR$(32+x0)
<44> 130 PRINT esc$"0"esc$"f"esc$"q"esc$"y"esc$"u"esc$"
     w"esc$"22"rv$clr$
<63> 140 PRINT FN yx$(1,0)CHR$(134)CHR$(138)CHR$(137) F
     N yx$(0,2)CHR$(134):PRINT FN yx$(0,3);STRING$(84,C
     HR$(138));CHR$(140):PRINT FN yx$(1,87)CHR$(131)CHR
     $(138)CHR$(140)
<16> 150 FOR y=2 TO 30:PRINT FN yx$(y,0)CHR$(133) FN yx
     $(y,89)CHR$(133):NEXT y:PRINT FN yx$(30,89)CHR$(13
     7)
<24> 160 PRINT FN yx$(30,0)CHR$(131)CHR$(138)CHR$(140):
     PRINT FN yx$(30,88)CHR$(138) FN yx$(30,87)CHR$(134
     )
<68> 170 PRINT FN yx$(31,2)CHR$(131);STRING$(84,CHR$(13
     8))CHR$(137);esc$"H"
<15> 180 PRINT ro$ FN yx$(1,7) STRING$(76,"_")
<33> 190 PRINT FN yx$(2,7)rv$" * JOYCE-FUNKTIONSTASTENB
     ELEGUNGSPROGRAMM. -. von Andreas Schmidt. ñ 1986 *
      "ro$
<77> 200 PRINT esc$"X"CHR$(32+4)CHR$(32+2)CHR$(32+25)CH
     R$(32+85)
<19> 210 CLOSE:FOR x=1 TO 8:ftasten$(x)="":NEXT x:filen
     ame$=""
<75> 220 PRINT clr$
<31> 230 PRINT FN yx$(7,15)rv$" 1- "ro$". Neue Funktion
     stastenbelegung erstellen"
<11> 240 PRINT FN yx$(9,15)rv$" 2- "ro$". Bereits vorha
     ndene Funktionstastenbelegung ändern"
<38> 250 PRINT FN yx$(11,15)rv$" 3- "ro$". Programm bee
     nden"
<48> 260 PRINT FN yx$(15,15)"Bitte wählen Sie durch Ein
     gabe des Menüpunktes: ";rv$" "ro$
<97> 270 a$=INKEY$:IF a$="" THEN 270
<77> 280 a=VAL(a$):IF a<1 OR a>3 THEN 270
< 5> 290 PRINT FN yx$(15,63)a$
<86> 300 IF a=3 THEN PRINT esc$"e"esc$"0"clr$;:SYSTEM
<66> 310 IF a=2 THEN 690
<71> 320 PRINT clr$:FOR x=1 TO 8:PRINT FN yx$((x*2)+2,6
     )"F"MID$(STR$(x),2,1)": "ftasten$(x):PRINT FN yx$(
     (2*x)+3,10)STRING$(69,"▒"):NEXT x
<64> 330 PRINT FN yx$(25,15)"Drücken Sie zum Beenden de
     r Tastenbelegung bitte ↵EXIT↵"esc$"H"
<92> 340 sl=69:x=10:taste=1
<78> 350 ma=1:a$=ftasten$(taste):y=(taste*2)+2:GOSUB 82
     0:GOSUB 830:GOSUB 910
< 1> 360 ftasten$(taste)=a$
<59> 370 IF ab THEN taste=taste+1:IF taste>8 THEN taste
     =1
<97> 380 IF auf THEN taste=taste-1:IF taste<1 THEN tast
     e=8
<15> 390 IF can=0 THEN 440
<30> 400 PRINT FN yx$(22,34)"Abbruch (j/n)? ";:sl=1:GOS
     UB 830
<15> 410 al=1:a$="j":GOSUB 910:a$=LOWER$(a$):IF can THE
     N a$="n" ELSE:IF a$<>"j" AND a$<>"n" THEN 410
< 6> 420 IF a$="j" THEN 210
<22> 430 PRINT FN yx$(22,34)SPACE$(20)
< 5> 440 IF exit=0 THEN 350
<46> 450 PRINT FN yx$(22,27)"Tastenbelegungen okay (j/n
     )? ";:sl=1:GOSUB 830
< 0> 460 al=1:a$="j":GOSUB 910:a$=LOWER$(a$):IF can THE
     N a$="n" ELSE:IF a$<>"j" AND a$<>"n" THEN 460
<47> 470 IF a$="n" THEN PRINT FN yx$(22,27)SPACE$(30):s
     l=69:GOTO 350
<37> 480 '
<11> 490 ON ERROR GOTO 1420:PRINT clr$; FN yx$(7,8)"Unt
     er welchem Namen soll die Tastenbelegung gespeiche
     rt werden? ";:GOSUB 830
<49> 500 a$=filename$:sl=8:al=1:GOSUB 910:filename$=UPP
     ER$(a$):IF a$="" AND can=0 THEN 500 ELSE IF can TH
     EN 320
<39> 510 RESET:IF FIND$(a$+".key")="" THEN 550 ELSE PRI
     NT bell$ FN yx$(9,11)"Es existiert bereits eine Ta
     stenbelegung unter diesem Namen!"
<45> 520 PRINT FN yx$(11,30)"Überschreiben (j/n)? ";:sl
     =1:GOSUB 830
<89> 530 a$="j":al=1:GOSUB 910:a$=LOWER$(a$):IF can THE
     N a$="n" ELSE:IF a$<>"j" AND a$<>"n" THEN 530
<69> 540 IF a$="n" THEN 490
<43> 550 CLOSE:OPEN "O",1,filename$+".key":FOR x=1 TO 8
<67> 560 IF ftasten$(x)>"" THEN PRINT #1,"e #"MID$(STR$
     (80+x),2,2)" "CHR$(34)ftasten$(x)CHR$(34)
<55> 570 NEXT x
< 3> 580 IF ftasten$(1)>"" THEN PRINT #1,"02 N"CHR$(34)
     "^'#81'"CHR$(34)
<85> 590 IF ftasten$(2)>"" THEN PRINT #1,"02 S"CHR$(34)
     "^'#82'"CHR$(34)
<56> 600 IF ftasten$(3)>"" THEN PRINT #1,"00 N"CHR$(34)
     "^'#83'"CHR$(34)
```

```
<39> 610 IF ftasten$(4)>"" THEN PRINT #1,"00 S"CHR$(34)
     "^'#84'"CHR$(34)
<94> 620 IF ftasten$(5)>"" THEN PRINT #1,"73 N"CHR$(34)
     "^'#85'"CHR$(34)
<77> 630 IF ftasten$(6)>"" THEN PRINT #1,"73 S"CHR$(34)
     "^'#86'"CHR$(34)
< 4> 640 IF ftasten$(7)>"" THEN PRINT #1,"77 N"CHR$(34)
     "^'#87'"CHR$(34)
<86> 650 IF ftasten$(8)>"" THEN PRINT #1,"77 S"CHR$(34)
     "^'#88'"CHR$(34)
<93> 660 CLOSE
<32> 670 GOTO 210
<39> 680 '
<65> 690 ON ERROR GOTO 1460
<87> 700 PRINT clr$;:PRINT esc$"X"CHR$(32+4)CHR$(32+8)C
     HR$(32+25)CHR$(32+75)clr$"Auf Laufwerk A: vorhande
     ne Tastenbelegungen:":PRINT:RESET:IF FIND$("*.key"
     )>"" THEN:FILES "*.key"
<20> 710 PRINT esc$"X"CHR$(32+4)CHR$(32+2)CHR$(32+25)CH
     R$(32+85):IF FIND$("*.key")="" THEN PRINT FN yx$(7
     ,13)"Auf dieser Diskette sind keine Tastenbelegung
     en vorhanden!":GOSUB 1370:GOTO 210
<77> 720 PRINT:PRINT FN yx$(7,7)"Geben Sie bitte den Na
     men der zu verwendenden Tastenbelegung ein: ";:GOS
     UB 830
<84> 730 sl=8:al=1:GOSUB 900:filename$=UPPER$(a$):IF a$
     ="" AND can=0 THEN 730 ELSE IF can THEN 210
<60> 740 RESET:IF FIND$(a$+".key")="" THEN PRINT bell$
     FN yx$(9,8)"Es besteht keine Tastenbelegung unter
     diesem Namen auf der Diskette!":GOSUB 1370:GOTO 69
     0
<58> 750 OPEN "I",1,filename$+".key"
<65> 760 FOR x=1 TO 8:IF EOF(1) THEN x=8:GOTO 790
< 0> 770 INPUT #1,a$
<51> 780 a=VAL(MID$(a$,5,1)):ftasten$(a)=RIGHT$(a$,LEN(
     a$)-7):ftasten$(a)=LEFT$(ftasten$(a),LEN(ftasten$(
     a))-1)
<61> 790 NEXT x
<41> 800 GOTO 320
<27> 810 '
<63> 820 PRINT FN yx$(y,x);:RETURN
<73> 830 PRINT pos$;:RETURN
<33> 840 '
<50> 850 ' input-routine
<37> 860 '
<62> 870 b$="":PRINT esc$"k";
<12> 880 auf=0:ab=0:exit=0:can=0
<92> 890 RETURN
<52> 900 a$=""
<17> 910 GOSUB 870:GOTO 1170
<39> 920 aa$=INKEY$:IF aa$="" THEN 920
<44> 930 a=ASC(aa$)
<64> 940 IF a=127 AND LEN(b$)>0 THEN 1200
<25> 950 IF a>34 THEN 1080
<22> 960 IF a=7 AND LEN(a$)>0 THEN 1240
<17> 970 IF a=1 AND LEN(b$)>0 THEN 1250
<82> 980 IF a=6 AND LEN(a$)>0 THEN 1260
<50> 990 IF a=9 THEN 1270
```

```
<65> 1000 IF a=13 THEN 1290
<14> 1010 IF a=8 THEN can=1:GOTO 1290
<67> 1020 IF a=31 AND ma THEN auf=1:GOTO 1290
<31> 1030 IF a=30 AND ma THEN ab=1:GOTO 1290
<26> 1040 IF a=27 THEN exit=1:GOTO 1290
< 7> 1050 IF a=23 AND a$<>"" THEN 1220
<70> 1060 IF a=21 THEN PRINT esc$"k";SPACE$(LEN(a$+b$)+
     1):a$="":b$="":GOTO 1130
< 2> 1070 IF a<32 OR a=34 THEN 920
<20> 1080 IF a>126 THEN 920
<81> 1090 IF al THEN:IF a<48 AND a<>45 AND a<>47 AND a<
     >95 OR a>57 AND a<65 OR a>93 AND a<97 OR a>122 OR
     a>90 AND a<97 THEN 920
<42> 1100 IF LEN(a$+b$)=sl AND a$="" THEN 920
<84> 1110 b$=b$+aa$
<65> 1120 IF a$<>"" THEN 1160
<43> 1130 PRINT esc$"k";
<69> 1140 PRINT b$;rv$;" ";ro$
<43> 1150 GOTO 920
<86> 1160 a$=RIGHT$(a$,LEN(a$)-1)
<55> 1170 PRINT esc$"k";
<50> 1180 PRINT b$;rv$;LEFT$(a$+" ",1);ro$;RIGHT$(a$+"
     ",LEN(a$+")-1)" "
<55> 1190 GOTO 920
<17> 1200 b$=LEFT$(b$,LEN(b$)-1)
< 3> 1210 GOTO 1170
<70> 1220 IF LEN(a$+b$)<sl THEN a$=" "+a$:GOTO 1170
<48> 1230 IF RIGHT$(a$,1)<>" " THEN 1170 ELSE a$=" "+LE
     FT$(a$,LEN(a$)-1):GOTO 1170
<66> 1240 a$=RIGHT$(a$,LEN(a$)-1):GOTO 1170
<29> 1250 a$=RIGHT$(b$,1)+a$:b$=LEFT$(b$,LEN(b$)-1):GOT
     O 1170
< 1> 1260 b$=b$+LEFT$(a$,1):a$=RIGHT$(a$,LEN(a$)-1):GOT
     O 1170
<18> 1270 IF LEN(a$)<4 THEN b$=b$+a$:a$="":GOTO 1120
< 7> 1280 b$=b$+LEFT$(a$,4):a$=RIGHT$(a$,LEN(a$)-4):GOT
     O 1170
<63> 1290 PRINT esc$"k";
<32> 1300 PRINT b$;a$" "
<72> 1310 a$=b$+a$
<87> 1320 ma=0:al=0
<82> 1330 IF RIGHT$(a$,1)=" " THEN a$=LEFT$(a$,LEN(a$)-
     1):GOTO 1330
<81> 1340 IF a=13 THEN ab=1
< 0> 1350 RETURN
< 8> 1360 '
<88> 1370 PRINT FN yx$(24,39)rv$"-TASTE-";ro$
<87> 1380 a$=INKEY$:IF a$="" THEN 1380
<37> 1390 can=0:IF ASC(a$)=8 THEN can=1
<86> 1400 RETURN
<94> 1410 '
<41> 1420 PRINT clr$FN yx$(5,38)"Error:"ERR
<67> 1430 RESUME 1440
<14> 1440 GOSUB 1370:GOTO 490
< 7> 1450 '
<53> 1460 PRINT clr$FN yx$(5,38)"Error:"ERR
<40> 1470 RESUME 1480
<95> 1480 GOSUB 1370:GOTO 210
```


NEU NEU NEU

Erst testen-dann kaufen!

LocoPost

Text-/Adressenprogramm
Einbeziehung von LocoScript
Serienbriefe/Adressenverwaltung
einfachste Pull-Down-Menüsteuerung
viele autom. Routinen/ bei Serienbrief:
Handeingabe und Textblöcke sind möglich
DM 84.-

LITERDAT-Bücher, Musik, Video DM 39.-

LocoPost/LITERDAT-Testprogr. DM 8.95

HESCHCOM Weichselgartenstr. 13
8000 München 71 Tel.: 089/789377

Schneider Joyce-Software für Jedermann !!!!!

DICTIONARY	-	Wörterbuch engl.(ca. 40000 Vokabel)	DM 159,00
VEREIN	-	Vereinsverwaltung (bis 1400 Mitgl.)	DM 239,00
AGENTUR	-	Versicherungsverwaltung	DM 620,00
SALON	-	Friseur-Kundenverwaltung	DM 198,00
KASSE	-	Kassenbuch	DM 198,00
HAUS I	-	Hausverwaltung (bis 100 Einheiten)	DM 998,00
T-ADRESS	-	Adressverwaltung (ca. 1400 Adressen)	DM 189,00
MICA	-	CAD- und Zeichenprogramm	DM 198,00
SUPERCALC2	-	Kalkulationprogramm eng. (professionell)	DM 198,00
PROFIREM	-	Lager/Fakturierung/Adressverwaltung	DM 178,00
WÄRME	-	Wärmebedarfsrechnung DIN 4701/83	DM 298,00
LOHN/GEHALT	-	Lohn- u. Gehaltsabrechnung (bis 50 MA)	DM 298,00
DIAKARTEI	-	Dia-Kartei-Verwaltung (ca.25000 Dias)	DM 298,00
VIDEOTHEK	-	Videothekenverwaltung	DM 649,80
GOA	-	Privatarztabrechnung	DM 998,00

Schneider Joyce - 256 KB, 1 Laufwerk, Drucker, LocoSript
- Vereinsverwaltung (ca. 750 Mitglieder) gesamt DM 1.899,00

Weitere Software und Zubehör auf Anfrage (80 Pfg.-Briefmarke beilegen).Gerne werden wir Ihre Wünsche erledigen.
MS-DOS-Software für alle Bereiche - Bitte Anfragen - !!!!!!!!!!

Postfach 1241 * D-8058 ERDING
Tel. 08122-40529 * Mo-Fr 14.30-18.00 u. Sa 9.15-13.15 Uhr

Computer - Studio CBS GmbH

dBase II-Installationen

auf JOYCE, JOYCE PLUS und Festplatte

Das JOYCE Sonderheft Nr. 1 enthielt einen Beitrag zur zweckmäßigeren Ausnutzung des erweiterten RAM-Speichers und des zweiten Laufwerks beim Einsatz von dBase II auf dem JOYCE PLUS. Dieser Beitrag stellt eine Weiterentwicklung der dort gegebenen Hinweise dar, wobei einige nützliche Tips zur Optimierung von dBase-Abläufen gegeben werden sollen.

Folgendes wurde – in Kurzform ausgedrückt – vorgeschlagen:

1. Zusammenlegung der abzuarbeitenden dBase-Programme und der zugehörigen Datenbanken auf einer Diskette im Laufwerk B:

2. Beim »Booten« von dBase Verlagerung aller dBase Programm-Dateien (DBASE.COM, DBASEOVR.COM), sowie einiger erforderlicher CP/M PLUS Dateien von der Startdiskette in die RAM-Disk, mittels einer entsprechend formulierten PROFILE.SUB Datei.

3. Diese startet dBase in der RAM-Disk, wobei ein beim »Booten« ebenfalls von der Startdisk dorthin verlagertes, separat zu erstellendes dBase-Programm namens START.CMD aufgerufen wird. Dieses seinerseits macht mittels des dBase-Befehls »SET DEFAULT TO B« das untere Laufwerk für die Dauer der dBase-Anwendung zum Bezugslaufwerk und ruft in einer zweiten Befehlszeile das auf Diskette im Laufwerk B: liegende DBase-Programm auf, welches abgearbeitet werden soll. In der Folge greifen also die in B: liegenden dBase-Programme auf die ebenfalls in B: liegende zugehörige Datenbank zu.

Hier soll eine vorteilhafte Abwandlung dieser Konfiguration beschrieben werden. Ihre wesentlichen Merkmale sind:

1. Bereitstellung der dBase-Programme und der zugehörigen Datenbanken in separaten Laufwerken.

```
Scanning Directory...

Sorting  Directory...

Directory For Drive A:  User  0

     Name      Bytes   Recs   Attributes      Name      Bytes   Recs   Attributes
------------ ------- ------ ------------ ------------ ------- ------ ------------
BUECHER  CMD      1k      8 Dir RW       BUECHER  FRM      1k      4 Dir RW
BUECHER1 CMD      3k     20 Dir RW       BUECHER2 CMD      4k     32 Dir RW
BUECHER3 CMD      1k      4 Dir RW       BUECHER4 CMD      1k      4 Dir RW
BUECHER5 CMD      1k      4 Dir RW       BUECHER6 CMD      1k      4 Dir RW
BUECHER7 CMD      1k      4 Dir RW       BUECHER8 CMD      1k      4 Dir RW
DB       KEY      2k     16 Dir RW       DBASEOVR COM     41k    328 Dir RW
DIR      COM     15k    114 Dir RW       DO       COM     20k    156 Dir RW
J12DCPM3 EMS     40k    320 Dir RW       PAPER    COM      1k      8 Dir RW
PIP      COM      9k     68 Dir RW       PROFILE  SUB      1k      2 Dir RW
RENAME   FRM      1k      4 Dir RW       RESET    COM      1k      2 Dir RW
SETDEF   COM      4k     32 Dir RW       SETKEYS  COM      2k     16 Dir RW
SUBMIT   COM      6k     42 Dir RW

Total Bytes     =   158k  Total Records =    1196  Files Found =   23
Total 1k Blocks =   158   Used/Max Dir Entries For Drive A:   28/  64

                    Bild 1: Directory Startdiskette BUECHER
```

Bild 1: Directory Startdiskette BUECHER

2. Abarbeitung der dBase Programme in der RAM-Disk, mit Zugriff auf die Datenbanken von dort aus.

Das Ergebnis ist, sowohl bei Einsatz des JOYCE PLUS als auch bei zusätzlicher Verwendung eines Festplattensubsystems, eine noch schnellere Abarbeitung von dBase-Programmen, sowie eine bessere Übersicht, und die volle Ausnutzung der beim JOYCE PLUS im zweiten Laufwerk verfügbaren Speicherkapazität.

Zwei Vorbemerkungen

1. Wie andere Programme auch, laufen alle dBase-Befehlsdateien (Extension. CMD) in sich grundsätzlich auf der RAM-Disk schneller ab als auf Diskette.

2. Der Zugriff eines dBase-Programmes von der RAM-Disk auf einen Datenbestand, der auf einer Diskette in einem anderen Laufwerk (am besten B:) steht, ist wenngleich nicht ganz unabhängig von der internen Schreibweise der Programme selbst – ebenfalls in der Regel schneller als wenn Programme und zugehörige Datenbanken auf einer gemeinsamen Diskette liegen.

Um sich diese Tatsachen zu Nutze zu machen, legt man deshalb zweckmäßigerweise beim Start von dBase nicht nur DBASE.COM und die Overlay-Datei DBASEOVR.COM, sondern auch die dBase-Programme, welche man abarbeiten möchte, von der Startdiskette, auf der man sie bereithält, mittels einer PROFILE.SUB-Datei in die RAM-Disk. Dazu natürlich die zum »Booten« erforderlichen CP/M System-Dateien, sowie evtl. gewünschte CP/M-Hilfsdateien.

Letztere müssen natürlich nicht beim Start in die RAM-Disk verlagert werden, wenn man in der PROFILE.SUB-Datei mittels SETDEF.COM derart einen Suchpfad einrichtet, daß die CP/M-Hilfsdateien wie PIP.COM, TYPE.COM etc. unabhängig vom gewählten Bezugslaufwerk auch auf der in Laufwerk A: befindlichen Startdiskette gefunden werden. Die Verlagerung ist allerdings vorzuziehen, und entsprechend ist der SETCOM-Befehl unserer PROFILE.SUB-Datei derart formuliert, daß auch nach Wahl eines anderen Bezugslaufwerks und Diskettenwechsel im Laufwerk A: auf die CP/M-Hilfsdateien in der RAM-Disk zugegriffen werden kann (siehe Bild 2).

Man kann nun mit Hilfe der PROFILE.SUB-Datei automatisch bis zum gewünschten dBase-Programm durchstarten, ohne daß eine zusätzliche dBase-Start-Datei (START.CMD im Sonderheft) eingerichtet und aufgerufen werden muß. Der durch die erhöhte Zahl an Kopiervorgängen verursachte Zeitverlust beim Start wird durch die höhere Ablaufgeschwindigkeit bei der anschließenden dBase-Anwendung mehr als wettgemacht. Auch vermeidet man den Einsatz des dBase-Befehls »SET DEFAULT TO«, der für die Dauer der dBase-Anwendung das Laufwerk B: zum Bezugslaufwerk macht. Diese Maßnahme hat nämlich oftmals einen Verlust an Orientierung zur Folge und macht zusätzliche Manipulationen erforderlich, wenn nicht lediglich ein spezielles dBase-Programm ablaufen soll, sondern man eventuell auch zwischenzeitlich im dBase-Einzelbefehlsmodus tätig werden möchte. Beim hier vorgestellten System-Start bleibt für die gesamte Dauer der dBase-Arbeit die RAM-Disk das Bezugslaufwerk, sowohl bei Rückkehr zum Betriebssystem als auch bei Rückkehr zum dBase-Prompt. Das wird jedem von Vorteil sein, der eigene dBase-Programme schreibt oder vorhandene Programme abändern möchte. Wie oft stellt sich bei der dBase-Anwendung der

```
SETDEF A:,* ÄORDER = (SUB,COM) TEMPORARY=M:Ü
SETKEYS DB.KEY
PIP
<M:=DO.COM
<M:=DBASEOVR.COM
<M:=*.CMD
<M:=*.FRM
<M:=PIP.COM
<
M:
DO BUECHER


Bild 2: Datei PROFILE.SUB
```

Bild 2: Datei PROFILE.SUB

Wunsch ein, hier oder dort schnell einmal Veränderungen vorzunehmen, sei es um eine bessere Ablaufgechwindigkeit zu erzielen, schönere Menueanzeigen zu gestalten oder auch um irgendwelche Maßnahmen zur »Narrensicherheit« in seine dBase-Programme einzubauen. Um bei Verlassen seines dBase-Programmes zum dBase-Prompt zu gelangen, schafft man sich zweckmäßigerweise im Hauptmenue jedes seiner dBase-Hauptprogramme neben dem üblichen Ausgang zum Betriebssystem (mittels »QUIT«) zusätzlich einen solchen zu dBase, wozu der dBase-Befehl »CANCEL« dient. Dieser beendet das laufende dBase-Programm, und man gelangt bei unserer Anordnung zum dBase-Prompt im Laufwerk M: zurück. Im dBase-Einzelbefehlsmodus kann man nun dort z.B. mittels »MODIFY COMMAND« die beim Start dorthin verlagerten Befehls-Dateien direkt zur Änderung aufrufen, statt erneut dBase starten zu müssen und erst mittels »SET DEFAULT TO B« und »MODIFY COMMAND«, bzw. mit zusätzlicher Angabe der Laufwerkskennung (MODIFY COMMAND B:<Dateiname>) die abzuändernden dBase-Befehlsdateien aufrufen zu können. Man gewinnt zudem wieder an Geschwindigkeit, da auch der Befehl MODIFY COMMAND in der RAM-Disk schneller arbeitet als auf Diskette. Erfahrungsgemäß ist es ohnehin mit einer Änderung am Programm meist nicht getan, so daß sich der Gewinn an Zugriffs- und Änderungsgeschwindigkeit bei »MODIFY COMMAND« oft multipliziert, wenn die Befehlsdateien in M: liegen. Nach Abschluß einer solchen Überarbeitung und Austestung muß man natürlich zur Sicherung seiner dBase-Programme irgendwann ins Betriebssystem zurückkehren und die revidierten Programme mittels PIP.COM auf seine Start-Diskette übertragen. Zwar kostet das wieder etwas Zeit, aber die spart man gegenüber der Programm-Korrektur auf Diskette wiederum dadurch ein, daß nicht die von dBase bei Programm-Überarbeitungen automatisch angelegten Backup-Dateien (Extension .BAK) zusätzlich gelöscht werden müssen, etwa, wenn der Platz auf der Diskette knapp zu werden droht, oder weil sie einfach störend sind.

Die Gefahr, daß ein Programm bei der Überarbeitung in der RAM-Disk zu Verlust geht, etwa durch Zusammenbruch wegen fehlerhafter Syntax, läßt sich niedrig halten, wenn man im Falle von Zweifeln an den eigenen Programmierkünsten zwischendrin schnell einmal seine in Arbeit befindliche dBase-Programmdatei irgendwohin hinüberkopiert, am vorteilhaftesten nach B:, wo auf der Datendiskette meist noch etwas Speicherplatz verfügbar ist. Man hält sich hiermit mehrere Versionen seiner Programmdatei »auf Lager«, bis man das gewünschte Endresultat der Überarbeitung erreicht hat. Daß es in dBase eine Reihe von Tricks gibt, mit denen sich der Cursor auf einem nach Programmabsturz erstarrten Bildschirm wieder flott machen läßt, sei nur am Rande erwähnt.

Letztlich bleibt während der Korrekturversuche in M: auch die Originalfas-

```
§ 1,23 say "BIBLIOTHEKSKATALOG"
§ 5,20 say "1. Buch hinzufügen"
§ 6,20 say "2. Eintrag ändern"
§ 7,20 say "3. Eintrag löschen"
§ 9,20 say "4. Buchtitel suchen"
§ 10,20 say "5. Bücher nach Autoren anzeigen"
§ 11,20 say "6. Bücher nach Sachgebieten anzeigen"
§ 12,20 say "7. Bücher nach Standort anzeigen"
§ 13,20 say "8. Verzeichnis aller Bücher ausdrucken"
§ 15,24 say "<B>etriebssystem"
§ 17,24 say "<D>Base"
§ 19,20 say "Geben Sie bitte Ihre Wahl ein" get option picture "!"
read
§ 23,0 say "Bitte warten"
do case
case option="B"
erase
quit
case option="D"
erase
cancel

Bild 3: Hauptmenue und Teile des Programms BUECHER.CMD (1. Lösung)
```

Abb. 3

sung stets auf der Startdiskette verfügbar, während bei mehrfachen Änderungen auf Diskette jeweils nur die letzte Programm-Version als Backup-Datei erhalten bleibt. Das Originalprogramm geht ja schon bei der zweiten Überarbeitung zu Verlust, wenn dBase erneut eine Backup-Datei anlegt, welche das, bei der ersten Korrektur zur Backup-Datei umfunktionierte Original überschreibt.

Da wir schon zum Thema Programm-Erstellung in dBase II abgeschweift sind, ein weiterer Hinweis hierzu: Bei größeren Umarbeitungen, und noch mehr für die Neu-Erstellung eines dBase-Programmes, sollte man ohnehin besser auf MODIFY COMMAND verzichten und statt des höheren Komforts und der größeren Bearbeitungsgeschwindigkeit halber, mit einem Textverarbeitungsprogramm arbeiten. dBase .CMD-Dateien sind reine ASCII-Files. In Protext z.B. kann man sie direkt als Text in Bearbeitung nehmen; in LocoScript können sie in eine Schablone wie beispielsweise PAGENUM.CEN eingelesen werden (f7 = Modus »Text einfügen« im Bearbeitungsmodus), wo dann alle Änderun-

JOYCE-ZWEITLAUFWERKE

Moderne TEAC-Laufwerke, 2*80 Track (1 MB unformatiert, 720 kb format.) problemlos anzuschließen, da anschlußfertig montiert in formschönem Gehäuse (Farbe passend zum Joyce), 5.25" Laufwerk wahlweise mit 40/80 Tr.-Umschalter und eingebautem Netzteil!

3.5" Laufwerk	DM 359,00
5.25" Laufwerk mit eig. Netzteil	DM 449,00
40/80 Track-Umschalter (nur 5.25") Aufpreis	DM 20,00
Programm MsCopy 5.25" Disk	DM 49,00
5.25" Qualitätsdisketten 10er Pack (96 tpi)	DM 28,00
3.5" Disketten 10er Pack (135 tpi)	DM 38,00
Programme:	
FIBU-STAR Plus	DM 298,00
1000 Kontenbl. (endlos)	DM 49,00

Unterhaltungsprogramme:

The Pawn	DM 65,90	Aftershock	DM 49,90
Batman	DM 39,90	Scrabble	DM 59,90
Trivial Persuite	DM 53,90	3 D Chyruss II Chess	DM 45,90
Jewels of Darkness	DM 65,90	3 D Clock Chess	41,90

Das Disk-Utility MsCopy ermöglicht das Beschreiben u. Lesen von Ms-Disketten. Hierdurch können problemlos Dateien IBM PC oder kompatiblen Geräten übertragen werden. Datenfiles von Multiplan oder dBase können direkt übernommen werden. (5.25" Laufwerk mit 40/80 Trackumschalter ist Voraussetzung).

Bitte kostenloses Informationsmaterial u. Katalog anfordern. Der Versand erfolgt per Nachnahme + 6,00 DM Versandkostenpauschale. Alle Preise sind freibleibend.

Soft- u. Hardwareversand
6690 St. Wendel 8, Fasanenweg 2,
0 68 56 / 5 04

ULRIKE BECKER

tel. Bestellung
tägl. ab 17.00 Uhr
Sa. v. 9.00 – 11.00 Uhr

EASY FIBU

Sachkontenbuchhaltung mit Vor- und Mehrwertsteuerberechnung

EASY FAKTURIERUNG

Sofortfakturierung mit Artikel und Kundenverwaltung

Die einfach zu bedienende, bewährte Profisoftware für Selbständige und Kleinbetriebe. Je DM 570,–.

Gruse Elektronik GmbH,
Stieglitzweg 1, 3300 Braunschweig,
Tel.: 05 31 / 35 00 65

```
§ 1,23 say "BIBLIOTHEKSKATALOG"
§ 5,20 say "1. Buch hinzufügen"
§ 6,20 say "2. Eintrag ändern"
§ 7,20 say "3. Eintrag löschen"
§ 9,20 say "4. Buchtitel suchen"
§ 10,20 say "5. Bücher nach Autoren anzeigen"
§ 11,20 say "6. Bücher nach Sachgebieten anzeigen"
§ 12,20 say "7. Bücher nach Standort anzeigen"
§ 13,20 say "8. Verzeichnis aller Bücher ausdrucken"
§ 15,20 say "X. Ende"
§ 17,20 say "Geben Sie bitte Ihre Wahl ein" get option picture "!"
read
§ 23,0 say "Bitte warten"
do case
case option="X"
erase
store " " to ende
do while.not.ende $"DB"
§ 9,21 say "Zurück zu (D)Base oder (B)etriebssystem ?" get ende picture "!"
read
enddo (while.not.ende)
erase
if ende="D"
cancel
else
quit
endif (ende)
```

Bild 4: Hauptmenue und Teile des Programms BUECHER.CMD (Alternativ-Losung)
(durch die "do while"-Schleife wird keine andere Eingabe als "D" oder "B" akzeptiert, weshalb die anschliessende Abfrage mit einer einzigen "if-endif" Bedingung auskommt)

Abb. 4

gen höchst komfortabel durchführbar sind. Man denke zum Beispiel an den Einsatz der »Suchen/Ersetzen«-Funktion (f7/f8 im Bearbeitungsmodus), wenn Programme durch Abänderung in den Menues und mittels Namenswechsel der aufzurufenden Datenbanken für den Zugriff auf eine andere als die ursprüngliche Datenbank zugerichtet werden sollen. Nach Abschluß aller Änderungen setzt man die COMMAND-Dateien dann von LocoScript aus (f7=Modus in der Dateiverwaltung) in unformatierte ASCII-Files um.

Aber zurück zu den Vorteilen des Starts eines dBase Programmes im Laufwerk M:: Hält man die abzuarbeitenden dBase .CMD-Dateien zur Umlegung ins Laufwerk M: auf der Startdiskette bereit, statt gemeinsam mit Datenbank und Indizes auf der im Laufwerk B: liegenden Diskette, so bleibt der gesamte Speicherplatz der Datendiskette für die Datenbestände frei. Je größer die zu verwaltenden Datenbänke werden, desto mehr wird (neben schnellem Ablauf der Programme) dieser Gesichtspunkt eine Rolle spielen. Natürlich müssen bei dieser Zugriffsform die dBase-Befehlsdateien entsprechend umformuliert werden, damit die im Laufwerk B: liegende Datenbank nebst Indizes korrekt eröffnet werden kann.

Die hier dargestellte Art des Starts einer dBase-Anwendung geht davon aus, daß (sinnvollerweise) für jede Datenbank eine separate Datendiskette für das Laufwerk B: angelegt ist. Dann ist es zweckmäßig, auch für jede Datenbank eine separate Startdiskette vorzusehen. Falls jedoch mehrere Datenbanken und zugehörige Indizes auf einund derselben Diskette im Laufwerk B: zu bearbeiten sind, kann man sich den Zugang zu allen gemeinsam durch entsprechende Formulierung der PROFILE.SUB Datei und zusätzliche Erstellung eines allen Programmen vorangestellten dBase-Hauptprogrammes mit entsprechendem Hauptmenue eröffnen, wobei Letzteres dann zu Beginn der dBase-Anwendung aus der PROFILE.SUB Datei aufgerufen wird, und in dem wiederum die Hauptprogramme zur Bearbeitung der verschiedenen Datenbanken als Unterprogramme angewählt werden können.

Bei der anschließenden konkreten Darstellung unseres dBase-Startmodus halten wir uns, der besseren Vergleichsmöglichkeiten wegen, wieder an die mit dBase II mitgelieferten Demonstrationsprogramme zur Datenbank BUECHER, wobei die Anmerkung erlaubt sei, daß dieselben weder besonders erhellend sind, noch im geringsten die im dBase mögliche Eleganz der Programmgestaltung wiedergeben.

Unsere STARTDISKETTE müßte mindestens die folgenden Dateien enthalten:

J12GCPM3.EMS
DO.COM (=dbase.com)
DDBASEOVR.COM
DB.KEY
SETKEYS.COM
SETDEF.COM
PROFILE.SUB
PIP.COM
SUBMIT.COM
BUECHER.CMD
BUECHER1.CMD
BUECHER2.CMD
BUECHER3.CMD
BUECHER4.CMD
BUECHER5.CMD
BUECHER6.CMD
BUECHER7.CMD
BUECHER8.CMD
BUECHER.FRM
RAEUME.FRM

Das sind, kurz gesagt, sämtliche dBase System-, Befehls-, Report- und evtl. Formatdateien, zusammen mit den grundsätzlich benötigten CP/M Systemdateien und der »Startdatei« (PROFILE.SUB).

Dazu legt man auf die Startdiskette eventuell je nach verbleibender Kapazität weitere gewünschte CP/M-Hilfsdateien wie TYPE.COM, etc., und zweckmäßigerweise die im JOYCE Sonderheft Nr. 1 angebotene Datei RESET.COM. Diese ermöglicht nach Verlassen von dBase die Rückstellung der Tastaturbelegung von dBase auf das Betriebssystem, was unerläßlich ist wenn z.B. mittels des Editors RPED.BAS Veränderungen an der PROFILE.SUB-Datei vorgenommen werden sollen. In den meisten Fällen wird auf der Startdiskette hinreichend Platz für all diese Dateien vorhanden sein (Bild 1). Es gilt dann zu überlegen, ob man sich die CP/M-Hilfsdateien ebenfalls beim Start in die RAM-Disk verlagern will. Damit nimmt zwar der Startvorgang erheblich mehr Zeit in Anspruch, aber andererseits hat man bei Rückkehr von dBase ins Betriebssystem stets alle CP/M-Dateien im direkten Zugriff.

Wir erstellen unsere PROFILE.SUB-Datei mit dem Texteditor RPED.BAS und geben ihr folgendes Aussehen (siehe Bild 2):

SETDEF M:, ÄORDER=(SUB,COM)*
TEMPORARY=M:Ü

(womit sichergestellt ist, daß die CP/M-Dateien jederzeit in der RAM-Disk gefunden werden und M: zum Laufwerk für vorübergehend gebildete Dateien gemacht wird)

SETKEYS DB.KEY
PIP
<M:=DO.COM
<M:=DBASEOVR.COM
<M:=.CMD*

(damit werden alle BUECHER.CMD Dateien nach M: verlagert)

<M:=.FRM*

(entsprechende Verlagerung der Report-Dateien)

<M:.COM*

(Verlagerung der CP/M-Dateien, falls gewünscht)

M:
DO BUECHER

Der letzte Befehl der PROFILE.SUB Datei ruft jetzt das BUECHER Hauptmenue in Laufwerk M: auf. Um von dort entsprechend den obigen Ausführungen auf die in Laufwerk B: liegenden Datenbestände zugreifen zu können, müssen nun die zugehörigen Un-

terprogramme (BUECHER1.CMD bis BUECHER8.CMD) mittels »MODIFY-COMMAND« umgeschrieben werden: Bei Eröffnung einer Datei (hier: BUECHER) in einem anderen als dem Default-Laufwerk muß sowohl vor den Namen der betreffenden Datei als auch vor den Namen des evtl. zu eröffnenden Index die Kennung des »datentragenden« Laufwerkes (hier: B:) gesetzt werden:

USE B:BUECHER bzw.
USE B:BUECHER INDEX B:BTITEL

Beim Fehlen der Laufwerkskennung vor dem Index-Namen kann der Index nicht eröffnet werden; bei gleichzeitiger Eröffnung mehrerer Indizes muß jedem der Indexnamen die Laufwerkskennung vorangestellt werden, z. B:

USE B:BUECHER INDEX
B:TITEL,
B:AUTOR

Die Mehrzahl der Befehlsdateien des Beispiel-Programmes BUECHER arbeitet im Zugriff auf die Datenbank BUECHER.DBF selbst. Als einzige Indexdatei wird BTITEL.NDX ständig aktualisiert (und muß deshalb neben der Datenbank BUECHER.DBF auf Diskette im Laufwerk B: liegen). Im Unterprogramm BUECHER8.CMD (Ausdrucken von Bücherlisten) werden einige Indizes während der ListenErstellung neu generiert. Diese kann man auf jedem beliebigen Laufwerk ablegen, was wiederum durch Voranstellung der Laufwerkskennung vor dem Indexnamen bewerkstelligt wird.
Beispiel:

USE B: BUECHER INDEX ON
<Ausdruck> TO B:AUTOREN

Ohne Angabe einer Laufwerkskennung wird der Index im Bezugslaufwerk (hier M:) abgelegt. Dies wäre im Fall des BUECHER8.CMD-Unterprogrammes durchaus angebracht, da bei der Anlage dieser Anwendung die neu gebildeten Indizes nur temporär benötigt werden und bei Erweiterungen oder Änderungen des Datenbestandes immer neu zu erstellen sind. Hierzu ist anzumerken, daß bei größeren Datenbeständen – die BUECHER-Datenbank ist in ihrem geringen Umfang nicht representativ – sorgfältig zu erwägen ist, ob nicht die häufiger benötigten Indizes vor Eingabe der ersten Daten erstellt, dann ständig offengehalten und aktualisiert werden sollten. Je mehr Indizes eröffnet sind, desto länger dauert andererseits die Abspeicherung bei Neu-Eingabe oder Änderung von Datensätzen. Es gilt also für jede Anwendung zu entscheiden, ob der unvermeidliche Zeitaufwand eher bei der Index-Erstellung im Bedarfsfalle wie in den Beispiel-Programmen, oder verteilt über die Dauer der Bearbeitung einer Datenbank zu verschmerzen ist. Entscheidet man sich für die gleichzeitige Aktualisierung mehrerer Indizes, dann müssen diese natürlich mit auf der Diskette im Laufwerk B: bereitstehen.

Zur korrekten Abarbeitung der Report-Dateien (Extension .FRM) sind im Beispiel BUECHER keine Änderungen in der Laufwerkskennung erforderlich sofern die Index-Dateien im Laufwerk M: abgelegt werden. In anderen dBase-Anwendungen kann der Zugriff auf

```
HRDINS.O
SETDEF A:,* &ORDER = (SUB,COM) TEMPORARY=M:U
SETKEYS DB.KEY
SUBMIT DOBUECH
```

Bild 5: Datei PROFILE.SUB mit Einbindung von Vortex WD 2000

Abb. 5

Report-Dateien erheblichen Zeitaufwand verursachen, weshalb dBase weitere Wege zur direkten Daten-Darstellung im Druck anbietet, auf die aber hier nicht näher eingegangen werden soll. Jedoch sei erwähnt, daß man auch ebenso elegant unter Verzicht auf Report-Dateien und direkten Ausdruck die gesuchten Daten zunächst in einer ASCII-Zwischendatei ablegen kann, um sie – wie oben für den Fall der Programmerstellung geschildert – in eine Schablone in LocoScript oder einem anderen Textverarbeitungsprogramm einzulesen, wo man ihnen jedes gewünschte Layout und jede Verschönerung aufprägen kann.

Um nun in den Vorteil der angesprochenen Möglichkeit der schnellen und direkten Änderung seiner dBase-Befehlsdateien in der RAM-Disk zu kommen, sollte man wie erwähnt in seinen Programmen zwei Abbruchs-Wege vorsehen. Im Beispielsfalle wird das Hauptprogramm BUECHER.CMD etwa wie folgt abgeändert (siehe Bild 3): Im Hauptmenue (»BIBLIOTHEKS-KATALOG«) wählt man statt des Abbruchs

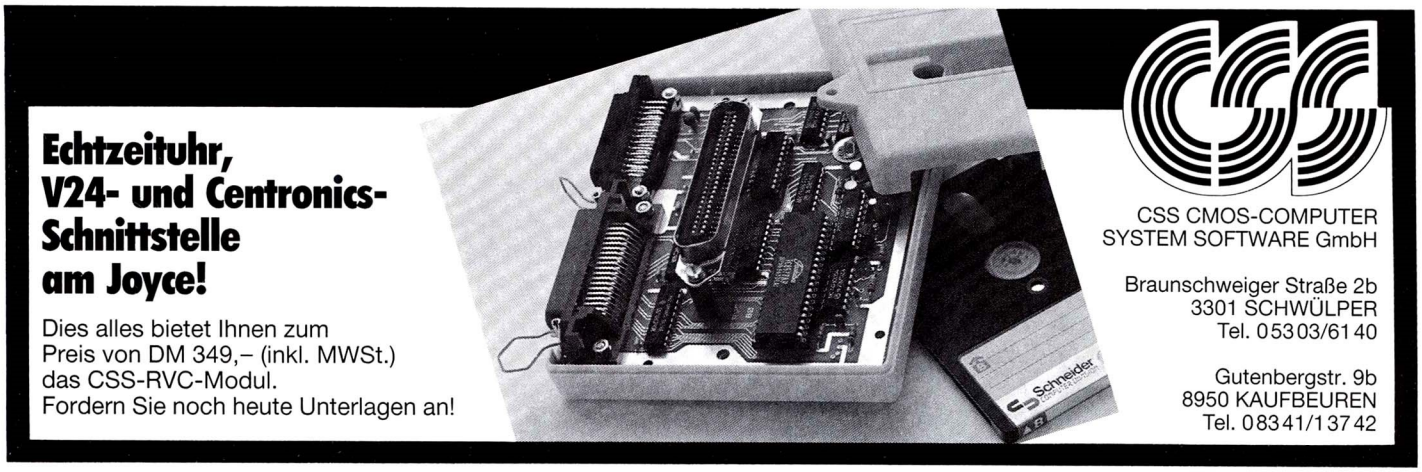
§ 15,20 say "X. Ende"

etwa die Formulierung:

§ 15,24 say "<B>etriebssystem"

und ändert weiter unten das zugehörige

case option = "X"



```
PIP
<M:=DO.COM
<M:=DBASEOVR.COM
<M:=BUECHER*.*
<
M:
DO BUECHER
```

Bild 6: Datei DOBUECH.SUB (Kopiervorgänge und dBase-Start mit WD 2000)

case option = "B"

um. Dann hängt man eine Zeile zum Ausstieg nach dBase an, etwa so:

§ *17,24 say* "<D>Base".

Die zugehörigen Befehlszeilen nach dem *do case* würde dann lauten:

case option = "B"
erase
quit
case
option = "D"
erase
cancel

In Hauptmenue muß dann nur noch das

"Geben Sie bitte Ihre Wahl ein..."

um zwei Zeilen nach unten orientiert werden: statt

§ *17,20*

heißt es dann:

§ *19,20.*

Der Vollständigkeit halber sei erwähnt, daß als Alternative der Befehl *case option = "X"* – ohne Änderung im Hauptmenue – derart erweitert werden kann, daß durch Aufruf eines kleinen Untermenues der Ausgang nach dBase oder ins Betriebssystem eröffnen wird (Bild 4).

Auch beim Vorhandensein eines Festplattensubsystems ergeben sich mit der vorgestellten Konfiguration die schnellsten dBase-Abläufe. Dazu einige zusätzliche Hinweise: Je nach Wahl der Festplatten-Partition (Einteilung in logische Laufwerke), wiederum mit Ablegung der dBase-Systemdateien und -programme getrennt von den Datenbanken, können die erarbeiteten dBase-Programme eventuell sogar ohne jede Änderung übernommen werden. Nur die PROFILE.SUB-Datei muß zum Zwecke der Einbindung der Harddisc in jedem Falle umgeschrieben werden. Wir legen unsere Programme und System-Dateien wieder auf eine Partition, die bei Harddisc-Betrieb als Laufwerk A: angesprochen wird, und die Datenbanken auf eine andere, als Laufwerk B: anzusprechende Partition. Da in den beim Start ausgelösten Kopiervorgängen dann auf die Festplatte zugegriffen wird, läuft der Programmstart obendrein sehr viel schneller ab als von der Startdiskette. Auf dieser werden die dBase-Programme eigentlich nicht mehr benötigt, man hält sie dort gleichsam nur als Sicherheitskopie vor.

Unter allen JOYCE-Benutzern werden am ehesten die dBase-Anwender die Vorzüge einer Festplatte zu schätzen wissen, denn trotz der beim JOYCE PLUS im Laufwerk B: verfügbaren 720 KByte gelangen sie beim Anwachsen einer Datenbank schon bald an die Grenzen der Speicher-Kapazität: Während man im LocoScript seine Texte aufspalten und mit Hilfe des Seitenzählungsmenues dennoch als kontinuierliche Einheit anlegen und zum Ausdruck bringen kann, ist es im dBase nicht möglich, eine Datenbank auf verschiedene Datenträger (sprich Disketten) aufzuteilen und sukzessive abzuarbeiten. Man ist gezwungen, die Datenbank vor Erschöpfung der Speicherkapazität nach irgendwelchen Kriterien aufzutrennen. Da bringt nur ein Winchester-Laufwerk Abhilfe. Für den JOYCE und JOYCE PLUS werden seit einiger Zeit Festplattensysteme angeboten, mit Systemdiskette und ausführlichem Handbuch.

Allerdings kommt beim Festplattenbetrieb das Problem besonders zum tragen, daß ein durch die PROFILE.SUB-Datei vorgegebener Stapeljob bei der ersten Fehlermeldung abgebrochen wird (die Einbindung der Harddisk in das System geschieht auch auf diese Weise, da das Betriebssystem CP/M grundsätzlich nur von Laufwerk A: aus gestartet werden kann).

Abhilfe ist leicht möglich durch Formulierung einer elementaren vierzeiligen PROFILE.SUB-Datei, mit der lediglich die Festplatte eingebunden und die System-Grundeinstellungen vorgenommen werden, dann anschließend eine zusätzlich einzurichtende SUBMIT-Datei aufgerufen wird, welche die weiteren Befehlsfolgen enthält. Die oben angeführte PROFILE.SUB-Datei und müßte dann folgendes Aussehen haben (Bild 5, am Beispiel der Vortex WD 2000):

```
SETDEF M:,* ÄORDER = (SUB,COM) TEMPORARY=M:Ü
SETKEYS DB.KEY
PIP
<M:=*.COM
<M:=*.CMD
<M:=*.FRM
<
M:
DO BUECHER
```

Bild 7: Datei PROFILE.SUB (Alternativ-Lösung)

HRDINS.0

(oder .1 usw., je nach gewünschter Partition der Festplatte, wobei HRDINS.COM als "Initialisierungsprogramm" zur Installation der Harddisc auf der Startdiskette vorhanden sein muß).

*SETDEF M:,*ÄORDER=(SUB,COM)*
TEMPORARY=M:Ü
SETKEYS DB.KEY
SUBMIT DOBUECH

(oder CP/M, oder Ähnliches, je nach Inhalt, Zweck, und Bezeichnung der nachfolgend auszuführenden SUBMIT-Datei)

Die SUBMIT-Datei »DOBUECH.SUB« würde in unserem Beispiel zum Start des Programmes BUECHER alle PIP-Befehlsfolgen aus unserer ersten PROFILE.SUB-Datei enthalten und den Aufruf des dBase-Programmes BUECHER (Bild 6).

Mit einer derartigen Konfiguration hätte man sich einer optimalen dBase-Anwendung auf dem JOYCE PLUS ein beträchtliches Stückchen angenähert, aber es bleibt ein weites Feld für Experimente.

(Dr. K. Stratemann)

JOYCE DATABOX

— Das ist die Software zum Sonderheft 2/87 —

DATABOX:

— mehr als der übliche Softwareservice

— bringt alle Programme des Joyce-Sonderheftes auf 3“ Diskette

— trägt das Titelbild des gleichzeitig erscheinenden Sonderheftes.

Für alle PCW's als 3“ Diskette



Die Databox platzt aus allen Nähten! Über 480 kB an Daten stehen auf drei Diskettenseiten zur Verfügung: alle Programme und Dateien aus dem Sonderheft finden Sie hier lauffähig vor.

Inhalt:

Diskette 1:

- 3D-Plotter
- Pascal-Compiler
- dBase-CMDs
- dBase-Handbuch
- Kybernetik-Lernspiel
- Quickregister
- Spaltendruck LocoScript
- Entscheidungsgenerator
- Grafmod-Erweiterung
- Adressverwaltung
- Archiv-Verwaltung
- Balkenmenue
- Funktionszeichner LOGO
- Kalender
- Labelprinter
- Lister f. Basicprogramme
- Funktionstastenprogramm
- Suburbia: Brettspiel ähnlich MONOPOLY

u.a.m., insgesamt ca. 310 kB auf 2 Seiten

Diskette 2:

Alle Dateien für die Turbo-Pascal-Grafik-Toolbox: Quellcode und Demo-Dateien im Source- und Kommandoformat – 170 kB Daten!

Diskette 1: **30, –** Diskette 2: **24, –**

Kombipack Diskette 1 und Diskette 2: **48, –**

Coupon ausschneiden, auf Postkarte aufkleben oder per Brief senden an:

DMV-Verlag
Postfach 250
3440 Eschwege

»Joyce-Bestellservice«
Knüller für Joyce-Fans

Ich mache von Ihrem Angebot Gebrauch und bestelle hiermit:

401	Stck.	**»Praktische Textverarbeitung mit Joyce«** — ein Buch-Disketten-Paket für praxisnahe Anwendungen mit Joyce nur	**89,– DM**
301	Stck.	**Sonderheft Joyce Nr. 1**	**20,– DM**
303	Stck.	**Databox** zum Sonderheft Joyce Nr.1 — viel Software für wenig Geld, Diskette 3“	**30,– DM**
NEU			
302	Stck	**Sonderheft Joyce Nr. 2**	**20,– DM**
304	Stck.	Databox-Diskette Nr. 1 zum Joyce SH 2	**30,– DM**
305	Stck.	Databox-Diskette Nr. 2 zum JOyce SH 2	**24,– DM**
306	Stck.	Kombipack, enthält Diskette 1 + 2 zum Joyce Sonderheft Nr. 2	**48,– DM**
215	Stck.	**Joyce Programmsammlung Vol.1** — Hochwertige Software zum Tiefstpreis, Diskette 3“	**59,– DM**
216	Stck.	**Joyce Programmsammlung Vol.2** — Hochwertige Software zum Tiefstpreis, Diskette 3“	**49,– DM**

Gesamtbetrag

zzgl. DM 3,– Porto/Verpackung. Per Nachnahme zzgl. Nachnahmegebühr. (In das Ausland ist Nachnahme nicht möglich.)

☐ Den Betrag bezahle ich mittels des beigefügten Verrechnungsschecks.

☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum _____ Unterschrift (bei Minderjährigen des gesetzl. Vertreters)

Nachdem sich Heft 11/1986 mit dem Drucken eines CP/M-Handbuchs aus dem Helpfile beschäftigt hat, möchte ich Ihnen eine Möglichkeit darstellen, das Gleiche für dBASE-II zu tun.

Auf der dBASE-Diskette (v 2.41) befindet sich ein File namens DBASEMSG.TXT, welches die Texte enthält, die mit HELP on-line abzurufen sind. Das Helpfile ist wesentlich einfacher aufgebaut als das von CP/M, es handelt sich um ein einfaches sequentielles Textfile.

Die einzelnen Stichworte sind an einem * als erstem Zeichen der Zeile zu erkennen, gefolgt vom Stichwort (Großbuchstaben) und RETURN. Es folgen die Textzeilen, jeweils durch RETURN abgeschlossen und eventuell weitere Stichworte.

Die Ausgabe bei der on-line Helpfunktion erfolgt solange, bis im Helpfile ein EXIT auftritt, welches eines oder mehrere Stichworte beenden kann. Einige Worte zum Inhalt von DBASEMSG.TXT:

1. Fehler: Es konnte ein Fehler im File festgestellt werden, der wirklich stört. Wird mit HELP CREATE Hilfe für das Kreieren von Datenbanken angefordert, so fehlt die erste Textzeile!

Dies liegt an einem fehlenden RETURN nach dem Stichwort, das sich aber mit einem Texteditor (ED.COM, RPED oder WORDSTAR) nachträglich einfach einfügen läßt.

2. Verwunderliches: Schaut man sich den gesamten Helptext an, wird man einige Texte finden, die speziell auf den HP150-Computer zugeschnitten sind. Der Helptext ist speziell für diesen Rechner geschrieben worden!

```
ACCEPT

> ACCEPT  — Eingabe  einer Zeichenkette in eine bestimmte tempo-
            räre Variable, die dann vom Typ "C"  ist. Als Prompt
            erscheint ein Doppelpunkt; leere Engabe (d.h. RETURN)
            belegt Varibale mit einem Leerzeichen)

  Syntax: ACCEPT :"<Zeichenkette>": TO <tempVar>

  Beispiele: . ACCEPT "Wie ist Ihr Name?" TO NAME
             Wie ist Ihr Name?: (Antwort wird in NAME gespeichert)
           . ? NAME
             (Antwort wird auf dem Bildschirm angezeigt)
```

Abb. 1: eine mit DBHELP.BAS ausgedruckte Manualseite...

dBASE-Manual anders erstellt...

Das stört eigentlich wenig, wenn man nur weiß, daß der HP-150 eine MS-DOS-Maschine ist und deshalb die Prozedurfiles nicht die Extension CMD, sondern PRG haben. Jedenfalls ist laufend von PRG-Files die Rede.

Die im Helptext vorhandenen Stichworte HP150, KEYBOARD, LOCAL, TOUCH und SOFTKEYS kann man also getrost vergessen. Ebenso vergessen kann man zwei ausführliche Beschreibungen über ANWENDERPROGRAMME und BEISPIELE, die für den JOYCE nicht erhältlich sind.

Drucken eines dBASE-Manuals: Das BASIC-Programm DBHELP.BAS druckt mit Hilfe von DBASEMSG .TXT ein übersichtliches dBASE-Manual.

Die Seiten werden durchnumeriert, die Stichworte fett und vergrößert gedruckt und am Ende des Helptextes wird ein Inhaltsverzeichnis mit Seitenzahlen angefügt. Wer bei jedem Stichwort mit einer neuen Seite anfangen möchte, benötigt 80 Blatt Papier; wer darauf verzichtet, kommt mit 32 Blatt aus.

Eigenes Helpfile für dBASE ? Wer viele Probleme mit dBASE löst, wird sich bestimmt über die Möglichkeit freuen, eigene Texte an das vorhandene Textfile anzuschließen oder aber ein komplett neues Helpfile zu schreiben.

Mit jedem Texteditor läßt sich ein solches File schreiben, das von dBASE problemlos verarbeitet wird, sofern man sich an die Syntax

***Stichwort Text *EXIT**

hält.

(P.Horstmann)

```
LISTING >DBHELP  <, REMARK = >REM<.

<54> 1000 REM "DBHELP.BAS"
<57> 1010 REM druckt ein dBASE-Manual vom Help-File "DB
     ASEMSG.TXT"
<60> 1020 REM Peter Horstmann, 2805 Stuhr, 1987
<63> 1030 REM
<44> 1040 DIM index$(100),index(100):REM Stichworte und
      Seitenzahlen für Inhaltsverzeichnis
<26> 1050 esc$=CHR$(27):.......... REM Escape
<70> 1060 cls$=esc$+"E"+esc$+"H":.. REM Clear Screen
<62> 1070 elite$=esc$+"M":......... REM Druckerumschalt
     ung auf Elite-Schrift, 12 Zeichen/Zoll
<52> 1080 links$=esc$+"l"+CHR$(10): REM Linker Rand 10
<13> 1090 fettan$=esc$+"E":........ REM Fettschrift an
```

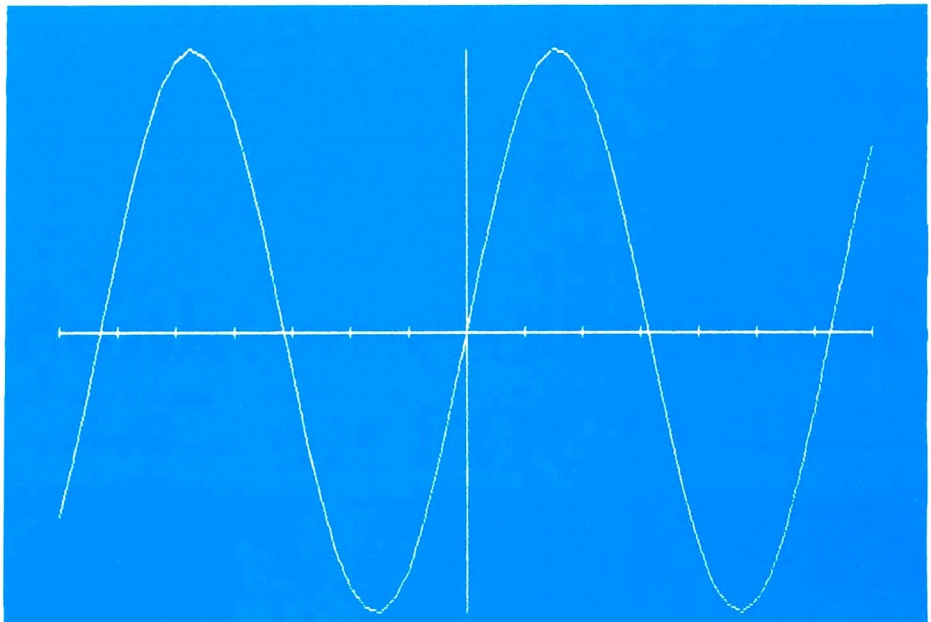
```
< 5> 1100 fettaus$=esc$+"F":....... REM Fettschrift aus
< 8> 1110 gran$=CHR$(14):.......... REM Vergößerte Schr
     ift an
< 2> 1120 graus$=CHR$(20):......... REM zurück auf Norm
     alschrift
< 7> 1130 LPRINT elite$+links$:.... REM Druckereinstell
     ung
< 7> 1140 PRINT cls$;"Ausdruck eines dBASE-Manuals... (
     Peter Horstmann, Stuhr)"
<68> 1150 PRINT..... "-------------------------------
     ----------------------":PRINT:PRINT
<52> 1160 INPUT "Ist das dBASE-Helpfile 'DBASEMSG.TXT'
     vorhanden ? ",q$
< 6> 1170 IF UPPER$(q$)<>"J" THEN PRINT "Bitte zuerst '
     DBASEMSG.TXT' kopieren und dann dieses Programm ne
```

```
     u starten !":END
< 9> 1180 PRINT:INPUT "Auf welchem Laufwerk (A:,B:,M:)
     ? ",drive$
<52> 1190 IF LEN(drive$)=1 THEN drive$=drive$+":": REM
     Nur Buchstabe eingegeben ?
<60> 1200 IF LEN(drive$)>0 THEN IF INSTR(1,"ABM$",LEFT$
     (drive$,1))=0 THEN PRINT "Unzulässiges Laufwerk !"
     :GOTO 1180
<90> 1210 datei$=drive$+"DBASEMSG.TXT"
<29> 1220 gefunden$=FIND$(datei$)
<89> 1230 IF gefunden$="" THEN PRINT "Datei nicht gefun
     den !":GOTO 1160
<75> 1240 PRINT:INPUT "Wünschen Sie einen Seitenvorschu
     b für jedes neue Hauptstichwort (J/N) ? ",vor$
<19> 1250 ON ERROR GOTO 1350
<63> 1260 OPEN "I",1,datei$
<63> 1270 IF UPPER$(vor$)<>"J" THEN GOSUB 1610:REM 1.Se
     itenkopf drucken
< 1> 1280 LINE INPUT #1,t$:..................... REM Zeil
     e lesen
<86> 1290 IF t$="*EXIT" THEN ein=0:GOTO 1280:. REM Ende
      eines Stichwortes
<39> 1300 IF LEFT$(t$,1)="*" AND NOT LEFT$(t$,2)="**" T
     HEN GOSUB 1500:GOTO 1280:REM Neues Stichwort
<50> 1310 LPRINT t$
<29> 1320 zeile=zeile+1
<92> 1330 IF zeile>60 THEN GOSUB 1610:........ REM Seit
     e voll
<39> 1340 GOTO 1280
<75> 1350 REM Fehlerbehandlung
< 5> 1360 CLOSE
<85> 1370 IF ERR=62 THEN 1390:.................. REM Date
     iende erreicht
<23> 1380 PRINT "Fehler";ERR;"in Zeile";ERL;"gefunden.
     Programm abgebrochen.":END
<87> 1390 REM Inhaltsverzeichnis drucken
< 3> 1400 t$="*INHALTSVERZEICHNIS"
```

```
< 9> 1410 GOSUB 1500
<62> 1420 FOR i=1 TO zaehler-1
<57> 1430 LPRINT "......... ";
<61> 1440 LPRINT USING "Ö.............................
     Ö";index$(i);
<90> 1450 LPRINT USING "####";index(i)
<43> 1460 zeile=zeile+1
< 7> 1470 IF zeile>60 THEN GOSUB 1610:........ REM Seit
     e voll
< 5> 1480 NEXT i
< 2> 1490 END
<64> 1500 REM Neues Stichwort
<90> 1510 ein=ein+1:....... REM Hauptstichwort: ein=1,
     Unterstichworte: ein>1
<93> 1520 IF ein=1 AND UPPER$(vor$)="J" THEN GOSUB 1610
< 5> 1530 x=INSTR(1,t$,">"):REM Sonderbehandlung für da
     s Stichwort CREATE (Fehler in DBASEMSG.TXT)
<24> 1540 IF x>0 THEN h$=LEFT$(t$,x-1):h1$=RIGHT$(t$,LE
     N(t$)-x+1) ELSE h$=t$:h1$=""
< 9> 1550 h$=STRING$((ein-1)*3," ")+RIGHT$(h$,LEN(h$)-1
     ):REM Unterstichworte 3 Zeichen einrücken
<60> 1560 LPRINT:LPRINT gran$+fettan$+h$+fettaus$+graus
     $:LPRINT
<84> 1570 zeile=zeile+3
< 2> 1580 zaehler=zaehler+1:index$(zaehler)=h$:index(za
     ehler)=seite: REM Stichwort und Seitenzahl merken
<95> 1590 IF h1$<>"" THEN LPRINT h1$:zeile=zeile+1
<90> 1600 RETURN
<69> 1610 REM Neue Seite
< 2> 1620 seite=seite+1
<56> 1630 LPRINT CHR$(12);"DBASEMSG.TXT";SPC(25);"- Sei
     te";seite;"-"
<11> 1640 LPRINT STRING$(80,"-")
<45> 1650 LPRINT:LPRINT:LPRINT
<14> 1660 zeile=6
<12> 1670 RETURN
```


Inserenten Joyce Sonderheft 2/87

ABD-Computer 61, 83
BBG 9
U. Becker 111
Büro für Softwareentwicklung 83
CMZ-Verlag 13
Computer-Studio Erding 109
Computer-Studio Reiss 63
CSS GmbH 113
Data-Becker 107
DMV 19,31,33,59,85,87,115,123
Effekta 105
Göddecker 85
Gruse 111
Hashagen 31
Heschcom 109
Keil 31
KMS-Soft 101
Kotulla 61
Micromarket 85
Naujoks 63
Rose 105
Signum-Medien-Verlag 37
Softdesign 9
Sunshine-Software 93
Star Division 2
TG-Soft 37
Van der Zalm 101
Vortex 124
Werder 13



Funktionenzeichnen mit LOGO

Auf den CPC's sowie der JOYCE läuft die LOGO-Version von Digital Research in etwas abgemagerter Form. Trotzdem stehen für die JOYCE mit etwa 140 Befehlen, von denen allerdings im Handbuch eine ganze Anzahl überhaupt nicht beschrieben sind, alle, für effektives Programmieren, wichtigen Primitiven zur Verfügung, was mit dem folgenden Programm »funzeich« gezeigt werden soll. Daneben bietet LOGO ein ausgezeichnetes Graphiksystem, das bei BASIC oder PASCAL mit dem GSX-System erst seit kurzem – und relativ kompliziert – zur Verfügung steht.

Mit dem Programm »funzeich« lassen sich mit der JOYCE unter LOGO die meisten mathematischen Funktionen auszeichnen. Unter Beachtung der LOGO-Syntax ist die Verwendung aller implementierter, sowie auch selbst definierter, arithmetischer LOGO-Funktionen zulässig.

Das Programm ist aus einer Reihe von Prozeduren zusammengesetzt, die sich z.T. durch ihre Namen selbst erklären. Im folgenden Text sind die Prozedurnamen jeweils in "" gesetzt.

Nach »eingabe« der Funktion F(x) werden mit »zeichenliste« die y-Werte berechnet. Die Verwendung der eingegebenen Funktion zur Berechnung der y-Werte wird dadurch möglich, daß die Prozedur »x« (op :x) dem 'x' in der Funktionsvorschrift den jeweils aktuellen Wert der Variablen x zuordnet. Mit »koord« erfolgt die Erstellung des Roh-Koordinatensystems in das nach Umwandlung der x/y-Werte in Bildschirmkoordinaten mit Hilfe von »dotrech« und Skalierung der x- und y-Achsen durch »skalax« und »skalay« der Funktionsgraph mit »graph« eingezeichnet wird. Die Steuerung des Gesamtablaufes wird von der Hauptprozedur »funzeich« übernommen. Mit »EXTRA PTR« kann schließlich eine Hardcopy, des in seiner Höhe optimal ausgenutzten Bildschirms, angefertigt werden.

Nach dem Einladen des Programmes »funzeich« (und ev. des Programmes »funktion«) ? load "funzeich (RETURN) (? load "funktion (RETURN)) wird »funzeich« gestartet ? funzeich (RETURN) und die Funktion F(x) eingegeben. Anschließend werden die Grenzen für die x-Werte und die Anzahl der zu berechnenden y-Werte erfragt. Die Zeichnung wird umso genauer, je größer die Zahl der y-Werte ist. Jedoch kann deren Anzahl in Abhängigkeit von der zu bearbeitenden Funktion in der Regel 60 bis 70 nicht überschreiten, da sonst die Fehlermeldung »out of LOGO stack« erfolgt. Dies hängt damit zusammen, daß LOGO selbst viel Speicherplatz benötigt, das Programm relativ lang ist und, daß alle Wertepaare vor dem Bildschirmaufbau zwischengespeichert werden. Jedoch ist die Anzahl der Stützpunkte mit 60 bis 70 für nicht oszillierende Kurven völlig ausreichend.

Die berechneten minimalen und maximalen y-Werte für das gewählte Intervall xmin/xmax werden zur optimalen Festlegung des Koordinatensystems auf dem Bildschirm ausgegeben. Danach können Koordinatenursprung und Skalierung der Achsen sinnvoll gewählt und auf Anfrage eingegeben werden. Unter frei wählbarem Namen läßt sich der Bildschirminhalt abspeichern.

Die Abb. 1 und 2 geben als Beispiele zwei Kurven mit ihren zugehörigen Parametern wieder. Bei einem Vergleich der Ausführungszeiten für die beiden abgebildeten Funktionen mit den, von der Konzeption her sehr unterschiedlichen, Programmen »funzeich« in LOGO und dem »Funktions-Plotter« von Jens Vygen ('Pascal International' 3/87) in PASCAL schneidet LOGO allerdings schlecht ab. So werden in LOGO für die Beispiele der beiden Abbildungen 3 bzw. 7 mal längere Ausführungszeiten gemessen.

Zur Eingabe des Programms ist zu beachten, daß alle im Programmausdruck angegebenen »Backslashs« (\ ), die die Trennereigenschaft nachfolgender Zeichen aufheben, mit der Taste »f3« oder »↑Q« eingebenen werden. Die Prozedur »adresse« wird für den Programmablauf nicht benötigt.

Die zusätzliche Datei »funktion« zeigt einmal eine Anzahl von selbstdefinierten Funktionen die nach Zuladen zum Programm »funzeich« in den Arbeitsspeicher dessen Möglichkeiten erweitern und sie zeigt andererseits, wie einfach und kurz in LOGO fehlende Funktionen wie Quadratwurzel (»sqrt«) oder Fakultät (»fak« (Fakultät rekursiv) bzw. »fakit« (Fakultät iterativ)) erzeugt werden können. Zu den beiden letzteren ist zu sagen, daß die rekursive Prozedur von der Programmierung her zwar eleganter ist als die iterative, jedoch wegen des hohen Speicherbedarfs nur 30! gegenüber 96! (iterativ) zu berechnen vermag. Bedeutung der selbstdefinierten Funktionen:

```
abs x   : Absolutwert von x
sgn x   : Signum von x
od      : True wenn x ungerade
ggt x y : Größter gemeinsamer Teiler
kgv x y : Kleinstes gemeinsames Vielfaches
bog x   : Umrechnung von Gradmaß in Bogenmaß
gra x   : Umrechnung von Bogenmaß in Gradmaß
tan x   : Tan x
sqrt x  : Quadratwurzel von x
fak x   : Fakultät von x
fakit x : Fakultät von x
```

LOGO, dem zu Unrecht das Image der Kindersprache anhaftet, ist anders als BASIC oder PASCAL aber, wie aus dem Programm hervorgeht, extrem leistungsfähig und höchsten Ansprüchen genügend.

Es ist allerdings in seiner vollen Breite nicht so leicht zu erlernen wie es das JOYCE-Benutzerhandbuch glauben machen möchte.

Eine ausgezeichnete Information zu dieser Programmiersprache, die auch auf mathematisch/naturwissenschaftlichem Gebiet einsetzbar ist, wird gegeben in:
»D.Senftleben«,
PROGRAMMIEREN IN LOGO.
Vogel-Verlag,
Würzburg 1986. (Helmut Schmidt)

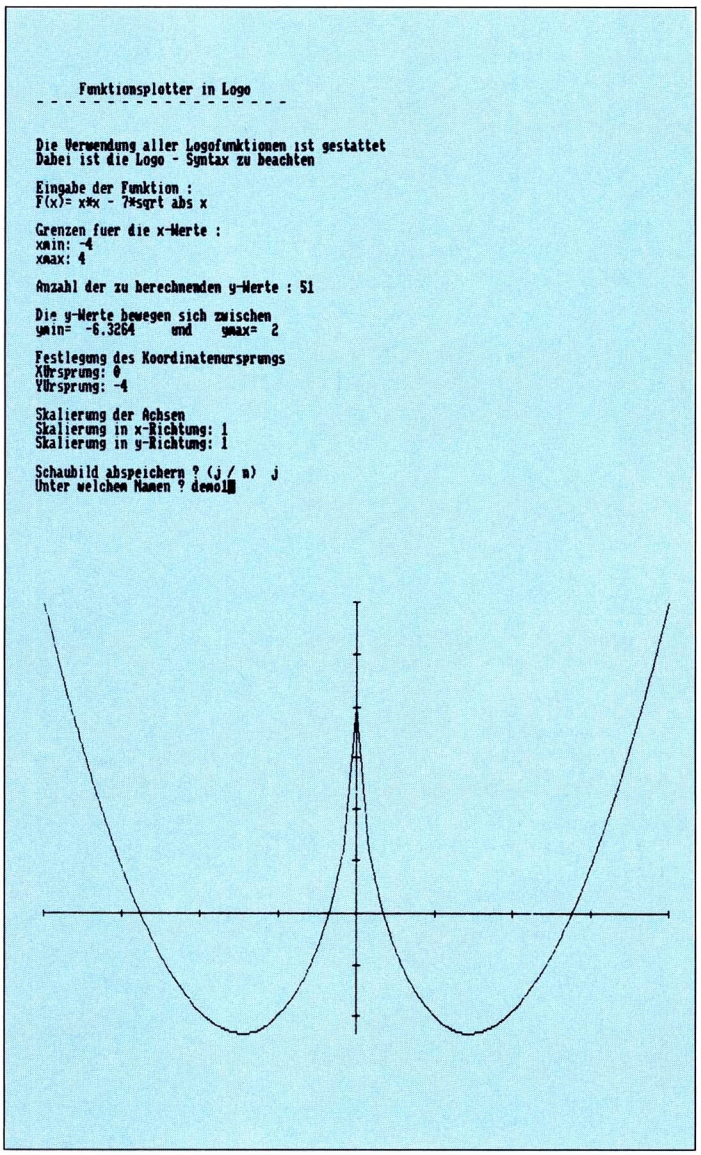


Abb. 1

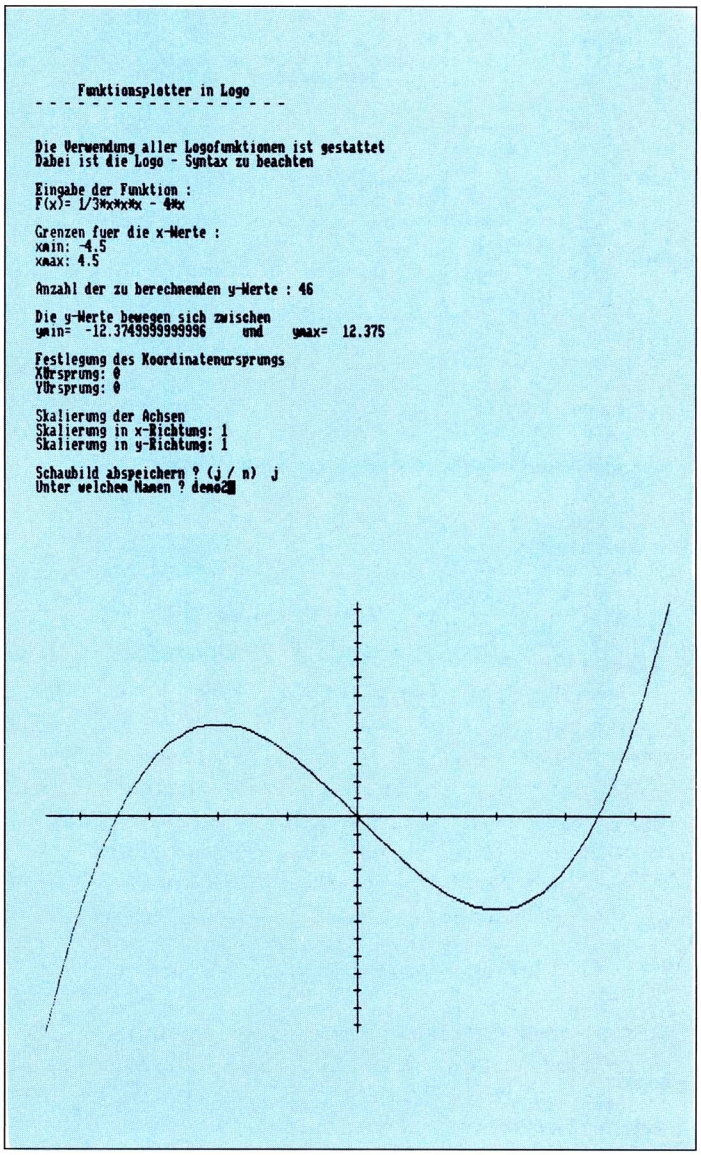


Abb. 2

```
to adresse
pr [Helmut Schmidt]
pr [7150 Backnang]
end
to eingabe
(local "funliste "xmin "xmax "anzschritte)
ts ct
pr [\ \ \ \ \ \ Funktionsplotter in Logo]
pr [- - - - - - - - - - - - - - - - - - -]
pr   []   pr   []   pr   [Die   Verwendung   aller
Logofunktionen ist gestattet]
pr [Dabei ist die Logo - Syntax zu beachten] pr []
pr [Eingabe der Funktion :]
type "F\(x\)\=\  make "funliste rl pr []
pr [Grenzen fuer die x\-Werte :]
type "xmin:\  make "xmin rq
type "xmax:\  make "xmax rq pr []
type "Anzahl\ der\ zu\  berechnenden\ y\-Werte\ :\
make "anzschritte rq
op (list :funliste :xmax :xmin :anzschritte)
```

```
end
to x
op :x
end
to zeichenliste :eingabe
(local "x "y  "schrittweite  "zlpunkte "ymin "ymax
"xur "yur "dx "dy)
make "zlpunkte []
make "schrittweite (/ (-  item  3  :eingabe item 2
:eingabe) (- item 4 :eingabe 1))
make "x item  2  :eingabe  make  "ymin  run item 1
:eingabe make "ymax :ymin
repeat  item  4  :eingabe  [make  "y  run  item  1
:eingabe make "zlpunkte (se  :zlpunkte :x :y) make
"x :x + :schrittweite  if  :y  > :ymax [make "ymax
:y] if :y < :ymin [make "ymin :y]]
pr [] pr [Die y\-Werte bewegen sich zwischen]
pr (list  "ymin\=\   :ymin  "\  \  \   "und\  \  \
"ymax\=\  :ymax)
pr [] pr [Festlegung des Koordinatenursprungs]
```

```
type "XUrsprung:\  make "xur rq
type "YUrsprung:\  make "yur rq
pr [] pr [Skalierung der Achsen]
type "Skalierung\ in\ x\-Richtung:\  make "dx rq
type "Skalierung\ in\ y\-Richtung:\  make "dy rq
pr [] type [Schaubild abspeichern ? (j / n) \ ]
label "r make "saveflag rc if not or (lc :saveflag
= "j) (lc :saveflag = "n) [go "r]
pr :saveflag
if lc :saveflag = "j [type "Unter\ welchem\ Namen\
?\  make "savename rq]
op (list item  2  :eingabe  item  3 :eingabe :ymin
:ymax :xur :yur :dx :dy :zlpunkte)
end
to zeichne :zeichenliste
(local "xfaktor "yfaktor "xmin "ymin)
make "xmin item 2 :zeichenliste
make "ymin item 3 :zeichenliste
make "xfaktor (/ 700 (-  item 1 :zeichenliste item
2 :zeichenliste))
make "yfaktor (/ 520 (-  item 4 :zeichenliste item
3 :zeichenliste))
fs cs ht koord graph item 9 :zeichenliste
end
to dotrech :koordliste
op (list  (-  (*  (-  item  1  :koordliste  :xmin)
:xfaktor) 350) (- (* (-  item 2 :koordliste :ymin)
:yfaktor) 260))
end
to koord
(local "y "x "pkt)
pu setpos dotrech (list  item 5 :zeichenliste item
3 :zeichenliste)
pd setpos dotrech (list  item 5 :zeichenliste item
4 :zeichenliste)
   setpos dotrech (list item  5 :zeichenliste item
6 :zeichenliste)
   setpos dotrech (list item  2 :zeichenliste item
6 :zeichenliste)
   setpos dotrech (list item  1 :zeichenliste item
6 :zeichenliste)
skalax skalay
end
to graph :punktliste
(local "n)
pu setpos dotrech (list item  1 :punktliste item 2
:punktliste) pd
make "n 3
repeat (- / count :punktliste 2 1) [setpos dotrech
(list item :n :punktliste item :n + 1 :punktliste)
make "n :n + 2]
end
to skalay
make "x  item  5  :zeichenliste  make  "y  item  6
:zeichenliste
repeat  (/  (-   item   4   :zeichenliste  item  6
:zeichenliste) item 8 :zeichenliste) [make "y :y +
item 8 :zeichenliste pu make "pkt
 dotrech (list :x :y) setpos (list - item 1 :pkt 4
item 2 :pkt) pd setpos (list  + item 1 :pkt 4 item
2 :pkt)]
make "x  item  5  :zeichenliste  make  "y  item  6
:zeichenliste
repeat  (/  (-   item   6   :zeichenliste  item  3
:zeichenliste) item 8  :zeichenliste)  [make "y :y
- item 8 :zeichenliste pu make "pkt
 dotrech (list :x :y) setpos (list - item 1 :pkt 4
item 2 :pkt) pd setpos (list  + item 1 :pkt 4 item
2 :pkt)]
end
to skalax
make "x  item  5  :zeichenliste  make  "y  item  6
:zeichenliste
repeat  (/  (-   item   1   :zeichenliste  item  5
:zeichenliste) item 7 :zeichenliste) [make "x :x +
item 7 :zeichenliste pu make "pkt
 dotrech (list :x :y) setpos  (list item 1 :pkt (-
item 2 :pkt 4))  pd  setpos  (list  item 1 :pkt (+
item 2 :pkt 4))]
make "x  item  5  :zeichenliste  make  "y  item  6
:zeichenliste
repeat  (/  (-   item   5   :zeichenliste  item  2
:zeichenliste) item 7  :zeichenliste)  [make "x :x
- item 7 :zeichenliste pu make "pkt
 dotrech (list :x :y) setpos  (list item 1 :pkt (-
item 2 :pkt 4))  pd  setpos  (list  item 1 :pkt (+
item 2 :pkt 4))]
end
to funzeich
(local "dummy "savename "saveflag)
type word char 27 0
label "anfang
recycle
cs zeichne zeichenliste eingabe
if lc :saveflag = "j [savepic :savename]
label "warte if not keyp [go "warte]
ss make "dummy rc
type [Noch eine Funktion ? (j / n)]
if lc rc = "j [go "anfang]
end
```

```
to abs :x
if :x < 0 [op -:x] [op :x]
end
to sgn :x
if :x = 0 [op 0] [op :x / abs :x]
end
to odd :x
op not remainder :x 2 = 0
end
to ggt :x :y
make "x abs :x
make "y abs :y
if and (not :x = 0) (not :y = 0) [go "s] [go "e]
label "s
make "x remainder :x :y
if not :x = 0 [make "y remainder :y :x]
if not or (:x = 0) (:y = 0) [go "s]
label "e
op :x + :y
end
to kgv :x :y
op (quotient :x ggt :x :y) * :y
end
to bog :x
op :x / 57.295
end
to tan :x
op (sin :x) / cos :x
end
to sqrt :x
(local "y "a)
if :x = 0 [op 0]
make "a :x
label "anfang
make "y :x
make "x (0.5 * (:x + :a / :x))
if abs :y - :x > 1.e-06 [go "anfang]
op :x
end
to gra :x
op :x * 57.295
end
to fak :x
if :x = 0 [op 1] [op :x * fak :x - 1]
end
to fakit :x
local "z make "z 1
repeat :x [make "z :z * :x make "x :x - 1] op :z
end
```

Grafik auf dem JOYCE:

GRAFMOD 1

mit neuem Layout

Hat es Sie bisher auch immer ein wenig gestört, daß man mit dem Programm GRAFMOD1 (veröffentlicht in PC INTERNATIONAL 1/1987, S. 96–103) zwar recht komfortable Diagramme erstellen und verwalten kann, daß die Säulendarstellung in den Diagrammen aber noch einiges zu wünschen übrig ließ, weil sie noch nicht differenziert genug war? Dann kann Ihnen jetzt mit einem Ergänzungsprogramm geholfen werden.

Als GRAFMOD1 entstand, war noch sehr wenig über das »Innenleben« des JOYCE und die Zugriffsmöglichkeiten auf interne Programmroutinen veröffentlicht.
Aus diesem Grunde wurden die Diagramme beim GRAFMOD1-Programm noch mit »herkömmlichen« Mitteln, d.h. unter Verwendung der Standardzeichen, auf den Bildschirm »gezaubert«. Dadurch ließen sich Ungenauigkeiten in der Höhendarstellung der Säulen nicht vermeiden, da immer nur »volle« Zeichen (und keine halb- oder viertelhohen usw.) ausgegeben werden konnten (Abb. 1).

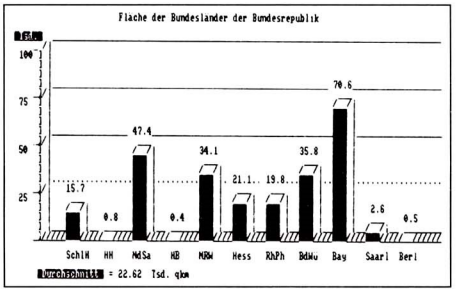


Abb.1: Eine mit GRAFMOD1 erstellte Grafik. (Sie wurde mit den in der Datei »GRAFMOD1.1« unter der Nr. 5 gespeicherten Daten erstellt.)

Nachdem im 1. JOYCE Sonderheft (4/87 S. 87f.) inzwischen 'verraten' worden ist, wie man den Zeichensatz des JOYCE beliebig verändern kann, ist es nun auch möglich, für das GRAFMOD1-Programm einen eigenen Zeichensatz zu 'stricken', durch den die Säulenhöhen so differenziert dargestellt werden, wie es der Bildschirm zuläßt (Abb. 2).
Wenn Sie auch in den 'Genuß' dieser Neuerung kommen möchten, brauchen Sie lediglich das Listing abzutippen und unter einem beliebigen Namen (z.B. "GMNEU") abzuspeichern. Dann laden Sie GRAFMOD1 und binden den neuen Teil mit dem Befehl »merge "gmneu"« in das alte Programm ein (nicht umgekehrt!).

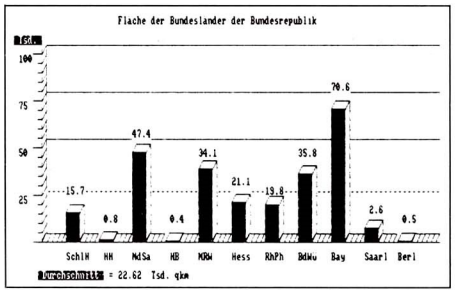


Abb.2: Die gleiche Grafik mit GRAFMOD2 erstellt.

Dies funktioniert allerdings nur, wenn Sie eine Version von GRAFMOD1 besitzen, die dem Listing in PC 1/87 genau entspricht! Andernfalls müßten Sie vor, während oder nach dem Abtippen das neue Listing an Ihre Version des Programms anpassen (Zeilennummern angleichen, Variablennamen ändern usw.).

Anmerkungen zum LISTING

Das Listing enthält sämtliche Zeilen zwischen Z. 10 und Z. 3250, in denen sich gegenüber der alten Version etwas geändert hat (Variablennamen, Wertzuweisungen usw.), dazu ein paar neue Zeilen und eine neue Programmroutine (ab Z. 6000), mit der beim Start des Programms der Zeichensatz des JOYCE ab der Zeichennummer 193 (Großes E Akut) in Grafikzeichen verwandelt wird (wie dies genau geschieht ist im JOYCE Sonderheft S. 87f. ausführlich beschrieben. Die dort vorgestellte Programmroutine findet sich hier in den Zeilen 7000–7280 wieder.)
Die für die einzelnen Grafikzeichen maßgeblichen Zahlenwerte stehen in den (DATA-) Zeilen ab Zeile 8000.
Die Zeilen 8010 bis 8120 werden Ihnen auf den ersten Blick etwas seltsam vorkommen. Durch diese besondere Verteilung der Zahlen auf die DATA-Zeilen ist es möglich, statt mit 288 Zahlen mit nur 71 Zahlen die 48 verschiedenen Zeichen für die Säulenspitzen herzustellen.

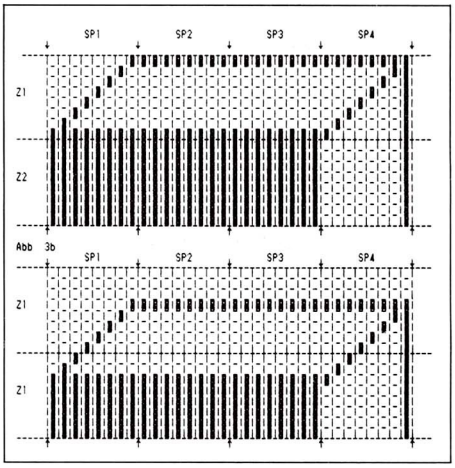


Abb.3

Wie aus Abb. 3 nämlich hervorgeht, bestehen die oberen beiden Bildreihen einer Säulenspitze aus insgesamt 8 Bildschirmzeichen (Z1/SP1 + Z1/SP2 + usw.), wobei jeweils die beiden inneren Zeichen (SP2 und SP4) identisch sind. Folglich müssen für eine Säulenspitze 6 verschiedene Bildschirmzeichen zur Verfügung stehen. Um acht verschiedene Säulenhöhen zu erreichen, benötigt man demnach 48 verschiedene Zeichen.
Wenn Sie das neue GRAFMOD-Programm gestartet haben, bleibt der geänderte Zeichensatz so lange erhalten, bis Sie den Rechner erneut starten. Falls Sie die DATA-BOX zum Sonderheft besitzen, können Sie den Zeichensatz jedoch auch mit dem darauf vorhandenen CP/M-Programm »ORIGINAL.COM« wiederherstellen. (Klaus Wiltsch)

```
LISTING >GRAFMOD+<, REMARK = >REM<.


<35> 10 '*************** INITIALISIERUNG ****************
     *
<31> 20 PRINT CHR$(27);"E";CHR$(27);"H";"Einen Moment b
     itte, der Zeichensatz wird geändert!":WIDTH 255,90
     :GOSUB 7120:GOSUB 6020
<48> 23 RESTORE 8140
<31> 25 FOR n=1 TO 15:READ zeichencode:GOSUB 7010:NEXT
<11> 300 '====== Zeichenelemente für Diagramm =========
     =
<13> 310 zchn.1$=STRING$(3,208)+CHR$(212)
<59> 320 zchn.2$=STRING$(84,201)
<23> 330 zchn.3$=STRING$(83,204)
```

```
<87> 340 zchn.4$=STRING$(83,206)
< 1> 350 '============ Titel/Copyright ================
     =
<94> 360 PRINT cl$;status.aus$;c.aus$
<48> 370 PRINT FNpkt$(5,34,STRING$(30,95))
<58> 380 PRINT FNpkt$(6,33,"/"+SPACE$(29)+"/"+CHR$(255)
     )
<68> 390 PRINT FNpkt$(7,32,"/"+SPACE$(29)+"/ "+CHR$(255
     ))
<57> 400 PRINT FNpkt$(8,31,"/"+SPACE$(29)+"/. "+CHR$(25
     5))
<12> 410 PRINT FNpkt$(9,30,"/"+SPACE$(29)+"/.. "+CHR$(2
     55))
<27> 420 PRINT FNpkt$(10,30 FNinv$(SPACE$(30))+".. /")
< 4> 430 PRINT FNpkt$(11,30,CHR$(255)+".... G R A F M O
```

```
      D. 2..... "+CHR$(212)+". /")
<23> 440 PRINT FNpkt$(12,30,FNinv$(SPACE$(30))+" /")
< 9> 450 PRINT FNpkt$(20,34,"(c) 1987 Klaus Wiltsch"):P
     RINT:PRINT:PRINT:PRINT:PRINT
<76> 740 PRINT TAB(36) FNinv$(u.an$+". -. "+u.aus$);"..
     . Wdhg.: Diagramm ohne Durchschnittsangabe"
<38> 2720 '************ DIAGRAMM ZEICHNEN ***************
     *
<88> 2730 PRINT FNpkt$(0,0,txt.prog$)
<98> 2740 li.mi!=ROUND(mittelwert!*100/teiler1*20/skala
     !,1)'-> Mittelwertlinie festlegen
<44> 2745 li.rest=ROUND((li.mi!-INT(li.mi!))*80/10,0)
<25> 2760 PRINT FNpkt$(2,8,txt$(1))'------------> Tite
     l
<82> 2770 '============== X-Achse ====================
     =
<49> 2780 PRINT FNpkt$(25,6,zchn.2$);FNpkt$(25,6,u.an$+
     "/"+u.aus$)
< 5> 2790 PRINT FNpkt$(27,8,txt$(2))'-----> Beschriftun
     g
<85> 2800 '============== Y-Achse ====================
     =
<72> 2810 PRINT FNpkt$(25,4,u.an$+CHR$(32)+CHR$(205)+u.
     aus$)
<31> 2820 FOR zeile=24 TO 5 STEP -1
<10> 2830... PRINT FNpkt$(zeile,5,CHR$(205)+CHR$(203)+z
     chn.3$)
<12> 2840 NEXT
<60> 2850 '------- Markierungen auf der Y-Achse -------
     -
<95> 2860 FOR zeile=20 TO 5 STEP -5
<83> 2870... PRINT FNpkt$(zeile,4,CHR$(95)+CHR$(205)+CH
     R$(203)+zchn.4$)
<25> 2880 '
<27> 2890 NEXT
< 3> 2900 '
<33> 2910 '------- Beschriftung d. Y-Achse ------------
     -
<92> 2920 txt.y=(skala!/4)*teiler1/100/teiler2
< 6> 2930 FOR zeile=21 TO 5 STEP -5
<57> 2940... PRINT FNpkt$(zeile,0,STR$(txt.y))
<90> 2950... txt.y=txt.y+(skala!/4)*teiler1/100/teiler2
<20> 2960 NEXT
< 8> 2970 PRINT FNpkt$(4,0,FNinv$(bereich$))
<82> 2980 '======== Mittelwertlinie zeichnen ==========
<95> 2985 IF line.control=1 THEN 3010
<22> 2987 IF li.rest=0 THEN li.mi!=li.mi!-1
<11> 2990 PRINT FNpkt$(24-INT(li.mi!),7,STRING$(83,CHR$
     (193+li.rest)))
<19> 3000 '========== Säulen zeichnen =================
     =
<29> 3010 schritt=ROUND(82/tabelle,0):tabulator=5+ROUND
     ((82-schritt*(tabelle-1))/2,0)
<54> 3020 FOR nr=1 TO tabelle
<93> 3024... zw.wert!=ROUND(wert(nr)*100/teiler1*20/ska
     la!,1)
<21> 3028... y.wert=INT(zw.wert!):y.rest!=(zw.wert!-y.w
     ert):top=ROUND(y.rest!*8,0)*6
<39> 3032... zchn.eo$=CHR$(207+top)+CHR$(209+top)+CHR$(
     209+top)+CHR$(211+top)
<93> 3036... zchn.eu$=CHR$(208+top)+CHR$(210+top)+CHR$(
     210+top)+CHR$(212+top)
<74> 3040... beschriftung$=MID$(STR$(ROUND(wert(nr)/tei
     ler2,7)),2,6)
< 9> 3050... IF wert(nr)=0 OR(top=0 AND y.wert<1) THEN
     3130 ELSE IF top=0 THEN y.wert=y.wert-1
<14> 3060..... FOR x=25 TO 26-y.wert STEP -1
< 0> 3070..... PRINT FNpkt$(x,tabulator,zchn.1$)
<27> 3080..... NEXT
< 8> 3090... PRINT FNpkt$(x,tabulator,zchn.eu$)
<74> 3100... PRINT FNpkt$(x-1,tabulator,zchn.eo$)
<51> 3110... PRINT FNpkt$(25,tabulator+3,u.an$+"/"+u.au
     s$)
<38> 3120... GOTO 3180
<81> 3130... PRINT FNpkt$(25,tabulator+1,u.an$+STRING$(
     2,206)+u.aus$)
< 6> 3140.. ' PRINT FNpkt$(24,tabulator,CHR$(32)+STRING
     $(2,95))
<78> 3150 '---- Werte zentr. über die S. schreiben ----
<10> 3155... IF wert(nr)=0 THEN 3190
<32> 3160... PRINT FNpkt$(23,tabulator+1+INT((4-LEN(bes
     chriftung$))/2),beschriftung$)
<69> 3170... GOTO 3190
<27> 3180... PRINT FNpkt$(x-2,tabulator+INT((5-LEN(besc
     hriftung$))/2),beschriftung$)
<51> 3190... tabulator=tabulator+schritt
<88> 3200 NEXT
<12> 3210 '======== Untere Zeile schreiben ============
     =
<17> 3215 IF line.control=1 AND INSTR(einheit$,"%")=0 T
     HEN PRINT FNpkt$(29,0,einheit$):RETURN:ELSE IF. li
     ne.control=1 THEN RETURN
<27> 3220 IF progr$="5" THEN PRINT FNpkt$(29,5,FNinv$("
      Angaben in Prozent! ")+". 100% =");summe.0#;" ";e
     inheit$:RETURN
<40> 3230 IF INSTR(einheit$,"%")<>0 THEN 3250
<57> 3240 PRINT FNpkt$(29,5,FNinv$(" Durchschnitt ")+"
     =");ROUND(mittelwert!/teiler2,2);bereich$;einheit$
< 0> 3250 RETURN
<19> 3260 '^^^^^ Ende v. Diagramm zeichnen ^^^^^^^^^^^^
     ^
<36> 6000 '======= Veränderung des Zeichensatzes für GR
     AFMOD1 =============
<57> 6010 '--- Abspeichern des Zeichensatzes für die "S
     äulenspitzen" ------
<52> 6020 zeichencode=206
<73> 6030 FOR zaehler1=1 TO 8
<53> 6040 ON zaehler1 GOTO 6050,6060,6070,6080,6090,610
     0,6110,6120
<89> 6050 RESTORE 8080:GOTO 6130
<79> 6060 RESTORE 8010:GOTO 6130
<98> 6070 RESTORE 8020:GOTO 6130
<18> 6080 RESTORE 8030:GOTO 6130
<37> 6090 RESTORE 8040:GOTO 6130
<28> 6100 RESTORE 8050:GOTO 6130
<47> 6110 RESTORE 8060:GOTO 6130
<66> 6120 RESTORE 8070:GOTO 6130
<44> 6130 FOR zaehler2=1 TO 6
<84> 6140 zeichencode=zeichencode+1
<28> 6150 IF zaehler2=3 OR zaehler2=5 THEN 6160 ELSE 61
     70
<54> 6160 FOR n=1 TO 8:READ x:NEXT
<47> 6170 GOSUB 7010
< 1> 6180 NEXT zaehler2
<85> 6190 NEXT zaehler1
<87> 6200 RETURN
<48> 7000 '---------- Zeichensatz verändern ----------
     -------------
<79> 7010 FOR i=0 TO 7
<82> 7020 READ x:POKE tabstart+i,x
<97> 7030 NEXT
<73> 7040 charstart=zeichencode*8+&HB800+2^16
<17> 7050 highaddr=INT(charstart/256)
<18> 7060 lowaddr=charstart-highaddr*256
<76> 7070 POKE &HF510,lowaddr
< 6> 7080 POKE &HF511,highaddr
<43> 7090 CALL codestart
<86> 7100 RETURN
<40> 7110 '---- Initialisierung des Zeichengenerators -
     -----
<54> 7120 MEMORY &HF4FF
<84> 7130 RESTORE 7200
<94> 7140 codestart=&HF500
<30> 7150 tabstart=&HF515
<83> 7160 FOR i=0 TO 20
<21> 7170 READ x:POKE codestart+i,x
<15> 7180 NEXT
<14> 7190 RETURN
<30> 7200 DATA &H01,&H09,&HF5
<35> 7210 DATA &HCD,&H5A,&HFC
<46> 7220 DATA &HE9,&H00
<12> 7230 DATA &HC9
<10> 7240 DATA &H21,&H15,&HF5
<93> 7250 DATA &H01,&H08,&H00
<56> 7260 DATA &H11,&H00,&H00
<44> 7270 DATA &HED,&HB0
<27> 7280 DATA &HC9
<26> 8000 '--------- Data für "Säulenspitzen"----------
     ------
<55> 8010 DATA 0
<58> 8020 DATA 0
<61> 8030 DATA 0
<64> 8040 DATA 0
<67> 8050 DATA 0
<70> 8060 DATA 0
<73> 8070 DATA 0
<61> 8080 DATA 1,2,4,8,16,32,64,255,255,255,255,255,255
     ,255,255,255
<28> 8090 DATA 0,0,0,0,0,0,0,0
<27> 8100 DATA 255,0,0,0,0,0,0,255,255,255,255,255,255,
     255,255,255
< 6> 8110 DATA 0,0,0,0,0,0,0,0
<93> 8120 DATA 255,3,5,9,17,33,65,129,1,1,1,1,1,1,1,1
<78> 8130 '--------- Data für sonstige Diagrammzeichen
     ------
<77> 8140 DATA 206,255,0,0,0,0,0,0
<42> 8150 DATA 205,1,1,1,1,1,1,255
<27> 8160 DATA 204,136,0,0,0,0,0,0
<25> 8170 DATA 203,1,3,5,9,17,33,65,129
<17> 8180 DATA 202,1,1,1,1,1,1,1
<25> 8190 DATA 201,255,2,4,8,16,32,64,255
< 1> 8200 DATA 200,136,60,0,0,0,0,0
<36> 8210 DATA 199,136,0,60,0,0,0,0
<42> 8220 DATA 198,136,0,0,60,0,0,0
<48> 8230 DATA 197,136,0,0,0,60,0,0
<54> 8240 DATA 196,136,0,0,0,0,60,0
<60> 8250 DATA 195,136,0,0,0,0,0,60
<66> 8260 DATA 194,136,0,0,0,0,0,60
<66> 8270 DATA 193,60,0,0,0,0,0,0
<23> 8280 DATA 255,192,192,192,192,192,192,192,192
```

Das Rauschgift-dezernat gibt bekannt:



Nach dem Genuß dieser Kombinationsdroge sind mittlerweile einige hundert Joyce-Benutzer bei Verlag und Autor als Abhängige registriert. Besondere Kennzeichen der Süchtigen: stundenlanges Verweilen vor ihrem Computer, massenweiser Papierverbrauch, Reduzierung der sozialen Kontakte, Verzicht auf regelmäßige Mahlzeiten und stark eingeschränkte verbale Kommunikation. Als ungewöhnlichen Nebeneffekt registrierten die Drogenfahnder leicht anschwellenden Bankkonten der Verfallenen: offensichtlich ist die Droge nach dem einmaligen Erwerb wiederverwendbar, so daß von einer echten Beschaffungskriminalität bisher nichts bekannt ist. Dazu hinterläßt das Rauschmittel, sozusagen als Nebenwirkung, bei den Triebhaften verblüffende Erfolge in deren beruflicher Schreibarbeit. Schwere Negative Begleiterscheinungen wie Abstürze, Fensterstürze oder Sturztrunkenheit konnte ein medizinisches Untersuchungsteam nicht feststellen. Die Beobachtungen werden in den kommenden Monaten fortgesetzt.

Aus dem Inhalt:

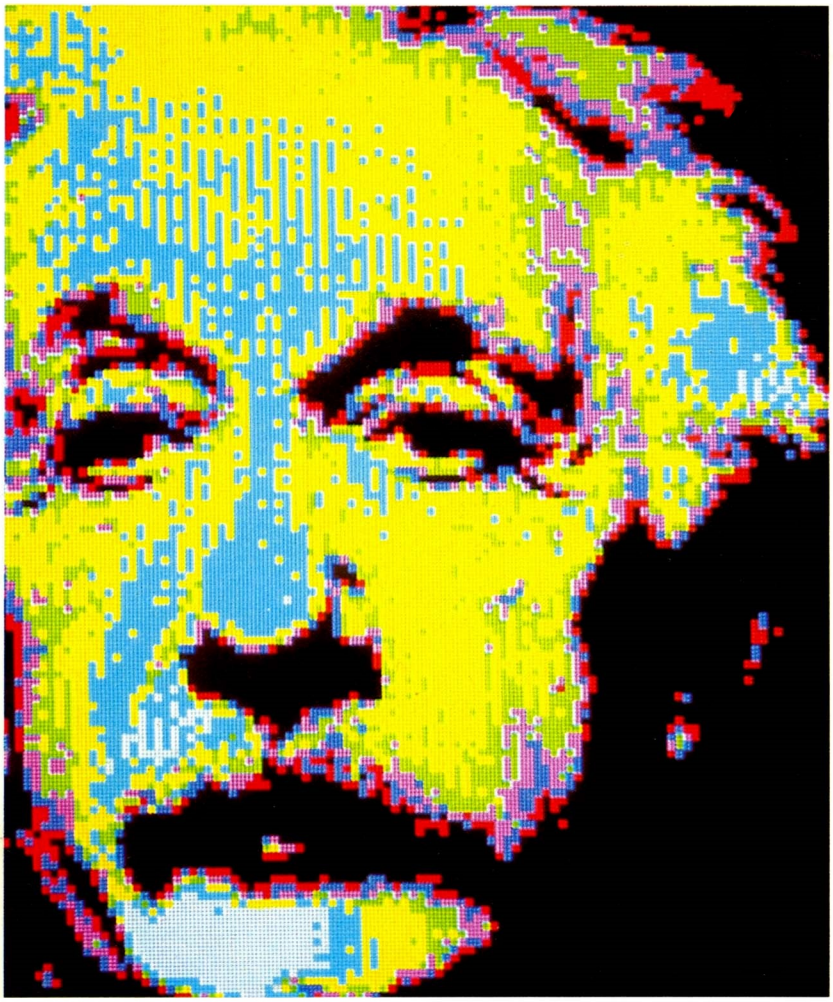
- LocoScript Spezial – Softwaretraining für Fortgeschrittene
- Fehler im System: Wie rette ich meinen Text?
- Joyce-Tasteninstallationsdatei für das Programm WordStar
- Aleatorische Poetik: Der Computer dichtet
- Auf Diskette: über 50 Dateien mit Schablonen, Briefen, Postkarten, Serien-Rundschreiben, Formularen, Etiketten, Druckbeispiele, Schriften, Bildschirm-Installationen u. v. m.

Leinen-Hardcover, 207 Seiten, 3"-Diskette
89,– DM (unverbindliche Preisempfehlung)

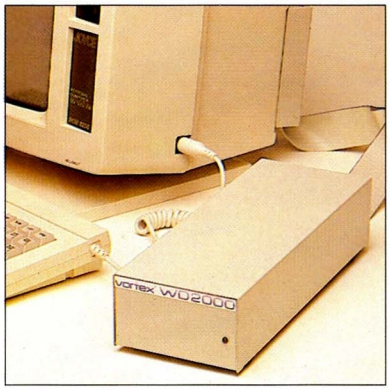
Zu Beziehen über den Computerfachhandel, den guten Fachbuchhandel oder direkt beim Verlag.
Händleranfragen erwünscht.

Ein DMV-Buch+3"-Diskette!

DMV-Verlag, Postfach 250, 3440 Eschwege



...UND PLÖTZLICH WIRD DER SCHNEIDER JOYCE ZUM ECHTEN TEXTSYSTEM.



Um es ganz offen zu sagen:

Was fangen Sie mit einem Textverarbeitungssystem an, das weniger als 10.000 DIN A4 Seiten Speicherkapazität hat?

Deshalb unsere Empfehlung:
Rüsten Sie Ihren Schneider Joyce mit unserem Festplattensystem **WD 2000** auf, damit Sie ihn tatsächlich professionell nutzen können.

Übrigens: Die vortex-Festplatte WD 2000 mit 20 MB Speicherkapazität paßt auch zu allen anderen Schneider CPC-Computern.

Der Infoscheck bringt weitere Informationen!

J/4/87

I·N·F·O-S·C·H·E·C·K

Bitte schicken Sie mir umgehend Unterlagen für die vortex-Festplatte WD 2000 für den Schneider Joyce.

_____

_____

_____

vortex Computersysteme GmbH
Falterstraße 51-53 · 7101 Flein · Telefon (07131) 52061

vortex
COMPUTERSYSTEME

...UND PLÖTZLICH HABEN SIE EINEN PROFI-COMPUTER