

Toolkits zum PC-1500

Das von unserer Artikelserie zum Sharp PC-1500 her bereits bekannte Autoren-Team beschreibt nachfolgend exklusiv für M+K einige nützliche Software-Erweiterungen zu diesem Gerät. Es handelt sich um zusätzliche Editierhilfen, schneller ablaufende Kassettenoperationen und eine bemerkenswerte BASIC-Erweiterung.

Die Befehlssatz-Erweiterung zum PC-1500 namens Toolkit sieht von aussen recht unscheinbar aus: ein mattglänzendes Aluminiumkästchen mit den Aussenmassen 8x7,5x2 cm. An der einen Längsseite ragt die 60-polige Steckerleiste zum Anschluss an den PC-1500 oder den Drucker/Plotter CE-150 heraus. Auch zwei deutlich beschriftete Buchsen (EAR/MIC) für den Kassettenrekorder sind vorhanden. Im Gegensatz zu älteren Toolkit-Modellen (die keine EAR-Buchse haben), kann man mit dem

**Albert A. Eggenberger
Lukas Zeller**

Gerät in der vorliegenden Version auch ohne CE-150 die Kassettenbefehle von Tool 2 (siehe unten) nutzen.

Beim Anschluss an den PC-1500 oder an der Rückseite des CE-150 ist (wie bei allen anderen Peripheriegeräten auch) darauf zu achten, dass der Computer ausgeschaltet ist, sonst gibt es einen «Absturz». Weil der Boden des eingesteckten Toolkit nicht bündig mit der Rechnerunterseite ist, bleibt ein Zwischenraum von etwa 5 mm zwischen Tisch und Tool, und die Steckerleiste wird bei versehentlichem Druck auf das Gehäuse leider etwas strapaziert. Diese Unschönheit könnte der Hersteller z.B. durch angeklebte Gummifüsschen beheben.

Im Tool befindet sich neben der Elektronik die Kassettenrekorderbuchsen ein 8 KByte CMOS-EPROM, das mit einem oder mehreren zwei KByte langen Softwareblöcken programmiert ist (siehe Kästchen). Unser Testgerät war mit vier der derzeit fünf erhältlichen Softwaremodulen voll bestückt: TOOL 1, TOOL 2, TOOL 3 und RWE-BASIC. Eine Zusammenstellung aller Befehle mit Kurzbeschreibung zeigt Tabelle 1.

TOOL 1: Editierhilfe

Wer mit MERGE arbeitet, kennt das Problem: Wird ein Programm von der Kassette zum bestehenden Programm dazugeladen, so ist nach-

her nur noch dieses Programm editierbar; die anderen Programme können nur noch laufengelassen werden. Geübte «PEEKer und POKEer» können mit «Umbiegen von Pointern» (so der Fachjargon) dieses Hindernis beseitigen. Viel eleganter und benutzerfreundlicher geht das mit TOOL 1: Mit PROGRAM kann jedes der BASIC-Programme im Speicher zum sogenannten «aktiven Programm-Modul», d.h. editierbar gemacht werden. Dazu können aber auch ohne MERGE weitere Module eröffnet oder einzelne Module gelöscht werden. Ebenso kann ein Modul in zwei aufgespalten oder zwei aufeinanderfolgende Module können zu einem verkettet werden. Auch Listbefehle (PLIST, PLAST), die das aktive Programm-Modul betreffen, sind vorhanden, da mit dem normalen LIST die Zeilenangaben immer auf das erste Programm-Modul bezogen werden.

Mit diesen zusätzlichen Befehlen ermöglicht TOOL 1 sogenannte modulare Programmierung, d.h. die Aufteilung eines Problems in einzelne, klar definierte Aufgaben. Jede Aufgabe wird in einem Modul separat gelöst; zum Schluss werden die Module miteinander verknüpft. In einem Anhang der Bedienungsanleitung sind die Grundlagen der modularen Programmierung erläutert.

TOOL 1 enthält aber auch noch andere Befehle, die man beim PC-1500 oft vermisst hat: DELETE löscht mehrere Programmzeilen auf einmal und RENUMBER ermöglicht die Neu-numerierung von Programmen, wobei alle GOTO-, GOSUB-, RESTORE- und THEN-Anweisungen den neuen Zeilennummern angepasst werden. Bei unserem Testgerät waren die GSB- und FN-Aufrufe des RWE-BASIC bei RENUMBER noch nicht berücksichtigt; laut Angaben des Herstellers wurde das jedoch unterdessen geändert.

CHANGE und FIND sind vor allem in langen Programmen (z.B. auch Assembler-Quellentext) sehr hilfreich, wenn es z.B. darum geht, eine Variable umzubenennen.

Für den Fall, dass man versehentlich NEW tippt, kann man das Programm mit KEEP zurückholen. Nach

ERROR 44 (Lesefehler) macht KEEP den schon geladenen Teil des Programms editierbar. Wunder kann man jedoch von KEEP nicht erwarten; durch unüberlegtes POKEen verunstaltete Programme können auch mit KEEP nicht gerettet werden.

TOOL 2: Schnelle Kassettenoperationen

Die Tool 2-Kassettenbefehle entsprechen von der Funktion her den SHARP-Befehlen weitgehend. Neu hinzugekommen ist die Möglichkeit, ein einzelnes Programmmodul separat abzuspeichern, ebenso den Befehl VERIFY, der nicht nur CLOAD? ersetzt, sondern auch das bei Sharp fehlende CLOAD M? und INPUT #? enthält. Bei FSAVE V/FLOAD V besteht leider die Einschränkung, dass abgespeicherte Standard-Stringvariablen (A\$,Z\$) nur wieder in Standard-Variable zurückgelesen werden können; dasselbe gilt auch für die Arrays @(*) und @\$(*): Sie können nicht in einen dimensionierten Array zurückgelesen werden, was bei der Sharp-Software möglich ist.

Im Vergleich mit Standard-Kassettenbefehlen laufen die TOOL 2-Befehle nach Herstellerangaben fünf- bis zwanzigmal schneller. Wir haben zwei Tests durchgeführt, wobei die Zeitangaben ohne den Vorsparn (bei Sharp ca. 8 sec, bei TOOL 2 ca. 1 sec) zu verstehen sind, um die tatsächliche Übertragungsrate aufzuzeigen.

BASIC-Programm mit einer Länge von 16'458 Bytes:

Sharp: 19 Min. 32 Sek.
TOOL 2: 50 Sekunden

Verschiedene Variablen:
A,B,A\$,B\$,VA(1,1),TX\$(1,1),TX\$,AB

Sharp: ca. 32 Sekunden
TOOL 2: ca. 16 Sekunden

Im ersten Test erweist sich TOOL 2 als etwa 23,5 mal, beim zweiten nur noch doppelt so schnell wie die Sharp-Software. Das liegt daran, dass zwischen den einzelnen Variableninhalten jeweils ein Zwischenspann kommt, da Zeit für die Auffindung der nächsten Variablen benötigt wird. Sobald man längere Variablen abspeichert, kommt die Geschwindigkeit von TOOL 2 wieder voll zum Zug.

Die Ansprüche an den Kassettenrekorder und das Bandmaterial sind etwas höher als bei der Sharp-Soft-

PPC/HHC

| Belehl | Kurzbeschreibung | Tabelle 1 | programm- kontrolliert | manuell | Funktion |
|---|--|-----------|---------------------------|---------|----------|
| Tool 1 (Editor-Hilfe) | | | | | |
| APPEND | Eröffnet ein neues Programm-Modul | | | • | |
| CHANGE | Ersetzt «Ausdruck 1» durch «Ausdruck 2» im aktiven Programm-Modul | | | • | |
| DELETE | Löscht im aktiven Programm-Modul die angegebene(n) Zeile(n) | | | • | |
| ERASE | Löscht das angegebene Programm-Modul | | | • | |
| FIND | Sucht «Ausdruck» im aktiven Programm-Modul | | | • | |
| KEEP | Ermöglicht das Editieren eines unvollständig eingelesenen oder gelöschten Programm-Moduls | | | • | |
| LINK | Verbindet das aktive Programm-Modul und das nachfolgende Modul zu einem Programm-Modul | | | • | |
| PLAST | Listet die letzte Zeile im aktiven Programm-Modul | | | • | |
| PLIST | Listet die angegebene Zeile im aktiven Programm-Modul | | | • | |
| PROGRAM | Macht das angegebene Programm-Modul aktiv | | | • | |
| RENUMBER | Numeriert das aktive Programm-Modul neu | | | • | |
| SPLIT | Unterteilt das aktive Programm-Modul bei der angegebenen Zeile in zwei unabhängige Programm-Module | | | • | |
| Tool 2 (Tape-Operation) | | | | | |
| FCHAIN | Erweitertes CHAIN | | • | | |
| FLOAD | Entspricht der Instruktion CLOAD | | | • | |
| FLOAD P | Erweitertes MERGE | | • | • | |
| FLOAD M | Entspricht der Instruktion CLOAD M | | • | • | |
| FLOAD V | Entspricht der Instruktion INPUT # | | • | • | |
| FSAVE | Entspricht der Instruktion CSAVE | | • | • | |
| FSAVE P | Speichert nur das aktive Programm-Modul auf Band | | • | • | |
| FSAVE M | Entspricht der Instruktion CSAVE M | | • | • | |
| FSAVE V | Entspricht der Instruktion PRINT # | | • | • | |
| VERIFY | Vergleicht die Daten auf dem Band mit den im Computer gespeicherten Informationen; es können alle Datenarten überprüft werden | | • | • | |
| Tool 3 (weitere Funktionen) | | | | | |
| CLR | Setzt den Inhalt der angegebenen Variablen, bzw. DIM-Felder auf Null | | • | • | • |
| DEC | Berechnet den Dezimalwert einer als Zeichenfolge angegebenen hexadezimalen Zahl | | • | • | • |
| ERL | Liefert die Zeilennummer, in welcher zuletzt ein ERROR auftrat | | • | • | • |
| ERN | Liefert die Nummer des zuletzt aufgetretenen ERRORs | | • | • | • |
| FRC | Liefert den Nachkommateil eines numerischen Ausdrucks | | • | • | • |
| FRE | Erweitertes MEM; berücksichtigt DIM-Felder und 2-Zeichen-Variablen | | • | • | • |
| HEXS | Wandelt den Wert eines numerischen Ausdrucks in die hexadezimale Zeichenfolge um | | • | • | • |
| INSTR | Liefert die Position des Beginns des Such-Ausdrucks im Haupt-Ausdruck | | • | • | • |
| PGM | Liefert die Nummer des aktiven Programm-Moduls | | • | • | • |
| PSIZE | Liefert die Anzahl Bytes, die das aktive Programm-Modul im Speicher belegt | | • | • | • |
| PURGE | Löscht die angegebenen 2-Zeichen-Variablen, bzw. DIM-Felder, aus dem Speicher | | • | • | |
| REDIM | Dimensioniert bereits bestehende DIM-Felder neu | | • | • | |
| RESUME | Ermöglicht, zusammen mit ON ERROR, die Fortsetzung des Programmablaufs nach einem ERROR mit der nächstfolgenden Programm-Anweisung | | • | | |
| STRINGS | Liefert ein beliebiges Zeichen oder eine beliebige Zeichenfolge so oft wie angegeben | | • | • | • |
| SWAP | Vertauscht die Inhalte der angegebenen Variablen | | • | • | |
| VKEEP | Ermöglicht die Adressierung von 2-Zeichen-Variablen und DIM-Feldern nach CLEAR, NEW, PURGE oder RUN | | | • | |
| VLIST | Listet 2-Zeichen-Variablen und DIM-Felder mit Namen, Datenart, Dimension(en) und Länge auf | | | • | |
| RWEbasic (Strukturdirektiven und Funktionen) | | | | | |
| DO ... WHILE ... LOOP | Strukturschleife | | • | | |
| EXIT | Sofortiger Ausstieg aus der Strukturschleife | | • | | |
| IF ... THEN ... ELSE ... ENDIF | Erweiterung von IF ... THEN | | • | | |
| SELECT ... CASE ... ENDSELECT | Mehrfachverzweigung | | • | | |
| SUB ... SUBEND | Unterprogrammdefinition | | • | | |
| GSB | Unterprogrammaufruf für mit SUB ... ENDSUB delinierte Unterprogramme | | • | | |
| SUBCLR | Löscht lokale Variable nach Fehlermeldung oder BREAK innerhalb des SUB-Programmes | | • | • | |
| EFFN | Funktionen definieren | | • | | |
| FN | Aufruf definierter Funktionen | | • | • | • |
| FAC | Fakultät | | • | • | • |
| INTEGRAL | Integral nach der Simpsonschen Regel | | • | • | • |
| ROUND | Dient zum Runden von Zahlen | | • | • | • |

ware. Unsere Tests liefen problemlos mit einem normalen Monogerät und Agfa Fel C60-Kassetten.

Wird TOOL 2 mit dem CE-150 zusammen betrieben, so können zwei Kassettengeräte ferngesteuert werden, arbeitet man mit dem Toolkit allein, muss der Bandlauf von Hand geschaltet werden.

TOOL 3: Diverse Befehle

Von den TOOL 3-Befehlen brauchen nur wenige näher erläutert zu werden, da die Funktion bei den meisten in Tabelle 1 genügend beschrieben ist.

Mit REDIM lassen sich bereits dimensionierte Felder in ihrer Ausdehnung verändern. Dazu ein Beispiel: Es ist ein Array AB(1,1) definiert. Dann wird REDIM AB(2,1) ausgeführt. AB hat nun eine »Zeile« mehr als vorher; die Inhalte der ursprünglichen Elemente (AB(0,0), AB(0,1), AB(1,0), AB(1,1)) wurden dabei nicht verändert.

Bei Textfeldern kann auch die Zeichenanzahl verändert werden, ebenso bei Textvariablen mit zwei Buchstaben im Namen: Mit ABS*» und REDIM ABS*80 wird eine Variable ABS erzeugt, die 80 Zeichen fassen kann.

Die Bedienungsanleitungen zu TOOL 1 bis TOOL 3 sind gut gegliedert. Mit vielen kommentierten Beispielen werden auch komplizierte Befehle anschaulich erklärt. Einzig die Syntaxdiagramme sind manchmal etwas unübersichtlich.

Das RWE-BASIC

Der vierte Softwareblock in unserem Toolkit ist das RWE-BASIC, das von einer kleinen Softwarefirma in Deutschland entwickelt wurde. Die darin enthaltenen Befehle unterstützen vor allem strukturierte Programmierung.

Doch zuerst ein Wort zu den Nicht-Strukturbefehlen. FAC liefert die Fakultät einer Zahl, z.B.: FAC 5 = 1*2*3*4*5 = 120. ROUND dient zum Runden einer Zahl auf die mit USING festgelegte Anzahl Nachkommastellen, z.B. ergibt USING «.###» und danach A=ROUND(78/9) für A den Wert 8.6667. Mit ROUND kann aber auch das USING-Format festgelegt werden, ohne Dutzende von #s eingeben zu müssen. Die INTEGRAL-Funktion berechnet das Integral einer Funktion nach der Simpsonschen Näherung.

Doch nun zu den Strukturbefehlen des RWE-BASIC. Im Testgerät hatten

wir eine alte Version mit einigen Fehlern. Unterdessen wird eine verbesserte und auch an TOOL 1-RENUMBER angepasste Version geliefert; die Syntax von FN ist dabei verändert worden. Da wir die neue Software nicht mehr rechtzeitig beschaffen konnten, stützen wir uns bei Hinweisen betreffend die neue Version nur auf Angaben des Herstellers.

Der Vorteil der strukturierten Programmierung ist einerseits die Unabhängigkeit von Zeilennummern, andererseits werden gut strukturierte Programme übersichtlicher und (sobald man sich in den Struktur-Anweisungen auskennt) auch leichter verständlich.

IF#...THEN...ELSE...ENDIF ist eine Erweiterung des normalen IF...THEN: Wenn die Bedingung nach IF# erfüllt ist, so wird der Programmteil zwischen THEN und ELSE ausgeführt (auch wenn es mehrere Zeilen sind), und danach läuft das Programm nach ENDIF weiter. Ist die Bedingung nicht erfüllt, so kommt der Programmteil ab ELSE zur Ausführung. Eine Anwendung zeigt Listing 1.

DO...WHILE...LOOP: Der Programmteil zwischen DO und LOOP wird solange ausgeführt, bis die Bedingung bei WHILE nicht mehr erfüllt ist. Als Beispiel zeigen wir ein Zahlenratespiel (Listing 1). Man beachte, dass das Programm kein einziges GOTO enthält. Die Leerstrings in den Zeilen 40 bis 80 dienen nur zum Einrücken der Befehle.

SELECT...CASE...ENDSELECT, Mehrfachverzweigung. Der Ausdruck nach SELECT wird bei jedem CASE für den Vergleich herangezogen. Dasselbe Zahlenratespiel wie in Listing 1 zeigt auch Listing 2, hier jedoch mit SELECT programmiert.

DEFFN und FN: Mit DEFFN kann eine Funktion definiert werden. Ein Beispiel:

```
10:KUBIKWURZEL=DEFFN
  (AB)=A^(1/3)
```

Diese Funktion kann nun mit FN (KUBIKWURZEL,X) aufgerufen werden und liefert als Ergebnis die Kubikwurzel von X. Anstelle von KUBIKWURZEL kann auch die Zeilennummer (10) gesetzt werden. Bei der neuen Version des RWE-BASIC lautet der Aufruf:

```
(FN KUBIKWURZEL):(X)
```

Die Variable AB in Zeile 10 (siehe oben) ist lokal, das heisst, sie wird nur während der Berechnung der

```
10:REM Zahlenratespiel
20:A=RND(100):REM A muss erraten werden
30:DO
40:" INPUT "Versuch=";B
50:" WHILE A>B:REM Ausstieg aus der
  Schleife wenn A=B
60:" IF A>B THEN PAUSE "zu klein"
70:" ELSE PAUSE "zu gross"
80:" ENDIF
90:LOOP
100:PRINT "Bravo, richtig !"
110:END
```

Listing 1

Funktion kurz angelegt und nachher sofort wieder gelöscht. Eine globale (d.h. zuvor schon benützte) Variable mit dem Namen AB geht dabei nicht verloren. Es können mit DEFFN auch Stringfunktionen oder Funktionen mit mehreren Argumenten definiert werden (Listing 3).

SUB...SUBEND und GSB. Um diese Befehle detailliert zu beschreiben, reicht hier der Platz einfach nicht. Darum soll eine Anwendung dieser Befehle anhand von Listing 4 erklärt werden.

Das Unterprogramm »RAHMEN« ist leicht zu verstehen, wenn man einmal von Zeile 100 absieht: Auf dem Drucker wird der Inhalt von TX\$, umrahmt von dem Zeichen in RA\$, ausgedruckt. Danach pipst es BE mal. Nun zu Zeile 100: SUB TX\$,RA\$,BE legt fest, dass das Unterprogramm drei Parameter hat, die beim Aufruf übergeben werden müssen. Das angehängte »CN« definiert CN als lokale Variable, d.h. sie wird beim Aufruf des Unterprogramms angelegt und bei SUBEND wieder gelöscht. Wird nun RUN 10 ausgeführt, so passiert folgendes: Der String »PC-1500« wird in die lokale Variable TX\$ geladen, »*« in RA\$ und 3 in BE. Dann wird das Unterprogramm ausgeführt. Bei SUBEND werden alle lokalen Variablen gelöscht (also TX\$,RA\$,BE und CN). Bei RUN 30 geschieht beinahe dasselbe. Im Unterschied zu Zeile 10 wird in 40 nicht einfach ein Text übergeben, sondern die Variable TES. Bei SUBEND wird nun TX\$ vor dem Löschen nach TES zurücktransportiert.

In SUB und GSB sind noch weitere Möglichkeiten eingebaut, z.B. die Übergabe eines Variablenfeldes.

```
10:A=RND(100)
20:INPUT "Versuch=";B
30:SELECT B
40:CASE <APAUSE "zu klein"
50:CASE >APAUSE "zu gross"
60:CASE =APRINT "Bravo, richtig !":
  END
70:ENDSELECT
80:GOTO 20
```

Listing 2

PPC/HHC

```
10:REM Textfunktion
20:"KLAMMER"DEFFN (AB*)="( "+AB*+" )"
30:REM Funktion mit mehreren Paramet
  *n
40:"PYTHAGORAS"DEFFN (A),B)=SQRT (A
  *A)+B*B)
FN ("KLAMMER", "PC-1500")
( PC-1500 )

FN ("PYTHAGORAS", 3, 4)
```

Listing 3

Zusammenfassend kann man sagen, dass das RWE-BASIC echtes strukturiertes Programmieren mit vielen Möglichkeiten (vor allem was lokale Variablen anbetrifft) bietet, die sonst nur in höheren Programmiersprachen vorhanden sind.

Leider ist die Anleitung nicht so anschaulich wie bei den TOOLS; man muss von strukturierter Programmierung schon eine deutliche Ahnung haben, um die Erklärungen zu verstehen. Insbesondere werden die Fachausdrücke wie Rekursion, globale/lokale Variablen etc. nirgends erklärt. Vermisst haben wir auch genauere Angaben über den Speicherverbrauch bei lokalen Variablen. Wie wir mittels Hexmonitor ermitteln konnten, legt das RWE-BASIC bei SUB-Aufrufen als Variablen «getarnte» Tabellen an, die bei SUBEND angeben, welche Variablen entfernt werden müssen. Diese Tabellen bewirken bei VLIST die Anzeige von seltsamen Variablennamen.

Schlusswort

Gesamthaft gesehen ist das getestete Gerät eine sehr leistungsfähige und gut durchdachte Software-

```
10:GSB "RAHMEN": "PC-1500", "A", 3
20:END
30:TEX="SHARP"
40:GSB "RAHMEN":TEX, "+", 5
50:LPRINT TEX
100:"RAHMEN"SUB TX*,RA*,BE:CN
110:FOR CN=1TO LEN TX*+4:LPRINT RA*::
  NEXT CN:LPRINT
120:LPRINT RA*+" "+TX*+" "+RA*
130:FOR CN=1TO LEN TX*+4:LPRINT RA*::
  NEXT CN:LPRINT
140:LPRINT
150:BEEP BE
160:TEX="FERTIG"
170:SUBEND

*****
* PC-1500 *
*****
RUN 30
*****
+ SHARP +
*****
```

FERTIG

Listing 4

Was - wo - wieviel?

fp. Nachfolgend geben wir einige wichtige Daten zu den getesteten Modulen wieder. M+K ist leider nicht in der Lage, über weitere Details Auskunft zu geben. Leser wenden sich bitte für alle Fragen an die Lieferanten: SRS Ing. Rudolf W. Fankhauser, 8213 Neunkirch (für Schweiz und Ausland ausser BRD). Ing. Rudolf W. Fankhauser, Postfach 1115, D-7893 Jestetten.

TOOL 1 «Editor-Hilfe» Fr. 102.--

TOOL 2 «Fast Tape Operationen» Fr. 102.--

TOOL 3 «Funktionen» Fr. 102.--

Hardware (Alu-Kästchen, Zubehör) Fr. 168.--

Hersteller dieser Artikel: TRAMsoft, Rümlang (CH)

RWEbasic Fr. 75.--

Hersteller: R. + W. Eckstein, Zirndorf (BRD)

Erweiterung. Die Programme sind offenbar genauestens getestet; wir haben trotz intensivem Suchen keinen Fehler finden können, der nicht bereits von den Autoren entdeckt und in neueren Versionen behoben worden ist!

Der einzige Haken am Toolkit ist für den Normalverbraucher der Preis des Geräts. Ein mit vier Softwaremodulen ausgerüsteter Toolkit kostet nämlich annähernd soviel wie ein PC-1500. «Zu teuer?», fragt man sich mit Recht, und ist versucht, die Frage mit ja zu beantworten. Doch das wäre ein vorschnelles Urteil. Es gilt, um den Preis fair beurteilen zu können, auch zu berücksichtigen, dass der Toolkit von einer kleinen Schweizer Firma entwickelt und nur in kleinen Stückzahlen gefertigt und verkauft wird. So gesehen ist der Preis für den an Japan-Massenprodukte gewohnten Konsumenten zwar hoch, aber nicht überrissen. □

nach wie vor die bessere Lösung, denn Textverarbeitungsprogramme zeigten noch nicht die Anwenderfreundlichkeit eines Textsystems. M+K meint: Einverstanden - aber dies muss ja nicht so bleiben, gegenwärtige vermeintliche Standards sind doch erheblich im Fluss. Und: die Preise eines PC, der ungleich mehr kann als nur Texte verarbeiten, sind ja bald einmal niedriger als die eines Textautomaten. Für einen potentiellen Kunden bietet sich zur Zeit also eine recht heikle Entscheidungslage. □

Schulsoftware-Katalog Schweiz

(220/tp) Rechtzeitig zur Didacta konnte er herausgebracht werden, der lange versprochene Schulsoftware-Katalog Schweiz vom Schulbuchverlag Klett. Das Büchlein hat 114 Seiten und ist unterteilt in folgende Kapitel: Informatik (-Unterricht), Mathematik, Physik/Chemie/Biologie, Sprachen, Verwaltung, Sonstiges. In diesen Sparten werden insgesamt 147 Programme oder Programmpakete angeboten, die von Lehrern oder kommerziellen Autoren aus der Schweiz stammen. Die Beschreibung der Programme ist kurz und korrekt wie auch die Angaben über Konfiguration usw. Im Anhang finden wir ein wieder nach Kategorien gegliedertes, alphabetisches Softwareregister (wobei man aus Programmtiteln ja in der Regel wenig klug wird...), ein Verzeichnis der verwendeten Geräte sowie der kommerziellen Anbieter. Klett versteht sich mit diesem Katalog ausschliesslich als Marktplatz, auf dem sich Käufer und Verkäufer selber einig werden müssen. Der Katalog soll periodisch erweitert werden. □



Ein klares Ja

(238/tp) Einerseits forciert Olivetti in ihrer Werbung immer noch die Textsysteme, andererseits kommen von Olivetti leistungsfähige PCs mit Textverarbeitungsprogrammen. Wir wollten es von Olivetti an einer Pressekonferenz genau wissen: Wie sieht der in Europa grösste Hersteller die Zukunft der reinen Textsysteme? Für Kleinbetriebe seien Textsysteme