# edv aspekte

3|89
Herausgegeben
von der Redaktion
rechentechnik
datenverarbeitung
DDR 5.00 M


## Modul-bausteine

## für rechner-gestützte Arbeitsplätze

# Inhalt


**Grundlagen**

01. Wurzelprogramm für eine Softwarelösung (KS) 2

02. Unterprogramm Schaltereinstellungen (KS) 2

03. Unterprogramm zur Festlegung von Anfangswerten für globale Variable (KS) 3

04. Beispiel zur Eröffnung der Arbeit (KS) 3

05. Unterprogramm zur Überprüfung eines persönlichen Codes eines jeden Nutzers (KS) 4

06. Unterprogramm zur Begrüßung eines Nutzers (KS) 4

07. Unterprogramm zur Vorbereitung der Arbeit (KS) 5

08. Unterprogramm zur Plausibilitätskontrolle eines eingegebenen Datums (KS) 7

09. Unterprogramm zum Abschluß der Arbeiten (KS) 7

10. Modul zur Reaktion auf Fehler (KS) 10

11. Funktionsmodul zur Anzeige von Hilfetexten (KS) 10

**Menüs**

12. Gestaltung eines aktualisierbaren Rahmens für die Menügestaltung (KS) 11

13. Modul zur Absicherung einer qualitätsgerechten Arbeit (KS) 12

14. Unterprogramm zur Gestaltung eines Aufgabengrundmenüs für 8-Bit-PC (KS) 13

15. Unterprogramm zur Gestaltung eines Aufgabengrundmenüs unter Berücksichtigung der Möglichkeiten der 16-Bit-PC (KS) 16

16. Konkordanz einer Zeichenkette unter Berücksichtigung des Kommas als Trennzeichen (KS) 21

**Dateiarbeit**

17. Übernahme einer Auswahl aus einem Directory in eine Datenbankdatei (KS) 22

18. Bereitstellen der Attribute einer vorzugebenden Datei (JS) 22

19. Setzen der Attribute einer vorzugebenden Datei (JS) 23

20. Demonstration der Nutzung der Moduln aus den Beispielen 18 und 19 (JS) 24

21. Erfragen einer bereits existierenden Datei und Bereitstellen ihrer Grundinformationen (KS) 24

22. Erfragen einer noch nicht existierenden Datei und Bereitstellen ihrer Grundinformationen (KS) 26

23. Erfragen einer noch nicht existierenden Datei und Bereitstellen ihrer Grundinformationen (KS) 27

24. Einrichten einer Diskette mit einer Diskettenkennzeichnung und Erfassen dieser Bezeichnung in einem Katalog (KS) 29

**Strukturbeschreibungsdateien**

25. Arbeit mit einer Strukturbeschreibungsdatei, um flexible Dateistrukturen zu nutzen (KS) 35

**Datenerfassung**

26. Ein universelles Datenerfassungs- und -pflegeprogramm (KS) 41

**Blättern in Dateien**

27. Blättern in einer Datenbankdatei und Auswählen eines Elementes mit Hilfe einer Balkentechnik (KS)

**Erzeugung von Befehlsfolgen für andere Systeme**

28. Unterprogramm zur Fortführung der Datenbereitstellung zur Berechnung eines Tilgungsplanes (KS) 58

29. Modul zur wahlweisen Zusammenstellung von Auswahlbedingungen (KS) 63

30. Die Funktion INKEY() als Beispiel für die Einbindung eines Assembler-Programms in ein Datenbankprogramm (siehe auch Beispiele 18 und 19) (JS) 64


**Die in diesem Heft zusammengestellten Beispiele sind ausschließlich mit dem Ziel ausgewählt, methodische Aspekte der Umsetzung von Teilaufgaben einer beliebigen Aufgabenklasse in informationstechnologische Strukturen und Lösungsbausteine einer komplexen Softwarelösung zu demonstrieren. Die Beispiele wurden unter Berücksichtigung international üblicher Beschreibungsformalismen in einer sprachlichen Form präsentiert, die eine geeignete Anpassung an die realen Bedingungen einer praktischen Aufgabenstellung, an den Einsatz eines konkreten Datenbanksystems unter Berücksichtigung der Anforderungen einer konkreten Hardware notwendig machen. Autor und Verlag können keine Garantie übernehmen, daß die nach der nutzereigenen Adaption der methodischen Beispiele gewonnenen Softwarelösungen beim Nutzer interpretierbare Ergebnisse liefern müssen. Die im Zusammenhang mit einigen methodischen Beispielen erwähnten Programmsysteme stehen gegenwärtig nicht für eine kommerzielle Nachnutzung zur Verfügung und sind auch zukünftig nicht dafür vorgesehen.**


**Die Auswahl der Beispiele sowie die Gestaltung des Schriftsatzes erfolgte mit einem Textprogramm auf PC 1715 durch Herrn Doz. Dr. sc. Kuno Schmidt (KS), Gülzower Str. 39, Berlin, 1144. Mitautor ist Jürgen Schenk (JS). Interessenten wenden sich bitte an Herrn Doz. Dr. sc. Kuno Schmidt.**

edv
aspekte

8. Jahrgang 3/1989

*Verlag Die Wirtschaft Berlin*
Am Friedrichshain 22 Berlin 1055
Verlagsdirektor: Dieter Grüneberg

*edv-aspekte*
Zeitschrift für spezielle Themen
der Informationsverarbeitung,
herausgegeben von der Redaktion
rechentechnik/datenverarbeitung,
1055 Berlin, Am Friedrichshain 22
Chefredakteur: Franz Loll 4 38 73 41
Redakteur: Claudia Schulz 4 38 73 16
Sekretariat: 4 38 72 33
Fernschreiber: 114 566
Titelgestaltung: Marlies Hawemann

*Redaktionsschluß: 31. 5. 1989*
Lizenz des Presseamtes beim Vorsitzenden
des Ministerrates der DDR Nr. 1529
*edv-aspekte*
Erscheinungsweise vierteljährlich zum Bezugspreis DDR 5,00 M je Heft
EDV-Artikel-Nr. 1331
Auslandspreise sind dem Zeitschriftenkatalog des Außenhandelsbetriebes
Buchexport zu entnehmen.
*Satz:* Verlag Die Wirtschaft, Berlin
*Druck:* (140) „Neues Deutschland", Berlin
*Anzeigenverwaltung:*
Berliner Verlag,
Karl-Liebknecht-Str. 29
Berlin 1056 Telefon: 2 70 33 02
*Anzeigenannahme:*
Berliner Verlag und Annahmestellen
in Berlin und in den Bezirken
Zur Zeit gültige Anzeigenpreisliste Nr. 12
*Im Ausland:*
INTERWERBUNG GmbH – Gesellschaft
für Werbung und Auslandsmessen der DDR,
Hermann-Duncker-Str. 89 Berlin 1157

*Bestellungen aus der DDR sind an den Postzeitungsvertrieb zu richten.*
Inkasso-Zeitraum: vierteljährlich

*Im Ausland:*
In den sozialistischen Ländern nur der zuständige Postzeitungsvertrieb. In allen anderen Staaten der örtliche Buch- und Zeitschriftenhandel. Bestellungen des Buch- und Zeitschriftenhandels sind zu richten an

BUCHEXPORT
Volkseigener Außenhandelsbetrieb der DDR,
DDR – Leninstr. 16, Leipzig 7010,
Postfach 160
oder an Verlag Die Wirtschaft,
DDR – Am Friedrichshain 22, Berlin 1055



# *Modulbausteine für rechnergestützte Arbeitsplätze*

Mit dem breiten Einsatz von Personalcomputern in der Volkswirtschaft der DDR wächst das Interesse an methodischen Hinweisen und Lösungsbeispielen für den Umgang mit Datenbanksystemen. Für die Personalcomputer der 8-Bit- und 16-Bit-Klasse stehen den Nutzern leistungsfähige Programmsysteme zur Verfügung, deren Befehlsvorrat kombatibel zu den internationational verbreitetsten Datenbanksystemen ist.
Diese Eigenschaften relationaler Datenbanksysteme führten zu einer breiten Anwendung vor allem bei Nutzern, die keine speziellen Informatik- und Programmierkenntnisse und -erfahrungen besitzen. Es ist nur allzu verständlich, daß aus diesem Kreis der Anwender heraus der Wunsch immer stärker wird, mit einer geeigneten methodisch-methodologischen Beispielsammlung ihre Bemühungen zu unterstützen, sich programmiermethodisch zu vervollkommnen und an Hand methodologisch ausgewählter Beispiele aus der Praxis das allgemeine Nutzungsniveau von Informationsbanken zu heben, die individuellen Programmierleistungen zu rationalisieren und zu intensivieren und schließlich die vielfältigen Möglichkeiten von Datenbanksystemen maximal auszuschöpfen.
Das vorliegende Heft möchte diesen vielfachen Wünschen der Anwender moderner PC-Technik Rechnung tragen und eine den oben beschriebenen Zielstellungen entsprechende Beispielsammlung präsentieren. Dabei wird von der Tatsache ausgegangen, daß in der Praxis sowohl Gerätetechnik der 8-Bit-Klasse (PC 1715, Bürocomputer etc.) als auch der 16-Bit-Klasse (A7100, A7150, EC1834 etc.) im Einsatz ist. Alle Beispiele lassen sich für beide Klassen von Geräten anpassen bei Gewährleistung einer im wesentlichen einheitlichen Nutzeroberfläche, gegebenenfalls aber mit geringfügigen Einschränkungen des Leistungsverhaltens.
Die Beispiele wurden thematisch so gewählt, daß mit ihrer Hilfe einheitliche Rahmenlösungen für (beliebige) rechnerunterstützte Arbeitsplätze effektiv gestaltet werden können. Dazu werden im ersten Abschnitt die grundlegenden Bausteine bereitgestellt, die eine einheitliche Rahmentechnologie für die informationellen Prozesse des Arbeitsplatzes gewährleisten. Der zweite Abschnitt stellt verschiedene Lösungsvarianten für Menüangebote und die dafür notwendigen Hilfsleistungen bereit. Der dritte Abschnitt behandelt einige häufig auftretende Grundbausteine der Dateiarbeit wie z. B. das Erfragen und Eröffnen einer Datei oder das Einrichten und Katalogisieren von Disketten. Am letzteren Beispiel wird darüber hinaus das prinzipielle automatische Zusammenwirken verschiedener Nutzerprogrammsysteme mit einem Datenbanksystem demonstriert. In den weiteren Abschnitten werden Lösungen vorgestellt, die komplexere Arbeitsabläufe beschreiben wie der Einsatz von Strukturbeschreibungsdateien für die Nutzung flexibler Dateistrukturen, die Erfassung und Pflege zweier voneinander abhängiger Datenbestände, das beliebige Blättern in einer Datenbankdatei mit Auswahl eines beliebigen Objektes und die Erzeugung von Befehlsfolgen für andere Nutzerprogrammsysteme innerhalb einer Datenbanklösung.

*Doz. Dr. sc. Kuno Schmidt*

## 1. Grundlagen für arbeitsplatzbezogene Programmlösungen

Die in diesem Abschnitt zusammengestellten Beispiele gehen davon aus, daß in der Praxis des Personalcomputereinsatzes zu einem überwiegenden Teil Softwarelösungen für rechnerunterstützte Arbeitsplätze gestaltet werden müssen. Das sind solche Lösungen, die komplexe Arbeitsprozesse eines konkreten Arbeitsplatzes unterstützen. Im Gegensatz zu den sogenannten "Insellösungen", die isolierte, in sich begrenzte Aufgabenstellungen zum Inhalt haben, handelt es sich bei der Software für rechnerunterstützte Arbeitsplätze um offene Programmsysteme mit integrierender Aufgabenstellung. Sie bedürfen einer strengen Architektur, wenn sie die an sie zu stellenden Forderungen erfüllen sollen. Einen wesentlichen Beitrag dazu liefert der praktisch realisierte modulare Aufbau des Systems. In den Beispielen dieses Abschnittes werden solche Moduln zusammengestellt, die bei der Eröffnung und beim Abschluß einer derartigen Softwarelösung für einen rechnerunterstützten Arbeitsplatz von Bedeutung sein können.

```
* Beispiel 1:
* Ein Wurzelprogramm für eine Softwarelösung eines
* rechnerunterstützten Arbeitsplatzes.
*
** MODULE (Bezeichnung des rechnerunterstützten
**                                   Arbeitsplatzes)
**
** DESCRIPTION
**
** Der Modul realisiert im Rahmen der Grundaufgaben
** für einen rechnerunterstützten Arbeitsplatz das
** Wurzelprogramm für die Softwarelösung des betref-
** fenden Arbeitsplatzes.

SET ECHO OFF
SET TALK OFF

* Rücksetzen aller Laufwerke

RESET

* Modulnfolge

DO schalter.PRG

* Siehe Beispiel 2

DO anfwert.PRG
* Siehe Beispiel 3

DO eroeffne.PRG

* Siehe Beispiel 4

DO nutzcode.PRG

* Siehe Beispiel 5

DO begruess.PRG

* Siehe Beispiel 6

DO arbvorbe.PRG

* Siehe Beispiel 7

DO grundmen.PRG

* Siehe Beispiel 12

RETURN

* Bemerkung: Für ein effektives Laufzeitverhalten
* dieses Wurzelprogramms ist es zweckmäßig, anstelle
* der DO-Anweisung die Befehlsfolgen selbst in den
* Quelltext des Wurzelprogramms einzufügen (mit Aus-
* nahme der letzten Befehlsdatei). In der gegebenen
** Darstellung wird allerdings die modulare Struktur
** des Wurzelprogramms deutlich.
```

```
* Beispiel 2:
* Unterprogramm Schaltereinstellungen

* MODULE schalter
*
* DESCRIPTION
**
** Der Modul realisiert im Rahmen der Grundaufgaben
** für einen rechnerunterstützten Arbeitsplatz das
** Einstellen von Grundstellungen der Programmschal-
** ter für die Belange des betreffenden Arbeits-
** platzes.

SET CARR ON
SET COLO OFF
SET CONF OFF
SET CONS ON
SET DELE ON
SET EJEC OFF
SET ESCA ON
SET EXAC ON
SET LINK OFF
SET RAW  OFF
* Die folgenden Schalter sind nur bei Programmlösun-
* gen für 16-Bit-Personalcomputer einsetzbar.
```

```
SET SAFE ON
SET BELL OFF
SET DEBU OFF
SET DELE OFF
SET SCOREBOARD OFF
SET CURSORMOVE OFF
SET DATE GERMAN
SET BELL OFF
SET FUNCTION 2 TO CHR(23)
SET FUNCTION 3 TO CHR(30)
SET FUNCTION 4 TO CHR(31)
SET FUNCTION 9 TO CHR(29)

RETURN
```

*** Beispiel 3:**
** Unterprogramm zur Festlegung von Anfangswerten für globale Variable.*
*
** MODULE anfwert
**
** **DESCRIPTION**
**
*** Der Modul realisiert im Rahmen der Grundaufgaben ** für einen rechnerunterstützten Arbeitsplatz das ** Einstellen von Anfangswerten für eine Reihe von ** (globalen) Programmvariablen.*

```
* Die folgenden Bildschirmsteuerzeichenfolgen gelten
* ausschließlich für den PC 1715.
*
* Cursor eine Position nach links und blinkende Dar-
* stellung ein. Beachten Sie bitte, daß ^ und B zwei
* verschiedene Zeichen sind. Es handelt sich hier
* also nicht um CTRL-B. Analoges gilt für die anderen
* Zeichenfolgen.

* Cursor eine Position nach links und blinkende Dar-
* stellung ein

STOR CHR(8)+CHR(27)+"^B" TO blk1

* Cursor eine Position nach links und inverse Dar-
* stellung ein

STOR CHR(8)+CHR(27)+"^P" TO inv1

 Cursor eine Position nach links und blinkende
* inverse Darstellung ein

STOR CHR(8)+CHR(27)+"^R" TO bli1

* blinkende Darstellung ein

STOR CHR(27)+"^B" TO blk

* inverse Darstellung ein

STOR CHR(27)+"^P" TO inv

* blinkende inverse Darstellung ein

STOR CHR(27)+"^R" TO bli

* normale Darstellung ein

STOR CHR(27)+"^§" TO norm

* Hinweis 1: Für andere Personalcomputersysteme ent-
* nehmen Sie bitte die entsprechenden Steuerzeichen-
* folgen dem betreffenden Systemhandbuch.
* Hinweis 2: Analoge Steuerzeichenfolgen können Sie
* für Ihren Drucker vereinbaren.

STOR (Zeitmaß) TO time

* (Zeitmaß) ist ein willkürlicher Zeitmaßstab inner-
* halb Ihrer Softwarelösung.

* Hinweis 3: Die Nutzung der eingeführten Variablen
* wird in den nachfolgenden Beispielen gezeigt.

RETURN
```

*** Beispiel 4:**
** Unterprogramm zur Eröffnung der Arbeit eines * rechnerunterstützten Arbeitsplatzes.*
*
** MODULE eroeffne
**
** **DESCRIPTION**
**
*** Der Modul realisiert im Rahmen der Grundaufgaben ** eines rechnerunterstützten Arbeitsplatzes das ** Eröffnungsbild einer Softwarelösung. Es wird das ** Titelbild gestaltet.*

```
* Eröffnungsbild

ERAS
?
?
? "        Rechnerunterstützter Arbeitsplatz"
?
?
?
? "    "+inv+" ( Bezeichnung des Arbeitsplatzes ) "+
                                                    norm
?
?
?
```

```
?
?
?
? "Copyright (c) 1986 Doz. Dr.-Ing. sc.nat. Kuno
                                                Schmidt"
? "Alle Rechte vorbehalten."

* determinierte Zeitschleife

STOR 0 TO j
DO WHIL j < 2*time
   STOR j+1 TO j
ENDD

RETURN
```

*** Beispiel 5:**
** Unterprogramm zur Überprüfung eines persönlichen*
** Codes eines jeden Nutzers*

```
*
** MODULE nucode
**
** DESCRIPTION
**
* Der Modul realisiert im Rahmen der Grundaufgaben
* eines rechnerunterstützten Arbeitsplatzes das
* Überprüfen des Berechtigungscodes eines zugelas-
* senen Nutzers.

SET COLO OFF
STORE '              ' TO namei
STORE '    ' TO codei
STORE (Kontrollcode)

* Der Kontrollcode dient der Prüfung, ob das vorlie-
* gende Programm original ist oder durch ein anderes
* gleichen Namens ersetzt wurde.

ERASE
@ 2,1 SAY "Geben Sie bitte Ihren"+INV+"Namen"+NORM+
                                                "an:"
SET INTENSITY ON
@ 2,37 GET namei PICTURE 'AAAAAAAAAAAAAAAAAAAA'
READ
SET INTENSITY OFF
@ 4,1 SAY "Geben Sie bitte Ihren"+INV+"Nutzercode"+
                                        NORM+"an:"
SET CONSOLE OFF
ACCEPT TO codei

* Der eingegebene persönliche Nutzungscode kann nach
* einem beliebigen Algorithmus in den im System
* verwalteten Nutzungscode umgewandelt werden. Der
* Algorithmus ist an dieser Stelle einzublenden.
SET EXACT ON
LOCATE FOR NAME = namei &bed
SET EXACT OFF
 RELEASE bed, codei1
IF EOF

   * Nutzer ist nicht angemeldet

   SET CONSOLE ON
   SET ESCAPE OFF
   ERASE
   ?
   ? "Leider hatte ich bisher nicht das Vergnügen,
                                        Sie begrüßen"
   ? "zu dürfen."
   ? "Entschuldigen Sie bitte! Aber aus diesem Grunde
                                        ist mir eine"
   ? "Zusammenarbeit mit Ihnen zur Zeit nicht
                                            möglich."
   ?
   ? "Wenden Sie sich bitte an Ihren Vorgesetzten."
   ?
   STORE 0 TO j
   DO WHILE j < 2*time
      STORE j+1 TO j
   ENDDO

   * Abbruch der Softwarelösung

   CLEAR
   ERASE
   QUIT
ENDIF

* Bediener ist zur Nutzung legitimiert

IF legitimati = 'A'
   STOR T TO c1
ELSE
   STOR F TO c1
ENDI
USE
SET CONSOLE ON

RETURN
```

*** Beispiel 6:**
** Unterprogramm zur Begrüßung eines Nutzers*

```
*
** MODULE begruess
**
** DESCRIPTION
**
** Der Modul realisiert im Rahmen der Grundaufgaben
** eines rechnerunterstützten Arbeitsplatzes das Be-
** grüßungsbild einer Softwarelösung
```

```
* Begrüßungsbild

ERAS
?
? "Ich begrüße Sie am rechnerunterstützten
                                        Arbeitsplatz"
?
? "   "+inv+" (( Bezeichnung des Arbeitsplatzes )) "+
                                                  norm
?
? "und hoffe auf eine erfolgreiche Zusammenarbeit"
? "bei der Lösung Ihrer Aufgaben mit Hilfe der ... ."
?
? "Falls Sie sich zunächst über mein Leistungs-"
? "angebot informieren wollen, empfehle ich Ihnen, "
? "mit der Grundaufgabe 9 zu beginnen."
?
? "Wenn sie nun die Taste"+inv+"'ET'"+norm+"bedienen"
? "würden, können wir mit der Arbeit beginnen."
?               "+inv+" V i e l  E r f o l g !"+norm
?

* determinierte Zeitschleife, um das Bild eine
* Mindestzeit stehen zu lassen

STOR 0 TO j
DO WHIL j < time
   STOR j+1 TO j
ENDD

* nutzerabhängige Zeitschleife

WAIT

RETU
```

**Beispiel 7:**

```
* Unterprogramm  zur  Vorbereitung  der  Arbeit  am
* rechnerunterstützten Arbeitsplatz
*
** MODULE arbvorbe
**
** DESCRIPTION
**
** Das Modul realisiert im  Rahmen der Grundaufgaben
** eines rechnerunterstützten Arbeitsplatzes notwen-
** dige Aktivitäten zur Absicherung einer qualitäts-
** gerechten Arbeit am Arbeitsplatz. Es werden:
** ein wählbares  Zeitmaß für das Anzeigen von Bild-
** schirmausschriften ausgewählt,
** das aktuelle  Datum erfaßt und auf Plausibilität
** geprüft,
** die zu benutzenden Laufwerke erfragt und
** die  Bezeichnungen  der eingelegten Disketten ge-
** prüft und festgehalten.

*  Zeitmaß für die determinierten Zeitschleifen
* erfassen
ERAS
?
? "Welche"+inv+"Lesegeschwindigkeit"+norm+
                                  "bevorzugen Sie heute?
?
? inv1+"(1)"+norm+"langsam"
? inv1+"(2)"+norm+"mittel --> Standardeinstellung"
  inv1+"(3)"+norm+"schnell"
?

* Hinweis: Für Erstnutzer und zur erneuten Einarbei-
* tung ist Fall 1 zweckmäßig.  Für die Routinenutzung
* sollten Sie Fall 3 bevorzugen.

? inv1+"Wählen Sie nach Ihren Bedürfnissen."+norm
?
WAIT TO fall
DO CASE
  CASE fall = '1'
      STOR 80 TO time

  CASE fall = '3'
      STOR 20 TO time

  OTHE
      STOR 40 TO time

ENDC

* Voraussetzung  schaffen für den  Wiedereintritt in
* das Programmsystem  nach einem Verlassen über QUIT
* TO ...  mit Rückkehrabsicht,

STOR 'variable.MEM' TO datei1
IF FILE(datei1)
   DELE FILE &datei1
ENDI

* Erfragen der zu nutzenden Laufwerke und der
* Diskettenbezeichnungen

STOR T TO entsch
DO WHIL entsch
   STOR F TO entsch

   * Standardmäßig  wird angenommen, daß  alle Daten-
   * bankdateien auf  Laufwerk 'A' liegen,  alle Be-
   * fehlsdateien  jedoch auf  Laufwerk 'B'.  Die
   * Standardeinstellung  ist für den Nutzer  jedoch
   * nicht bindend.
   STOR "A" TO lw1
   STOR "B" TO lw
```

```
STOR "              " TO DISKA
STOR "              " TO DISKB
ERAS
§ 2,1 SAY "Geben Sie bitte das für die"+inv+
 "(Datenbankdateien)"+norm+"von Ihnen zu nutzende"
§ $+1,1 SAY "Diskettenlaufwerk vor: "+inv+
                          "(A ö B ö C ö D)"+norm
SET INTE ON

* Der Schalter bewirkt (bei entsprechender Instal-
* lation  Ihres  Datenbanksystems eine  Inversdar-
* stellung des Eingabemaskenbereiches.

§ $+1,1 GET lw1 PICT "A"
READ
SET INTE OFF
ENDI

STOR T TO entsch
DO WHIL entsch

   * Bzgl. Diskettenbezeichnung siehe auch
   * Abschnitt 3

   § $+1,1 SAY "Geben Sie bitte die"+inv+      .
           " Bezeichnung "+norm+"der in"+inv+" "+
          lw1+" "+norm+"eingelegten Diskette an:"
   SET INTE ON
   § $+1,1 GET diska PICT "XXXXXXXXXXXX"
   READ
   SET INTE OFF
   STOR !(lw1) TO lw1
   STOR !(diska) TO diska
   STOR lw1+":"+diska TO datei1
   STOR .NOT. FILE(datei1) TO entsch
   IF entsch
      ERAS
      § 2,1 SAY bli1+"A c h t u n g!"+norm+" Sie
          haben eine falsche Diskette eingelegt!"
      § $+1,1 SAY "Korrigieren Sie Ihre
           vorgegebene Diskettenbezeichnung oder"
      § $+1,1 SAY "wechseln Sie Ihre Diskette in
                Laufwerk"+inv+" "+lw1+" "+norm+"."
      § $+1,1 SAY "Drücken Sie danach die Taste"+
                                  +inv+" (ET) "+norm
      DISP FILE ON &lw1 LIKE *.*
      WAIT
      ERAS
      RESE
   ENDI
ENDD
ERAS
§ 2,1 SAY "Geben Sie bitte das für die"+inv+
         " Kommandodateien der Karteiarbeit"+norm
§ $+1,1 SAY "von Ihnen zu nutzende Disketten
        laufwerk vor: "+inv+"(A ö B ö C ö D)"+norm
SET INTE ON
§ $+1,1 GET lw PICT "A"
READ
SET INTE OFF
STOR T TO entsch
DO WHIL entsch
   STOR F TO entsch
   § $+1,1 SAY "Geben Sie bitte die"+inv+
            " Bezeichnung "+norm+"der eingelegten
                                   Diskette an:"
   SET INTE ON
   § $+1,1 GET diskb PICT "XXXXXXXXXXXX"
   READ
   SET INTE OFF
   STOR !(lw) TO lw
   STOR !(diskb) TO diskb
   STOR lw+":"+diskb TO datei1
   STOR .NOT. FILE(datei1) TO entsch
   IF entsch
      ERAS
      § 2,1 SAY bli1+"A c h t u n g!"+norm+
         " Sie haben eine falsche Bezeichnung der"
      § $+1,1 SAY "Kommandodateidiskette
                                      angegeben!"
      § $+1,1 SAY "Geben Sie die richtige
          Bezeichnung erneut vor, nachdem Sie die"
      § $+1,1 SAY "Taste"+inv+" (ET) "+norm+
                                "gedrückt haben."
      DISP FILE ON &lw LIKE *.*
      WAIT
      ERAS
   ENDI
ENDD
IF lw = lw1
   ERAS
   § 2,1 SAY bli1+"A c h t u n g! "+norm+
         " Sie haben zweimal das gleiche Laufwerk
                                      angegeben!"
   § $+1,1 SAY "Wiederholen Sie Ihre Eingaben."
   STOR 0 TO j
   DO WHIL j < time
      STOR j+1 TO j
   ENDD
   STOR T TO entsch
ENDI
ENDD

* Erfassen des aktuellen Datums

STOR "  .  .  " TO datum
ERAS
§ 2,1 SAY "Geben Sie mir bitte das"+inv+
          " heutige Datum "+norm+"in der Form"+inv+
                                " TT.MM.JJ "+norm
§ $+1,1 SAY "vor."
SET INTE ON
```

```
§ $+1,1 GET datum PICT "##.##.##"
READ
SET INTE OFF

* Das Modul datumkon führt eine Plausibilitätskon-
* trolle des eingegebenen Datumswertes vor. Damit
* wird gesichert, daß stets ein Datum für die Arbeit
* am rechnerunterstützten Arbeitsplatz zur Verfügung
* steht (siehe Beispiel 8).

STOR lw+":datumkon.CMD" TO datei1
DO &datei1
SET DATE TO &datum

* Bereitstellen einer vollständigen Datumsangabe und
* in anglistischer Darstellung (für Sortier- und
* Rechercnezwecke)

STOR $(datum,1,6)+"19";$(datum,7,2) TO datumv
STOR "19"+$(datum,7,2)+$(datum,4,2)+$(datum,1,2) TO
                                              datuma

* Löschen der lokalen Variablen

RELE entsch, fall, j, datei1

RETURN
```

*** Beispiel 8:**
** Unterprogramm zur Plausibilitätskontrolle eines*
** eingegebenen Datums*

```
*
** MODULE datumkon
**
** DESCRIPTION
**
** Der Modul realisiert die Plausibilitätskontrolle
** eines eingegebenen Datums

STOR T TO error1
DO WHIL error1
   STOR VAL($(datum,1,2)) TO tag
   STOR VAL($(datum,7,4)) TO jahr
   DO CASE
      CASE monat = 1 .OR. monat = 3 .OR. monat = 5
                  .OR. monat = 7 .OR. monat = 8 .OR.
                           monat = 10 .OR. monat = 12
         IF tag >= 1 .AND. tag <= 31
            STOR F TO error1
         ELSE
            STOR "Tage" TO dat
         ENDI

      CASE monat = 4 .OR. monat = 6 .OR. monat = 9
                                    .OR. monat = 11
         IF tag >= 1 .AND. tag <= 30
            STOR F TO error1
         ELSE
            STOR "Tage" TO dat
         ENDI

      CASE monat = 2
         IF tag >= 1 .AND. tag <= 28
            STOR F TO error1
         ELSE
            STOR "Tage" TO dat
         ENDI

      OTHE
         STOR "Monats" TO dat
   ENDC

   IF error1
      STOR "  .  .  " TO datum
      ERAS
      § 2,1 SAY bli1+"A c h t u n g!"+norm+
                        " Fehlerhafte "+dat+"angabe."
      § $+2,1 SAY inv1+"Bitte Datumsangabe erneut
                                        vorgeben"+norm
      SET INTE ON
      § $+1,1 GET datum PICT '##.##.##'
      READ
      SET INTE OFF
   ENDI
ENDD
RELE monat, tag, jahr, dat, error1

RETU
```

*** Beispiel 9:**
** Unterprogramm zum Abschluß der Arbeiten am rech-*
** nergestützten Arbeitsplatz (Bezeichnung des Ar-*
** beitsplatzes)*

```
*
** MODULE aktioend
**
** DESCRIPTION
** Das Modul realisiert den Abschluß der Arbeiten
** am rechnergestützten Arbeitsplatz (Bezeichnung
** des Arbeitsplatzes)

* Abfrage, ob dieser Modul wirklich aktiviert werden
* soll

ERAS
?
? "Wollen Sie die Arbeit am rechnergestützten
                                    Arbeitsplatz:"
? inv1+"' ( Bezeichnung des rechnergestützten
                              Arbeitsplatzes )'"+norm
```

```
? "beenden?"+inv+"J/N"+norm
?
WAIT TO antwort
IF !(antwort) = 'N'
   RETU
ENDI

* Anlegen von Sicherungsdateien

ERAS
?
? "Wollen Sie eine"+inv+"Sicherungsdatei"+norm+
                          "für Ihre Daten anlegen?"
? inv1+"J/N"+norm
?
WAIT TO antwort
IF !(antwort) = 'J'

   * Grundmenü zur Arbeit mit Sicherungsdateien

   STOR T TO whol
   DO WHIL whol
      ERAS
      ?
      ? "Welche"+inv+"geänderte Datei"+norm+
                                "wollen Sie sichern:"
   ?
   ? inv1+"(1)"+norm+" Datei der Stammdaten des
         rechnergestützten Arbeitsplatzes ... ."
    ? inv1+"(2)"+norm+" Datei der
              b e r e c h n e t e n  Daten ... ."
    ? inv1+"(3)"+norm+" Datei der ... ."

    * Hier weitere zu sichernde Dateien einfügen.

    ? inv1+"(0)"+norm+" Keine Datei sichern."
    ?
    ? inv1+"Wählen Sie Ihre zu sichernde Datei."+
                                              norm
    ?
    WAIT TO fall

    * Die einzelnen zu sichernden Dateien

    DO CASE

       CASE fall = '1'
          STOR '(Dateibezeichnung)' TO datnam
          STOR " Stammdaten des rechnergestützten
                      Arbeitsplatzes ... ." TO text
       CASE fall = '2'
          STOR '(Dateibezeichnung)' TO datnam
          STOR " b e r e c h n e t e  Daten ...."
                                           TO text
       CASE fall = '3'
          STOR '(Dateibezeichnung)' TO datnam
          STOR " ... ." TO text

       * Hier weitere Fälle einfügen

       CASE fall = '0'
          STOR F TO whol
          LOOP

       OTHE
          ERAS
          ?
          ? bli1+"A c h t u n g!"+norm+" Sie haben
                  eine nicht existierende Aufgabe"
          ? "ausgewählt."
          ? "Wählen Sie erneut eine der
               aufgeführten Dateien, nachdem Sie"
          ? "die Taste"+inv+"ET"+norm+"gedrückt
                                              haben."
          ?
          WAIT
          LOOP
    ENDC

    * Frage nach Laufwerk und Dateiname

     STOR 'A' TO lw1
     STOR T TO entsch
     DO WHIL entsch
        ERAS
        § 2,1 SAY "Bitte Bezeichnung der "+inv+
                                   "Datei der "+norm
        § $+1,1 SAY inv1+text+norm+"vorgeben."
        § $+3,1 SAY "Von welchem Laufwerk?"
        SET INTE ON
        § $,26 SAY " " GET lw1 PICT 'X'
        § $-2,1 SAY " " GET datnam PICT 'XXXXXXXX'
        READ
        SET INTE OFF
        STOR !(datnam) TO datnam
        STOR !(lw1) TO lw1

        * Plausibilitätstest für Laufwerk

        IF lw1 < 'A' .OR. lw1 > 'D'
           ERAS
           ?
           ? bli1+"A c h t u n g!"+norm+" Sie haben
                    ein falsches Laufwerk angegeben."
           ? "Wählen Sie erneut das richtige
                          Laufwerk, nachdem Sie die"
           ? "Taste"+inv+"ET"+norm+"gedrückt haben."
           ?
           WAIT
           LOOP
        ENDI
        IF lw1 <> lw
```

```
      STOR lwl+":"+TRIM(datnam)+".DBF" TO datei
      STOR .NOT. FILE(datei) TO entsch

      * gewünschte Datei nicht vorhanden

      IF entsch
         ERAS
         ?
         ? blil+"A c h t u n g!"+norm+
              " Sie haben eine falsche Diskette
                                    eingelegt"
         ? "oder den Dateinamen falsch
                                    angegeben."
         ?
         DISP FILES ON &lwl LIKE *.*
         STOR 0 TO j
         DO WHIL j < time
            STOR j+1 TO j
         ENDD
         ?
         ? "Korrigieren Sie den Dateinamen oder
                        legen Sie die richtige"
         ? "Diskette ein."
         ?
         USE
         WAIT
         RESE &lwl
      ENDI
   ELSE
      USE
      ERAS
      ?
      ? blil+"A c h t u n g!"+norm+" Sie haben
            ein falsches Laufwerk angegeben."
      ? "Wählen Sie erneut das richtige
                     Laufwerk, nachdem Sie die"
      ? "Taste"+inv+"ET"+norm+"gedrückt haben."
      ?
      WAIT
   ENDI
ENDD

* gewünschte Datei wird gesichert

USE &datei
STOR lwl+":"+datnam+".BAK" TO datei1
IF FILE(datei1)
   DELE FILE &datei1
ENDI
COPY TO &datei1
ERAS
?
? "Ich habe für die Datei der "+text
? "eine"+inv+"Sicherungskopie"+norm+"angelegt."
?
STOR 0 TO j
      DO WHIL j < time
         STOR j+1 TO j
      ENDD
   ENDD

   * Ein abschließender Scherz zur Entspannung

   ERAS
   ?
   ? "Geben Sie mir rasch noch etwas ein:"
   ?
ELSE
   ERAS
   ?
   ? invl+"Auch G u t:"+norm+" Geben Sie mir rasch
                                    noch etwas ein:"
   ?
ENDI
WAIT TO zahl
DO CASE

   CASE (zahl < 'z' .AND. zahl > 'a') .OR.
                        (zahl < 'Z' .AND. zahl > 'A')

      § 10,1 SAY "Das war der Buchstabe:"
      § 10,27 SAY inv+zahl+norm USIN 'XXXXXXX'

   CASE zahl < '9' .AND. zahl > '0'

      § 10,1 SAY "Das war die Ziffer:"
      § 10,25 SAY inv+zahl+norm USIN 'XXXXXXX'

   OTHE

      § 10,1 SAY "Das war das Sonderzeichen:"
      § 10,31 SAY inv+zahl+norm USIN 'XXXXXXX'

ENDC
STOR 0 TO j
DO WHIL j < time
   STOR j+1 TO j
ENDD

* Verabschiedung

§ $+1,1 SAY invl+"D a n k e!"+norm
§ $+1,1 SAY "Wenn Sie zufrieden waren, sagen Sie es
                                          weiter!"
§ $+1,1 SAY "Andernfalls teilen Sie Ihre Wünsche dem
                                       ... mit!"
§ $+2,22 SAY inv+"T S C H Ü E Ü E S Z!"+norm
STOR 0 TO j
DO WHIL j < 3*time
   STOR j+1 TO j
ENDD
```

```
* Ein abschließender Service

ERAS
@ 2,1 SAY "Noch ein kleiner Service.
@ $+1,1 SAY "Möchten Sie ein weiteres Programmsystem
                              nutzen?"+inv+"J/N"+norm
@ $+1,1 SAY " "
WAIT TO antwort
IF !(antwort) = 'N'
   ERAS
   CLEAR
   QUIT
ELSE
   @ $+1,1 SAY "Geben Sie den"+inv+"Namen Ihres
                                 Programms"+norm+"an:"
   @ $+1,1 SAY " "
   ACCE TO datnam
   @ $+1,1 SAY "Von welchem"+inv+"Laufwerk"+norm+
                  "soll das Programm geladen werden?"
   @ $+1,1 SAY inv1+"(B ö C ö D)"+norm
   @ $+1,1 SAY " "
   WAIT TO lw
   STOR !(lw)+":" TO datei
   STOR !(TRIM(datnam)) TO datei2
   STOR lw1+":PIP" TO datei1
   STOR "; Bitte"+inv+"gewünschte Diskette"+norm+
        "in das angegebene Laufwerk"+norm+"einlegen,"+
                     inv+"ET"+norm+"drücken" TO text
   ERAS
   QUIT TO '&text.', '&datei1.', '&datei.', '&datei2'
ENDI
```

*** Beispiel 10:**

** Modul zur Reaktion auf Fehler*

```
*
** MODULE ( Bezeichnung Fehlerreaktionsmodul )
**
```

**** DESCRIPTION**

```
**
```

*** Das Modul gestaltet die Reaktion auf Systemfehler*
*** und sorgt für einen ordnungsgemäßen Abschluß der*
*** gesamten Softwarelösung bei einem instabilen Zu-*
*** stand. Die möglichen Reaktionen können innerhalb*
*** dieses Moduls arbeitsplatzabhängig gestaltet wer-*
*** den.*
*** Die Nachrichten werden im Mentorfenster der Menü-*
*** rahmenorganisation angezeigt.*

```
PARAMETERS m_wnummer, m_ende

PRIVATE m_antwort, m_i, m_is

m_ende=.T.
m_antwort=" "
m_wnummer=CURWIN()
WSELECT m_wnummer+2
WSAVE
WDISPLAY m_wnummer+2
SET BELL ON
SET INTENSITY ON
@ 0,0 SAY CHR(7)
@ 0,0 SAY "Fehler "+LTRIM(STR(ERROR(),3))+":"+
                                    LEFT(MESSAGE(),60)
@ 1,0 SAY SUBSTR(MESSAGE(),61,100)+". Abschluß mit
                ESC-Taste!" GET m_antwort PICTURE "X"
@ 1,0 SAY CHR(7)
READ
SET INTENSITY OFF
SET BELL OFF
WDISPLAY m_wnummer+2
WRESTOR
WSELECT m_wnummer

* Abschließen aller eröffneten Dateien

CLOSE ALL

RETURN
```

*** Beispiel 11:**

** Funktionsmodul zur Anzeige von Hilfetexten.*

```
*
** MODULE hilfefun
**
```

**** DESCRIPTION**

```
**
```

*** Der Modul aktiviert das Hilfesystem mit der Datei*
*** (Bezeichnung der Hilfetextdatenbankdatei).DBF der*
*** Hilfetexte Tx_0, Tx_1, Tx_2 und Tx_3 im Nut-*
*** zungsbereich 10 für die Bedürfnisse des rechner-*
*** unterstützten Arbeitsplatzes ( Bezeichnung des*
*** rechnerunterstützten Arbeitsplatzes ).*
*** Als Hilfefenster wurde Fenster 1 (w_inhalt_1)*
*** Modul arbvorb2 (siehe Beispiel 13) festgelegt:*
*** SET USERHELP TO WINDOW w_inhalt_1.*
*** Das Fenster umfaßt 4 Zeilen zu je 78 Zeichen.*

```
FUNCTION hilfefun

PARAMETERS m_prognam, m_get_nr, m_nothing

PRIVATE m_merkmal, m_select

m_select=STR(SELECT(), 2)
m_merkmal=m_prognam+SUBSTR(STR(m_get_nr+100, 3),2, 2)
SELECT 10
GO TOP
SEEK m_merkmal
IF FOUND()
   ? tx_0
```

```
   ? tx_1
   ? tx_2
   ? tx_3
   ?? ""
ELSE
   SET BELL ON
   ? CHR(7)
   ?? CENTER("Selbsterklärend, deshalb keine Hilfe
                                    vorgesehen!",78)
   SET BELL OFF
ENDIF
m_nothing=""
SELECT &m_select

RETURN m_nothing


*****************************************************
```

```
-----------------------------------------------------
            aspekte-vorschau  4/89
-----------------------------------------------------
```

Logisches Programmieren-
Softwarewerkzeuge für P 8000
---

Mit dem P 8000 compact vom Kombinat Elektro-Apparate-Werke "Friedrich Ebert" Berlin steht der DDR-Wirtschaft ein weiterer leistungsfähiger MS-DOS-fähiger 16-Bit-Computer mit großer Einsatzbreite zur Verfügung.
Für die zunehmende Zahl der Nutzer stellt edv-aspekte 4/89 Softwarewerkzeuge für diesen Rechner vor. Die Auswahl der Beiträge gibt P-8000-Neulingen wie auch erfahrenen Nutzern Lernhilfen und Impulse für neue Einsatzmöglichkeiten.
Den Schwerpunkt bildet die komplette Sprachbeschreibung von HU-PROLOG, einschließlich einer Beispielsammlung zum Programmieren mit PROLOG.
Im weiteren werden folgende Themen behandelt:
- Portierung von PASCAL-Software mit PCC
- Modula 2 für den P 8000
- Der Modula-2-Compiler im Betriebssystem VMX auf ESER-Rechnern
- PROGRESS - Eine Spezifikationssprache für PROgrammierte GRaph-Ersetzungs-SySteme
- Ein moderner Editor für UNIX-kompatible Betriebssysteme
- SPIDER - Ein System zur Datenübertragung
- Erfahrungen bei der Vernetzung von mehreren P 8000 unter WEGA mittels UUCP

```
*****************************************************
```

## 2. Menügestaltung und Menüauswertung

In den folgenden methodischen Beispielen werden Menüangebote behandelt, verschiedene Möglichkeiten der Umsetzung in geeignete Programmsteuerfolgen dargestellt und die Voraussetzungen für ihre qualitätsgerechte Nutzung entwickelt.

*** Beispiel 12:**
** Gestaltung eines aktualisierbaren Rahmens für die*
** Menügestaltung*

```
*
** MODULE menurahm
**

**
```

*** Das Modul realisiert einen aktualisierbaren Rahmen für die Gestaltung des Mensch-Maschine-Dialogs innerhalb einer Softwarelösung für den rechnerunterstützten Arbeitsplatz ( Bezeichnung des rechnerunterstützten Arbeitsplatzes ). Es werden vier Bereiche vorgesehen (von oben nach unten):*
*** 1. Titel- (Status-)fenster: Umfang: 1 Zeile*
*** Oberschrift Aufgabenabschnitt, aktuelles Datum*
*** 2. Mitteilungsfenster: Umfang: 4 Zeilen*
*** Allgemeine Informationen, Hilfetexte, inhaltliche Anzeigen zur Auswahl, etc.*
*** 3. Arbeitsfenster: Umfang: 12 Zeilen*
*** Menüangebote, Dialogführung, Anzeige von Arbeitsergebnissen mit Möglichkeiten des beliebigen Blätterns, Gestaltung von inhaltlichen Entscheidungs- und Bewertungssituationen, etc.*
*** 4. Mentorfenster: Umfang: 2 Zeilen*
*** Führung des Nutzers, organisatorische Hinweise ,Ablaufentscheidungen durch den Nutzer, Einflußmöglichkeiten des Nutzers auf den Arbeitsablauf, etc.*

```
 PARAMETERS m_text, m_farbe_1, m_farbe_2, m_ende

PRIVATE m_colum, m_i, m_farbe_11, m_farbe_12,
                      m_farbe_13, m_color, m_text5


m_ende=.F.
m_color=ISCOLOR
WSELECT 0
IF m_color

   * (Die Ausführung des Hilfsprogramms siehe
   * Beispiel 16)

  DO konkorda WITH m_farbe_1, m_farbe_11,
                   m_farbe_12, m_farbe_13, m_ende
```

```
   IF m_ende
      RETURN
   ENDIF
   SET COLOR TO &m_farbe_11, &m_farbe_12, m_farbe_13
ENDIF
m_text5=CHR(213)+REPLICATE(CHR(205),78)+CHR(184)
§ 0,0 CLEAR
§ ROW(),0 SAY m_text5
m_text5=CHR(179)+REPLICATE(" ",78)+CHR(179)
§ ROW()+1,0 SAY m_text5
IF m_color
   DO konkorda WITH m_farbe_2, m_farbe_11,
                    m_farbe_12, m_farbe_13, m_ende
   IF m_ende
     RETURN
   ENDIF
   SET COLOR TO &m_farbe_11, &m_farbe_12, &m_farbe_13
ENDIF
m_colum=5
§ ROW(),m_colum SAY  m_text
m_colum=68
§ ROW(),m_colum SAY date()
IF m_color
  DO konkorda WITH m_farbe_1, m_farbe_11,
                   m_farbe_12, m_farbe_13, m_ende
  IF m_ende
     RETURN
   ENDIF
   SET COLOR TO &m_farbe_11, &m_farbe_12, &m_farbe_13
ENDIF
m_text5=CHR(198)+REPLICATE(CHR(205),78)+CHR(181)
§ ROW()+1,0 SAY m_text5
m_text5=CHR(179)+REPLICATE(" ",78)+CHR(179)
FOR m_i=1 TO 4
   § ROW()+1,0 SAY m_text5
NEXT
m_text5=CHR(198)+REPLICATE(CHR(205),78)+CHR(181)
§ ROW()+1,0 SAY m_text5
m_text5=CHR(179)+REPLICATE(" ",78)+CHR(179)
FOR m_i=1 TO 12
   § ROW()+1,0 SAY m_text5
NEXT
m_text5=CHR(198)+REPLICATE(CHR(205),78)+CHR(181)
§ ROW()+1,0 SAY m_text5
m_text5=CHR(179)+REPLICATE(" ",78)+CHR(179)
§ ROW()+1,0 SAY m_text5
§ ROW()+1,0 SAY m_text5
m_text5=CHR(212)+REPLICATE(CHR(205),78)+CHR(190)
§ ROW()+1,0 SAY m_text5

RETURN
```

*** Beispiel 13:**

** Modul zur Absicherung einer qualitätsgerechten*
** Arbeit an rechnerunterstützten Arbeitsplatz (*
** Bezeichnung des rechnerunterstützten Arbeitsplatzes ) unter Berücksichtigung der Menügestaltung*
** (Ergänzung der Aufgaben des Moduls aus dem Beispiel 7 unter Berücksichtigung der Möglichkeiten*
** der 16-Bit-Personalcomputer)*
***
*** MODULE arbvorb2*
****
*** **DESCRIPTION***
****
*** Das Modul aktiviert den aktualisierbaren Rahmen*
*** für die Gestaltung des Mensch-Maschine-Dialogs*
*** innerhalb einer Softwarelösung für den rechnergestützten Arbeitsplatz ( Bezeichnung des rechnergestützten Arbeitsplatzes ), vereinbart die Fenster für die im vorherigen Beispiel fixierten*
*** verschiedenen Arbeitsbereiche, aktiviert die Reaktion auf interne Systemfehlerzustände, setzt*
*** Schalterstellungen und Funktionen, deklariert und eröffnet die benötigten Fenster und iniziiert die*
*** Hilfefunktion für den rechnerunterstützten Arbeitsplatz (Bezeichnung des rechnerunterstützten Arbeitsplatzes ).*

```
PARAMETERS m_farbe_1, m_farbe_2, m_farbe_3,
           m_farbe_4, m_farbe_5, m_farbe_6, m_ende

PUBLIC m_nothing

PRIVATE m_text, m_farbe_11, m_farbe_12, m_farbe_13,
        m_prognam, m_select, m_wnummer, m_color

* Fehlerbehandlung, Abbruchmöglichkeit
* (Ausführung des Fehlerreaktionsmodul siehe
* Beispiel 10)
* Beachten Sie bitte,  daß Sie die folgende Programm
* zeile erst nach den Testarbeiten aktivieren sollten

ON ERROR DO ( Bezeichnung Fehlerreaktionsmodul ).PRG
                              WITH m_wnummer, m_ende

* Anfangswerte setzen

m_ende=.F.
m_color=ISCOLOR
m_prognam="( Bezeichnung des Moduls zur Realisierung
       des Aufgabenabschnittes der Softwarelösung )"
m_nothing=" "
m_wnummer=0
m_text="( Bezeichnung des Aufgabenabschnittes der
                                    Softwarelösung )"

* Menürahmen gestalten (siehe Beispiel 12)

DO menurahm WITH m_text, m_farbe_1, m_farbe_2, m_ende
IF m_ende
```

```
   RETURN
ENDIF

* benötigte Window erklären und eröffnen

IF m_color
   DO konkorda WITH m_farbe_3, m_farbe_11,
                    m_farbe_12, m_farbe_13, m_ende
   IF m_ende
      RETURN
   ENDIF
   WSET WINDOW w_inhalt_1 TO 3, 2, 6, 76
         COLOR &m_farbe_11, &m_farbe_12, &m_farbe_13
   DO konkorda WITH m_farbe_4, m_farbe_11,
                    m_farbe_12, m_farbe_13, m_ende
   IF m_ende
      RETURN
   ENDIF
   WSET WINDOW w_inhalt_2 TO 8, 2, 19, 76
         COLOR &m_farbe_11, &m_farbe_12, &m_farbe_13
   DO konkorda WITH m_farbe_5, m_farbe_11,
                    m_farbe_12, m_farbe_13, m_ende
   IF m_ende
      RETURN
   ENDIF
   WSET WINDOW w_inhalt_3 TO 21, 2, 22, 76
         COLOR &m_farbe_11, &m_farbe_12, &m_farbe_13
ELSE
   WSET WINDOW w_inhalt_1 TO 3, 2, 6, 76
   WSET WINDOW w_inhalt_2 TO 8, 2, 19, 76
   WSET WINDOW w_inhalt_3 TO 21, 2, 22, 76
ENDIF

DO WHILE WACTIVE()
   m_wnummer=m_wnummer+1
   WSELECT m_wnummer
ENDDO

WSET FRAME OFF
WUSE w_inhalt_1
WSELECT m_wnummer+2
WUSE w_inhalt_3
WSELECT m_wnummer+1
WUSE w_inhalt_2

* Schalterstellungen und Funktionen setzen

SET SCOREBOARD OFF
SET CURSORMOVE OFF
SET DATE GERMAN
SET BELL OFF
SET FUNCTION 2 TO CHR(23)
SET FUNCTION 3 TO CHR(30)
SET FUNCTION 4 TO CHR(31)
SET FUNCTION 9 TO CHR(29)
SET FUNCTION 10 TO CHR(27)

* Hilfeunterstützung setzen
* Die Datei der Hilfetexte hat folgende Struktur:
* pr_name C 10
* get_nr  N  3 0
* tx_0    C 78
* tx_1    C 78
* tx_2    C 78
* tx_3    C 78

* Für umfangreichere Hilfetexte sind weitere Felder
* vorzusehen und die Hilfefensterdefinition entspre
* chend zu erweitern.

SET USERHELP ON
WSELECT m_wnummer
SET USERHELP TO WINDOW w_inhalt_1
m_select=STR(SELECT())
SELECT 10
IF FILE("( Bezeichnung Datei Hilfetexte ).NDX")
   USE ( Bezeichnung Datei Hilfetexte ).DBF INDEX
               ( Bezeichnung Datei Hilfetexte ).NDX
ELSE
   USE ( Bezeichnung Datei Hilfetexte ).DBF
   INDEX ON pr_name+SUBS(STR(VAL(get_nr)+100,3),2,2)
            TO ( Bezeichnung Datei Hilfetexte ).NDX
   USE ( Bezeichnung Datei Hilfetexte ).DBF INDEX
               ( Bezeichnung Datei Hilfetexte ).NDX
ENDIF
GO TOP

RETURN
```

*** Beispiel 14:**
** Unterprogramm zur Gestaltung eines Aufgabengrund-*
** menüs (für 8-Bit_Personalcomputer)*

```
**
** MODULE grundmen
**
```

**** DESCRIPTION**

```
**
** Das Modul realisiert das Grundmenü für das Aufga-
** benangebot des rechnergestützten Arbeitsplatzes
** (Bezeichnung des rechnergestützten Arbeitsplat-
** zes)

* Vorbereitung des Moduls als Kopfprogramm einer
* Modulfolge

IF FILE("variable.MEM")
   REST FROM variable.MEM
   SET DATE TO &datum
   DELE FILE variable.MEM
ENDI

* Menüangebot gestalten
```

```
STOR T TO entsch
DO WHIL entsch
   ERAS
   ?
   ?          "+inv+" *** ( Bezeichnung des rechnerge
   stützten Arbeitsplatz ) - Grundaufgaben *** "+norm
   ?                        "+inv+" Uebersicht "+norm
   ? inv1+"(1)"+norm+" Erfassen von Stammdaten "
   ? inv1+"(2)"+norm+" Erfassen von Bewegungsdaten "
   ? inv1+"(3)"+norm+" Auswahl von Daten zur
                                        Bearbeitung "
   ? inv1+"(4)"+norm+" Obernahme von berechneten
                        Daten in die Datenbank ... "
   ? inv1+"(5)"+norm+" Kontrolle des Bearbeitungs
                     standes von Aufgabenstellungen "
   ? inv1+"(6)"+norm+" Aktualisierung der Daten der
                                      Datenbank ... "
   ? inv1+"(7)"+norm+" Analysen und Berichte "
   ? inv1+"(8)"+norm+" Dienst- und Havarieaufgaben
            zum rechnergestützten Arbeitsplatz ... "
   ? inv1+"(9)"+norm+" Informationsdienste zum
                    rechnergestützten Arbeitsplatz"
   ?       und zur zugehörigen Softwarelösung"

   * An dieser Stelle können weitere Aufgaben
   * eingefügt werden

   ? inv1+"(0)"+norm+" Abschluß der Arbeiten am
                rechnergestützten Arbeitsplatz ... "
   ?
   ? inv1+"Wählen Sie Ihre Aufgabe:"+norm
   ?
   WAIT TO auswahl

   * Erfragen des Laufwerkes für die Befehlsdatei
   STOR T TO entsch
   DO WHIL entsch
      ?
      ? "Von welchem Laufwerk soll das Programm
                                      geladen werden? "
      ? inv1+"(B ó C ó D)"+norm
      ?
      WAIT TO lw
      STOR !(lw) TO lw

      * Plausibilitätstest für Laufwerk

      IF lw < 'A' .OR. lw > 'D'
         ERAS
         ?
         ? bli1+"A c h t u n g!"+norm+" Sie haben ein
                        falsches Laufwerk angegeben."
         ? "Wählen Sie erneut das richtige Laufwerk,
                                    nachdem Sie die"
         ? "Taste"+inv+"ET"+norm+"gedrückt haben."
         ?
         WAIT
      ELSE
         STOR F TO entsch
      ENDI
   ENDD
   STOR T TO entsch

   * Verzweigung in die verschiedenen Grundaufgaben
   * des rechnerunterstützten Arbeitsplatzes ...

   DO CASE

      CASE auswahl = '1'

         * Erfassung der Stammdaten

         STOR lw+":(Befehlsdateiname).CMD" TO datei
         IF FILE(datei)
            DO &datei
            STOR F TO entsch
         ELSE
            STOR "(Bezeichnung)/Datenerfassung"
                                              TO text
         ENDI

      CASE auswahl = '2'

         * Erfassung der Bewegungsdaten

         STOR lw+":(Befehlsdateiname).CMD" TO datei
         IF FILE(datei)
            DO &datei
            STOR F TO entsch
         ELSE
            STOR "(Bezeichnung)/Datenerfassung"
                                              TO text
         ENDI

      CASE auswahl = '3'

         * Auswahl von Daten zur Bearbeitung

         STOR lw+":(Befehlsdateiname).CMD" TO datei
         IF FILE(datei)
            DO &datei
            STOR F TO entsch
         ELSE
            STOR "(Bezeichnung)/Datenbearbeitung"
                                              TO text
         ENDI

      CASE auswahl = '4'

         * Archivierung von Daten

         STOR lw+":(Befehlsdateiname).CMD" TO datei
```

```
   IF FILE(datei)
      DO &datei
      STOR F TO entsch
   ELSE
      STOR "(Bezeichnung)/Datenerfassung"
                                        TO text
   ENDI

CASE auswahl = '5'

   * Kontrolle des Bearbeitungsstandes von
   * Aufgaben

   STOR lw+":(Befehlsdateiname).CMD" TO datei
   IF FILE(datei)
      DO &datei
      STOR F TO entsch
   ELSE
      STOR "(Bezeichnung)/Datenbearbeitung"
                                        TO text
   ENDI

CASE auswahl = '6'

   * Aktualisierung der Daten

   STOR lw+":(Befehlsdateiname).CMD" TO datei
   IF FILE(datei)
      DO &datei
      STOR F TO entsch
   ELSE
      STOR "(Bezeichnung)/Datenerfassung"
                                        TO text
   ENDI

CASE auswahl = '7'

   * Analysen und Berichte

   STOR lw+":(Befehlsdateiname).CMD" TO datei
   IF FILE(datei)
      DO &datei
      STOR F TO entsch
   ELSE
      STOR "(Bezeichnung)/Datenbearbeitung"
                                        TO text
   ENDI

CASE auswahl = '8'

   * Programmdiskettenwechsel für den Zweig
   * Dienst- und Havarieaufgaben

   STOR lw+":utilidhp.SUB" TO datei
   IF FILE(datei)
      STOR "A:variable.MEM" TO DATEI
      RELE auswahl, text, j, entsch
      SAVE TO &datei
      QUIT TO 'A:', '&datei. &lw.'
   ELSE
      STOR "(Bezeichnung)/Utilities" TO text
   ENDI

CASE auswahl = '9'

   * Programmdiskettenwechsel für den Zweig
   * Informationsdienste

   STOR lw+":utiliinf.SUB" TO datei
   IF FILE(datei)
      STOR "A:variable.MEM" TO DATEI
      RELE auswahl, text, j, entsch
      SAVE TO &datei
      QUIT TO 'A:', '&datei. &lw.'
   ELSE
      STOR "(Bezeichnung)/Utilities" TO text
   ENDI

CASE auswahl = '0'

   * Abschluß der Arbeiten

   STOR lw+":(Befehlsdateiname).CMD" TO datei
   IF FILE(datei)
      RELE auswahl, text, j, entsch
      DO &datei
   ELSE
      STOR "(Bezeichnung)/Datenerfassung/
            Datenbearbeitung/Utilities" TO text
   ENDI

OTHE

   * eine nicht existierende Aufgabe

   ERAS
   ?
   ? bli1+"A c h t u n g!  "+norm+" Sie haben
            eine nicht existierende Aufgabe"
   ? "ausgewählt!"
   ?
   ? "Wählen Sie erneut eine der aufgeführten
                                    Aufgaben, "
   ? "nachdem Sie die Taste"+inv+"ET"+norm+
                              "gedrückt haben."
   ?
   WAIT
ENDC
IF entsch

   * Es wurde eine falsche Programmdiskette
   * eingelegt,
```

```
      ERAS
      ?
      ? blil+"A c h t u n g! "+norm+" Sie haben eine
                        falsche Diskette eingelegt!"
      ? "Sie hat folgenden Inhalt:"
      ?
      DISP FILE ON &lw LIKE *.*
      ?
      STOR 0 TO j
      DO WHIL j < time
         STOR j+1 TO j
      ENDD
      ? "Legen Sie bitte die Diskette"+inv+text+norm
      ? "nach Aufforderung in Laufwerk"+inv+lw+":"+
                                          norm+"ein."
      ?
      ? "Wenn Sie den Text gelesen haben, drücken Sie
                 bitte die Taste"+inv+" (ET) "+norm
      ?
      WAIT

      * Organisation eines Programmdiskettenwechsels

      STOR "variable.MEM" TO datei
      RELE auswahl, text, j, entsch
      SAVE TO &datei
      STOR lw+":progerro.SUB" TO datei
      QUIT TO "A:", "SUBM &datei. &lw."
   ENDI
   STOR ' ' TO text
ENDD

RETU
```

*** Beispiel 15:**

** Unterprogramm zur Gestaltung eines Aufgabengrundmenüs (unter Berücksichtigung der Möglichkeiten der 16-Bit-Personalcomputer)*

```
*
** MODULE grundmen
**
** DESCRIPTION
**
** Das Modul realisiert das Grundmenü für das Aufga-
** benangebot des rechnergestützten Arbeitsplatzes
** "Projektierung".
** In diesem Beispiel wird die Fenstertechnik für
** die Menüauswahl mit Hilfe eines Balkens demon-
** striert.

PARAMETERS m_farbe_1, m_farbe_6, m_nummer, m_datum,
                                      m_time, m_ende

PRIVATE m_color, m_farbe_11, m_farbe_12, m_farbe_13,
            m_entsch, m_1zeile, m_2zeile, m_zeiles,
                  m_auswahl, m_whol, m_get_nr, m_prname
PRIVATE m_zeile1, m_spalte1, m_spalte2, m_fall,
                        m_whol1, m_text5, m_key, m_inp

* Einrichten für Rückkehrorganisation

IF FILE("variable.MEM")
   REST FROM variable.MEM
   SET DATE TO &m_datum
   DELE FILE variable.MEM
ENDI

* Anfangswerte setzen

m_ende=.F.
m_color=ISCOLOR()
m_prname="PROJGMEN"
m_1zeile=8
m_2zeile=8
m_zeile=0
m_farbe_11=""
m_farbe_12=""
m_farbe_13=""
m_zeile1=21
m_spalte1=1
m_spalte2=11
m_fall='1'
m_text5=" "

* Balken als Fenster (mit vorgebbarer Farbgebung)
* darstellen.

IF m_color
   DO konkorda WITH m_farbe_6, m_farbe_11,
                        m_farbe_12, m_farbe_13, m_ende
   IF m_ende
      RETURN
   ENDIF
ENDIF

* Menüangebot gestalten

m_entsch=.T.
DO WHIL m_entsch
   WSELECT m_nummer
   WDISPLAY  m_nummer
   § 1,0 SAY CENTER (" *** Projektierung -
                              Grundaufgaben *** ", 78)
   § 2,0 SAY CENTER (" Uebersicht ", 78)
   WSELECT m_nummer+1
   § 0,1 SAY "Erarbeitung Preisangebot einschließlich
                                          Stückliste"
   § 1,1 SAY "Erarbeitung verbindliches Preisangebot
                 einschließlich Leistungsbeschreibung"
   § 2,1 SAY "Auswahl von Projektdaten zur technolo-
                                  gischen Aufbereitung"
```

```
@ 3,1 SAY "Übernahme von berechneten Daten in die
                                        Datenbank "
@ 4,1 SAY "Kontrolle des Bearbeitungsstandes von
                                  Aufgabenstellungen "
@ 5,1 SAY "Aktualisierung der Daten der Datenbank"
@ 6,1 SAY "Analysen und Berichte "
@ 7,1 SAY "Dienst- und Havarieaufgaben zum
                       rechnergestützten Arbeitsplatz "
@ 8,1 SAY "Informationsdienste zum
   rechnergestützten Arbeitsplatz 'Projektierung'"
@ 9,1 SAY "und zur zugehörigen Softwarelösung"

* An dieser Stelle können weitere Aufgaben
                                  eingefügt werden

@ 10,1 SAY "Abschluß der Arbeiten am
   rechnergestützten Arbeitsplatz 'Projektierung'"
WSELECT m_nummer+2
WDISPLAY m_nummer+2
@ 0,0 SAY "Balken hoch Balken tief Aufgabe
                                        bestätigen"
WSELECT 0
@ 23,13 SAY "Balken: Cursor links, Cursor rechts;
                                  Wiederholung: ET."

* Definieren des Darstellungsfensters (bei Farbe
                                  bzw. bei Nichtfarbe)

IF m_color
   WSET WINDOW w_inhalt_4 TO m_1zeile, 1,
                                        m_2zeile, 76 ;
   COLOR &m_farbe_11, &m_farbe_12, &m_farbe_13
ELSE
   WSET WINDOW w_inhalt_4 TO m_1zeile, 1,
                                        m_2zeile, 76
ENDIF
WSELECT m_nummer+3
WUSE w_inhalt_4
m_whol=.T.
DO WHILE m_whol
   DO CASE

      CASE m_zeile = 0

         @ 0,1 SAY "Erarbeitung Preisangebot
                        einschließlich Stückliste"

      CASE m_zeile = 1

         @ 0,1 SAY "Erarbeitung verbindliches
Preisangebot einschließlich Leistungsbeschreibung"

      CASE m_zeile = 2

         @ 0,1 SAY "Auswahl von Projektdaten zur
                     technologischen Aufbereitung"

      CASE m_zeile = 3

         @ 0,1 SAY "Übernahme von berechneten
                        Daten in die Datenbank "

      CASE m_zeile = 4

         @ 0,1 SAY "Kontrolle des Bearbeitungs
               standes von Aufgabenstellungen "

      CASE m_zeile = 5

         @ 0,1 SAY "Aktualisierung der Daten der
                                        Datenbank "

      CASE m_zeile = 6
         @ 0,1 SAY "Analysen und Berichte "

      CASE m_zeile = 7

         @ 0,1 SAY "Dienst- und Havarieaufgaben
            zum rechnergestützten Arbeitsplatz "

      CASE m_zeile = 8

         @ 0,1 SAY "Informationsdienste zum
rechnergestützten Arbeitsplatz 'Projektierung'"
         @ 1,1 SAY "und zur zugehörigen
                                  Softwarelösung"

      * An dieser Stelle können weitere Aufgaben
      * eingefügt werden

      CASE m_zeile = 9

         @ 0,1 SAY "Abschluß der Arbeiten am
rechnergestützten Arbeitsplatz 'Projektierung'"
      ENDCASE

      * Abfrage der Tasteneingabe

      whol1=.T.
      DO WHILE whol1

         * Definieren des Darstellungsfensters (bei
         * Farbe bzw. bei Nichtfarbe)

         IF m_color
            WSET WINDOW w_inhalt_5 TO m_zeile1,
                  m_spalte1, m_zeile1, m_spalte2 ;
            COLOR &m_farbe_11, &m_farbe_12,
                                        &m_farbe_13
         ELSE
            WSET WINDOW w_inhalt_5 TO m_zeile1,
                  m_spalte1, m_zeile1, m_spalte2
         ENDIF
```

```
WSELECT m_nummer+4
WUSE w_inhalt_5

DO CASE

   CASE m_fall = '1'

      § 0,0 SAY "Balken hoch"

   CASE m_fall = '2'

      § 0,0 SAY "Balken tief"

   CASE m_fall = '3'

      § 0,0 SAY "Aufgabe bestätigen"

ENDCASE
m_key=0
DO WHILE m_key = 0
   m_key=INKEY()
ENDDO

DO CASE

   CASE m_key = 4

      DO CASE

         CASE m_fall = '1'

            m_spalte1=13
            m_spalte2=23
            m_fall='2'

         CASE m_fall = '2'

            m_spalte1=25
            m_spalte2=42
            m_fall='3'

         CASE m_fall = '3'

            m_spalte1=1
            m_spalte2=11
            m_fall='1'

      ENDCASE

   CASE m_key = 19

      DO CASE

         CASE m_fall = '1'

            m_spalte1=25
            m_spalte2=42
            m_fall='3'

         CASE m_fall = '2'

            m_spalte1=1
            m_spalte2=11
            m_fall='1'

         CASE m_fall = '3'

            m_spalte1=13
            m_spalte2=23
            m_fall='2'

      ENDCASE

   CASE m_key = 13

      DO CASE

         CASE m_fall = '1'

            m_inp=5

         CASE m_fall = '2'

            m_inp=24

         CASE m_fall = '3'

            m_inp=13
            WSELECT 0
            m_text5=REPLICATE(CHR(205),54)
            § 23,13 SAY m_text5
            m_text5=""
            WSELECT m_nummer+2
            m_fall='1'
            m_spalte1=1
            m_spalte2=11

      ENDCASE
      whall=.F.

   ENDCASE

   WCLOSE m_nummer+4

ENDDO

DO CASE

   CASE m_inp = 5

      * Kursor nach oben
```

```
m_zeile=m_zeile-1
IF m_zeile < 0
   m_zeile=0
ENDIF
DO CASE

   CASE m_zeile = 0

      m_1zeile=8
      m_2zeile=8

   CASE m_zeile = 1

      m_1zeile=9
      m_2zeile=9

   CASE m_zeile = 2

      m_1zeile=10
      m_2zeile=10

   CASE m_zeile = 3

      m_1zeile=11
      m_2zeile=11

   CASE m_zeile = 4

      m_1zeile=12
      m_2zeile=12

   CASE m_zeile = 5

      m_1zeile=13
      m_2zeile=13

   CASE m_zeile = 6
      m_1zeile=14
      m_2zeile=14

   CASE m_zeile = 7

      m_1zeile=15
      m_2zeile=15

   CASE m_zeile = 8

      m_1zeile=16
      m_2zeile=17

   CASE m_zeile = 9

      m_1zeile=18
      m_2zeile=18

ENDCASE
```

```
CASE m_inp = 24

   * Kursor nach unten

   m_zeile=m_zeile+1
   IF m_zeile > 9
      m_zeile=9
   ENDIF
   DO CASE

      CASE m_zeile = 0

         m_1zeile=8
         m_2zeile=8

      CASE m_zeile = 1

         m_1zeile=9
         m_2zeile=9

      CASE m_zeile = 2

         m_1zeile=10
         m_2zeile=10

      CASE m_zeile = 3

         m_1zeile=11
         m_2zeile=11

      CASE m_zeile = 4

         m_1zeile=12
         m_2zeile=12

      CASE m_zeile = 5
         m_1zeile=13
         m_2zeile=13

      CASE m_zeile = 6

         m_1zeile=14
         m_2zeile=14

      CASE m_zeile = 7

         m_1zeile=15
         m_2zeile=15

      CASE m_zeile = 8

         m_1zeile=16
         m_2zeile=17

      CASE m_zeile = 9
```

```
            m_1zeile=18
            m_2zeile=18

      ENDCASE

   CASE m_inp = 13

      * Auswahl einer Aufgabe

      m_whol=.F.
      IF m_zeile = 9
         m_auswahl="0"
      ELSE
         m_auswahl=LTRIM(STR(m_zeile+1,2))
      ENDIF

ENDCASE
WCLOSE m_nummer+3
ENDDO
WDISPLAY m_nummer
WDISPLAY m_nummer+1
WDISPLAY m_nummer+2

* Verzweigung in die verschiedenen Grundaufgaben
* des rechnerunterstützten Arbeitsplatzes
* Projektierung

DO CASE

   CASE m_auswahl = '1'

      * Erarbeitung eines Preisangebotes
      * einschließlich Stückliste

      IF FILE("projerf.PRG")
         DO projerf.PRG WITH m_nummer, m_farbe_1,
                        m_farbe_6, m_time, m_ende
         IF m_ende
            RETURN
         ENDIF
         m_entsch=.F.
      ELSE
         m_text5="Projektierung/Preisangebot"
      ENDI

   CASE m_auswahl = '2'

      * Erarbeitung des verbindlichen
      * Preisangebotes einschließlich
      * Leistungsbeschreibung

      IF FILE("projerf.PRG")
         DO projerf.PRG WITH m_nummer, m_farbe_1,
                        m_farbe_6, m_time, m_ende
         IF m_ende
            RETURN
         ENDIF
         m_entsch=.F.
      ELSE
         m_text5="Projektierung/verbindliches
                                    Preisangebot"
      ENDI

   CASE m_auswahl = '3'

      * Auswahl von Daten zur technologischen
      * Aufbereitung

      IF FILE("projtech.PRG")
         DO projtech.PRG WITH m_nummer, m_farbe_1,
                        m_farbe_6, m_time, m_ende
         IF m_ende
            RETURN
         ENDIF
         m_entsch=.F.
      ELSE
         m_text5="Projektierung/technologische
                                    Aufbereitung"
      ENDI
   CASE m_auswahl = '8'

      * Dienst- und Havarieaufgaben
      m_datei="projdhp.PRG"
      IF FILE(m_datei)
         DO projdhp.PRG WITH m_nummer, m_farbe_1,
               m_farbe_6, m_datum, m_time, m_ende
      ELSE
         m_text5="Projektierung/Utilities"
      ENDI

   CASE m_auswahl = '9'

      * Informationsdienste

      m_datei="projinf.PRG"
      IF FILE(m_datei)
         DO projinf WITH m_nummer, m_farbe_1,
                        m_farbe_6, m_time, m_ende
      ELSE
         m_text5=
               "Projektierung/Informationsdienste"
      ENDI
      CASE m_auswahl = '0'

         * Abschluß der Arbeiten

         m_datei="projend.PRG"
         IF FILE(m_datei)
            DO projend WITH m_nummer, m_farbe_1,
                        m_farbe_6, m_time, m_ende
         ELSE
         m_text5="Projektierung/Abschlußleistung"
```

```
      ENDI

   OTHE

      * eine nicht existierende Aufgabe wurde
      * gewählt

      WSELECT m_nummer+2
      WDISPLAY m_nummer+2
      § 0,0 SAY "A c h t u n g!  Sie haben eine
          nicht existierende Aufgabe ausgewählt!"
      § 1,0 SAY "Wählen Sie erneut, nachdem Sie
                  die Taste <ET> gedrückt haben."
      m_whol1=.T.
      DO WHILE m_whol1
         m_inp=0
         DO WHILE m_inp = 0
            m_inp=INKEY()
         ENDDO
         IF m_inp = 13
            m_whol1=.F.
            m_whol=.F.
         ENDIF
         m_inp=0
      ENDDO
      m_entsch=.F.
ENDCASE
IF m_entsch

   * Die Befehlsdatei ist nicht verfügbar.

   WSELECT m_nummer+2
   WDISPLAY m_nummer+2
   § 0,0 SAY "A c h t u n g! Die Befehlsdatei "+
                  m_text5+" ist nicht verfügbar."
   § 1,0 SAY "Das Diskettenverzeichnis hat
                                folgenden Inhalt:"
   WSELECT m_nummer+1
   WSAVE
   WDISPLAY m_nummer+1
   DIR *.*
   SLEEP(3)
   WSELECT m_nummer+2
   WDISPLAY m_nummer+2
   § 0,0 SAY "Wählen Sie nach Aufforderung das
          Diskettenverzeichnis für die Aufgabe "
   § 1,0 SAY m_text5
   WSELECT m_nummer+1
   WDISPLAY m_nummer+1
   m_get_nr=2
   m_pfad="
          "
   SET USERHELP ON
   SET INTENSITY ON
   § 4,1 SAY "Geben Sie Suchpfade an:"
   § 5,1 GET m_pfad PICTURE "XXXXXXXXXXXXXXXXXXXX
                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
          ; HELP helpfun(m_pr_name, m_get_nr, m_nothing)
         READ
         SET USERHELP OFF
         SET INTENSITY OFF
         SET PATH TO &m_pfad
         WDISPLAY m_nummer+1
         WRESTORE
         m_text5=" "
      ENDIF
      m_entsch=.T.
      m_1zeile=8
      m_2zeile=8
      m_zeile=0
ENDDO

RETURN
```

```
* Beispiel 16:
* Konkordanz einer Zeichenkette unter Berücksichti-
* gung des Komma als Trennzeichen
*
** MODULE konkorda
**
** DESCRIPTION
**
** Das Modul realisiert die Zerlegung einer Zei-
** chenkette in Worte unter Berücksichtigung des
** Kommas als Trennzeichen

PARAMETERS m_text, m_text1, m_text2, m_text3, m_ende

PRIVATE m_position, m_text1

m_ende=.F.
m_position=AT(",", m_text)
IF m_position = 0
   m_text1=m_text
   m_text2=" "
   m_text3=" "
ELSE
   m_text1=SUBSTR(m_text,1, m_position-1)
   m_text_1=SUBSTR(m_text, m_position+1, LEN(m_text))
   m=position=AT(",", m_text_1)
   IF m_position = 0
      m_text2=m_text_1
      m_text3=" "
   ELSE
      m_text2=SUBSTR(m_text_1, 1, m_position-1)
      m_text3=SUBSTR(m_text_1, m_position+1,
                                    LEN(m_text_1))
   ENDIF
ENDIF
RETURN
```

## 3. Dateiarbeit

Die folgenden Beispiele zeigen verschiedene Möglichkeiten der Arbeit mit Dateien. Dabei werden solche Aufgaben wie das Erfragen einer Dateibezeichnung, das Setzen und Lesen von Dateiattributen, die Nutzung von Alternatedateien oder die automatische Vergabe und Erfassung von Diskettennamen behandelt.

```
* Beispiel 17:
* Übernahme einer Auswahl aus einem Directory in
* eine Datenbankdatei
*
** MODULE datverz
**
** DESCRIPTION
**
** Das Modul übernimmt eine vorgebbare Auswahl von
** Dateien aus einem Dateiverzeichnis in eine Daten-
** bankdatei.

PARAMETERS m_dirausw, m_datei, m_ende

PRIVATE m_select, m_nr, m_nrs, m_fname

m_ende=.F.
m_select=STR(SELECT())

* Übernahme einer Auswahl aus einem Directory

SELECT 9
SET ALTERNATE TO hdatei.TXT
SET CONSOLE OFF
SET ALTERNATE ON
DIR &m_dirausw
SET ALTERNATE OFF
SET CONSOLE ON
CLOSE ALTERNATE

* Aufbereitung des Directory für die Darstellung

USE &m_datei
APPEND FROM hdatei.TXT SDF
GO TOP
DELETE
GO BOTTOM
SKIP -3
DELETE NEXT 4
PACK
m_fname=FIELD(1)
REPLACE ALL &m_fname WITH SUBSTR(&m_fname, 1, 18)+
   SUBSTR(&m_fname, 20, 18)+SUBSTR(&m_fname, 39, 18)+
                                    SUBSTR(&m_fname, 58, 18)

* Abschlußorganisation

USE
ERASE hdatei.TXT
SELECT &m_select

RETURN
```

```
* Beispiel 18:
* Bereitstellen der Attribute einer vorzugebenden
* Datei
*
** MODULE getattrb
**
** DESCRIPTION
**
** Das Modul stellt die Attribute einer vorzugeben-
** den Datei bereit.
** Dazu wird die Nummer des Bits (1 .. 11) vorgege-
** ben (Bit Nr. 9 ist die R/O- und Bit Nr. 10 die
** SYS-Eigenschaft). Wenn das Attribut gesetzt ist,
** wird 1 geliefert, andernfalls 0. Wenn die Datei
** nicht existiert, wird als Ergebnis der Wert 255
** geliefert.
**
** PARAMETERS
**
** CHARACTER INPUT  VARIABLE  file   'Bezeichnung der
**                              abzufragenden Datei'
** NUMERICAL INPUT  VARIABLE  bitnr  'Nummer des
**                             festzustellenden Bits'
** NUMERICAL OUTPUT VARIABLE  result 'Ergebnis der
**                                         Abfrage'

IF $(file, 2, 1) = ':'
   STORE CHR(RANK(file)-64) TO fcb
ELSE
   STORE CHR(0) TO fcb
ENDIF
STORE $('.', file) TO l
STORE fcb+$($(file, $(':', file)+1, l-1-
      $(':', file))+'        ', 1, 8)+$($(file, l+1,
            LEN(file)-l)+'   ', 1, 3)+CHR(0) TO fcb

* Directory-Eintrag suchen und in die Variable FCB
* kopieren
*
*   ld   c,(hl)
*   ld   b,00h
*   inc  hl
*   push bc
*   push hl
*   push hl
```

```
*   ld   de,0a800h
*   push de
*   ld   c,1ah
*   call 5          ; BDOS-Ruf: DMA-Adresse setzen
*   pop  hl
*   ex   (sp),hl
*   ex   de,hl
*   ld   c,11h
*   call 5          ; BDOS-Ruf: ersten Eintrag suchen
*   cp   0ffh
*   jr   nz,M1
*   pop  hl         ; kein Eintrag gefunden!
*   pop  hl
*   ld   (hl),0ffh
*   pop  bc
*   ret
*M1:sla  a          ; Position des Eintrages im DMA-
*                                 Puffer berechnen
*   sla  a
*   sla  a
*   sla  a
*   sla  a
*   pop  hl
*   add  a,l
*   ld   l,a
*   pop  de
*   pop  bc
*   ldir            ; gefundenen Eintrag in die
*                        Parametervariable kopieren
*   ret

POKE 42752,78,6,0,35,197,229,229,17,00,168,213,14,26,
  205,5,0,225,227,235,14,17,205,5,0,254,255,32,6,225,
  225,54,255,193,201,203,39,203,39,203,39,203,39,203,
                  39,225,133,111,209,193,237,176,201

SET CALL TO 42752
CALL fcb
IF RANK(fcb) = 255
   STORE 255 TO result
ELSE
   STORE INT(RANK($(fcb, bitnr+1, 1))/128) TO result
ENDIF
RELEASE fcb,1

RETURN



* Beispiel 19:
* Setzen der Attribute einer vorzugebenden Datei
*
** MODULE setattri
**
** DESCRIPTION
** Das Modul setzt die Attribute einer vorzugebenden
** Datei auf einen vorzugebenden Wert.
** Dazu ist die Nummer des zu setzenden Bit (1 ...
** 11) (Bit Nr. 9 ist die R/O- und Bit Nr. 10 die
** SYS-Eigenschaft) und der Wert des zu setzenden
** Bit (0 oder 1) anzugeben.
** Wenn die Datei nicht existiert, wird als Ergebnis
** der Wert 255 geliefert.
**
** PARAMETERS
**
** CHARACTER INPUT  VARIABLE file   'Bezeichnung der
**                                    abzufragenden Datei'
** NUMERICAL INPUT  VARIABLE bitnr  'Nummer des
**                                   festzustellenden Bit'
** NUMERICAL INPUT  variable bit    'Wert des Bit'
** NUMERICAL OUTPUT VARIABLE result 'Ergebnis der
**                                              Abfrage'

IF $(file, 2, 1) = ':'
   STORE CHR(RANK(file)-64) TO fcb
ELSE
   STORE CHR(0) TO fcb
ENDIF
STORE $('.', file) TO l
STORE fcb+$($(file, $(':', file)+1,l-1-$(':', file))+
 '        ', 1, 8)+$($(file, l+1, LEN(file)-l)+'   ',
                                   1, 3)+CHR(0) TO fcb
STORE RANK(fcb) TO l

*
* Directory-Eintrag suchen und in die Variable FCB
*                                              kopieren
*
*   ld   c,(hl)
*   ld   b,00h
*   inc  hl
*   push bc
*   push hl
*   push hl
*   ld   de,0a800h
*   push de
*   ld   c,1ah
*   call 5          ; BDOS-Ruf: DMA-Adresse setzen
*   pop  hl
*   ex   (sp),hl
*   ex   de,hl
*   ld   c,11h
*   call 5          ; BDOS-Ruf: ersten Eintrag suchen
*   cp   0ffh
*   jr   nz,M1
*   pop  hl         ; kein Eintrag gefunden!
*   pop  hl
*   ld   (hl),0ffh
*   pop  bc
*   ret
*M1:sla  a          ; Position des Eintrages im DMA-
```

```
*                                   Puffer berechnen
*   sla  a
*   sla  a
*   sla  a
*   sla  a
*   pop  hl
*   add  a,l
*   ld   l,a
*   pop  de
*   pop  bc
*   ldir            ; gefundenen Eintrag in die
*                         Parametervariable kopieren
*   ret
*

POKE 42752,78,6,0,35,197,229,229,17,00,168,213,14,26,
  205,5,0,225,227,235,14,17,205,5,0,254,255,32,6,225,
  225,54,255,193,201,203,39,203,39,203,39,203,39,203,
                  39,225,133,111,209,193,237,176,201

SET CALL TO 42752
CALL fcb

IF RANK(fcb) = 255
   STORE 255 TO result
ELSE
   STORE CHR(L)+$(fcb, 2, 12) TO fcb
   STORE RANK($(fcb, bitnr+1, 1)) TO l
   STORE 1-128*INT(l/128)+128*bit TO l
   STORE $(fcb, 1, bitnr)+CHR(L)+$(fcb, bitnr+2, 12-
                                        bitnr) TO fcb

*  Dateiattribute entsprechend der Variablen FCB
*  setzen
*
*   inc  hl
*   push hl
*   ex   de,hl
*   ld   c,leh
*   call 5          ; BDOS-Ruf: Dateiattribute setzen
*   pop  hl
*   ld   a,(hl)
*   ret

   POKE 42752,35,229,235,14,30,205,5,0, 225,119,201
   CALL fcb

   STORE RANK(fcb) TO result
ENDIF
RELEASE fcb,l

RETURN
```

```
* Beispiel 20:
* Demonstration der Nutzung der Moduln aus den Bei-
* spielen 18 und 19.
** MODULE ATTRIB
**
** DESCRIPTION
**
** Dieses Modul ist ein Beispiel für die Benutzung
** der Moduln SETATTRB und GETATTRB.

SET TALK OFF
ERASE
STORE 'B:T.CMD' TO FILE
STORE 9 TO BITNR
STORE 1 TO BIT
DO WHILE T
   § 1,1 SAY 'File  ' GET FILE
   § 3,1 SAY 'BitNr ' GET BITNR PICTURE '99'
   § 5,1 SAY 'Wert  ' GET BIT PICTURE '9'
   READ
   IF BITNR(1 .OR. BITNR)11
      RETURN
   ENDIF
   IF BIT(2
      DO SETATTRB
      § 9,1 SAY 'Erfolg:'
      § 9,8 SAY SUCC
      WAIT
   ENDIF
   DO GETATTRB
   § 11,1 SAY 'Resultat:'
   § 11,10 SAY RESULT
ENDD

RELEASE FILE,BITNR,BIT,SUCC,RESULT
STOP
```

```
* Beispiel 21:
* Erfragen einer bereits existierenden Datenbankda-
* tei und bereitstellen ihrer Grundinformationen
*
** MODULE datfrag1
**
** DESCRIPTION
**
** Das Modul stellt eine Auswahl aus dem aktuellen
** Dateiverzeichnis bereit und gestattet die Auswahl
** entweder über ein Blättern im angezeigten Datei-
** verzeichnis oder durch direkte Eingabe der Datei-
** bezeichnung. Im letzteren Falle wird geprüft, ob
** die angegebene Datei im aktuellen Dateiverzeich-
** nis aufgeführt ist. Wurde eine falsche Dateibe-
** zeichnung gewählt, kann eine weitere Datei ausge-
** wählt werden. Gleichzeitig besteht die Höglich-
```

```
** keit, über die Taste F10 die Softwarelösung abzu-
** brechen.

PARAMETERS m_muster, m_seldbf, m_wnummer, m_farbe_6,
                          m_prognam, m_dbfbez, m_ende

PRIVATE m_text5, m_datnam, m_pfad, m_antwort, m_key,
                        m_get_nr, m_weiter, m_entsch2

* Bestimmen  des Datenbankverzeichnisses und Anzeigen
* der ersten drei Zeilen im ersten Fenster der  Soft
* warelösung  zum rechnerunterstützten Arbeitsplatz (
* Bezeichnung des rechnerunterstützten Arbeitsplatzes
* ).

WSELECT m_wnummer
WDISPLAY m_wnummer
§ 2,0 SAY "Einen Moment bitte! Ich bin gleich
                                                soweit."
m_datei="softdir.dbf"
DO datverz WITH m_muster, m_datei, m_ende
IF m_ende
   RETURN
ENDIF
SELECT 9
USE &m_datei
m_text5=" *** Verzeichnis Ihrer Datenbanken *** "
§ 0,0 SAY m_text5
DISPLAY NEXT 3 OFF
USE

* Auswahl der aktuellen Datenbank unter wahlweisem
* Blättern im Datenbankverzeichnis

m_weiter=.T.
DO WHILE m_weiter
   WSELECT m_wnummer+1
   WDISPLAY m_wnummer+1
   § 3,0 SAY "Wählen Sie Ihre Datenbank aus oder"
   § 4,0 SAY "geben Sie einen Datenbanknamen vor,"
   § 5,0 SAY "deren Daten bearbeitet werden sollen."
   WSELECT m_wnummer+2
   WDISPLAY m_wnummer+2
   m_datnam="        "
   m_dbfbez="              "
   § 0,0 SAY "Auswahl aus Ihrem Datenbankverzeichnis:
                                    F3; Abbruch der "
   § 1,0 SAY "( Softwarelösung ): F10; weiter:
                                          Leertaste"
   m_entsch2=.T.
   DO WHILE m_entsch2
      m_key=0
      DO WHILE m_key = 0
         m_key=INKEY()
      ENDDO
      DO CASE
         * Taste F10

         CASE m_key = -9
            WSELECT m_wnummer+2
            WSAVE
            DO progfi WITH m_wnummer, m_ende
            IF m_ende
               RETURN
            ENDIF
            WSELECT m_wnummer+2
           WRESTORE

         * Taste F3

         CASE m_key = -2
            WSELECT m_wnummer
            WSAVE

            * Bezl. des Moduls blattdat siehe
            * Beispiel 27

            DO blattdat WITH 3, 72, m_datei,
                    m_wnummer, 18, m_dbfbez, m_text5,
                                    m_farbe_6, m_ende
            IF m_ende
               RETURN
            ENDIF
            WSELECT m_wnummer
            WRESTORE
            m_entsch2=.F.

         * Leertaste

         CASE m_key = 32

            m_entsch2=.F.

      ENDCASE
      m_key=0
   ENDDO
   IF m_dbfbez = "              "
      WSELECT m_wnummer+2
      WDISPLAY m_wnummer+2
      WDISPLAY m_wnummer+1
      § 0,0 SAY "Hilfeinformation. F1."
      WSELECT m_wnummer+1
      m_get_nr=2
      SET USERHELP ON
      § 4,0 GET m_datnam PICTURE "!!!!!!!!"
          HELP tabghi(m_pr_name, m_get_nr, m_nothing)
      READ
      SET USERHELP OFF
      m_dbfbez=RTRIM(m_datnam)+".DBF"
      m_get_nr=3
      m_pfad="
      SET USERHELP ON
```

```
      § 6,0 SAY "Geben Sie eventuelle Suchpfade an:"
      § 7,0 GET m_pfad PICTURE "XXXXXXXXXXXXXXXXXXXXX
     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
          HELP tabghi(m_pr_name, m_get_nr, m_nothing)
      READ
      SET USERHELP OFF
      SET PATH TO &m_pfad
   ENDIF
   IF .NOT. FILE(m_dbfbez)
      WSELECT m_wnummer+2
      WDISPLAY m_wnummer+1
      WDISPLAY m_wnummer+2
      m_antwort="J"
      SET CONFIRM OFF
      § 0,0 SAY "Ihre angegebene Datenbankdatei
                              existiert im angegebenen "
      § 1,0 SAY "Dateiverzeichnis nicht. Wollen Sie
      erneut wählen? (J/N)" GET m_antwort PICTURE "!"
      READ
      SET CONFIRM ON
      WDISPLAY m_wnummer+2
      IF m_antwort = "N"
         § 0,0 SAY "Die Arbeit der ( Softwarelösung )
                                       wird abgebrochen!"
         SLEEP(4)
         m_ende=.T.
         RETURN
      ELSE
         § 0,0 SAY "Hilfeinformation: F1."
      ENDIF
   ELSE
      m_weiter=.F.
   ENDIF
m_key=0
ENDDO
SELECT 9
USE
ERASE &m_datei
SELECT &m_seldbf
USE &m_dbfbez
WSELECT m_wnummer
WDISPLAY m_wnummer+1
WDISPLAY m_wnummer+2
WDISPLAY m_wnummer
§ 0,0 SAY "Sie benutzen Ihre Datenbank "
§ 1,0 SAY DBF() PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX
         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
SLEEP(3)

RETURN
```

*** Beispiel 22:**

** Erfragen einer noch nicht existierenden Daten-*
** bankdatei und Bereitstellen ihrer Grundinforma-*
** tionen*

```
** MODULE datfrag2
**
** DESCRIPTION
**
** Der Modul stellt eine Auswahl aus dem aktuellen
** Dateiverzeichnis bereit und erfragt durch direkte
** Eingabe die Dateibezeichnung. Es wird geprüft, ob
** die angegebene Datei im aktuellen Dateiverzeich-
** Das Modul stellt eine Auswahl aus dem aktuellen
** Dateiverzeichnis bereit und erfragt durch direkte
** Eingabe die Dateibezeichnung. Es wird geprüft, ob
** die angegebene Datei im aktuellen Dateiverzeich-
** nis aufgeführt ist. Ist das der Fall, kann wahl-
** weise die bereits existierende Datei gelöscht
** oder eine neue Dateibezeichnung gewählt werden.

PARAMETERS m_muster, m_seldbf, m_wnummer, m_prognam,
                m_dbfbez, m_strucbez, m_pfad, m_ende

PRIVATE m_text5, m_datnam, m_antwort, m_key, m_datei,
                        m_get_nr, m_entsch, m_entsch1

* Bestimmen des Verzeichnisses aller bereits existie
* renden Dateien, die zum gleichen Dateityp wie die
* zu erfragende Datei gehören und Anzeigen der ersten
* drei Zeilen im ersten Fenster der (Softwarelösung).

WSELECT m_wnummer
WDISPLAY m_wnummer
§ 2,0 SAY "Einen Moment bitte! Ich bin gleich
                                              soweit."
m_datei="softdir.dbf"
DO datverz WITH m_muster, m_datei, m_ende
IF m_ende
   RETURN
ENDIF
SELECT 9
USE &m_datei
m_text5=" *** Verzeichnis Ihrer ( ... )dateien *** "
§ 0,0 SAY m_text5
DISPLAY NEXT 3 OFF
USE

* wahlweises Blättern im ( ...)dateiverzeichnis

WSAVE
WSELECT m_nummer+2
WDISPLAY m_nummer+2
§ 0,0 SAY "Blättern im Verzeichnis Ihrer ( ... )
                                        dateien: PgDn;"
§ 1,0 SAY "Hilfeinformation: F1."

* Erfragen der Bezeichnung der zu kreierenden Datei

m_entsch=.T.
DO WHILE m_entsch
```

```
m_entsch1=.T.
DO WHILE m_entsch1
   WSELECT m_wnummer+1
   § 1,0 SAY "Geben Sie Ihre Bezeichnung für die
                        zu kreierende Datei vor."
   m_datnam="          "
   m_get_nr=2
   SET USERHELP ON
   § 3,0 GET m_datnam PICTURE "!!!!!!!!"
        HELP tabghi(m_prname, m_get_nr, m_nothing)
   READ
   SET USERHELP OFF
   m_key=READKEY()
   IF m_key = 263 .OR. m_key = 7
      WSELECT m_wnummer
      WSAVE
      DO blatdat1 WITH 3, 72, m_datei, m_wnummer,
                                    m_text5, m_ende
      IF m_ende
         RETURN
      ENDIF
      m_datnam="          "
      WSELECT m_wnummer
      WRESTOR
   ENDIF
   IF m_datnam ()  "
      m_entsch1=.F.
   ENDIF
   m_key=0
ENDDO
m_datbez=RTRIM(m_datnam)+".(Dateityp)"

WSELECT m_wnummer+2
WDISPLAY m_wnummer+2

IF FILE(m_datbez)
   § 0,0 SAY "Achtung! Unter Ihrer angeführten
              Bezeichnung existiert bereits eine"
   m_get_nr=4
   m_antwort = "N"
   SET CONFIRM OFF
   SET USERHELP ON
   § 1,0 SAY "( ... )datei. Soll diese Datei
    gelöscht werden? (J/N) " GET m_antwort PICTURE
    "!" HELP tabghi(m_prname, m_get_nr, m_nothing)
   § 1,65 SAY "Hilfe: F1."
   READ
   SET USERHELP OFF
   SET CONFIRM ON
   IF m_antwort = "J"
      ERASE &m_datbez
      WDISPLAY m_wnummer+2
      § 0,0 SAY "Die vorhandene ( ... )datei wurde
                                        gelöscht!"
      SLEEP(2)
      SELECT &m_seldbf
      CREATE &m_datbez FROM &m_strucbez
      m_entsch=.F.
      WDISPLAY m_wnummer+2
      § 0,0 SAY "Die ( ... )datei wurde unter der
                              Bezeichnung "+m_datbez
      m_text5="im Verzeichnis "+m_pfad+
                                    " angelegt. "
      § 1,0 SAY m_text5 PICTURE "AAAAAAAAAAAAAAAAA
            AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                                 AAAAAAAAAAAAAAAA"
      SLEEP(3)
   ELSE
      WDISPLAY m_wnummer+2
      § 0,0 SAY "Geben Sie eine andere Bezeichnung
                        für Ihre ( ... )datei vor."
      SLEEP(3)
   ENDIF
ELSE
   SELECT 7
   CREATE &m_datbez FROM &m_strucbez
   m_entsch=.F.
   WDISPLAY m_wnummer+2
   § 0,1 SAY "Die ( ... )datei wurde unter der
                              Bezeichnung "+m_datbez
   m_text5="im Verzeichnis "+m_pfad+" angelegt. "
   § 1,1 SAY m_text5 PICTURE "AAAAAAAAAAAAAAAAAAA
         AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                              AAAAAAAAAAAAAAAAAA"
   SLEEP(3)
ENDIF
WDISPLAY m_wnummer+2
WRESTORE
ENDDO
WDISPLAY m_wnummer+1
SELECT 9
USE
ERASE &m_datei

RETURN
```

*** Beispiel 23:**

** Erfragen einer noch nicht existierenden Datenbankdatei und Bereitstellen ihrer Grundinformationen.*
** Das Modul entstand aus dem Modul aus Beispiel 22 und soll zeigen, wie aus einer Programmlösung für 16-Bit-Personalcomputer eine Lösung für 8-Bit-Personalcomputer abgeleitet werden kann. Allerdings können dabei nicht alle Leistungen sinnvoll übertragen werden. Es ist jedoch ersichtlich, daß der Kern der Lösung und die damit verbundene Nutzeroberfläche erhalten bleiben. Gleichzeitig soll eine Simulation einer sehr einfachen Fenstertechnik für die Nutzung auf 8-Bit-Personalcompu-*

```
* tern demonstriert werden.  Sie beruht auf der Aus-
* nutzung  der variablen Cursorsteuerung in  Verbin-
* dung mit einer Leitadressenprogrammierung.
*
** MODULE datfrag3
**
** DESCRIPTION
**
** Das  Modul stellt eine Auswahl aus dem  aktuellen
** Dateiverzeichnis bereit und erfragt durch direkte
** Eingabe die Dateibezeichnung. Es wird geprüft, ob
** die  angegebene Datei im aktuellen Dateiverzeich-
** nis aufgeführt ist.  Ist das der Fall, kann wahl-
** weise  die  bereits existierende  Datei  gelöscht
** oder eine neue Dateibezeichnung gewählt werden.

* Bestimmen des Verzeichnisses aller bereits existie
* renden  Dateien,  die zum gleichen Dateityp wie die
* zu erfragende Datei gehören und Anzeigen der ersten
* drei Zeilen im ersten Fenster der (Softwarelösung).

* WSELECT m_wnummer
§ 3,0 SAY CHR(24)
* WDISPLAY m_wnummer
§ $+1,0 SAY CHR(24)
§ $+1,0 SAY CHR(24)
§ $,0 SAY "Einen Moment bitte! Ich bin gleich
                                              soweit."
§ $+1,0 SAY CHR(24)
§ 3,0
STOR "softdir.dbf" TO m_datei
DO datverz
IF m_ende
   RETURN
ENDIF
SELECT SECO
USE &m_datei
STOR " *** Verzeichnis Ihrer ( ... )dateien *** "
                                            TO m_text5
§ $,0 SAY m_text5
DISPLAY NEXT 3 OFF
USE

* wahlweises Blättern im ( ...)dateiverzeichnis

* WSELECT m_nummer+2
§ 21,0 SAY CHR(24)
* WDISPLAY m_nummer+2
§ $+1,0 SAY CHR(24)
§ $-1,0 SAY "Blättern im Verzeichnis Ihrer ( ... )
                                        dateien: F3;"

* Erfragen der Bezeichnung der zu kreierenden Datei

STOR T TO m_entsch
DO WHILE m_entsch
STOR T TO m_entsch1
DO WHILE m_entsch1
   * WSELECT m_wnummer+1
   § 8,0 SAY CHR(24)
   § $+1,0 SAY "Geben Sie Ihre Bezeichnung für die
                          zu kreierende Datei vor."
   STOR "         " TO m_datnam
   § $+2,0 GET m_datnam PICTURE "!!!!!!!!"
   READ
   IF m_datnam () "         "
      STOR F TO m_entsch1
   ENDIF
ENDDO
STOR TRIM(m_datnam)+".(Dateityp)" TO m_datbez

* WSELECT m_wnummer+2
* WDISPLAY m_wnummer+2
§ 21,0 SAY CHR(24)
§ $+1,0 SAY CHR(24)
IF FILE(m_datbez)
   § 21,0 SAY "Achtung! Unter Ihrer angeführten
             Bezeichnung existiert bereits eine"
   STOR "N" TO m_antwort
   § $+1,0 SAY "( ... )datei. Soll diese Datei
    gelöscht werden? (J/N) " GET m_antwort PICTURE
                                                "!"
   READ
   IF m_antwort = "J"
      ERASE &m_datbez
      * WDISPLAY m_wnummer+2
      § 21,0 SAY CHR(24)
      § 22,0 SAY CHR(24)
      § 21,0 SAY "Die vorhandene ( ... )datei wurd
                                          gelöscht!"
      * SLEEP(m_time)
      STOR 0 TO m_i
      DO WHIL m_i ( m_time
         STOR m_i+1 TO m_i
      ENDDO
      SELECT PRIM
      CREATE &m_datbez FROM &m_strucbez
      STOR F TO m_entsch
      * WDISPLAY m_wnummer+2
      § 21,0 SAY CHR(24)
      § 22,0 SAY CHR(24)
      § 21,0 SAY "Die ( ... )datei wurde unter der
                                 Bezeichnung "+m_datbez
      § $+1,0 SAY "angelegt."
      * SLEEP(m_time)
      STOR 0 TO m_i
      DO WHIL m_i ( m_time
         STOR m_i+1 TO m_i
      ENDDO
   ELSE
      * WDISPLAY m_wnummer+2
      § 21,0 SAY CHR(24)
```

```
      § 22,0 SAY CHR(24)
      § 21,0 SAY "Geben Sie eine andere Bezeichnun
                        für Ihre ( ... )datei vor."
      * SLEEP(m_time)
      STOR 0 TO m_i
      DO WHIL m_i < m_time
         STOR m_i+1 TO m_i
      ENDDO
   ENDIF
ELSE
   SELECT PRIM
   CREATE &m_datbez FROM &m_strucbez
   STOR F TO m_entsch
   * WDISPLAY m_wnummer+2
   § 21,0 SAY CHR(24)
   § 22,0 SAY CHR(24)
   § 21,0 SAY "Die ( ... )datei wurde unter der
                              Bezeichnung "+m_datbez
   § $+1,0 SAY "angelegt."
   * SLEEP(m_time)
   STOR 0 TO m_i
   DO WHIL m_i < m_time
      STOR m_i+1 TO m_i
   ENDDO
ENDIF
* WDISPLAY m_wnummer+2
§ 21,0 SAY CHR(24)
§ 22,0 SAY CHR(24)
ENDDO
* WDISPLAY m_wnummer+1
§ 8,0 SAY CHR(24)
STOR 0 TO m_i
DO WHIL m_i < 11
   § $+1,0 SAY CHR(24)
   STOR m_i+1 TO m_i
ENDDO
SELECT SECO
USE
DELE FILE &m_datei

RETURN
```

*** Beispiel 24:**

** Einrichten einer Diskette mit einer Diskettenkennzeichnung und Erfassen dieser Bezeichnungen einem Katalog. Gleichzeitig wird die automatische Kopplung von verschiedenen Programmsystemen mit Hilfe von SUBM-Dateien demonstriert. Im vorliegenden Beispil werden SUBM-Dateien automatisch erzeugt und aktiviert. Der automatische Abarbeitungsprozeß führt am Ende in Ihre Datenbanklösung zurück. Dieses Beispiel umfaßt mehrere Moduln.*

```
*
*
** MODULE dvererdi
**
** DESCRIPTION
**
** Das Modul erfragt die Bezeichnung der einzurich-
** tenden Diskette, das benutzte Betriebssystem, die
** initialisierte Diskettenkapazität und die Disket-
** tenseite. Diese Informationen werden für eine
** SUBM-Aufgabendatei bereitgestellt in die in die-
** ser Datei festgelegte Aufgabenfolge aktiviert
** (siehe nächstes Modul).

* Vorbereitung als Kopfprogramm eines Programmzweiges

STOR "VARIABLE.MEM" TO DATEIA
IF FILE(DATEIA)
   REST FROM &DATEIA
   SET DATE TO &DATUM
   DELE FILE &DATEIA
ENDI

* Abfrage, ob eröffnete Aufgabe wirklich bearbeitet
* werden soll

ERAS
?
? "Wollen Sie die Aufgabe:"
? INV1+" Einrichten von neu initialisierten
                                        Disketten "+NORM
? "bearbeiten?"+INV+" J/N "+NORM
?
WAIT TO ANTWORT
IF !(ANTWORT) = "N"
   RETURN
ENDI
STOR '         ' TO DISKNAMEI
STOR 'A  'TO DISKSEITE
STOR 'SCP1715 ' TO DISKINIT
STOR '780' TO DISKKAPAZ
STOR " einzurichtenden" TO PAR1
STOR " Katalog" TO PAR2

* Laufwerkkontrolle

STOR LW+":DVERERD5.CMD" TO DATEI
DO &DATEI

* Erfassen Kennzeichnung erste Diskette

ERAS
§ 2,1 SAY "Geben Sie den"+INV+" Namen "+NORM+"Ihrer
  ersten"+INV+" einzurichtenden Diskette "+NORM+"an:"
§ 4,1 SAY "Welche"+INV+" Diskettenseite "+NORM+
                              "wollen Sie einrichten?"
§ 5,1 SAY "(Vorderseite oder beide Seiten:"+INV+
                  " A "+NORM+"Rückseite:"+INV+" B "+NORM
```

```
§ 7,1 SAY "Auf welchem"+INV+" Gerät "+NORM+"und in
     welchem"+INV+" Betriebssystem "+NORM+"wurde die"
§ 8,1 SAY "Diskette initialisiert?"
§ 10,1 SAY "Welche"+INV+" Diskettenkapazität "+
                      NORM+"haben Sie initialisiert?"
SET INTE ON
§ 3,1 GET DISKNAMEI PICT 'XXXXXXXX'
§ 6,1 GET DISKSEITE PICT 'A'
§ 9,1 GET DISKINIT PICT 'XXXXXXXX'
§ 11,1 GET DISKKAPAZ PICT '###'
§ 12,1
READ
SET INTE OFF

* Pause für Diskettenwechsel

ERAS
?
? "Legen Sie Ihre"+INV+" Katalogdatenbankdiskette "+
       NORM+"in Laufwerk"+INV+" "+LW1+" "+NORM+"ein."
? "Drücken Sie danach die Taste"+INV+" (ET) "+NORM"
?
WAIT

* Technologische Hinweise für den Nutzer

ERAS
?
? "Legen Sie bitte erst nach Aufforderung Ihre erste
                          einzurichtende Diskette in"
? "Laufwerk"+INV+" "+LW+" "+NORM+"ein."
? "Beachten Sie dann die weiteren Hinweise."
? "Nachdem Ihre Disketten eingerichtet wurden, kehrt
                         die Programmsteurung in das"
? "Disketten- und Dateiverwaltungssystem zurück.
                        Ihre eingerichteten Disketten"
? "werden dann in Ihre Katalogdatenbank eingetragen."
?
? "A c h t u n g! Das Programm läuft automatisch!
                         Benutzen Sie während dieses"
? "Ablaufes keine Taste ohne Aufforderung! Damit wird
                             der automatische Ablauf"
? "unterbrochen."
?
? "Beginnen Sie in einem solchen Falle Ihre Arbeit
                             zweckmäszig von vorn!"
?
? "Wenn Sie die Information gelesen haben, drücken
                                    Sie bitte die"
? "Taste"+INV+" (ET) "+NORM
?
WAIT
STOR "DVERERDI" TO DATRUECK
STOR LW1+":INTEGRIE &LW.:DVERERDI.SUB &LW.
     &DISKNAMEI. &DISKSEITE. &DISKINIT. &DISKKAPAZ."
                                        TO DATEI
RELE DISKNAMEI, DISKSEITE, DISKINIT, DISKKAPAZ, PAR1,
                                                PAR2
SAVE TO &DATEIA
QUIT TO 'A:', '&DATEI'
```

```
MODULE dvererdi.sub

DESCRIPTION

SUBM-Programmdatei dvererdi.sub zur automatischen
Abwicklung fogender Teilaufgaben:
Kennzeichnung der Diskette durch zwei leere Dateien
mit Kennung der Diskette, Diskettenkapazität, Be-
triebssystemhinweis (und evt. Diskettenseite),
Einrichten einer leeren Datei mit dem Namen GO.COM,
Übertragung des aktuellen Betriebssystems auf die
Systemspuren der neu einzurichtenden Diskette,
Erfasaung der eingerichteten Dateien zum Zwecke der
Katalogisierung und automatischen Nachweisführung.
Das Programm wird beim Verlassen des Moduls dvererdi
aktiviert und übernimmt die im Modul dvererdi er-
faßte Diskettenkennzeichnung.

;
; Legen Sie bitte in Laufwerk^Ä^P $1 ^Ä^§Ihre zu
                                  erfassende Diskette ein.
; Drücken Sie danach die Taste^Ä^P (ET) ^Ä^§.
;
PIP
DIENST
SAVE $1:-$2.$3 3d00 0
SAVE $1:$4.$5 3d00 0
SAVE $1:GO.COM 3d00 0
READ 0 1 3d00 96 ÄpÖ
WRITE $1: 0 1 3d00 96
SETRO $1: ***
DIR $1: ÄUÖ
EXIT
ERA KATALOG.KAT
PIP A:=KATALOG.KATÄG150
KATALOG $1
DVERERDI
INTEGRIE DVERERD1.SUB $1 $2 $3 $4 $5
```

```
MODULE dvererdi.bas

DESCRIPTION

Das folgende BASIC-Programm dvererdi.bas erzeugt
eine weitere SUBM-Aufgabendatei dvererd1.sub mit
folgendem wahlweisem Inhalt:
entweder: zur Wiederholung für eine weitere einzu-
```

*richtende Diskette gleicher Inhalt wie dvererdi.sub (Zyklenbildung innerhalb einer SUBM-Aufgabendatei), oder: automatische Rückkehr in das Disketten und Dateiverwaltungssystem.*
*Die Entscheidung erfolgt über einen Nutzerdialog.*

```
10 OPEN "O", #1, "A:DISKERF1.SUB"
20 PRINT CHR$(12);: PRINT: PRINT
30 PRINT "Soll eine weitere Diskette erfaszt werden?"
                  +CHR$(27)+"^P J/N "+CHR$(27)+"^S";
40 ANTWORT$ = INKEY$
50 IF ANTWORT$ = "" THEN GOTO 40
60 PRINT ANTWORT$
70 IF ANTWORT$ = "J" OR ANTWORT$ = "j" THEN GOTO 110
   ELSE IF ANTWORT$ = "N" OR ANTWORT$ = "n" THEN GOTO
                                        80 ELSE GOTO 30
80 PRINT# 1, "$1:": PRINT# 1, ";Legen Sie bitte in
       Laufwerk"+CHR$(27)+"^P $1 "+CHR$(27)+"^SIhre
                              Programmdiskette ein."
90 PRINT# 1, ";Drücken Sie danach die Taste"+
      CHR$(27)+"^P (ET) "+CHR$(27)+"^S.": PRINT# 1,
                                              "A:PIP"
100 PRINT# 1, "$1:( Datenbanksystem )
                         $1:DVERERF1.CMD": GOTO 460
110 INPUT "Geben Sie Ihre Bezeichnung für die
    einzurichtende Diskette vor (maximal 6 Zeichen)";
                                              DBEZ$
120 IF LEN(DBEZ$) > 6 THEN GOTO 110
130 PRINT
140 INPUT "Geben Sie die Diskettenkapazität der
  einzurichtenden Diskette vor (maximal 3 Zeichen):";
                                              DKAP$
150 IF LEN(DKAP$) > 3 THEN GOTO 140
160 PRINT
220 PRINT# 1, ";Legen Sie bitte in Laufwerk"+
     CHR$(27)+"^P $1 "+CHR$(27)+"^SIhre zu erfassende
                                    Diskette ein."
230 PRINT# 1, ";Drücken Sie danach die Taste"+
      CHR$(27)+"^P (ET) "+CHR$(27)+"^S.": PRINT# 1,
                                              "A:PIP"
240 PRINT# 1, "SAVE 0 $1:-F"+DBEZ$+"."+DKAP$
330 PRINT# 1, "DVERERDI"
340 PRINT# 1, "KATALOG"
350 PRINT# 1, "SD"
360 PRINT# 1, "INTEGRIE DISKERF1.SUB $1 $2 $3
370 PRINT CHR$(12);: PRINT: PRINT: PRINT "Rufen Sie
        beim Programm KATALOG bitte Aufgabe 3 auf."
380 PRINT "Katalogisieren Sie nur die eine Dikette"
390 PRINT "Stellen Sie auch keine weitere Aufgabe
                        für das Programm KATALOG"
400 PRINT "Legen Sie, falls erwünscht, bitte eine
                 weitere einzurichtende Diskette in"
410 PRINT "das von Ihnen gewählte Laufwerk ein"
420 PRINT "Schlieszen Sie danach das Programm mit
                                          CTRL-C ab"
430 PRINT: PRINT "Wenn Sie diese Zeilen gelesen
                    haben, drücken Sie die Taste (ET)."
440 BZ$ = INKEY$
450 IF BZ$ = "" THEN GOTO 440
460 CLOSE
470 END
```

```
** MODULE dvererd1
**
** DESCRIPTION
**
** Das Modul übernimmt automatisch die im Rahmen
** der SUBM-Datei(en) erfaßten Diskettenkennzeich-
** nungen und verwaltet sie in entsprechenden Kata-
** logen.

SET ECHO OFF
SET TALK OFF
SET COLO OFF

* Einrichten des Moduls als Kopfprogramm eines Pro-
* grammzweiges

STOR "VARIABLE.MEM" TO DATEIA
IF FILE(DATEIA)
   REST FROM &DATEIA
   SET DATE TO &DATUM
   DELE FILE &DATEIA
ENDI

* Übernahme der zu katalogisierenden Diskettenkenn-
* zeichnungen bzw. Dateibezeichnungen

ERAS
?
? "Einen Moment bitte!"
IF DATRUECK = 'DVERERDI'
? "Ich übernehme erst die zu katalogisierenden Disket
ELSE
? "Ich übernehme erst die zu katalogisierenden Datei
ENDI
?

* Uebernahme der Datei KATALOG.KAT in die Hilfsdatei1

? "Uebernahme der  Datei KATALOG.KAT in eine
                                          Hilfsdatei"
?
STOR LW1+":KATALOG.KAT" TO DATEI
STOR LW1+":DVERERH1.$$$" TO DATEI1
STOR LW+":DVERWAS3.DBF" TO DATEI2
SELE PRIM
CREA &DATEI1 FROM &DATEI2
USE &DATEI1
APPE FROM &DATEI SDF
```

```
STOR 0 TO A, B, C
SELE PRIM
GO TOP

* Selektieren der Angaben zu den freien Disketten-
* kapazitäten

DO WHILE (A <> 1 .OR. B <> 8 .OR. C <> 10) .AND.
                                          .NOT. EOF
   STOR §('++',$(P.DATNAME,1,2)) TO A
   STOR §('K',P.DATNAME) TO B
   STOR §('FRE',P.DATNAME) TO C
   IF A <> 1 .OR. B <> 8 .OR. C <> 10
      SKIP
   ENDI
ENDD
DO WHILE A = 1 .AND. B = 8 .AND. C = 10 .AND
                                          .NOT. EOF
   STOR # TO ANFZEIL
   STOR TRIM(P.USER-P.LAENGE-P.EMPTY-P.ATRIBUT) TO
                                          DIRKOPADRI
   LOCA NEXT 200 FOR §('-',$(P.DATNAME,1,1)) > 0
   STOR P.DATNAME TO DATNAMEI
   LOCA NEXT 200 FOR §('++',$(P.DATNAME,1,2)) > 0
   STOR # TO ENDZEIL
   IF EOF
      STOR STR(ENDZEIL-ANFZEIL+1,4) TO ANZAHL
   ELSE
      STOR STR(ENDZEIL-ANFZEIL,4) TO ANZAHL
   ENDI
   GO ANFZEIL
   REPL NEXT &ANZAHL P.DISKNAME WITH DATNAMEI,
                        P.DIRKOPADR WITH DIRKOPADRI
   SKIP
   STOR §('++',$(P.DATNAME,1,2)) TO A
   STOR §('K',P.DATNAME) TO B
   STOR §('FRE',P.DATNAME) TO C
ENDD
RELE ANFZEIL, ENDZEIL, DIRKOPADRI, ANZAHL

* Erfragen der Datei der freien Kapazitäten

STOR LW+":DVERWAD2.CMD" TO DATEI
DO &DATEI
SELE SECO
USE &DATEI INDEX &DATEI1, &DATEI2

* Uebernahme der freien Kapazitäten in die
* Datenbank der freien Kapazitäten

STOR 0 TO A, B, C
SELE PRIM
GO TOP
DO WHILE (A <> 1 .OR. B <> 8 .OR. C <> 10) .AND.
                                          .NOT. EOF
   STOR §('++',$(P.DATNAME,1,2)) TO A
   STOR §('K',P.DATNAME) TO B
   STOR §('FRE',P.DATNAME) TO C
   IF A <> 1 .OR. B <> 8 .OR. C <> 10
      SKIP
   ENDI
ENDD
ERAS
?
? "Die zu katalogisierenden Disketten verfügen über
                              folgende freie Kapazitäten"
? "-----------------------------------------------
                        ----------------------------"
?
SELE PRIM
DO WHILE A = 1 .AND. B = 8 .AND. C = 10 .AND.
                                          .NOT. EOF
   SELE SECO
   IF §('.',P.DISKNAME) > 0
      STOR $(P.DISKNAME,1,§('.',P.DISKNAME)-1) TO BED
   ELSE
      STOR P.DISKNAME TO BED
   ENDI
   FIND '&BED'
   IF # = 0
   APPEND BLANK
      REPL S.DISKNAME WITH P.DISKNAME
   ENDI
   STOR $(P.DATNAME,1,7) TO HW
   DO WHIL §('++',HW) > 0
      STOR $(HW,3,7) TO HW
   ENDD
   REPL S.FREE WITH STR(VAL(HW),3), S.DATE WITH
                                          DATUMA
   ? "   "+S.DISKNAME+"   "+S.FREE+" KByte"
   STOR 0 TO J
   DO WHIL J < TIME
      STOR J+1 TO J
   ENDD
   SELE PRIM
   REPL P.DISKNAME WITH '          '
   SKIP
   LOCA NEXT 200 FOR §('++',$(P.DATNAME,1,2)) > 0
   STOR §('++',$(P.DATNAME,1,2)) TO A
   STOR §('K',P.DATNAME) TO B
   STOR §('FRE',P.DATNAME) TO C
ENDD

* Erfragen der Datei der Katalogdatenbank

STOR LW+":DVERWAD1.CMD" TO DATEI
DO &DATEI
SELE SECO
USE &DATEI

* Uebernahme der zu katalogisierenden Dateien in die
* Katalogdatenbank
```

```
ERAS
?
? "Übernahme der zu katalogisierenden Disketten und
                                                Dateien"
?
STOR "                                          " TO
                                                BEMERKI
STOR "          " TO DESKRI1
STOR "          " TO DESKRI2
STOR "OFF" TO SCHUTZI
STOR LW1+":DVERERH2.$$$" TO DATEI1
STOR LW1+":DVERERH3.$$$" TO DATEI2
IF FILE(DATEI1)
   DELE FILE &DATEI1
ENDI
SELE PRIM
INDEX ON P.DISKNAME+P.DATNAME TO &DATEI1
COPY TO &DATEI2
USE &DATEI2
GO TOP
DO WHIL .NOT. EOF

   * Bearbeitung einer einzelnen Diskette

   SELE PRIM
   GO TOP
   DO WHIL P.DISKNAME = '          ' .AND.
                                        .NOT. EOF
   SKIP
   ENDD
   IF .NOT. EOF
      STOR P.DISKNAME TO DISKI, DISKI1
      SELE SECO
      LOCA FOR DISKI $ S.DISKNAME
      IF .NOT. EOF
         STOR LW+":DVERERD8.CMD" TO DATEI
         DO &DATEI
      ELSE
         STOR LW+":DVERERD9.CMD" TO DATEI
         DO &DATEI
      ENDI
      ERAS
      ?
      ? "Die Dateien der Diskette"+INV+" "+DISKI1+
        " "+NORM+"wurden vollständig katalogisiert."
   ELSE
      ERAS
      ?
      ? BLI1+"A c h t u n g!"+NORM+"Es liegen keine
                     zu katalogisierenden Disketten"
      ? "und Dateien (mehr) vor."
   ENDI
   STOR 0 TO J
   DO WHIL J < TIME
      STOR J+1 TO J
   ENDD
   STOR F TO ENTSCH
   SELE PRIM
   ENDD

* Abschlußorganisation

STOR LW1+":????????.$$$" TO DATEI
DELE FILE &DATEI
IF DATRUECK = 'DVERERDI'
   ERAS

   ?
   ? "Wollen Sie"+INV+" weitere Disketten "+NORM+
                     "einrichten?"+INV+" J/N "+NORM
   ?
   WAIT TO ANTWORT
   IF !(ANTWORT) = 'J'
      STOR LW+":REDABAS "+LW+":DVERERDI.CMD" TO DATE
      STOR '         ' TO DATRUECK
      RELE DISKI, DESKRI1, DESKRI2, SCHUTZI, BEMERKI
                                                A, B, C
      SAVE TO VARIABLE.MEM
      QUIT TO '&DATEI'
   ENDI
ELSE
   ERAS

   ?
   ? "Wollen Sie"+INV+" weitere Dateien "+NORM+
            "einer"+INV+" weiteren Diskette "+NORM
   ? "katalogisieren?"+INV+" J/N "+NORM
   ?
   WAIT TO ANTWORT
   IF !(ANTWORT) = 'J'
      ERAS

      ?
      ? "Legen Sie Ihre"+INV+"
       Katalogdatenbankdiskette "+NORM+"in Laufwerk"
                        INV+" "+LW1+" "+NORM+"ein."
      ? "Drücken Sie danach die Taste"+INV+" (ET) "+
                                              NORM+"."
      ?
      WAIT
      ERAS

      ?
      ? "Im Interesse einer sorgfältigen und vor
                allem schnelleren Katalogisierung"
      ? " (d.h. alphabetische Einordnung der Dateien
                        für jede Diskette separat"
      ? "(zweistufige Einordnung), sollten Sie
   zweckmäßig nur immer"+INV+" eine Diskette "+NORM
      ? "katalogisieren. Das ist allerdings keine
                                    Einschränkung."
```

```
* Abschlußorganisation

STOR LW1+":????????.$$$" TO DATEI
DELE FILE &DATEI
IF DATRUECK = 'DVERERDI'
   ERAS
   ?
   ? "Wollen Sie"+INV+" weitere Disketten "+NORM+
                      "einrichten?"+INV+" J/N "+NORM
   ?
   WAIT TO ANTWORT
   IF !(ANTWORT) = 'J'
      STOR LW+":REDABAS "+LW+":DVERERDI.CMD" TO DATE
      STOR '         ' TO DATRUECK
      RELE DISKI, DESKRI1, DESKRI2, SCHUTZI, BEMERKI
                                              A, B, C
      SAVE TO VARIABLE.MEM
      QUIT TO '&DATEI'
   ENDI
ELSE
   ERAS
   ?
   ? "Wollen Sie"+INV+" weitere Dateien "+NORM+
             "einer"+INV+" weiteren Diskette "+NORM
   ? "katalogisieren?"+INV+" J/N "+NORM
   ?
   WAIT TO ANTWORT
   IF !(ANTWORT) = 'J'
      ERAS
      ?
      ? "Legen Sie Ihre"+INV+"
       Katalogdatenbankdiskette "+NORM+"in Laufwerk"
                         INV+" "+LW1+" "+NORM+"ein."
      ? "Drücken Sie danach die Taste"+INV+" (ET) "+
                                             NORM+"."
      ?
      WAIT
      ERAS
      ?
      ? "Im Interesse einer sorgfältigen und vor
                  allem schnelleren Katalogisierung"
      ? " (d.h. alphabetische Einordnung der Dateien
                          für jede Diskette separat"
      ? "(zweistufige Einordnung), sollten Sie
    zweckmäszig nur immer"+INV+" eine Diskette "+NORM
      ? "katalogisieren. Das ist allerdings keine
                                    Einschränkung."
      ?
      ? "Legen Sie bitte erst nach Aufforderung Ihre
                   zu katalogisierende Diskette in"
      ? "Laufwerk"+INV+" "+LW+" "+NORM+"ein. Beachten
                   Sie dann die weiteren Hinweise."
      ?
      ? "Nachdem die Dateien Ihrer Diskette erfaszt
                  wurden, kehrt die Programmsteürung"
      ? "in das Disketten- und Dateiverwaltungssystem
                                    zurück. Ihre erfaszten Dateien"
      ? "werden dann in Ihre Katalogdatenbank
                                          eingetragen."
      ?
      ? "A c h t u n g!  Beachten Sie bitte, dasz der
                  weitere Programmablauf automatisch"
      ? "erfolgt. Benutzen Sie bitte in dieser Zeit
                 die Tatstatur nur nach Aufforderung."
      ?
      ? "Wenn Sie die Information gelesen haben,
                                drücken Sie bitte die"
      ? "Taste"+INV+" (ET) "+NORM
      ?
      WAIT
      RELE DISKI, DESKRI1, DESKRI2, SCHUTZI, BEMERKI,
                                              A, B, C
      SAVE TO VARIABLE.MEM
      STOR LW1+":INTEGRIE "+LW+":DVERERDA.SUB &LW" TO
                                                DATEI
      QUIT TO 'A:', '&DATEI'
   ENDI
ENDI
ERAS
?
? "Wollen Sie eine"+INV+"weitere Aufgabe"+NORM+"der
                     Disketten- und Dateiverwaltung"
? "bearbeiten?"+INV+"J/N"+NORM
?
WAIT TO ANTWORT
IF !(ANTWORT) = 'J'
   RELE DISKI, DESKRI1, DESKRI2, SCHUTZI, BEMERKI,
                                              A, B, C
   SAVE TO VARIABLE.MEM
   STOR LW+":REDABAS "+LW+":DVERWAME.CMD" TO DATEI
   QUIT TO '&DATEI'
ELSE
   RELE DISKI, DESKRI1, DESKRI2, SCHUTZI, BEMERKI,
                                              A, B, C
   STOR LW+":DVERWAEN.CMD" TO DATEI
   DO  &DATEI
ENDI
```

## 4. Einsatz von Strukturbeschreibungsdateien

In sogenannten Strukturbeschreibungsdateien wird die Feldstruktur einer Datenbankdatei in Form von Datensätzen abgelegt. In einer solchen Datei lassen sich weitere wichtige Informationen hinterlegen, die im Zusammenhang mit den in der Datei beschriebenen Datenbankdateien stehen wie z. B. eingerichtete Indexdateien, angelegte Sicherungsdateien, auf die Datenbankdatein sich beziehende Reportdateien etc.

Darüber hinaus können die Informationen über die Feldcharakteristika einer Datenbankdatei vorteilhaft in verschiedenen sachlichen Zusammenhängen genutzt werden. Im vorliegenden Beispiel wird eine flexible Datenbankstruktur ermöglicht.

```
* Beispiel 25:
* Arbeit mit einer Strukturbeschreibungsdatei, um
* flexible Dateistrukturen zu nutzen.
*
** MODULE KARTMODS
**
** DESCRIPTION
**
** Der Modul realisiert im Rahmen der Grundaufgaben
** der Karteiarbeit das Modifizieren des Aufbaus
** eines Karteiblattes. Die Struktur eines Kartei-
** blattes ist in einer Strukturbeschreibungsdatei
** erfaßt, die den Dateityp .STR erhalten hat. Die
** Modifizierung des Karteiblatts kann in drei Rich-
** tungen erfolgen:
** Zum einen können einzelne Spalten im Karteiblatt
** stillgelegt (entfernt) oder die charakteristi-
** schen Größen Spaltencharakteristik, Spaltenbreite
** und eventuell vorhandene Dezimalstellenanzahl
** geändert werden.
** Zum zweiten können neue Spalten hinzugefügt und
** drittens können stillgelegte Spalten wieder reak-
** tiviert werden.
** Zunächst werden alle Spaltenangaben angezeigt und
** die gewünschten Veränderungen erfragt. An-
** schließend werden die eventuell neu aufzunehmen-
** den Spalten erfaßt.
** Die von der Strukturänderung der Kartei betroffe-
** nen Indexkarteien werden im Anschluß an die Ände-
** rung aktualisiert bzw. gelöscht.

  Anzeigen einer Warnung

STOR "N" TO ANTWORT
ERAS
§ 2,1 SAY BLI1+" A c h t u n g! "+NORM+" Bei dieser
                          Aufgabe ist Ihr Datenbestand"
§ $+1,1 SAY "gefährdet, da Sie die Struktur Ihrer
                                  Kartei ändern wollen."
§ $+1,1 SAY "Ich empfehle Ihnen, diese Aufgabe mit
                 einer"+INV+" Kopie Ihrer Kartei "+NORM
§ $+1,1 SAY "durchzuführen."
§ $+1,1 SAY "Benutzen Sie dazu die"+INV+" Aufgabe 5 "
                     +NORM+"innerhalb der Grundaufgabe"
§ $+1,1 SAY INV1+" 'Verwalten von eröffneten
                                        Karteien' "+NORM
§ $+2,1 SAY "Wollen Sie die gewählte Aufgabe
                       weiterführen?"+INV+" J/N "+NORM
SET INTE ON
§ $+1,1 GET ANTWORT PICT 'A'
READ
SET INTE OFF
IF !(ANTWORT) = "N"
   RETU
ENDI

* Eröffnung der Karteistrukturbeschreibungsdatei

STOR LW+":"+TRIM(DATNAM)+".STR" TO DATEI1
SELE SECO
USE &DATEI1
GO 1

* Erfragen der zu bearbeitenden Spalten

 STOR F TO FLAGE
 STOR "J" TO  ANTWORT
 STOR F TO ENTSCH
 ERAS
 § 2,1 SAY "Möchten Sie in Ihrer Kartei"+INV+" Spalten
                          ändern, entfernen oder "+NORM
 § $+1,1 SAY INV1+"reaktivieren "+NORM+"?"+INV+"J/N "+
                                                  NORM
 SET INTE ON
 § $+1,1 GET ANTWORT PICT 'A'
 SET INTE OFF
 READ
 IF !(ANTWORT) = 'J'
    STOR T TO ENTSCH
    DO WHIL .NOT. EOF
       IF S.FIELD:NAME () "ELEMMARKER" .AND.
                          S.FIELD:NAME () "BLATMARKER"
          ERAS
          IF *
             § 2,1 SAY "Auf Ihrer Karteikarte befindet
                        sich die stillgelegte Spalte:"
          ELSE
             § 2,1 SAY "Auf Ihrer Karteikarte befindet
                                  sich die Spalte:"
          ENDI
          § $+1,1 SAY INV1+"Spaltenbezeichnung: "+NORM
          § $,25 SAY S.FIELD:NAME USIN "XXXXXXXXXX"
          § $+1,1 SAY INV1+"Spaltencharakteristik: "+
          NORM
          § $,25 SAY S.FIELD:TYPE USIN "X"
          IF S.FIELD:TYPE = 'C' .OR. S.FIELD:TYPE ="N"
             § $+1,1 SAY INV1+"Spaltenbreite: "+NORM
             § $,25 SAY S.FIELD:LEN USIN "###"
             IF S.FIELD:TYPE = "N"
                § $+1,1 SAY INV1+"Anzahl
                                 Dezimalstellen: "+NORM
                § $,25 SAY S.FIELD:DEC USIN "###"
             ENDI
          ENDI
          IF *
```

```
      STOR "N" TO ANTWORT
      § $+1,1 SAY "Möchten Sie diese Spalte"+
INV+" reaktivieren "+NORM+"?"+INV+" J/N "+NORM
      SET INTE ON
      § $+1,1 GET ANTWORT PICT "A"
      SET INTE OFF
      READ
      IF !(ANTWORT) = "J"
         RECA
      ELSE
         LOOP
      ENDI
   ELSE
      STOR "N" TO ANTWORT
      § $+1,1 SAY "Möchten Sie diese Spalte"+
        INV+" stillegen (entfernen) "+NORM+"?"+
                                 INV+" J/N "+NORM
      SET INTE ON
      § $+1,1 GET ANTWORT PICT "A"
       SET INTE OFF
      READ
      IF !(ANTWORT) = "J"
         DELE
         SKIP
         STOR T TO FLAGE
         LOOP
      ENDI
   ENDI
   STOR "N" TO ANTWORT
   § $+1,1 SAY "Möchten Sie die"+INV+
         " charakteristischen Merkmale "+NORM+
                                  "dieser Spalte"
   § $+1,1 SAY "verändern?"+INV+" J/N "+NORM
   SET INTE ON
   § $+1,1 GET ANTWORT PICT "A"
   READ
   SET INTE OFF
   IF !(ANTWORT) = "J"
      § $+2,1 SAY "Bitte führen Sie"+INV+" Ihre
                        Aenderungen "+NORM+"aus:"
      STOR S.FIELD:TYPE TO ZEICHEN
      § $+1,1 SAY "Spaltencharakteristik: "
      § $+1,1 SAY "C bedeutet: Eintragung von
Worten, Wortfolgen oder Zahlen ohne Berechnung"
      § $+1,1 SAY "N bedeutet: Eintragung von
                              Zahlen mit Berechnung"
      § $+1,1 SAY "L bedeutet: Eintragung von
                     Bewertungen T(rue)/F(alse)"
      SET INTE ON
      § $+1,1 GET ZEICHEN PICT 'A'
      READ
      SET INTE OFF
      STOR !(ZEICHEN) TO ZEICHEN
      DO WHILE ZEICHEN <> 'C' .AND. ZEICHEN <>
                        'N' .AND. ZEICHEN <> 'L'
         STOR ' ' TO ZEICHEN
         SET INTE ON
         § $,1 GET ZEICHEN PICT 'A'
         READ
         SET INTE OFF
         STOR !(ZEICHEN) TO ZEICHEN
      ENDD
      REPL S.FIELD:TYPE WITH ZEICHEN
      IF S.FIELD:TYPE = 'L'
         REPL S.FIELD:LEN WITH 1
      ELSE
         SET INTE ON
         § $+1,1 SAY "Spaltenbreite:"
                  GET S.FIELD:LEN PICT '###'
         READ
         SET INTE OFF
         DO WHILE S.FIELD:LEN > 256 .OR.
                                 S.FIELD:LEN < 1
            REPL S.FIELD:LEN WITH 0
            SET INTE ON
             § $,1 SAY "Spaltenbreite:"
                  GET S.FIELD:LEN PICT '###'
            READ
         SET INTE OFF
      ENDD
      ENDI
      IF S.FIELD:TYPE = 'N'
         SET INTE ON
         § $+1,1 SAY "Anzahl Dezimalstellen:"
                  GET S.FIELD:DEC PICT '###'
         READ
         SET INTE OFF
         DO WHILE S.FIELD:DEC > 7 .OR.
                        S.FIELD:DEC < 0 .OR.
                  S.FIELD:DEC >= S.FIELD:LEN
            REPL S.FIELD:DEC WITH 0
            SET INTE ON
            § $,1 SAY "Anzahl Dezimalstellen:"
                  GET S.FIELD:DEC PICT '###'
            READ
            SET INTE OFF
         ENDD
      ELSE
         REPL S.FIELD:DEC WITH 0
      ENDI
   ENDI
   SKIP
   STOR T TO FLAGE
ENDD
ENDI

* Erfragen der neu aufzunehmenden Spalten

STOR # TO J
STOR "J" TO  ANTWORT
ERAS
```

```
§ 2,1 SAY "Möchten Sie in Ihrer Kartei"+INV+
                  " Spalten neu aufnehmen "+NORM+"?"
§ $+1,1 SAY INV1+"J/N "+NORM
SET INTE ON
§ $+1,1 GET ANTWORT PICT 'A'
SET INTE OFF
READ
IF !(ANTWORT) = 'J'
   GO TOP
   ERAS
   § 2,1 SAY "                          Die
                  Spaltenbezeichnungen Ihrer Kartei"
   STOR 0 TO K
   STOR 3 TO K1
   DO WHIL .NOT. EOF
      IF S.FIELD:NAME <> "BLATMARKER" .AND.
           S.FIELD:NAME <> "ELEMMARKER" .AND. .NOT. *
         IF K = 0
            § K1,1 SAY S.FIELD:NAME
                                  USING 'XXXXXXXXXX'
            § K1+1,1 SAY S.FIELD:TYPE USING 'X'
         ELSE
            § K1,K*11+1 SAY S.FIELD:NAME
                                  USING 'XXXXXXXXXX'
            § K1+1,K*11+1 SAY S.FIELD:TYPE  USING 'X'
         ENDI
         STOR K+1 TO K
         IF K= 6
            STOR 0 TO K
            STOR K1+2 TO K1
         ENDI
      ENDI
      SKIP
   ENDD
   STOR K1+2 TO K1
   GO TOP
   STOR T TO ENTSCH
   DO WHIL ENTSCH
      STOR F TO ENTSCH
      STOR '          ' TO FIELDNAME
      § K1,1 SAY LOESCH USING 'X'
      § $+1,1 SAY "Geben Sie die"+INV+" Bezeichnung
              der neuen nächsten Spalte "+NORM+"Ihres
                                        Karteiblattes"
      § $+1,1 SAY "vor. Drücken Sie die Taste"+INV+
       " (ET) "+NORM+", wenn Sie keine weitere Spalte
                                              anlegen."
      SET INTE ON
      § $+1,1 GET FIELDNAME PICT 'XXXXXXXXXX'
      SET INTE OFF
      READ
      IF '          ' $ FIELDNAME .AND. J > 1
         STOR "N" TO ANTWORT
         ERAS
         § 2,1 SAY "Einen kleinen Moment bitte!"
         § $+1,1 SAY "Ich ordne Ihre Angaben nur
                                              schnell ein."
      STOR T TO FLAGE
   ELSE
      STOR T TO ENTSCH
      GO 1
      DO WHIL .NOT. EOF
         IF FIELDNAME = S.FIELD:NAME
            § $+1,1 SAY BLI1+"A c h t u n g!"+
                        NORM+" Doppelte Angabe der
                                 Spaltenbezeichnung!"
            § $+1,1 SAY "Wiederholen Sie bitte
                                      Ihre Angaben."
            STOR 0 TO K2
            DO WHIL K2 <= TIME
               STOR K2+1 TO K2
            ENDD
            STOR T TO ENTSCH
            GO J
         ELSE
            SKIP
         ENDI
      ENDD
      STOR TRIM(FIELDNAME) TO FIELDNAME
      STOR LEN(FIELDNAME) TO LAENGE
      STOR 1 TO I
      DO WHIL I <= LAENGE
         STOR $(FIELDNAME,I,1) TO ZEICHEN
         IF ZEICHEN < " " .OR. (ZEICHEN > " "
          .AND. ZEICHEN < "0") .OR. (ZEICHEN > ":"
          .AND. ZEICHEN < "A") .OR. (ZEICHEN > "Z"
          .AND. ZEICHEN < "a") .OR. ZEICHEN > "z"
            § $+1,1 SAY BLI1+"A c h t u n g!"+NORM
                      +" Fehlerhafte Angabe der
                                 Spaltenbezeichnung!"
            § $+1,1 SAY "Wiederholen Sie bitte
                                      Ihre Angaben."
            STOR 0 TO K2
            DO WHIL K2 <= TIME
               STOR K2+1 TO K2
            ENDD
            STOR T TO ENTSCH
            STOR LAENGE TO I
         ENDI
         STOR I+1 TO I
      ENDD
      STOR T TO ENTSCH
      APPE BLAN
      STOR J+1 TO J
      REPL S.FIELD:NAME WITH !(FIELDNAME)
      § $+2,1 SAY "Charakterisieren Sie den"+INV+
            " Inhalt "+NORM+"dieser Spalte."+INV+
                              " C ö N ö L "+NORM
      § $+1,1 SAY "C bedeutet: Eintragung von
              Worten, Wortfolgen oder Zahlen ohne
                                      Berechnung"
      § $+1,1 SAY "N bedeutet: Eintragung von
```

```
                        Zahlen mit Berechnung"
§ $+1,1 SAY "L bedeutet: Eintragung von
                    Bewertungen T(rü)/F(alse)"
STOR ' ' TO ZEICHEN
SET INTE ON
§ $+1,1 GET ZEICHEN PICT 'A'
READ
SET INTE OFF
REPL S.FIELD:TYPE WITH !(ZEICHEN)
DO WHILE S.FIELD:TYPE () 'C' .AND.
                    S.FIELD:TYPE () 'N' .AND.
                          S.FIELD:TYPE () 'L'
   STOR ' ' TO ZEICHEN
   SET INTE ON
   § $+1,1 GET ZEICHEN PICT 'A'
   READ
   SET INTE OFF
   REPL S.FIELD:TYPE WITH !(ZEICHEN)
ENDD
IF S.FIELD:TYPE = 'L'
   REPL S.FIELD:LEN WITH 1, S.FIELD:DEC
                                        WITH 0
ELSE
   § $+2,1 SAY "Geben Sie die"+INV+"maximale
       Länge "+NORM+"dieser Eintragung an."+
                          INV+" (= 256 "+NORM
   SET INTE ON
   § $+1,1 GET S.FIELD:LEN PICT '###'
   READ
   SET INTE OFF
   DO WHILE S.FIELD:LEN ) 256 .OR.
                              S.FIELD:LEN ( 1
      REPL S.FIELD:LEN WITH 0
      SET INTE ON
      § $,1 GET S.FIELD:LEN PICT '###'
      READ
      SET INTE OFF
   ENDD
ENDI
IF S.FIELD:TYPE = 'N'
   § $+2,1 SAY "Geben Sie die"+INV+" Anzahl
    Dezimalstellen "+NORM+"dieser Eintragung
                        an."+INV+" (= 7 "+NORM
   SET INTE ON
   § $+1,1 GET S.FIELD:DEC PICT '###'
   READ
   SET INTE OFF
   DO WHILE S.FIELD:DEC ) 7 .OR.
                          S.FIELD:DEC ( 0 .OR.
                    S.FIELD:DEC )= S.FIELD:LEN
      REPL S.FIELD:DEC WITH 0
      SET INTE ON
      § $,1 GET S.FIELD:DEC PICT '###'
      READ
      SET INTE OFF
   ENDD
      ELSE
         REPL S.FIELD:DEC WITH 0
      ENDI
   ENDI
   IF K = 0
      § K1-2,1 SAY S.FIELD:NAME USIN 'XXXXXXXXXX'
      § K1-1,1 SAY S.FIELD:TYPE  USING  'X'
   ELSE
      § K1-2,K*11+1 SAY S.FIELD:NAME
                                    USIN 'XXXXXXXXXX'
      § K1-1,K*11+1 SAY S.FIELD:TYPE  USING  'X'
   ENDI
   STOR K+1 TO K
   IF K = 6
      STOR 0 TO K
      STOR K1+2 TO K1
   ENDI
ENDD
STOR T TO ENTSCH
REPL ALL S.FIELD:LEN WITH J FOR S.FIELD:NAME =
                                        "BLATMARKER"
GO 1
REPL INDDATBEZ1 WITH $(INDDATBEZ1,1,5)+STR(J,2)+
          .     $(INDDATBEZ1,8,5), MASDATBEZ WITH
     $(MASDATBEZ,1,5)+STR(J,2)+$(MASDATBEZ,8,5),
     REPDATBEZ1 WITH $(REPDATBEZ1,1,5)+STR(J,2)+
              $(REPDATBEZ1,8,5), ARCDATBEZ1 WITH
     $(ARCDATBEZ1,1,5)+STR(J,2)+$(ARCDATBEZ1,8,5)
REPL BAKDATBEZ1 WITH $(BAKDATBEZ1,1,5)+STR(J,2)+
              $(BAKDATBEZ1,8,5), KOPDATBEZ1 WITH
     $(KOPDATBEZ1,1,5)+STR(J,2)+$(KOPDATBEZ1,8,5)
ENDI

* Kartei nach Strukturänderung neu anlegen

IF FLAGE
   SELE PRIM
   STOR LW1+":"+DATNAM+".$$$" TO DATEI2
   COPY TO &DATEI2
   APPE FROM &DATEI2
   STOR "J" TO ANTWORT
   ERAS
   § 2,1 SAY BLI1+"  A c h t u n g!  "+NORM+" Prüfen
            Sie bitte noch einmal, ob Sie von Ihrer"
   § $+1,1 SAY "Kartei eine Kopie angefertigt haben.
                    Falls nicht, biete ich Ihnen die"
   § $+1,1 SAY "letztmalige Möglichkeit, diese Kopie
                                    noch nachzuholen."
   § $+1,1 SAY "Anderenfalls wird Ihre ursprüngliche
             Kartei durch die von Ihnen gewünschten"
   § $+1,1 SAY "Aenderungen neu eingerichtet."
   § $+2,1 SAY "Wollen Sie eine"+INV+" Kopie Ihrer
          Ursprungskartei "+NORM+"aufbewahren?"+INV+
                                          " J/N "+NORM
   SET INTE ON
   § $+1,1 GET ANTWORT PICT 'A'
```

```
   READ
   SET INTE OFF
   IF !(ANTWORT) = 'J'
      STOR LW+":KARTCOPY.CMD" TO DATEI3
      DO &DATEI3
   ENDI
   STOR LW1+":"+DATNAM+".DBF" TO DATEI
   USE
   DELE FILE &DATEI
   CREA &DATEI FROM &DATEI1
   USE &DATEI
   DELE FILE &DATEI2
   ERAS
   § 2,1 SAY "Ihre Kartei"+INV+"  "+DATNAM+".DBF "+
         NORM+"habe ich Ihren Wünschen entsprechend"
   § $+1,1 SAY "neu strukturiert und eingerichtet."
ELSE
   § 2,1 SAY "Da Sie trotz Aufruf dieser Aufgabe
                  keine Aenderungswünsche für Ihre"
   § $+1,1 SAY "Kartei"+INV+" "+DATNAM+".DBF "+NORM+
            "hatten, können Sie Ihre Kartei in der"
   § $+1,1 SAY "gewohnten Weise nutzen."
ENDI
STOR 0 TO I
DO WHILE I <= TIME
   STOR I+1 TO I
ENDD

* zugeordnete Indexdateien prüfen

IF FLAGE
   SELE SECO
   ERAS
   § 2,1 SAY "Bitte gedulden Sie sich noch!"
   § $+1,1 SAY "Ich aktualisiere erst noch Ihre
                     Indexkarteien, damit Sie auch"
   § $+1,1 SAY "weiterhin mit Ihrer Kartei sortiert
                                 arbeiten können."
   GO TOP
   ERAS
   § 2,1 SAY "                          Die
                Spaltenbezeichnungen Ihrer Kartei"
   STOR 0 TO K
   STOR 3 TO K1
   DO WHIL .NOT. EOF
      IF S.FIELD:NAME <> "BLATMARKER" .AND.
         S.FIELD:NAME <> "ELEMMARKER" .AND. .NOT. *
        IF K = 0
          § K1,1 SAY S.FIELD:NAME
                                 USING 'XXXXXXXXXX'
          § K1+1,1 SAY S.FIELD:TYPE  USING  'X'
        ELSE
          § K1,K*11+1 SAY S.FIELD:NAME
                                 USING 'XXXXXXXXXX'
          § K1+1,K*11+1 SAY S.FIELD:TYPE  USING 'X'
        ENDI
        STOR K+1 TO K
        IF K= 6
          STOR 0 TO K
          STOR K1+2 TO K1
        ENDI
      ENDI
      SKIP
   ENDD
   STOR K1+2 TO K1
   GO TOP
   STOR 0 TO INDDATANZ
   STOR 1 TO INDSCHALT, INDSN
   STOR VAL(§(INDDATBEZ1,1,2)) TO INDDATMAX
   STOR VAL(§(INDDATBEZ1,4,2)) TO INDSNMAX
   STOR VAL(§(INDDATBEZ1,3,1)) TO INDSCHALTM
   STOR F TO FLAGE
   DO WHIL INDDATANZ < INDDATMAX
      STOR INDDATANZ+1 TO INDDATANZ
      STOR INDSCHALT+1 TO INDSCHALT
      IF INDSCHALT > 2
         STOR INDSCHALT-2 TO INDSCHALT
         STOR INDSN+1 TO INDSN
      ENDI
      STOR STR(INDSCHALT,1) TO INDSCHALTS
      STOR S.INDSCHLUE&INDSCHALTS TO SCHLUESSEL
      REPL S.HILFSFELD WITH TEST(&SCHLUESSEL)
      DO WHIL S.HILFSFELD = 0
         STOR T TO FLAGE
         § K1,1 SAY LOESCH USING 'X'
         § $+1,1 SAY BLI1+"A c h t u n g!"+NORM+
             " Der bisher benutzte Schlüsselausdruck"
         § $+1,1 SAY INV1+SCHLUESSEL+" "+NORM+"ist
             nach der Strukturänderung Ihrer Kartei"+
                          INV+" fehlerhaft! "+NORM
         § $+1,1 SAY "Korrigieren Sie bitte Ihren
                              Sortierschlüssel oder "
         § $+1,1 SAY "drücken Sie einfach die Taste"+
      INV+" (ET) "+NORM+", falls Sie diesen Schlüssel"
         § $+1,1 SAY "nicht mehr benutzen wollen."
         STOR '                              ' TO
                                          SCHLUESSEL
         SET INTE ON
         § $+1,1 GET SCHLUESSEL PICT 'XXXXXXXXXXXXXXX
                                         XXXXXXXXXX'
         READ
         SET INTE OFF
         IF SCHLUESSEL = '                          '
            REPL S.HILFSFELD WITH 1
         ELSE
            REPL S.HILFSFELD WITH TEST(&SCHLUESSEL)
         ENDI
      ENDD
      IF SCHLUESSEL = '                          '
         IF # = INDSNMAX .AND. INDSCHALT = INDSCHALM
            REPL S.INDDATBEZ&INDSCHALTS
                                 WITH '          ',
```

```
         S.INDSCHLUE&INDSCHALTS WITH '          ',
         S.INDDISKBE&INDSCHALTS WITH
          S.INDDISKBE&INDSCHALTS WITH '          '
         STOR INDSCHALT-1 TO INDSCHALT
         IF INDSCHALT = 0
            STOR INDSCHALT+2 TO INDSCHALT
            IF INDSN > 1
               STOR INDSN-1 TO INDSN
            ENDI
         ENDI
         STOR  STR(INDSCHALT,1) TO INDSCHALTS
      ELSE
         STOR 9 TO SN
         GO INDSNMAX
         STOR STR(INDSCHALTM,1) TO INDSCHALTS
         STOR INDDATBEZ&INDSCHALTS TO INDDATBEZM
         STOR INDSCHLUE&INDSCHALTS TO INDSCHLUEM
         STOR INDDISKBE&INDSCHALTS TO INDDISKBEM
         STOR INDSCHALTM-1 TO INDSCHALTM
         IF INDSCHALTM = 0
            STOR INDSCHALTM+2 TO INDSCHALTM
            STOR INDSNMAX-1 TO INDSNMAX
         ENDI
         STOR STR(INDSCHALT,1) TO INDSCHALTS
         GO SN
         REPL INDDATBEZ&INDSCHALTS WITH INDDATBEZM
            ,INDSCHLUE&INDSCHALTS WITH INDSCHLUEM,
              INDDISKBE&INDSCHALTS WITH INDDISKBEM
      ENDI
      STOR INDDATMAX-1 TO INDDATMAX
      STOR INDDATANZ-1 TO INDDATANZ
      STOR INDSCHALT-1 TO INDSCHALT
      IF INDSCHALT = 0
         STOR INDSCHALT+2 TO INDSCHALT
         IF INDSN > 1
            STOR INDSN-1 TO INDSN
         ENDI
      ENDI
   ELSE
      REPL S.INDSCHLUE&INDSCHALTS WITH SCHLUESSEL
      STOR S.INDDATBEZ&INDSCHALTS TO DATEI1
      SELE PRIM
      INDEX ON &SCHLUESSEL TO &LW1.:&DATEI1
      SELE SECO
      IF INDSCHALT = 2
         SKIP
      ENDI
   ENDI
ENDD
IF FLAGE
   ERAS
   § 2,1 SAY "Ihre Indexkarteien wurden
     entsprechend der neuen Struktur Ihrer Kartei"
   § $+1,1 SAY "aktualisiert."
ELSE
   ERAS
      § 2,1 "Ihre Indexkarteien bleiben auch nach der
                        Strukturänderung Ihrer Kartei"
      § $+1,1 SAY "ohne Einschränkungen aktüll."
   ENDI                                        uell."
   STOR 0 TO I
   DO WHILE I (= TIME
      STOR I+1 TO I
   ENDD
ENDI
IF INDDATANZ = 2*J-1
   ERAS
   § 2,1 SAY BLI1+"H i n w e i s!"+NORM+" Das
           Verzeichnis für die Sortierschlüssel ist"
   § $+1,1 SAY "ausgeschöpft. Wenn Sie"+INV+" weitere
           Sortierschlüssel "+NORM+"für Ihre Kartei"
   § $+1,1 SAY "benötigen, können Sie zwei Wege
                                        beschreiten:"
   § $+2,1 SAY "1. Sie"+INV+" streichen Schlüssel "+
         NORM+"im Verzeichnis der Sortierschlüssel,"
   § $+1,1 SAY "   die Sie"+INV+" nicht mehr
            benötigen "+NORM+", (Wählen Sie dazu die
                                        entsprechende"
   § $+1,1 SAY "   Aufgabe im Rahmen des
                  rechnerunterstützten Arbeitsplatzes"
   § $+1,1 SAY "  "+INV+"'Allgemeine Karteiarbeit' "+
                                        NORM+".), oder"
   § $+2,1 SAY "2.  Sie legen eine"+INV+" Kopie Ihrer
   aktuellen Kartei "+NORM+"an. (Wählen Sie dazu die"
   § $+1,1 SAY "   entsprechende Aufgabe im Rahmen
            des rechnerunterstützten Arbeitsplatzes"
   § $+1,1 SAY "  "+INV+"'Allgemeine Karteiarbeit' "+
         NORM+".) und bearbeiten Sie diese Kopie mit"
   § $+1,1 SAY "   Ihren weiteren Sortierschlüsseln.
       Beachten Sie, daß Sie"+INV+" nach jeder "+NORM
   § $+1,1 SAY "  "+INV+"Aktualisierung "+NORM+"von
            Ihrer Mutterkartei neue aktuelle Kopien"
   § $+1,1 SAY "   anlegen müssen. (Wählen Sie dazu
            die entsprechende Aufgabe im Rahmen des"
   § $+1,1 SAY "   rechnerunterstützten
   Arbeitsplatzes"+INV+" 'Allgemeine Karteiarbeit' "+
                                        NORM+".)"

ENDI
GO TOP
REPL S.INDDATBEZ1 WITH STR(INDDATANZ-1,2)+
   STR(INDSCHALT,1)+STR(INDSN,2)+$(S.INDDATBEZ1,6,7)
USE
SELE PRIM
RELE ANTWORT, J, I, K, DATEI1, DATBEZ, INDDATANZ,

      AUSDRUCK, LAENGE, FIELDNAME, ENTSCH, ZEICHEN,
INDSCHALT, INDSCHALTS, INDSN, INDDATBEZM, INDSCHLUEM,
      INDDISKBEM, INDSCHALTM, INDSNMAX, INDSNMAX, SN

RETU
```

## 5. Datenerfassung

Datenerfassung und Datenpflege sind grundlegende Voraussetzungen für eine erfolgreiche und qualitätsgerechte Arbeit an einem rechnerunterstützten Arbeitsplatz. Diese Aufgaben haben aus sachlichen Gründen eine große Vielfalt von Erscheinungsformen und bedürfen daher im allgemeinen einer individuellen Programmierung. Programmierleistungen dieser Art sind sehr (zeit-)aufwendig. Das vorliegende komplexe Beispiel kann als Muster dienen, Datenerfassungs- und Pflegeprogramme in gewisser einheitlicher Weise zu gestalten. Der überwiegende Teil derartiger Datenerfassungs- und -pflegeprogramme benötigt die im vorliegenden Beispiel angeführten allgemeinen Komponenten. Aus dem speziell dargestelltem Muster eines Datenerfassungsbogens lassen sich alle anderen Muster in einfacher Weise durch Modifizierung relativ weniger Zeilen des Programmtextes ableiten.

*** Beispiel 26:**
** In diesem Beispiel wird eine Lösung gezeigt, die*
** es gestattet, universelle Datenerfassungs- und -*
** pflegeprogramme zu realisieren. Ausgangspunkt ist*
** die Verwaltung von Daten in einer Datei.An Hand*
** eines Schlüssels wird durch das Programm entschie-*
** den, ob eine Datenersterfassung oder eine Ergän-*
** zung eines bereits bestehenden Datenbestandes*
** erfolgen soll. In dem bereits erfaßten Datenbe-*
** stand kann zu jedem beliebigen Zeitpunkt der Da-*
** tenbearbeitung ein Blättern vorwärts und rückwärts*
** erfolgen. Dabei besteht stets die Möglichkeit, den*
** vorliegenden Datenbestand zu korrigieren oder zu*
** aktualisieren bzw. ausgewählte Datensätze stillzu-*
** legen ( mit einer außerhalb dieser Lösung gegebe*
** nen Möglichkeit, diese Datensätze wieder zu reak-*
** tivieren).*
** Das Beispiel besteht aus mehereren Moduln.*
*
** MODULE projlerf
**
** **DESCRIPTION**
**
*** Das Modul realisiert das Rahmenprogramm für die*
*** Projektierungsaufgabe "Leistungserfassung mit Ko-*
*** sten- und Mengenbewertung". Es werden die benö-*
*** tigten Dateien erfragt und eröffnet. An Hand*
*** einer Objektnummer werden die das zu bearbeitende*
*** Objekt betreffenden Grunddaten wie der maximale*
*** Objektwert, die Kostenstelle, der Kostenträger*
*** etc. bereitgestellt. Danach wird das Modul zur*
*** zur Leistungsbeschreibung aktiviert.*

```
PARAMETERS m_nummer, m_farbe_1, m_farbe_6, m_time, ;
           m_ende

PRIVATE m_antwort, m_selobdbf, m_selobdbf, m_dbfbez,
                                   m_get_nr, m_entschi, ;
        m_objnump, m_kumsumax, m_indbez, m_prname,
                                                m_ende1

* Anfangswerte setzen

m_ende=.F.
m_ende1=.F.
m_prname="PROJ1ERF"
m_selobdbf="8"
m_selsldbf="7"
m_dbfbez="
m_objnump="
m_kumsumax=0.00

* Abfrage, ob das Programm wirklich abgearbeitet
* werden soll

WSELECT m_nummer+2
WDISPLAY m_nummer+2
m_antwort="J"
m_get_nr=1
§ 0,0 SAY "Wollen Sie die Projektierungsaufgabe
                                        bearbeiten: "
§ 1,0 SAY "'Leistungserfassung mit Kosten- und
                              Mengenbewertung'? (J/N)" ;
GET m_antwort PICTURE "!" VALID (m_antwort = "J" ;
       .OR. m_antwort = "N") HELP projhi (m_prognam,
                                  m_get_nr, m_nothing)
READ
IF m_antwort = "N"
  RETURN
ENDIF

* Die Aufgabe wird abgearbeitet

WSELECT 0
§ 1,5 SAY "Leistungsangaben mit Kosten- und
                                      Mengenbewertung"

* Bestimmen des Teilverzeichnisses der
* Objektdateien und Anzeigen
* der ersten drei Zeilen im ersten Fenster der Soft-
* warelösung zum rechnerunterstützten Arbeitsplatz
* Projektierung.

DO projldat WITH m_nummer, m_dbfbez, m_farbe_6,
                                    m_time, m_ende

m_indbez=SUBSTR(m_dbfbez, 1, AT(".", m_dbfbez)-1)+
                                                ".NDX"

IF (.NOT. FILE(m_indbez))
```

```
   SELECT &m_selobdbf
   USE &m_dbfbez ALIAS datob
   INDEX ON datob->obj_num TO &m_indbez
   USE &m_dbfbez ALIAS datob INDEX &m_indbez
ELSE
   SELECT &m_selobdbf
   USE &m_dbfbez ALIAS datob INDEX &m_indbez
ENDIF
WSELECT m_nummer
WDISPLAY m_nummer+1
WDISPLAY m_nummer+2
WDISPLAY m_nummer
§ 1,1 SAY "Sie benutzen Ihre Objektdatenbank "
§ 2,1 SAY DBF() PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX
        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
SLEEP(m_time)

* Bestimmen des Teilverzeichnisses der Dateien der
* Leistungsbeschreibung
* und Anzeigen der ersten drei Zeilen im ersten
* Fenster der Softwarelösung zum
* rechnerunterstützten Arbeitsplatz Projektierung

DO proj3dat WITH m_nummer, m_dbfbez, m_farbe_6,
                                    m_time, m_ende

m_indbez=SUBSTR(m_dbfbez, 1, AT(".", m_dbfbez)-1)+
                                                ".NDX"
IF (.NOT. FILE(m_indbez))
   SELECT &m_selsldbf
   USE &m_dbfbez ALIAS datsl
   INDEX ON datsl->obj_num TO &m_indbez
   USE &m_dbfbez ALIAS datsl INDEX &m_indbez
ELSE
   SELECT &m_selsldbf
   USE &m_dbfbez ALIAS datsl INDEX &m_indbez
ENDIF
WSELECT m_nummer
WDISPLAY m_nummer+1
WDISPLAY m_nummer+2
WDISPLAY m_nummer
§ 1,1 SAY "Sie benutzen Ihre Datenbank der
                              Leistungsbeschreibung:"
§ 2,1 SAY DBF() PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX
        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
SLEEP(m_time)

* Bearbeiten eines Projektes

m_entsch1=.T.
DO WHILE m_entsch1

   * Erfassen der Charakteristika des zu
   * projektierenden Objektes

   DO projobj WITH m_nummer, m_objnump, m_kumsumax,
                                    m_farbe_6, m_time, ;
   m_ende1, m_ende
   IF (m_ende)
      RETURN
   ENDIF

   * Erfassung der zu projektierenden Leistungen

   IF (.NOT. m_ende1)
      DO projlei WITH m_nummer, m_objnump, m_kumsumax
                          ,m_farbe_6, m_time, m_ende
      IF m_ende
         RETURN
      ENDIF
   ENDIF

   WSELECT m_nummer+1
   WDISPLAY m_nummer
   WDISPLAY m_nummer+1
   WDISPLAY m_nummer+2
   SET USERHELP OFF
   SET INTENSITY OFF
   m_antwort="J"
   m_get_nr=19
   § 5,1 SAY "Wollen Sie ein weiteres Objekt
                                bearbeiten? (J/N)";
   GET m_antwort PICTURE "!" VALID (m_antwort = "J"
        .OR. m_antwort = "N") HELP projhi (m_prognam,
                                m_get_nr, m_nothing)
   READ
   IF m_antwort = "N"
      m_entsch1=.F.
   ENDIF
   m_get_nr=19

ENDDO

RETURN



** MODULE projlei
**
** DESCRIPTION
**
** Das Modul entscheidet automatisch, ob Ersterfas-
** sung von Leistungsbeschreibungen oder eine Erwei-
** terung bereits erfaßter Leistungsbeschreibungen
** erforderlich ist. Das betreffende Modul wird
** entsprechend aktiviert.

PARAMETERS m_nummer, m_objnump, m_kumsumax,
                          m_farbe_6, m_time, m_ende

PRIVATE m_color, m_1zeile, m_farbe_11, m_farbe_12,
                          m_farbe_13, m_prognam, ;
```

```
m_ispalte, m_fbez, m_i, m_i1, m_antwort, m_get_nr,
                                m_elnhaupt, m_entsch2, ;
m_spalte, m_kumsum, m_entsch1, m_entsch, m_text5

* Anfangswerte setzen

m_ende=.F.
m_color=ISCOLOR()
m_farbe_11=""
m_farbe_12=""
m_farbe_13=""
m_prognam="PROJLEI "
m_kumsum=0.00
m_elnhaupt="
m_entsch1=.T.
m_antwort="N"

* Erfassung der Projektleistungen

m_text5=CHR(198)+REPLICATE(CHR(205),11)+CHR(209)+
                    REPLICATE(CHR(205),29)+CHR(209)+
                    REPLICATE(CHR(205),10)+CHR(209)+
                    REPLICATE(CHR(205),11)+CHR(209)+
                    REPLICATE(CHR(205),11)+CHR(181)

WSELECT 0
§ 7,0 SAY m_text5
WSELECT m_nummer+1
WDISPLAY m_nummer+1
CLEAR GETS
§ 0,0 SAY "  ELN-Nr.  "CHR(179)+"    Leistungs
                     beschreibung          "+CHR(179)+
        "ME bez.auf"+CHR(179)+" Preis/ME  "+CHR(179)+
                                          "   Menge   "
m_text5=REPLICATE(CHR(205),11)+CHR(216)+
                    REPLICATE(CHR(205),29)+CHR(216)+
                    REPLICATE(CHR(205),10)+CHR(216)+
                    REPLICATE(CHR(205),11)+CHR(216)+
                              REPLICATE(CHR(205),11)
§ 1,0 SAY m_text5
m_text5=REPLICATE(" ",11)+CHR(179)+
                    REPLICATE(" ",29)+CHR(179)+
                    REPLICATE(" ",10)+CHR(179)+
                    REPLICATE(" ",11)+CHR(179)+
                              REPLICATE(" ",11)

FOR m_i2=2 TO 11
   § m_i2,0 SAY m_text5
NEXT m_i2
m_text5=CHR(212)+REPLICATE(CHR(205),11)+CHR(207)+
                    REPLICATE(CHR(205),29)+CHR(207)+
                    REPLICATE(CHR(205),10)+CHR(207)+
                    REPLICATE(CHR(205),11)+CHR(207)+
                     REPLICATE(CHR(205),11)+CHR(190)

WSELECT 0
§ 20,0 SAY m_text5
WSELECT m_nummer+1

* Bereitstellen der Projektleistungen für das
* ausgewählte Objekt

SELECT datsl
SEEK(m_objnump)
IF (.NOT. EOF())

   * Für das betreffende Objekt bereits Leistungen
   * projektiert

   m_entsch1=.T.
   m_fbez=datsl->obj_num
   DO WHILE ((m_objnump $ m_fbez) .AND. m_entsch1)
      FOR m_i=2 TO 11

         * Bereitstellen der bisher projektierten
         * Leistungen im Arbeitsfenster

         m_fbez=datsl->obj_num
         IF ((.NOT. (m_objnump $ m_fbez)) .OR. EOF())

            * Ende der Projektleistungen erreicht
            * Frage nach weiteren Projektleistungen
            * für das betreffende Objekt

            WSELECT m_nummer+2
            WDISPLAY m_nummer+2
            SET USERHELP ON
            SET INTENSITY ON
            m_antwort="N"
            m_get_nr=1
            § 0,0 SAY "Wollen Sie weitere Leistungen
                              projektieren? (J/N)" ;
            GET m_antwort PICTURE "!" VALID
                             (m_antwort = "J" .OR. ;
            m_antwort = "N") HELP projhi (m_prognam,
                               m_get_nr, m_notning)
            READ
            SET USERHELP OFF
            SET INTENSITY OFF

            IF m_antwort = "N"

               * Abschließen und weiteres Objekt
               * bearbeiten

               WDISPLAY m_nummer+1
               WDISPLAY m_nummer+2
               WSELECT 0
               m_text5=CHR(198)+
                     REPLICATE(CHR(205),78)+CHR(181)
               § 7,0 SAY m_text5
               m_text5=CHR(212)+
                  REPLICATE(CHR(205),78)+CHR(190)
               § 20,0 SAY m_text5
               WSELECT m_nummer+1
```

```
            m_i=22
            m_entsch1=.F.
         ELSE

            * Projektleistungen durch weitere
            * Leistungen ergänzen

            m_entsch1=.F.
            m_i1=m_i
            m_i=22
            LOOP
         ENDIF
      ELSE

         * Anzeigen der im Objekt bereits
         * projektierten Leistungen zur Korrektur

         DO projlart WITH m_nummer, m_objnump,
              m_kumsum, m_kumsumax, m_elnhaupt, ;
                   m_i, m_entsch1, m_time, m_ende
      ENDIF
      SKIP
   NEXT m_i
   IF m_entsch1
      WSELECT m_nummer+1
      § 2,0 CLEAR TO 11,77
      m_text5=REPLICATE(" ",11)+CHR(179)+
                     REPLICATE(" ",29)+CHR(179)+
                     REPLICATE(" ",10)+CHR(179)+
                     REPLICATE(" ",11)+CHR(179)+
                              REPLICATE(" ",11)
      FOR m_i2=2 TO 11
         § m_i2,0 SAY m_text5
      NEXT m_i2
   ENDIF
   m_key=0
ENDDO

* Weitere Leistungen projektieren

IF m_antwort = "J"
   DO proj4art WITH m_nummer, m_objnump, m_kumsum,
                        m_kumsumax, m_elnhaupt, ;
   m_i1, m_entsch1, m_time, m_ende
ENDIF
ELSE

* erstmalig Leistungen für ein neues Objekt
* projektieren

m_i=2
DO proj4art WITH m_nummer, m_objnump, m_kumsum,
                        m_kumsumax, m_elnhaupt, ;
m_i, m_entsch1, m_time, m_ende
ENDIF
RETURN
```

```
** MODULE projlart
**
** DESCRIPTION
**
** Das Modul bringt die bereits erfaßten Leistungs-
** beschreibungen für ein ausgewähltes Objekt im
** Arbeitsfenster zur Anzeige. Dabei sind gleichzei-
** tig Korrekturen aller angezeigten Informationen
** sowie die Stillegung einzelner Leistungen mög-
** lich. In diesem Datenbestand kann beliebig vor-
** wärts und rückwärts geblättert werden. Die Bewe-
** gung des Cursors auf eventuelle Korrekturstellen
** ist auf den im Arbeitsfenster angezeigten Daten-
** bestand beschränkt.
** Neue Leistungen können wahlweise an die Lei-
** stungsbeschreibungen angefügt werden. Der kumu-
** lierte Projektwert wird laufend angezeigt und
** hinsichtlich der Oberschreitung eines vorgegebe-
** nen maximalen Projektwertes geprüft. Bei Ober-
** schreitung des Maximalwertes kann der maximale
** Projektwert wahlweise erhöht werden, wobei dann
** weitere Leistungsbeschreibungen eingegeben werden
** können, oder die Eingabe weiterer Leistungsbe-
** schreibungen wird abgebrochen. Eine Bearbeitung
** eines weiteren Objektes ist anschließend sofort
** möglich. Die Erfassung der Leistungsdaten ist zu
** jedem Zeitpunkt einer neu zu erfassenden Leistung
** unterbrechbar. Beim Wiederanlauf des Programms
** kann der Erfassungsprozeß im Umfang des maximalen
** Projektwertes weiter fortgeführt werden.

PARAMETERS m_nummer, m_objnump, m_kumsum, m_kumsumax,
                                        m_elnhaupt, ;
           m_farbe_6, m_i, m_entsch1, m_time, m_ende

PRIVATE m_prognam, m_fbez, m_antwort, m_get_nr,
                                  m_entsch2, m_entsch4, ;
        m_key, m_eln, m_leist, m_meinh, m_preisme,
                                        m_menge, ;
        m_spalte, m_entsch3, m_maske
PRIVATE m_leist1, m_next, m_i1, m_wert, m_i_neu,
                                        m_wert_neu, ;
        m_i_alt, m_wert_alt, m_kumsum_a, m_spalte_a

* Anfangswerte setzen

m_ende=.F.
m_prognam="PROJ1ART"
m_wert=0.00
m_entsch3=.T.

* Anzeigen der Projektleistungen zur Korrektur

WSELECT m_nummer+2
WDISPLAY m_nummer+2
§ 0,0 SAY "Hilfeinformation: F1; keine weiteren
```

```
                              Projektleistungen: F10;"
§ 1,0 SAY "Korrekturhinweise: F3; nächste
                          Projektleistung: Leertaste."
WSELECT m_nummer
m_entsch2=.T.
DO WHILE m_entsch2

   * Wiederholung, bis richtige Taste gedrückt

   m_key=0
   DO WHILE m_key = 0
      m_key=INKEY()
   ENDDO

   DO CASE

      CASE m_key = -9

         * keine weitere Leistung anzeigen

         WDISPLAY m_nummer+1
         WDISPLAY m_nummer+2
         WSELECT 0
         m_text5=CHR(198)+
                     REPLICATE(CHR(205),78)+CHR(181)
         § 7,0 SAY m_text5
         m_text5=CHR(212)+
                     REPLICATE(CHR(205),78)+CHR(190)
         § 20,0 SAY m_text5
         WSELECT m_nummer+1
         m_entsch2=.F.
         m_entsch1=.F.
         m_i=22

      CASE m_key = -2

         * Korrekturhinweise

         WSELECT m_nummer
         WSAVE
         WDISPLAY m_nummer
         § 0,0 SAY "Die bereits erfaßten Projektdaten
          können unter Zuhilfenahme folgender Tasten"
         § 1,0 SAY "korrigiert werden: (Cursor
                     links/rechts) zur Markierung der
                                  Korrekturstelle,"
         § 2,0 SAY "(INS) schaltet Einfügen ein/aus,
        (DEL) löscht durch Cursor markiertes Zeichen"
         § 3,0 SAY "neues Zeichen überschreibt altes,
          wenn Einfügen (INSERT) aus, sonst einfügen"
         WSELECT m_nummer+2
         WSAVE
         WDISPLAY m_nummer+2
         § 0,0 SAY "Wenn Sie den Text gelesen haben,
                    drücken Sie bitte die Taste F10."
         m_entsch2=.T.
         DO WHILE m_entsch2
            m_inp=0
            DO WHILE m_inp = 0
               m_inp=INKEY()
            ENDDO
            IF m_inp = -9
               m_entsch2=.F.
            ENDIF
            m_key=0
         ENDDO
         m_entsch2=.T.
         WDISPLAY m_nummer+2
         WRESTORE
         WSELECT m_nummer
         WDISPLAY m_nummer
         WRESTORE

      CASE m_key = 32

         * die nächste Projektleistung anzeigen

         m_entsch2=.F.

   ENDCASE
   m_key=0
ENDDO

IF m_entsch1
   WSELECT m_nummer+2
   WDISPLAY m_nummer+2
   § 0,0 SAY "nst. Eintrag: ET; vorh. Eintrag:
          Cu li; nst. Leist.: Cu ti; Streichen: END;"
   § 1,0 SAY "vorh. Leist.: Cu ha; Blättern vorw.:
              PgDn; Blättern rückw.: PgUp; Hilfe: F1"
   m_fbez=datsl->obj_num
   m_eln=datsl->eln_num
   m_elnhaupt=SUBSTR(m_eln, 1, 5)
   m_eln=SUBSTR(m_eln, 6, 16)
   m_leist=datsl->leistbez
   m_leist1=SUBSTR(m_leist, 1, 29)
   m_meinh=datsl->maseinheit
   m_preisme=datsl->preismenge
   m_menge=datsl->menge
   WSELECT m_nummer
   § 2,48 SAY m_elnhaupt PICTURE "#####"
   WSELECT m_nummer+1
   m_spalte=1
   DO WHILE ((m_spalte < 6) .OR. (.NOT. m_entsch3))
      SET USERHELP ON
      SET INTENSITY ON
      CLEAR GETS
      m_entsch3=.T.

      DO CASE

         CASE m_spalte = 1
```

```
   IF (DELETED())
      § m_i,0 SAY "gestrichen "
      m_spalte=6
   ELSE
      m_get_nr=10
      § m_i,0 GET m_eln PICTURE
                              "##########" ;
      HELP projhi (m_prognam, m_get_nr,
                                  m_nothing)
   ENDIF
   § m_i,12 SAY m_leist1 PICTURE
            "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   WSELECT m_nummer
   WSAVE
   WDISPLAY m_nummer
   m_maske="!"+REPLICATE("X", 250)+"!"
   § 0,0 SAY "Leistungsbeschreibung: "
   § 0,23 SAY m_leist PICTURE &m_maske
   WSELECT m_nummer+2
   WSAVE
   WDISPLAY m_nummer+2
   § 0,0 SAY "Nach dem Lesen der
vollständigen Leistungsbeschreibung: F10."
   m_entsch4=.T.
   DO WHILE m_entsch4
      m_key=0
      DO WHILE m_key = 0
         m_key=INKEY()
      ENDDO
      IF m_key = -9
         m_entsch4=.F.
      ENDIF
      m_key=0
   ENDDO
   WSELECT m_nummer+2
   WDISPLAY m_nummer+2
   WRESTOR
   WSELECT m_nummer
   WDISPLAY m_nummer
   WRESTOR
   WSELECT m_nummer+1
   § m_i,42 SAY m_meinh PICTURE "XXXXXXXXXX"
   § m_i,53 SAY m_preisme PICTURE
                              "########.##"
   § m_i,65 SAY m_menge PICTURE
                              "#######.###"

CASE m_spalte = 2

   WSELECT m_nummer
   WSAVE
   WDISPLAY m_nummer
   m_maske="!"+REPLICATE("X", 250)+"!"
   m_get_nr=11
   § 0,0 SAY "Leistungs-"
   § 1,0 SAY "beschrei- "
   § 2,0 SAY "bung:
   SET USERHELP ON
   SET INTENSITY ON
   § 0,11 GET m_leist PICTURE &m_maske ;
   HELP projhi (m_prognam, m_get_nr,
                                  m_nothing)

CASE m_spalte = 3

   m_get_nr=12
   § m_i,42 GET m_meinh PICTURE
                              "XXXXXXXXXX" ;
   HELP projhi (m_prognam, m_get_nr,
                                  m_nothing)

CASE m_spalte = 4

   m_get_nr=13
   § m_i,53 GET m_preisme PICTURE
                              "########.##" ;
   HELP projhi (m_prognam, m_get_nr,
                                  m_nothing)

CASE m_spalte = 5

   m_get_nr=14
   § m_i,65 GET m_menge PICTURE
                              "#######.###" ;
   HELP projhi (m_prognam, m_get_nr,
                                  m_nothing)

ENDCASE
IF m_entsch3
   IF m_spalte = 2
      READ
      SET USERHELP OFF
      SET INTENSITY OFF
      WDISPLAY m_nummer
      WRESTORE
      WSELECT m_nummer+1
      m_leist1=SUBSTR(m_leist, 1, 29)
      § m_i,12 SAY m_leist1 PICTURE
               "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
   ELSE
      READ
   ENDIF
   m_key=READKEY()
   DO CASE

      CASE  m_key = 257 .OR. m_key = 1 .OR.
                                  m_key = 15 ;
         .OR. m_key = 271

         * Cursor rechts

         m_spalte=m_spalte+1
```

```
CASE  m_key = 256 .OR. m_key = 0

   * Cursor links

   m_spalte=m_spalte-1
   IF m_spalte < 1
      m_i=m_i-1
      IF m_i < 2
         m_i=2
         m_spalte=1
      ELSE
         SKIP -1
         IF (DELETED())
            @ m_i,0 SAY "gestrichen "
            m_leist=datsl->leistbez
            m_leist1=SUBSTR(m_leist,1,29)
            m_meinh=datsl->maseinheit
            m_preisme=datsl->preismenge
            m_menge=datsl->menge
            @ m_i,12 SAY m_leist1 PICTURE
          "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
            IF m_spalte = 2

               * Wahlweise Anzeige der
     * vollständigen Leistungsbeschreibung

               DO proj5art WITH m_nummer,
                           m_leist, m_ende

               IF m_ende
                  RETURN
               ENDIF
            ENDIF
            @ m_i,42 SAY m_meinh PICTURE
                              "XXXXXXXXXX"
            @ m_i,53 SAY m_preisme
                    PICTURE "########.##"
            @ m_i,65 SAY m_menge PICTURE
                             "#######.###"
            m_spalte=6
            m_entsch3=.F.
         ELSE
            CLEAR GETS
            m_fbez=datsl->obj_num
            m_eln=datsl->eln_num
            m_elnhaupt=SUBSTR(m_eln,1,5)
            m_eln=SUBSTR(m_eln, 6, 16)
            m_leist=datsl->leistbez
            m_leist1=SUBSTR(m_leist,1,29)
            m_meinh=datsl->maseinheit
            m_preisme=datsl->preismenge
            m_menge=datsl->menge
            m_spalte=1
            m_wert=m_menge*m_preisme
            m_kumsum=m_kumsum-m_wert
            @ 2,48 SAY m_elnhaupt PICTURE
                                  "#####"
            @ 1,23 SAY m_wert PICTURE
                              "#########.##"
            @ 1,36 SAY m_kumsum PICTURE
                            "############.##"
            WSELECT m_nummer+1
         ENDIF
      ENDIF
   ENDIF

CASE  m_key = 260 .OR. m_key = 4

   * Cursor hoch

   m_i=m_i-1
   IF m_i < 2
      m_i=2
   ELSE
      SKIP -1
      IF (DELETED())
         @ m_i,0 SAY "gestrichen "
         m_leist=datsl->leistbez
         m_leist1=SUBSTR(m_leist, 1, 29)
         m_meinh=datsl->maseinheit
         m_preisme=datsl->preismenge
         m_menge=datsl->menge
         @ m_i,12 SAY m_leist1 PICTURE
          "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
         IF m_spalte = 2

            * Wahlweise Anzeige der
     * vollständigen Leistungsbeschreibung

            DO proj5art WITH m_nummer,
                        m_leist, m_ende
            IF m_ende
               RETURN
            ENDIF
         ENDIF
         @ m_i,42 SAY m_meinh PICTURE
                           "XXXXXXXXXX"
         @ m_i,53 SAY m_preisme PICTURE
                           "########.##"
         @ m_i,65 SAY m_menge PICTURE
                           "#######.###"
         m_spalte=6
         m_entsch3=.F.
      ELSE
         CLEAR GETS
         m_fbez=datsl->obj_num
         m_eln=datsl->eln_num
         m_elnhaupt=SUBSTR(m_eln, 1, 5)
         m_eln=SUBSTR(m_eln, 6, 16)
         m_leist=datsl->leistbez
         m_leist1=SUBSTR(m_leist, 1, 29)
         m_meinh=datsl->maseinheit
         m_preisme=datsl->preismenge
```

```
            m_menge=datsl->menge
            m_wert=m_menge*m_preisme
            m_kumsum=m_kumsum-m_wert
            WSELECT m_nummer
            @ 2,48 SAY m_elnhaupt PICTURE
                                    "#####"
            @ 1,23 SAY m_wert PICTURE
                              "##########.##"
            @ 1,36 SAY m_kumsum PICTURE
                            "############.##"
            WSELECT m_nummer+1
         ENDIF
      ENDIF

   CASE  m_key = 261 .OR. m_key = 5

      * Cursor tief

      m_wert=m_menge*m_preisme
      SKIP
      m_fbez=datsl->obj_num
      IF m_fbez = m_objnump
         m_i=m_i+1
         IF m_i > 11
            m_i=11
            SKIP -1
         ELSE
            IF DELETED()
               @ m_i,0 SAY "gestrichen "
               m_leist=datsl->leistbez
               m_leist1=SUBSTR(m_leist,1,29)
               m_meinh=datsl->maseinheit
               m_preisme=datsl->preismenge
               m_menge=datsl->menge
               @ m_i,12 SAY m_leist1 PICTURE
             "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
               IF m_spalte = 2

                  * Wahlweise Anzeige der
          * vollständigen Leistungsbeschreibung

                  DO proj5art WITH m_nummer,
                            m_leist, m_ende
                  IF m_ende
                     RETURN
                  ENDIF
               ENDIF
               @ m_i,42 SAY m_meinh PICTURE
                                 "XXXXXXXXXX"
               @ m_i,53 SAY m_preisme
                        PICTURE "########.##"
               @ m_i,65 SAY m_menge PICTURE
                              "#######.###"
               m_spalte=6
               m_enisch3=.F.
            ELSE
               m_kumsum=m_kumsum+m_wert
               m_spalte=1
               CLEAR GETS
               m_fbez=datsl->obj_num
               m_eln=datsl->eln_num
               m_elnhaupt=SUBSTR(m_eln,1,5)
               m_eln=SUBSTR(m_eln, 6, 16)
               m_leist=datsl->leistbez
               m_leist1=SUBSTR(m_leist,1,29)
               m_meinh=datsl->maseinheit
               m_preisme=datsl->preismenge
               m_menge=datsl->menge
               m_whol=.T.
               WSELECT m_nummer
               @ 2,48 SAY m_elnhaupt PICTURE
                                       "#####"
               @ 1,23 SAY m_wert PICTURE
                                 "##########.##"
               @ 1,36 SAY m_kumsum PICTURE
                               "############.##"
               WSELECT m_nummer+1
            ENDIF
         ENDIF
      ELSE
         SKIP -1
      ENDIF

   CASE  m_key = 262 .OR. m_key = 6

      * Taste PgUp

      m_next=-(-m_i+2)+10
      m_i_alt=m_i
      m_wert_alt=m_wert
      m_kumsum_a=m_kumsum
      m_spalte_a=m_spalte
      FOR m_i1=1 TO m_next
         SKIP -1
         m_fbez=datsl->obj_num
         IF m_fbez = m_objnump
            IF (.NOT. (DELETED()))
               m_wert_neu=datsl->preismenge*
                                 datsl->menge
               m_kumsum=m_kumsum-m_wert_neu
            ENDIF
            m_spalte=1
            m_i_neu=2
         ELSE
            SKIP m_i1
            m_i1=22
            m_i_neu=m_i_alt
            m_spalte=m_spalte_a
            m_kumsum=m_kumsum_a
            m_wert_neu=m_wert_alt
         ENDIF
         m_key=0
```

```
   NEXT m_i1
   m_eln=datsl->eln_num
   m_elnhaupt=SUBSTR(m_eln, 1, 5)
   m_eln=SUBSTR(m_eln, 6, 16)
   m_leist=datsl->leistbez
   m_leist1=SUBSTR(m_leist, 1, 29)
   m_meinh=datsl->maseinheit
   m_preisme=datsl->preismenge
   m_menge=datsl->menge
   m_wert=m_wert_neu
   WSELECT m_nummer
   @ 2,48 SAY m_elnhaupt PICTURE "#####"
   @ 1,23 SAY m_wert PICTURE
                          "#########.##"
   @ 1,36 SAY m_kumsum PICTURE
                          "############.##"
   WSELECT m_nummer+1
   m_i=m_i_neu
   WSELECT m_nummer+1
   IF m_i1 < 20
      @ 2,0 CLEAR TO 11,77
      m_text5=REPLICATE(" ",11)+CHR(179)+
              REPLICATE(" ",29)+CHR(179)+
              REPLICATE(" ",10)+CHR(179)+
              REPLICATE(" ",11)+CHR(179)+
                        REPLICATE(" ",11)
      FOR m_i2=2 TO 11
         @ m_i2,0 SAY m_text5
      NEXT m_i2
   ENDIF

CASE  m_key = 263 .OR. m_key = 7

   * Taste PgDn

   m_i_alt=m_i
   m_wert_alt=m_wert
   m_kumsum_a=m_kumsum
   m_spalte_a=m_spalte
   m_wert_neu=datsl->preismenge*
                              datsl->menge
   FOR m_i1=m_i+1 TO 12
      SKIP
      m_fbez=datsl->obj_num
      IF (.NOT. (m_fbez = m_objnump))
         SKIP -(m_i1-m_i)
         m_i1=22
         m_i_neu=m_i_alt
         m_spalte=m_spalte_a
         m_kumsum=m_kumsum_a
         m_wert_neu=m_wert_alt
      ELSE
         IF (.NOT. (DELETED()))
            m_kumsum=m_kumsum+m_wert_neu
            m_wert_neu=datsl->preismenge*
                              datsl->menge
            ENDIF
            m_spalte=1
            m_i_neu=2
         ENDIF
         m_key=0
      NEXT m_i1
      m_fbez=datsl->obj_num
      m_eln=datsl->eln_num
      m_elnhaupt=SUBSTR(m_eln, 1, 5)
      m_eln=SUBSTR(m_eln, 6, 16)
      m_leist=datsl->leistbez
      m_leist1=SUBSTR(m_leist, 1, 29)
      m_meinh=datsl->maseinheit
      m_preisme=datsl->preismenge
      m_menge=datsl->menge
      m_wert=m_wert_neu
      WSELECT m_nummer
      @ 2,48 SAY m_elnhaupt PICTURE "#####"
      @ 1,23 SAY m_wert PICTURE
                          "#########.##"
      @ 1,36 SAY m_kumsum PICTURE
                          "############.##"
      WSELECT m_nummer+1
      m_i=m_i_neu
      WSELECT m_nummer+1
      IF m_i1 < 20
         @ 2,0 CLEAR TO 11,77
         m_text5=REPLICATE(" ",11)+CHR(179)+
                 REPLICATE(" ",29)+CHR(179)+
                 REPLICATE(" ",10)+CHR(179)+
                 REPLICATE(" ",11)+CHR(179)+
                           REPLICATE(" ",11)
         FOR m_i2=2 TO 11
            @ m_i2,0 SAY m_text5
         NEXT m_i2
      ENDIF

CASE  m_key = 259 .OR. m_key = 3

   * Taste END

   DELETE
   m_wert=m_menge*m_preisme
   m_kumsum=m_kumsum-m_wert
   WSELECT m_nummer
   @ 2,48 SAY m_elnhaupt PICTURE "#####"
   @ 1,23 SAY m_wert PICTURE
                          "#########.##"
   @ 1,36 SAY m_kumsum PICTURE
                          "############.##"
   WSELECT m_nummer+2
   WSAVE
   WDISPLAY m_nummer+2
   @ 0,1 SAY "Die durch den Cursor
markierte Leistung habe ich gestrichen."
   SLEEP(m_time)
```

```
            WDISPLAY m_nummer+2
            WSELECT m_nummer+1
            § m_i,0 SAY "gestrichen "

      ENDCASE
   ENDIF
   SET USERHELP OFF
   SET INTENSITY OFF
ENDDO

* Überprüfung des maximalen Objektwertes

IF m_entsch3
   m_wert=m_menge*m_preisme
   m_kumsum=m_kumsum+m_wert
   WSELECT m_nummer
   § 1,23 SAY m_wert PICTURE "##########.##"
   § 1,36 SAY m_kumsum PICTURE "##############.##"
   m_entsch=.T.
   DO WHILE m_entsch
      IF m_kumsum >= m_kumsumax
         WSELECT m_nummer+2
         WSAVE
         WDISPLAY m_nummer+2
         § 0,0 SAY CHR(7)
         § 0,0 SAY "Achtung! Ihr vorgegebener
        maximaler Objektwert wurde überschritten."
         SET USERHELP ON
         SET INTENSITY ON
         m_antwort="J"
         m_get_nr=8
         § 1,0 SAY "Wollen Sie den Objektwert
                          weiter erhöhen? (J/N)" ;
         GET m_antwort PICTURE "!" VALID
                          (m_antwort = "J" .OR. ;
         m_antwort = "N") HELP projhi (m_prognam,
                          m_get_nr, m_nothing)
         READ
         SET USERHELP OFF
         SET INTENSITY OFF
         IF m_antwort = "J"
            m_get_nr=9
            SET USERHELP ON
            SET INTENSITY ON
            WDISPLAY m_nummer+2
            § 0,1 SAY "Erhöhen Sie den
               vorgegebenen maximalen Objektwert."
            § 1,1 SAY "alter Objektwert: "+
                          LTRIM(STR(m_kumsumax))
            § 1,40 SAY "neuer Objektwert: " GET
            m_kumsumax PICTURE "##############.##" ;
                  HELP projhi (m_prname, m_get_nr,
                                    m_nothing)
            READ
            SET USERHELP OFF
            SET INTENSITY OFF
            SELECT datob
            REPLACE datob->obj_maxwer WITH
                                    m_kumsumax
            SELECT datsl
            WDISPLAY m_nummer+2
            WRESTORE
         ELSE
            SET USERHELP OFF
            SET INTENSITY OFF
            m_antwort="J"
            m_get_nr=10
            WDISPLAY m_nummer+2
            § 1,0 SAY "Wollen Sie bereits
                  projektierte Leistungen wieder
                        streichen? (J/N)" ;
            GET m_antwort PICTURE "!" VALID
                        (m_antwort = "J" .OR. ;
            m_antwort = "N") HELP projhi
                  (m_prognam, m_get_nr, m_nothin
            READ
            IF m_antwort = "N"
               WDISPLAY m_nummer+1
               WDISPLAY m_nummer+2
               WSELECT 0
               m_text5=CHR(198)+
                  REPLICATE(CHR(205),78)+CHR(181)
               § 7,0 SAY m_text5
               m_text5=CHR(212)+
                  REPLICATE(CHR(205),78)+CHR(190)
               § 20,0 SAY m_text5
               WSELECT m_nummer+1
               m_entsch1=.F.
               m_i=22
            ENDIF
         ENDIF
      ELSE
         SELECT datsl
         REPLACE datsl->obj_num WITH m_objnump,
           datsl->eln_num WITH m_elnhaupt+m_eln, ;
         datsl->leistbez WITH m_leist,
                  datsl->maseinheit WITH m_meinh, ;
         datsl->preismenge WITH m_preisme
         REPLACE datsl->menge WITH m_menge
         m_entsch=.F.
      ENDIF
      WSELECT m_nummer+1
   ENDDO
ENDIF

RETURN
```

## 6. Blättern in Datenbankdateien

Sichten und Auswählen von Daten aus (umfangreichen) Datenbeständen gehören zu den grundlegenden Aufgaben eines rechnerunterstützten Arbeitsplatzes. Die für gewöhnlich in Datenbanksystemen verfügbaren Befehle zur Realisierung derartiger Aufgaben sind entweder nicht ausreichend leistungsfähig (nur vorwärts orientiertes Anzeigen etc.) oder erfordern tiefere Kenntnisse des betreffenden Datenbanksystems.
Im vorliegenden Abschnitt wird ein Algorithmus demonstriert, der ein beliebiges Blättern in Datenbankdateien ermöglicht und gleichzeitig die Auswahl eines gewünschten Datums mittels Balkenmarkierung gestattet. Dieser Algorithmus ist in Verbindung mit anderen in diesem Heft demonstrierten Methoden sehr interessant koppelbar. Z. B. führt eine Verbindung dieses Blätteralgorithmus mit der Nutzung einer Strukturbeschreibungsdatei und einem Recherchekriterium oder einer Indexierung zu Möglichkeiten einer sehr gezielten Aufbereitung eines Datenbestandes für operative Analysen, Berichte und andere Darstellungen, wie sie für eine niveauvolle Arbeit eines rechnerunterstützten Arbeitsplatzes notwendig werden.

```
* Beispiel 27:
* Blättern in einer Datenbankdatei und Auswählen
* eines Elementes mit Hilfe einer Balkentechnik.
*
** MODULE blattdat
**
** DESCRIPTION
**
** Der Modul ermöglicht das beliebige Blättern in
** einer Datenbankdatei und gestattet die Auswahl
** eines durch einen Farbbalken markierten Datenele-
** mentes. Das Blättern erfolgt seitenweise nach
** vorwärts oder rückwärts vermittels der Bildta-
** sten. Eine Seite wird definiert als Inhalt eines
** Darstellungsfensters. Innerhalb einer Seite kann
** ein Farbbalken mittels der Cursortasten auf jedes
** Datenelement gestellt werden. Die Übernahme des
** gewünschten Datenelementes in eine Variable er-
** folgt vermittels der Taste (ET).

PARAMETERS m_zanz, m_zlaenge, m_bezdatei, m_wnummer,
           m_sbreite, m_inhalt, m_titel, m_farbe_6,
           m_ende

PRIVATE m_seite, m_snum, m_whol, m_snummax, m_snum1,
        m_snum2, m_select, m_select1, m_szanz,
        m_antwort, m_inp, m_farbe_11, m_farbe_12,
        m_farbe_13, m_text, m_text1, m_fname,
        m_laenge, m_whol1, m_zeile, m_spalte,
        m_color, m_lzeile, m_lspalte, m_datei

* Eröffnen des Darstellungsfensters

WSELECT m_wnummer
WSAVE
WDISPLAY m_wnummer
§ 0,0 SAY m_titel
m_lzeile=3
m_lspalte=2

* Test, ob das Darstellungsfenster breit genug ist

IF m_zlaenge (= 43
   ? CENTER("Die Fensterbreite muß mindestens 44
                        Zeichen betragen.", m_zlaenge)
   ? CENTER("Es erfolgt Abbruch!", m_zlaenge)
   ?? ""
   RETURN
ENDIF

* Anfangswerte setzen

m_antwort=" "
m_seite=1
m_snum=1
m_szanz=STR(m_zanz,2)
m_zeile=1
m_spalte=1
m_color=ISCOLOR()
IF m_color
   DO konkorda WITH m_farbe_6, m_farbe_11,
                    m_farbe_12, m_farbe_13, m_ende
ENDIF
m_datei=" "

* Eröffnen der darzustellenden Datenbank

m_select=STR(SELECT(),2)
SELECT 9
USE &m_bezdatei
GO BOTTOM
m_snummax=RECNO()
GO TOP
m_fname=FIELD(1)

* Die Datenbank ist leer

IF m_snummax = 0
   WDISPLAY m_wnummer
   § 0,0 SAY m_titel
   § 2,0 SAY CENTER("Die anzuzeigende Datei ist
                                leer!", m_zlaenge-1)
   § 2,m_zlaenge-1 SAY ""
   SLEEP(5)
```

```
   RETURN
ELSE

   * Die Datenbank umfaßt maximal eine Seite
   * (eine Seite entspricht einem Fensterinhalt.)

   IF m_snummax <= m_zanz
      m_whol=.T.
      DO WHILE m_whol
         WDISPLAY m_wnummer
         @ 0,0 SAY m_titel
         DISPLAY ALL OFF
         WSELECT m_wnummer+2
         WDISPLAY m_nummer+2
         @ 0,0 SAY "Seite 1,SATZ 1-"+
               STR(m_snummax,2)+" exit: F10; Balken:
                                 Cursor; Auswahl: ET"
         @ 1,m_zlaenge-1 SAY ""
         IF m_color
            WSET WINDOW w_inhalt_4 TO m_1zeile+
               m_zeile, m_1spalte+m_spalte, m_1zeile+
              m_zeile, m_1spalte+m_spalte+m_sbreite-1
                     COLOR &m_farbe_11, &m_farbe_12,
                                        &m_farbe_13
         ELSE
            WSET WINDOW w_inhalt_4 TO m_1zeile+
               m_zeile, m_1spalte+m_spalte, m_1zeile+
              m_zeile, m_1spalte+m_spalte+m_sbreite-1
         ENDIF
         WSELECT m_wnummer+3
         WUSE w_inhalt_4
         GO m_snum
         m_text1=&m_fname
         m_laenge=LEN(RTRIM(m_text1))+
                                    INT(m_sbreite/2)
         IF m_spalte < m_laenge-INT(m_sbreite/2)
            m_text=SUBSTR(RTRIM(m_text1), m_spalte,
                                        m_sbreite)
            @ 0,0 SAY m_text
         ELSE
            m_text=""
         ENDIF
         m_whol1=.T.
         DO WHILE m_whol1
            m_inp=0
            DO WHILE m_inp = 0
               m_inp=INKEY()
            ENDDO

            DO CASE

               * Taste F10

               CASE m_inp = -9
                  m_whol1=.F.
                  m_whol=.F.
                  WSELECT m_wnummer
                  WCLOSE m_wnummer+3

               * Kursor rechts

               CASE m_inp = 4
                  WSELECT m_wnummer
                  WCLOSE m_wnummer+3
                  m_spalte=m_spalte+m_sbreite
                  IF m_spalte > m_laenge
                     m_spalte=1
                  ENDIF

               * Kursor nach links

               CASE m_inp = 19
                  WSELECT m_wnummer
                  WCLOSE m_wnummer+3
                  m_spalte=m_spalte-m_sbreite
                  IF m_spalte < 1
                     m_spalte=(INT(m_laenge/
                           m_sbreite)-1)*m_sbreite+1
                  ENDIF

               * Kursor nach oben

               CASE m_inp = 5
                  WSELECT m_wnummer
                  WCLOSE m_wnummer+3
                  m_zeile=m_zeile-1
                  IF m_zeile < 1
                     m_zeile=m_snummax
                     GO BOTTOM
                  ELSE
                     SKIP -1
                  ENDIF

               * Kursor nach unten

               CASE m_inp = 24
                  WSELECT m_wnummer
                  WCLOSE m_wnummer+3
                  m_zeile=m_zeile+1
                  IF m_zeile > m_snummax
                     m_zeile=1
                     GO TOP
                  ELSE
                     SKIP
                  ENDIF

               * Obernahme des Fensterinhaltes in
               * die Variable m_inhalt durch <ET>

               CASE m_inp = 13
                  m_inhalt=m_text
                  m_whol1=.F.
```

```
            m_whol=.F.
            WSELECT m_wnummer
            WCLOSE m_wnummer+3

      ENDCASE

      IF m_inp = 4 .OR. m_inp = 5 .OR.
                    m_inp = 19 .OR. m_inp = 24
         IF m_color
            WSET WINDOW w_inhalt_4 TO m_izeile+
                  m_zeile, m_ispalte+m_spalte,
                  m_izeile+m_zeile, m_ispalte+
                           m_spalte+m_sbreite-1
               COLOR &m_farbe_11, &m_farbe_12,
                                  &m_farbe_13
         ELSE
            WSET WINDOW w_inhalt_4 TO m_izeile+
                  m_zeile, m_ispalte+m_spalte,
                  m_izeile+m_zeile, m_ispalte+
                           m_spalte+m_sbreite-1
         ENDIF
         WSELECT m_wnummer+3
         WUSE w_inhalt_4
         m_text1=&m_fname
         m_laenge=LEN(RTRIM(m_text1))+
                                 INT(m_sbreite/2)
         IF m_spalte < m_laenge-
                                 INT(m_sbreite/2)
            m_text=SUBSTR(RTRIM(m_text1),
                           m_spalte, m_sbreite)
            @ 0,0 SAY m_text
         ELSE
            m_text=""
         ENDIF
      ENDIF
      m_inp=0
   ENDDO
ENDDO
ELSE

   * Mehr als eine Seite

   m_whol=.T.
   DO WHILE m_whol
      WDISPLAY m_wnummer
      @ 0,0 SAY m_titel
      DISPLAY OFF NEXT &m_szanz
      m_snum1=m_snum
      m_snum=m_snum+m_zanz-1
      m_snum2=m_snum
      GO m_snum1

      * von erster Seite aus vorwärts blättern

      IF m_snum < 2*m_zanz-1
         WSELECT m_wnummer+2
         WDISPLAY m_nummer+2
         @ 0,0 SAY "Seite"+STR(m_seite,2)+";Satz"+
               STR(m_snum1,2)+"-"+STR(m_snum2,2)+
               ";Blättern: PgDn;exit: F10;Balken:
                           Cursor;Auswahl: ET"
         @ 0,m_zlaenge-1 SAY ""

         IF m_color
            WSET WINDOW w_inhalt_4 TO m_izeile+
                     m_zeile, m_ispalte+m_spalte,
                     m_izeile+m_zeile, m_ispalte+
                              m_spalte+m_sbreite-1
                  COLOR &m_farbe_11, &m_farbe_12,
                                     &m_farbe_13
         ELSE
            WSET WINDOW w_inhalt_4 TO m_izeile+
                     m_zeile, m_ispalte+m_spalte,
                     m_izeile+m_zeile, m_ispalte+
                              m_spalte+m_sbreite-1
         ENDIF
         WSELECT m_wnummer+3
         WUSE w_inhalt_4
         m_text1=&m_fname
         m_laenge=LEN(RTRIM(m_text1))+
                                 INT(m_sbreite/2)
         m_text=SUBSTR(RTRIM(m_text1), m_spalte,
                                    m_sbreite)
         @ 0,0 SAY m_text
         m_whol1=.T.
         DO WHILE m_whol1
            m_inp=0
            DO WHILE m_inp = 0
               m_inp=INKEY()
            ENDDO

            DO CASE

               * Taste F10

               CASE m_inp = -9
                  m_whol=.F.
                  m_whol1=.F.
                  WSELECT m_wnummer
                  WCLOSE m_wnummer+3

               * Taste PgDn

               CASE m_inp = 3
                  GO m_snum2
                  SKIP
                  m_snum1=m_snum
                  m_snum=m_snum+1
                  m_snum2= m_snum
                  m_seite=m_seite+1
                  m_zeile=1
                  m_whol1=.F.
```

```
      WSELECT m_wnummer
      WCLOSE m_wnummer+3

   * Kursor rechts

   CASE m_inp = 4
      WSELECT m_wnummer
      WCLOSE m_wnummer+3
      m_spalte=m_spalte+m_sbreite
      IF m_spalte ) m_laenge-
                        INT(m_sbreite/2)
         m_spalte=1
      ENDIF

   * Kursor nach links

   CASE m_inp = 19
      WSELECT m_wnummer
      WCLOSE m_wnummer+3
      m_spalte=m_spalte-m_sbreite
      IF m_spalte ( 1
         m_spalte=(INT(m_laenge/
             m_sbreite)-1)*m_sbreite+1
      ENDIF

   * Kursor nach oben

   CASE m_inp = 5
      WSELECT m_wnummer
      WCLOSE m_wnummer+3
      IF m_zeile ( 1
         m_zeile=m_zanz
         GO m_snum2
      ELSE
         SKIP -1
      ENDIF

   * Kursor nach unten

   CASE m_inp = 24
      WSELECT m_wnummer
      WCLOSE m_wnummer+3
      m_zeile=m_zeile+1
      IF m_zeile (=  m_zanz
         SKIP
      ELSE
         m_zeile=1
         GO m_snum1
      ENDIF

   * Obernahme des Fensterinhaltes in
   * die Variable m_inhalt durch (ET)

   CASE m_inp = 13
      m_inhalt=m_text
      m_whol1=.F.
      m_whol=.F.
      WSELECT m_wnummer
      WCLOSE m_wnummer+3

   * beliebige andere Taste

   OTHERWISE
      GO m_snum2
      m_snum=m_snum-m_zanz+1
      SKIP -m_zanz+1
      m_whol1=.F.
      WSELECT m_wnummer
      WCLOSE m_wnummer+3

ENDCASE

IF m_inp = 4 .OR. m_inp = 5 .OR.
          m_inp = 19 .OR. m_inp = 24
   IF m_color
      WSET WINDOW w_inhalt_4 TO
          m_izeile+m_zeile, m_ispalte+
           m_spalte, m_izeile+m_zeile,
        m_ispalte+m_spalte+m_sbreite-1
       COLOR &m_farbe_11, &m_farbe_12,
                          &m_farbe_13
   ELSE
      WSET WINDOW w_inhalt_4 TO
          m_izeile+m_zeile, m_ispalte+
           m_spalte, m_izeile+m_zeile,
        m_ispalte+m_spalte+m_sbreite-1
   ENDIF
   WSELECT m_wnummer+3
   WUSE w_inhalt_4
   m_text1=&m_fname
   m_laenge=LEN(RTRIM(m_text1))+
                        INT(m_sbreite/2)
   m_text=SUBSTR(RTRIM(m_text1),
                    m_spalte, m_sbreite)
   § 0,0 SAY m_text
ENDIF
m_inp=0
ENDDO
ELSE

   * von letzter Seite aus rückwärts
   * blättern

   IF m_snum )= m_snummax
      m_snum2=m_snummax
      WSELECT m_wnummer+2
      WDISPLAY m_nummer+2
      § 0,0 SAY "Seite"+STR(m_seite, 2)+";
                 Satz"+STR(m_snum1, 2)+"-"+
                STR(m_snum2, 2)+";Blättern:
                    PgTp;exit: F10;Balken:
                      Cursor;Auswahl: ET"
```

```
@ 0,m_zlaenge-1 SAY ""
IF m_color
   WSET WINDOW w_inhalt_4 TO
         m_izeile+m_zeile, m_ispalte+
          m_spalte, m_izeile+m_zeile,
        m_ispalte+m_spalte+m_sbreite-1
       COLOR &m_farbe_11, &m_farbe_12,
                          &m_farbe_13
ELSE
   WSET WINDOW w_inhalt_4 TO
         m_izeile+m_zeile, m_ispalte+
          m_spalte, m_izeile+m_zeile,
        m_ispalte+m_spalte+m_sbreite-1
ENDIF
WSELECT m_wnummer+3
WUSE w_inhalt_4
m_texti=&m_fname
m_laenge=LEN(RTRIM(m_texti))+
                        INT(m_sbreite/2)
IF m_spalte < m_laenge-
                        INT(m_sbreite/2)
   m_text=SUBSTR(RTRIM(m_texti),
                  m_spalte, m_sbreite)
   @ 0,0 SAY m_text
ELSE
   m_text="L"
ENDIF
m_whoii=.T.
DO WHILE m_whoii
   m_inp=0
   DO WHILE m_inp = 0
      m_inp=INKEY()
   ENDDO

   DO CASE

      * Taste F10

      CASE m_inp = -9
         m_whoi=.F.
         m_whoii=.F.
         WSELECT m_wnummer
         WCLOSE m_wnummer+3

      * Taste PgTp

      CASE m_inp = 18
         m_snum2=m_snum
         m_snum=m_snum-2*m_zanz+1
         m_snumi= m_snum
         m_seite=m_seite-1
         m_zeile=1
         GO m_snum
         m_whoii=.F.
         WSELECT m_wnummer
         WCLOSE m_wnummer+3

      * Kursor rechts

      CASE m_inp = 4
         WSELECT m_wnummer
         WCLOSE m_wnummer+3
         m_spalte=m_spalte+m_sbreite
         IF m_spalte > m_laenge-
                     INT(m_sbreite/2)
            m_spalte=1
         ENDIF

      * Kursor nach links

      CASE m_inp = 19
         WSELECT m_wnummer
         WCLOSE m_wnummer+3
         m_spalte=m_spalte-m_sbreite
         IF m_spalte < 1
            m_spalte=(INT(m_laenge/
             m_sbreite)-1)*m_sbreite+1
         ENDIF

      * Kursor nach oben

      CASE m_inp = 5
         WSELECT m_wnummer
         WCLOSE m_wnummer+3
         m_zeile=m_zeile-1
         IF m_zeile < 1
            IF "L" $ m_text
               m_zeile=m_snummax-
                              m_snumi
               GO BOTTOM
               SKIP -1
            ELSE
               m_zeile=m_snummax-
                           m_snumi+1
               GO BOTTOM
            ENDIF
         ELSE
            SKIP -1
         ENDIF
         m_text=""

      * Kursor nach unten

      CASE m_inp = 24
         WSELECT m_wnummer
         WCLOSE m_wnummer+3
         m_zeile=m_zeile+1
         IF ((m_zeile <= m_snummax-
               m_snumi+1) .AND. (.NOT.
                     ("L" $ m_text)))
            SKIP
         ELSE
            m_zeile=1
```

```
            GO m_snum1
         ENDIF
         m_text=""

      * Übernahme des Fensterinhaltes
      * in die Variable m_inhalt durch
      * (ET)

      CASE m_inp = 13
         m_inhalt=m_text
         m_whol=.F.
         m_whol1=.F.
         WSELECT m_wnummer
         WCLOSE m_wnummer+3

      * beliebige andere Taste

      OTHERWISE
         m_snum=m_snum-m_zanz+1
         GO m_snum
         m_whol1=.F.
         WSELECT m_wnummer
         WCLOSE m_wnummer+3
   ENDCASE
   IF m_inp = 4 .OR. m_inp = 5 .OR.
         m_inp = 19 .OR. m_inp = 24
      IF m_color
         WSET WINDOW w_inhalt_4 TO
          m_1zeile+m_zeile, m_1spalte+
           m_spalte, m_1zeile+m_zeile,
        m_1spalte+m_spalte+m_sbreite-1
       COLOR &m_farbe_11, &m_farbe_12,
                          &m_farbe_13
      ELSE
         WSET WINDOW w_inhalt_4 TO
          m_1zeile+m_zeile, m_1spalte+
           m_spalte, m_1zeile+m_zeile,
        m_1spalte+m_spalte+m_sbreite-1
      ENDIF
      WSELECT m_wnummer+3
      WUSE w_inhalt_4
      m_text1=&m_fname
      m_laenge=LEN(RTRIM(m_text1))+
                        INT(m_sbreite/2)
      IF m_spalte < m_laenge-
                        INT(m_sbreite/2)
         m_text=SUBSTR((RTRIM
                (m_text1)), m_spalte,
                            m_sbreite)
         § 0,0 SAY m_text
      ELSE
         m_text="L"
      ENDIF
   ENDIF
   m_inp=0
ENDDO
```

```
ELSE

   * von einer beliebigen Seite aus
   * vorwärts oder rückwärts blättern

   WSELECT m_wnummer+2
   WDISPLAY m_nummer+2
   § 0,0 SAY "Seite"+STR(m_seite, 2)+
            ";Satz"+STR(m_snum1, 2)+"-"+
        STR(m_snum2, 2)+";Blättern: PgTp,
          PgDn;exit: F10;Balken: Cursor;
                              Auswahl: ET"
   § 0,m_zlaenge-1 SAY ""

   IF m_color
      WSET WINDOW w_inhalt_4 TO
             m_1zeile+m_zeile, m_1spalte+
              m_spalte, m_1zeile+m_zeile,
           m_1spalte+m_spalte+m_sbreite-1
          COLOR &m_farbe_11, &m_farbe_1
                              &m_farbe_13
   ELSE
      WSET WINDOW w_inhalt_4 TO
             m_1zeile+m_zeile, m_1spalte+
              m_spalte, m_1zeile+m_zeile,
           m_1spalte+m_spalte+m_sbreite-1
   ENDIF
   WSELECT m_wnummer+3
   WUSE w_inhalt_4
   m_text1=&m_fname
   m_laenge=LEN(RTRIM(m_text1))+
                        INT(m_sbreite/2)
   m_text=SUBSTR(RTRIM(m_text1),
                    m_spalte, m_sbreite)
   § 0,0 SAY m_text
   m_whol1=.T.
   DO WHILE m_whol1

      m_inp=0
      DO WHILE m_inp = 0
         m_inp=INKEY()
      ENDDO

      DO CASE

         * Taste F10

         CASE m_inp = -9
            m_whol=.F.
            m_whol1=.F.
            WSELECT m_wnummer
            WCLOSE m_wnummer+3

         * Taste PgTp

         CASE m_inp = 18
```

```
      GO m_snum2
      SKIP -2*m_zanz+1
      m_snum2=m_snum
      m_snum=m_snum-2*m_zanz+1
      m_snum1= m_snum
      m_seite=m_seite-1
      m_zeile=1
      m_whol1=.F.
      WSELECT m_wnummer
      WCLOSE m_wnummer+3

* Taste PgDn

CASE m_inp = 3
      GO m_snum2
      SKIP
      m_snum1=m_snum
      m_snum=m_snum+1
      m_snum2= m_snum
      m_seite=m_seite+1
      m_zeile=1
      m_whol1=.F.
      WSELECT m_wnummer
      WCLOSE m_wnummer+3

* Kursor rechts

CASE m_inp = 4
      WSELECT m_wnummer
      WCLOSE m_wnummer+3
      m_spalte=m_spalte+m_sbreite
      IF m_spalte > m_laenge-
                    INT(m_sbreite/2)
      m_spalte=1
      ENDIF

* Kursor nach links

CASE m_inp = 19
      WSELECT m_wnummer
      WCLOSE m_wnummer+3
      m_spalte=m_spalte-m_sbreite
      IF m_spalte < 1
         m_spalte=(INT(m_laenge/
          m_sbreite)-1)*m_sbreite+1
      ENDIF

* Kursor nach oben

CASE m_inp = 5
      WSELECT m_wnummer
      WCLOSE m_wnummer+3
      m_zeile=m_zeile-1
      IF m_zeile < 1
         m_zeile=m_zanz
         GO m_snum2
      ELSE
         SKIP -1
      ENDIF

* Kursor nach unten

CASE m_inp = 24
      WSELECT m_wnummer
      WCLOSE m_wnummer+3
      m_zeile=m_zeile+1
      IF m_zeile <= m_zanz
         SKIP
      ELSE
         m_zeile=1
         GO m_snum1
      ENDIF

* Obernahme des Fensterinhaltes
* in die Variable m_inhalt durch
* (ET)

CASE m_inp = 13
      m_inhalt=m_text
      m_whol=.F.
      m_whol1=.F.
      WSELECT m_wnummer
      WCLOSE m_wnummer+3

* beliebige andere Taste

OTHERWISE
      GO m_snum
      m_snum=m_snum-m_zanz+1
      SKIP -m_zanz+1
      m_whol1=.F.
      WSELECT m_wnummer
      WCLOSE m_wnummer+3

ENDCASE
IF m_inp = 4 .OR. m_inp = 5 .OR.
      m_inp = 19 .OR. m_inp = 24
   IF m_color
```

**********

Hinweis:

Wir möchten unseren Lesern mitteilen, daß die edv-aspekte-Ausgabe 4/88

DCP-Software vom Kombinat Robotron

wieder lieferbar ist. Bestellungen richten Sie bitte an unsere Vertriebsabteilung.

**********

```
                    WSET WINDOW w_inhalt_4 TO
                     m_lzeile+m_zeile, m_lspalte+
                      m_spalte, m_lzeile+m_zeile,
                    m_lspalte+m_spalte+m_sbreite-1
                   COLOR &m_farbe_11, &m_farbe_12,
                                      &m_farbe_13
                  ELSE
                    WSET WINDOW w_inhalt_4 TO
                     m_lzeile+m_zeile, m_lspalte+
                      m_spalte, m_lzeile+m_zeile,
                    m_lspalte+m_spalte+m_sbreite-1
                  ENDIF
                  WSELECT m_nummer+3
                  WUSE w_inhalt_4
                  m_text1=&m_fname
                  m_laenge=LEN(RTRIM(m_text1))+
                                   INT(m_sbreite/2)
                  m_text=SUBSTR(RTRIM(m_text1),
                              m_spalte, m_sbreite)
                  § 0,0 SAY m_text
                ENDIF
                m_inp=0
              ENDDO
            ENDIF
          ENDIF
          m_inp=0
        ENDDO
      ENDIF
    ENDIF

* Abschlußorganisation

SELECT &m_select
USE &m_datei
WRESTORE

RETURN
```

## 7. Erzeugen von Befehlsfolgen für Kalkulationsprogramme, Textprogramme und Datenbanksysteme

Komplexe Aufgabenstellungen erfodern im allgemeinen den Einsatz verschiedener Programmsysteme mit unterschiedlicher Kommandosprache. Um den Nutzer solcher komplexer Aufgabenlöungsprozesse von belastender Routinebedienung zu entlasten, ist ein weitgehend automatisierter Programmablauf anzustreben. Im Beispiel 24 wurde bereits demonstriert, wie der automatische Aufruf verschiedener Nutzersysteme mit Hilfe des Dienstprogramms SUBM gestaltet werden kann. Die in diesem Abschnitt aufgeführten Beispiele demonstrieren, wie bereits im Datenbanksystem in Abhängigkeit von den zur Bearbeitung ausgewählten Daten konkrete Kommandofolgen für Kalkulationsprogramme und Textprogramme, aber auch für Datenbanksysteme selbst automatisch generiert werden können. Dabei entstehen Kommandodateien, die in bekannter Weise den betreffenden Nutzerprogrammsystemen in der Kommandozeile des Betriebssystems zur Bearbeitung übergeben werden können. Bei diesen Kommandodateien handelt es sich um ASCII-Textdateien, die zunächst im Datenbanksystem als gewöhnliche Datenbankdateien entwickelt und durch ein abschließendes Kopieren in derartige Textdateien überführt werden.

```
* Beispiel 28:
* Modul zur Fortführung der Datenbereitstellung zur
* Berechnung eines Tilgungsplanes.
*
** MODULE tilgpld1
**
** DESCRIPTION
**
** Die für die Erstellung eines Kredittilgungsplanes
** erforderlichen Berechnungen der Ratenbeträge, der
** progressiven oder degressiven Zinsen, der Rück-
** zahlungstermine der einzelnen Ratenbeträge und
** weiterer Berechnungen erfolgen nicht im Daten-
** banksystem sondern im Kalkulationsprogramm. An-
** schließend werden die erforderlichen Bank- und
** betrieblichen Dokumente erstellt (gegebenenfalls
** mehrsprachig). Die Bearbeitung dieser Arbeitsun-
** terlagen erfolgt nach der Erfassung bzw. Auswahl
** der erforderlichen Daten innerhalb des Daten-
** banksystems automatisch im Kalkulationsprogramm
** bzw. im Kombodruck des Textprogramms. Das vorlie-
** gende Modul stellt dazu die für die betreffenden
** Nutzersysteme notwendigen Kommandodateien auf der
** Grundlage der Erfordernisse der bereitstehenden
** Daten automatisch zusammen. Die Struktur der für
** die Zwischenspeicherung der einzelnen Kommando-
** zeilen erforderlichen Datenbankstrukturen sind
** aus dem Kontext des Programms ersichtlich. Zu
** beachten ist lediglich, daß jede Kommandozeile in
** der betreffenden Datenbank einen Datensatz bil-
** det. Beim Kalkulationsprogramm ist zusätzlich zu
** beacheten, daß Befehle und Daten in der Kommando-
** datei gemischt auftreten, aber unterschiedliche
** Datentypen aufweisen (die Kommandofolge muß als
** Zeichenkette vorliegen, während die Daten als
** numerische Werte übermittelt werden müssen). Das
** macht es erforderlich, eine Datenbankdatei mit
** zwei Feldern vom Typ numerisch und der Länge 12,
** 2 (maximal größter zu übermittelnder Betrag) bzw.
** vom Typ Zeichenkette und der Länge 33 (Längste
** auftretende Kommandofolge) einzurichten. Jeder
** Datensatz enthält nur ausschließlich in einem
** Feld einen Eintrag.
** Beim Kombodruck beachten Sie bitte den unter-
** schiedlichen Aufbau einer Befehlsdatei und einer
```

```
** Parameterdatei. Beide Typen werden im vorliegen-
** den Beispiel behandelt.

* Vorbereitung des Moduls als mögliches Kopfpro-
* gramm

IF FILE("variable.MEM")
   REST FROM variable.MEM
   SET DATE TO &m_datum
   DELE FILE variable.MEM
ENDI

* Zusammenstellung von Kommandofolgen für die Bear-
* beitung der Tilgungspläne

ERAS
?
? "Ich stelle jetzt die Daten für die Berechnung und
                                              den Druck"
? "des gewünschten Tilgungsplanes zusammen."
? "Bitte werden Sie nicht ungeduldig!"
?
* Löschen evt. vorhandener alter Kommandodateien

STOR m_lw1+":tilgplo7.CAL" TO m_datei
IF FILE(m_datei)
   DELE FILE &m_datei
ENDI
STOR m_lw1+":tilgplo6.TXT" TO m_datei
IF FILE(m_datei)
   DELE FILE &m_datei
ENDI
 . . .
STOR m_lw1+":tilgplo2.TXT" TO m_datei
IF FILE(m_datei)
   DELE FILE &m_datei
ENDI
STOR m_lw1+":tilgplo1.TXT" TO m_datei
IF FILE(m_datei)
   DELE FILE &m_datei
ENDI

* Kopplungsdatei zum Textprozessor erstellen und als
* ASCII-Datei bereitstellen (Parameterdatei für Kom-
* bodruck

? "Kopplungsdatei zum Textprozessor"
?
GO m_satznum
SELE SECO
STOR m_lw1+":tilgplhd.$$$" TO m_datei1
STOR m_lw1+":tilgpls2.DBF" TO m_datei2
CREA &m_datei1 FROM &m_datei2
USE &m_datei1
APPE BLANK
REPL elem1 WITH P.m_vnr
REPL elem2 WITH P.m_knr
IF !(P.m_ttp) = 'N'
   REPL elem4 WITH "Exportauftrag Nr.: "+P.m_enr
   REPL elem5 WITH "Rechnungseingang Nr.: "+P.m_ernr
ENDI
REPL elem3 WITH P.m_tlnr
REPL elem8 WITH "    T"
IF !(P.m_fr) = 'J'
   REPL elem6 WITH "Fracht "+TRIM(P.m_we)+": "+
                                        STR(P.m_fh,10,2)
   REPL elem7 WITH "Fracht VM: "+
                                   STR(P.m_fh*P.m_wu,10,2)
ENDI
REPL elem9 WITH STR(P.m_blt,3)
REPL elem10 WITH P.m_we
REPL elem11 WITH m_datnam
REPL elem12 WITH STR(m_satznum,3)
IF !(m_fft) = 'N'
   REPL elem13 WITH 'J'
ELSE
   REPL elem13 WITH 'N'
ENDI
REPL elem14 WITH 'Faelligkeit (+'+STR(m_blt,3)+
                                              ' Tage): '
REPL elem15 WITH 0
REPL elem16 WITH m_ttp
COPY TO &m_datei DELI WITH ,
USE
DELE FILE &m_datei1

* Erzeugung einer Kommandodatei für Kombodruck

STOR m_lw1+":tilgps14.DBF" TO m_datei2
CREA &m_datei1 FROM &m_datei2
USE &m_datei1
APPE BLAN
REPL zeile WITH '.FI '+m_lw+':tilgplt1.CMD'
APPE BLAN
REPL zeile WITH CHR(3)
APPE BLAN
REPL zeile WITH '.FI '+m_lw+':tilgplt2.CMD'
IF !(m_fft) = 'N'
   APPE BLAN

   * CTRL-C zum Blattwechsel

   REPL zeile WITH CHR(3)
   APPE BLAN
   REPL zeile WITH '.FI '+m_lw+':tilgplt3.CMD'
   APPE BLAN
   REPL zeile WITH CHR(3)
   APPE BLAN
   REPL zeile WITH '.FI '+m_lw+':tilgplt4.CMD'
ENDI
STOR m_lw1+":tilgplt5.CMD" TO m_datei
COPY TO &m_datei SDF
```

```
USE
DELE FILE &m_datei1

* Kopplungsdatei zum Tableau-Programm erstellen und
* als ASCII-Datei bereitstellen.
* (Berechnung der Tilgungsraten und Zinsen in ver-
* schiedenen Währungen sowie der Fälligkeitstermine)

? "Kopplungsdatei zum Tableau-Programm"
?
SELE SECO
STOR m_lw1+":tilgplsc.TXT" TO m_datei3
STOR m_lw1+":tilgpls3.DBF" TO m_datei4
IF FILE(m_datei3)
   DELE FILE &m_datei3
ENDI
CREA &m_datei1 FROM &m_datei4
STOR 5 TO m_zeile
USE &m_datei1
APPE BLANK
REPL elem1 WITH "/GM"
APPE BLANK
STOR "/L"+m_lw+":tilgplan,A" TO m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
REPL elem1 WITH "=B"+STR(m_zeile,1)
APPE BLANK
REPL elem2 WITH m_blt
APPE BLANK
IF PKR
   REPL elem2 WITH 1
ELSE
   REPL elem2 WITH 0
ENDI
APPE BLANK
IF !(m_fr) = 'J'
   REPL elem2 WITH 1
ELSE
   REPL elem2 WITH 0
ENDI
APPE BLANK
IF !(m_fft) = 'J'
   REPL elem2 WITH 1
ELSE
   REPL elem2 WITH 0
ENDI
APPE BLANK
REPL elem2 WITH m_fh
APPE BLANK
REPL elem2 WITH m_kh
APPE BLANK
REPL elem2 WITH m_wu
APPE BLANK
REPL elem2 WITH m_zs
APPE BLANK
REPL elem2 WITH VAL($(m_datum,1,2))
APPE BLANK
REPL elem2 WITH VAL($(m_datum,4,2))
APPE BLANK
REPL elem2 WITH VAL($(m_datum,7,4))
APPE BLANK
REPL elem2 WITH m_aber
APPE BLANK
REPL elem2 WITH m_azzr
APPE BLANK
IF m_pkr
   REPL elem2 WITH m_ar
ELSE
   REPL elem2 WITH 0
ENDI
IF .NOT. m_pkr
   APPE BLANK
   REPL elem2 WITH m_pr1
   APPE BLANK
   REPL elem2 WITH m_pr2
    . . .
   APPE BLANK
   REPL elem2 WITH m_pr9
   APPE BLANK
   REPL elem2 WITH m_pr10
ENDI
APPE BLANK
IF !($(m_kkred,2,1)) = 'M'
   STOR "/L"+m_lw+":tilgpld1,A" TO m_datei2
ELSE
   STOR "/L"+m_lw+":tilgpld2,A" TO m_datei2
ENDI
REPL elem1 WITH m_datei2
APPE BLANK
STOR "/S"+m_lw1+":tilgplhd.$$$,PVN23:P32" TO m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
STOR "/L"+m_lw1+":tilgplhd.$$$,A" TO m_datei2
REPL elem1 WITH m_datei2
STOR 0 TO m_i
IF !($(m_kkred,2,1)) = 'M'
   STOR VAL($(m_datum,4,2))+m_aber-m_azzr TO m_monat
ELSE
   STOR VAL($(m_datum,1,2))+m_aber TO m_tag
   STOR VAL($(m_datum,4,2)) TO m_monati
ENDI
DO WHIL m_i < m_az
   APPE BLANK
   REPL elem1 WITH "=Q12"
   APPE BLANK
   REPL elem2 WITH m_i+1
   IF !($(m_kkred,2,1)) = 'M'
      STOR m_monat+m_azzr TO m_monat
   ELSE
      STOR m_monati+INT((m_tag+m_i*m_azzr-0.5)/30) TO
                                                m_monat
   ENDI
```

```
   DO WHIL m_monat > 12
      STOR m_monat-12 TO m_monat
   ENDD
   APPE BLANK
   STOR "/L"+m_lw+":m_datum" TO elem
   DO CASE
      CASE m_monat = 1
         STOR elem+"JAN" TO elem
      CASE m_monat = 2
         STOR elem+"FEB" TO elem
           . . .
      CASE m_monat = 11
         STOR elem+"NOV" TO elem
      CASE m_monat = 12
         STOR elem+"DEZ" TO elem
   ENDC
   STOR elem+",PQ23:R35,Q23" TO elem
   REPL elem1 WITH elem
   STOR m_i+1 TO m_i
ENDD
APPE BLANK
STOR "/S"+m_lw1+":tilgplo7,A" TO m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
REPL elem1 WITH "/BN12:U59"
APPE BLANK
STOR "/L"+m_lw1+":tilgplo7,PN23:P32,N23,V" TO
                                              m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
STOR "/L"+m_lw1+":tilgplo7,PS23:U32,S23,V" TO
                                              m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
IF !(PRODEG) = 'D'
   IF !($(m_kkred,2,1)) = 'M'
      STOR "/L"+m_lw+":tilgplr1,A" TO m_datei2
   ELSE
      STOR "/L"+m_lw+":tilgplr2,A" TO m_datei2
   ENDI
ELSE
   IF !($(m_kkred,2,1)) = 'M'
      STOR "/L"+m_lw+":tilgplr3,A" TO m_datei2
   ELSE
      STOR "/L"+m_lw+":tilgplr4,A" TO m_datei2
   ENDI
ENDI
REPL elem1 WITH m_datei2
APPE BLANK
STOR "/S"+m_lw1+":tilgplhd.$$$,DA" TO m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
STOR "/L"+m_lw1+":tilgplhd.$$$,PC23:M32,C23,V" TO
                                              m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
STOR "/L"+m_lw+":tilgplkr,A" TO m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
STOR "/ODAA4:AH16,D"+m_lw1+":tilgplo2.TXT" TO
                                              m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
STOR "/ODAA4:AP16,D"+m_lw1+":tilgplo3.TXT" TO
                                              m_datei2
REPL elem1 WITH m_datei2
APPE BLANK
STOR "/ODAI4:AM4,D"+m_lw1+":tilgplo4.TXT" TO m_datei2
REPL elem1 WITH m_datei2
IF !(m_fft) = 'N'
   ? "Teilaufgabe Tilgungsplan Fracht"
   ?
   . . .
ENDI
APPE BLANK
REPL elem1 WITH "/QY"
COPY TO &m_datei3 SDF
USE
DELE FILE &m_datei1

* Kopplungsdatei für die wiederholte Bearbeitung der
* Teiltilgungspläne

IF AUSWAHL = '3'
   ? "Kopplungsdatei für die Teiltilgungspläne"
   ?
    . . .
ENDI
SELE PRIM
ERAS
? "Die auf Laufwerk"+inv+m_lw1+norm+"erzeugte Datei"+
                          inv+"tilgplsc.TXT"+norm+"ist"
? "mit dem"+inv+"Tableau-Programm"+norm+"weiter zu
                                              bearbeiten."
? "Dazu wird das laufende Datenbank-Programm
                                          'TILGUNGSPLAN' "
? "abgeschlossen und das Tableau-Programm gestartet!"
? "Nach Abschlusz des Tableau-Programms wird das"+
                                 inv+"Textprogramm"+norm
? "TP gestartet, um die erforderlichen Exemplare des"
? "berechneten Tilgungsplanes auszudrucken. Danach
                                              wird erneut"
? "das Datenbankprogrammsystem erreicht und das
                                              Programm 4"
? "im Hauptmenü aufgerufen. Darauf folgend kann ein"
                                    +inv+"weiterer"+norm
? inv1+"Tilgungsplan"+norm+"bearbeitet oder die"+inv+
                 "Arbeit abgeschlossen"+norm+"werden."
?
? "Wenn Sie den Text gelesen haben, drücken Sie die"
? "Taste"+inv+'ET'+norm+"!"
?
```

```
STOR 0 TO J
DO WHIL J < 2*TIME
   STOR J+1 TO J
ENDD
SAVE TO variable.MEM
IF AR = 10
   QUIT TO 'A:', '&m_lw.:SUBMIT &m_lw.:tilgplt1.SUB
                                        &m_lw &m_lw1'
ELSE
   QUIT TO 'A:', '&m_lw.:SUBMIT &m_lw.:tilgplt3.SUB
                                        &m_lw &m_lw1'
ENDI

RETU
```

*** Beispiel 29:**

** Modul zur wahlweisen Zusammenstellung von Auswahl-*
** bedingungen.*

```
*
** MODULE tilgplk2
**
```

**** DESCRIPTION**

```
**
```

*** Kontroll-, Analyse- und Rechercheaufgaben erfor-*
*** dern sehr differenzierte Sichten auf einen Daten-*
*** bestand. Im allgemeinen werden diese Sichten*
*** durch geeignete logische Bedingungen beschrieben.*
*** Sie setzen sich aus elementaren Entscheidungen*
*** auf der Grundlage der erfaßten einzelnen Daten-*
*** elemente zusammen, die entweder mit dem logischen*
*** UND oder mit dem logischen ODER verbunden werden.*
*** Das vorliegende Modul demonstriert am Beispiel*
*** der Auswahl von Tilgungsplänen den variablen*
*** Aufbau eines Recherchekriteriums. Schlüssel- oder*
*** andere Datenbankfelder werden mit vorgebbaren*
*** Werten verglichen. Dieser Vergleich wird als*
*** textlicher Ausdruck mit den sprachlichen Aus-*
*** drucksmitteln des Datenbanksystems dargestellt.*
*** Die so entstehenden Programmtextfragmente werden*
*** in geeigneten Variablen gesammelt. Sie können im*
*** Rahmen einer Macroanweisung im weiteren Programm-*
*** verlauf unmittelbar ausgewertet werden. Auf die-*
*** sem Wege ist ein außerordentlich flexibler Zu-*
*** griff zum Beispiel zu den Datenbeständen einer*
*** Datenbank möglich.*

```
* Anfangswerte setzen

STOR F TO m_beding
STOR ' ' TO m_bed

* Auswahl der Fälligkeitstermine

STOR T TO m_error
DO WHIL m_error
   ERAS
   ?
   ? "  "+inv+" *** Auswahl von Fälligkeiten von
                                Tilgungsraten *** "+norm
   ? "                    "+inv+" - Kriterienübersicht
                                                - "+norm
   ?
   ? inv1+"(1)"+norm+" Kontrolle im vorausliegenden
                                          Zeitabschnitt"
   ? inv1+"(2)"+norm+" Kontrolle im zurückliegenden
                                          Zeitabschnitt"
   ? inv1+"(3)"+norm+" Kontrolle in einem beliebig
                                vorgebbaren Zeitabschnitt"
   ?
   ? inv1+"Wählen Sie ein Auswahlkriterium:"+norm
   ?
   WAIT TO m_auswahl
   IF m_auswahl < '1' .OR. m_auswahl > '3'
      ERAS
      ?
      ? blil+"A c h t u n g!"+norm+" Sie haben eine
                               nicht existierende Aufgabe"
      ? "gewählt."
      ? "Wiederholen Sie bitte Ihre Angaben."
      ?
      STOR 0 TO m_j
      DO WHIL m_j < m_time
         STOR m_j+1 TO m_j
      ENDD
   ELSE
      STOR F TO m_error
   ENDI

   * Auswahl der Tilgungsratentypen

   STOR '00000/00000' TO m_enri
   STOR '00000' TO m_mrnri
   ERAS
   ?
   ? "    "+inv+" *** Alternative Einschränkung der
                                    Kontrolle *** "+norm
   ? "   Nicht benoetigte Informationen koennen
                                    übergangen werden."
   ?
   ? inv1+"(1)"+norm+" auf Tilgungsraten zum
                                              Gesamtkredit"
   ? inv1+"(2)"+norm+" auf Tilgungsraten zum
                                              Frachtkredit"
   ? inv1+"(3)"+norm+" auf Tilgungsraten zum
                                              Warenkredit"
   ? inv1+"(4)"+norm+" auf Tilgaungsraten zum
                                          Teiltilgungsplan:"
   @ 9,5 SAY "Exportauftragsnummer " GET m_enri PICT
                                              'XXXXX/XXXXX'
   @ 9,39 SAY "Markrechnungsnummer " GET m_mrnri PICT
```

```
                                                'XXXXX'
   READ
   § 10,1 SAY ' '
   ? inv1+"Wählen Sie Ihre gewünschten
                                  Tilgungsraten."+norm
   ?
   WAIT TO m_tpt
   IF m_tpt < '1' .OR. m_tpt > '4' .OR. (m_tpt = '4'
         .AND. m_enri = '00000/00000' .AND. m_mrnri =
                                                '00000')
      ERAS
      ?
      ? bli1+"A c h t u n g!"+norm+" Sie haben eine
                           nicht existierende Aufgabe"
      ? "gewählt."
      ? "Wiederholen Sie bitte Ihre Angaben."
      ?
      STOR 0 TO m_j
      DO WHIL m_j < m_time
         STOR m_j+1 TO m_j
      ENDD
   ELSE
      STOR  F TO m_error
   ENDI
ENDD
STOR ' ' TO m_bedp
STOR ' ' TO m_beds
DO CASE
   CASE m_tpt = '1'
      STOR "!(S.m_fr) = 'N' .AND. S.m_tpa = 0 " TO
                                                m_beds
   CASE m_tpt = '2'
      STOR "!(S.m_fr) = 'J' .AND. S.m_tpa = 0 " TO
                                                m_beds
   CASE m_tpt = '3'
      STOR "S.m_tpa = 0 " TO m_beds
   CASE m_tpt = '4'
      IF m_enri <> '00000/00000' .AND. m_mrnri <>
                                                '00000'
         STOR "P.m_enr $ m_enri .AND. P.m_mrnr $
                                    m_mrnri TO m_bedp
      ELSE
         IF m_enri <> '00000/00000'
            STOR "P.m_enr $ m_enri TO m_bedp
         ELSE
            IF m_mrnri <> '00000'
               STOR P.m_mrnr $ m_mrnri TO m_bedp
            ENDI
         ENDI
      ENDI
      STOR "S.m_tpa > 0 " TO m_beds
ENDC
ERAS
?
? "Wollen Sie die Kontrolle auf"+inv+"ausgewählte
                                  Tilgungsraten"+norm
? "beschränken?"+inv+"J/N"+norm
?
WAIT TO ANTWORT
IF !(ANTWORT) = 'J'
   STOR T TO m_beding
   STOR T TO WHOLUNG
   DO WHIL WHOLUNG
      ERAS
      STOR ' ' TO m_bed
      STOR '00000' TO m_lnri
      STOR '000000000000' TO m_vnri
      STOR '000000000000000000000000' TO m_knri
      STOR 'J' TO m_kkredi
      STOR '000' TO m_tlnri
      § 2,1 SAY BLK1+"Geben Sie Ihre
                        Auswahlbedingungen vor."+norm
      § 3,1 SAY "Die nicht zutreffenden Bedingungen
                         können übergangen werden."
      § 5,1 SAY inv1+"Auswahl für ein Land oder eine
                                  Ländergruppe"+norm
      § 6,1 SAY inv1+"Auswahl für eine
                                 Vertragsnummer"+norm
      § 7,1 SAY inv1+"Auswahl für eine LC-Nummer"+
                                                norm
      § 8,1 SAY inv1+"Auswahl nach kurzfristigen (J)
                                               oder"+norm
      § 9,1 SAY inv1+"langfristigen (N) Krediten"+
                                                norm
      § 10,1 SAY inv1+"Auswahl für eine
                          Teillieferungsnummer"+norm
      § 5,52 GET m_lnri PICT '99999'
      § 6,39 GET m_vnri PICT '999999999999'
      § 7,33 GET m_knri PICT
                          'XXXXXXXXXXXXXXXXXXXXXXXX'
      § 9,33 GET m_kkredi PICT 'X'
      § 10,45 GET m_tlnri PICT '999'
      READ
      IF '00000' <> m_lnri
         STOR m_bed+'P.LNR $ m_lnri .AND. ' TO m_bed
      ENDI
      IF '000000000000' <> m_vnri
         STOR m_bed+'P.m_vnr $ m_vnri .AND. ' TO
                                                m_bed
      ENDI
      IF '000000000000000000000000' <> m_knri
         STOR m_bed+'P.m_knr $ m_knri .AND. ' TO
                                                m_bed
      ENDI
      IF ' ' <> m_kkredi
         STOR m_bed+'$(P.m_kkred,1,1) $ m_kkredi
                                    .AND. ' TO m_bed
      ENDI
      IF '000' <> m_tlnri
         STOR m_bed+'P.m_tlnr $ m_tlnri .AND. ' TO
                                                m_bed
      ENDI
```

```
      IF ' ' = m_bed
         ERAS
         ?
         ? bli1+"A c h t u n g!"+norm+" Sie haben
                  k e i n e  Bedingungen gestellt!"
         ?
         ? "War das Ihre Absicht?"+inv+"J/N"+norm
         ?
         WAIT TO ANTWORT
         IF !(ANTWORT) = 'J'
            ?
            ? inv1+"Es werden keine Bedingungen
                                 berücksichtigt!"+norm
            STOR F TO WHDLUNG
            STOR F TO m_beding
            STOR ' ' TO m_bed
         ELSE
            ?
            ? "Geben Sie Ihre Bedingungen jetzt vor."
         ENDI
         STOR 0 TO m_j
         DO WHIL m_j < m_time
            STOR m_j+1 TO m_j
         ENDD
      ELSE
         IF LEN(m_bed) > 6
            IF $($(m_bed,LEN(m_bed)-6,7),' .AND. ') =
                                                   1
               STOR $(m_bed,1,LEN(m_bed)-7) TO m_bed
               STOR F TO WHDLUNG
            ENDI
         ENDI
      ENDI
   ENDD
ENDI
IF m_bed <> ' '
   STOR m_bedp+'.AND. '+m_bed TO m_bedp
ENDI
RELE m_lnri, m_vnri, m_knri, m_kkredi, m_tlnri

RETU
```

*** Beispiel 30:**
** Die Funktion INKEY() als Beispiel für die Einbin-*
** dung eines ASSEMBLER-Programms in ein Datenbank-*
** programm (siehe auch die Beispiele 18 und 19.)*

```
*
** MODULE inkey
**
** DESCRIPTION
**
** Dieser Modul führt eine Tastaturabfrage aus. Es
** liefert als Ergebnis das entsprechende Zeichen.
** Es erfolgt kein Echo.
**
** PARAMETERS
**
** CHARACTER OUTPUT variable m_charin   "gedrückte
**                                          Taste"
**
** ENDPAR
**
** ENDMOD

*  Abfrage Tastatur

*  inc  hl
*  push hl
*M:ld   c,06h
*  ld   e,0ffh
*  call 5          ; BDOS-Ruf: direkte Consol-Eingabe
*  cp   a,0
*  jr   z,M      ; Sprung, wenn keine Taste gedrückt
*  pop  hl
*  ld   (hl),a ; gelesene Taste in Variable speiche
*  ret
*
POKE 42752,35,229,14,6,30,255,205,5,0,254,0,40,245,
                                              225,119,201
SET CALL TO 42752
STORE ' ' TO m_charin
CALL m_charin

RETURN
```

**********

WICHTIGER HINWEIS

für unsere Leser außerhalb der Deutschen Demokratischen Republik

Wir erlauben uns, Sie rechtzeitig daran zu erinnern, Ihre Abonnementsbestellung vorzunehmen bzw. zu erneuern.

Die Bezugsmöglichkeiten finden Sie in dieser Ausgabe im Impressum auf der Seite 1.

Wir würden uns sehr freuen, wenn Sie unsere Zeitschrift weiterempfehlen und wir Sie auch künftig als Leser begrüßen könnten.

Die Redaktion

**********

# MÖGLICHKEITEN DER GESTALTUNG VON MENÜANGEBOTEN

## 1. Menüangebote mit Nummernauswahl

(siehe Programmierbeispiel 14)

| |
|---|
| (1) Erfassung von Stammdaten<br>(2) Erfassung von ...daten<br>(3) Kontrolle von ...<br>(4) Analyse von ...<br>...<br>(9) Dienst- und Havarieaufgaben<br>(9) Informationsdienste<br>(0) Abschlußorganisation |
| Wählen Sie die Nummer Ihrer Aufgabe: |

## 2. Menüangebote mit Lichtbalkenmarkierung

(siehe Programmierbeispiel 15)

| |
|---|
| Erfassung von Stammdaten<br>Erfassung von ...daten<br>Kontrolle von ...<br>Analyse von ...<br>...<br>Dienst- und Havarieaufgaben<br>Informationsdienste<br>Abschlußorganisation |
| Balken: Cursor hoch/tief; Aufgabenbestätigung: ET |

## 3. Menüangebote mit Fenstertechnik und Spezifizierungsmöglichkeiten

| Erfassung | Kontrolle | Analyse | ... | Dienst/ Havarie | Information |
|---|---|---|---|---|---|
| xxxxxxxx<br>xxxxxx<br>xxx<br>xxxx | xxxxxxxx<br>xxxx<br>xxxxxx | xxxxxxx<br>xxxx<br>xxxxxxx<br>xxxx<br>xxx | | xxxxx<br>xxx<br>xxxx | xxxxxx<br>xxxxxxxx<br>xxx |
| | | | | | |

aspekte **blickpunkt**



# FENSTERTECHNIKEN UND IHRE NUTZUNG

## 1. Grundmuster für die Aufteilung des Bildschirmes in Fenster für die Dialogführung

(siehe Programmierbeispiel 12)

| | |
|---|---|
| Statusfenster | max. 1 Zeile |
| Informationsfenster | max. 4 Zeilen |
| Arbeitsfenster | max. 12 Zeilen |
| Mentorfenster | max. 2 Zeilen |

## 2. Anwendung der Fenstertechnik im Programmsystem „Projektierung“

(siehe Programmierbeispiel 26)

Quantitative Leistungserfassung mit Kostenbewertung

Objektnummer xxxx-xx-xxxxx/Ox Kostenstelle xxxxxxxxxx
Objektbezeichnung ELN-Hauptnr. xxxxx Kostenträger xxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxx kum. Objektwert xxxxxxxxxxx.xx
Projekttyp xxxxxxxxxxxxxx Artikelwert xxxxxxxxxxx.xx

| ELN-Nummer | Artikelbezeichnung | ME | Preis/ME | Menge |
|---|---|---|---|---|
| xxxxxxxxxx | xxxxxxxxxxxxxxxxxxx | xx | xxxxxx.xx | xxxxxx.xxx |
| xxxxxxxxxx | xxxxxxxxxxxxxxxxxxx | xx | xxxxxx.xx | xxxxxx.xxx |
| xxxxxxxxxx | xxxxxxxxxxxxxxxxxxx | xx | xxxxxx.xx | xxxxxx.xxx |
| xxxxxxxxxx | xxxxxxxxxxxxxxxxxxx | xx | xxxxxx.xx | xxxxxx.xxx |

nächster Artikel Korrekturhinweise
kein weiterer Artikel

## 3. Einsatz der Fenstertechnik bei der Erfragung einer Datei

(siehe Programmbeispiele 21 und 22)

xxxxxxxxxxxxxxxxxxxxxx

xxxxxxxx.xxx xxxxxxxx.xxx xxxxxxxx.xxx xxxxxxxx.xxx
xxxxxxxx.xxx xxxxxxxx.xxx xxxxxxxx.xxx xxxxxxxx.xxx
xxxxxxxx.xxx xxxxxxxx.xxx xxxxxxxx.xxx

Wählen Sie Ihre Datenbank aus oder
geben Sie einen Datenbanknamen vor

Auswahl aus Dateiverzeichnis weiter
Abbruch der Aufgabe

ISSN DDR 0232-6833 · edv-aspekte · Berlin 8 (1989) 3 · S. 1–64