

# 9

## Fallstudien

In den folgenden Unterabschnitten wird die Makro-Erstellung/Programmierung anhand mehrerer Beispiele verdeutlicht. Sie stehen zum Teil nicht in unmittelbarem Zusammenhang mit der Textverarbeitung mit WordPerfect. Sie dienen lediglich dazu, anhand leichtverständlicher Anwendungen (z.B. Spiele) die Makro-Programmierung und das Zusammenspiel der Makro-Befehle zu verdeutlichen, und um Ihnen Anregungen für die eigene Programmierung zu geben. Die wichtigsten Zeilen eines Makros sind am linken Rand mit einer Zahl gekennzeichnet (diese Zahl darf bei der Programmierung nicht eingegeben werden). In den Erklärungen wird auf diese Zahl Bezug genommen.

Die ersten beiden Beispiele sind mit voller Absicht einfach, klein und übersichtlich gehalten, um Anfängern der Makro-Programmierung einen leichten Einstieg zu ermöglichen. Bei den Kalender-Makros werden dann alle Register der Makro-Programmierung gezogen. Nutzen Sie die dort verwendeten Befehle bzw. Routinen als Muster für eigene Anwendungen.

### 9.1 Einmaleins errechnen

**Aufgabenstellung:** Durch die Eingabe zweier Zahlen (Unter- und Obergrenze) werden die Zahlenkolonnen des ausgewählten Einmaleins am Bildschirm angezeigt. Wenn gewünscht, können Sie diese ausdrucken. Wenn Sie z.B. als Untergrenze 5 und als Obergrenze 10 eingeben, wird Ihnen das Einmaleins der Zahlen 5 bis einschließlich 10 errechnet und am Bildschirm angezeigt. Beenden Sie das Makro bei Bedarf während der Ausführung durch Klicken auf [Abbrechen]. Das Ergebnis wird in einem neuen Textfenster angezeigt.

**Schwerpunkte:** Schleifenprogrammierung, Zuordnungen, Text ausgeben.

#### Beispiel

1	*	5	=	5
2	*	5	=	10
3	*	5	=	15

4	*	5	=	20
5	*	5	=	25
6	*	5	=	30
7	*	5	=	35
8	*	5	=	40
9	*	5	=	45
10	*	5	=	50

### Makrodefinition: 1MALEINS.WCM

```

1      Application      (A1; "WordPerfect"; Default; "DE")
2      OnCancel         (Abbruch)
3      FileNew          ()
4      Display          (On!)
5      TabSet           (Relative!; {4.5403c; TabRight!; 5.5393c; TabLeft!; 7.5396c;
                        TabRight!; 8.5408c; TabLeft!; 10.5389c; TabRight!})
6      Label            (Anfang)
7      GetNumber        (Anfang;"Bitte Untergrenze eingeben";"1 x 1 errechnen")
8      GetNumber        (Ende;"Bitte Obergrenze eingeben";"1 x 1 errechnen")
9      If               (Anfang>Ende)
10     Prompt           (" F e h l e r";"Untergrenze ist größer als Obergrenze";1;;)
11     Beep
12     Wait             (30)
13     EndPrompt
14     Go               (Anfang)
15     EndIf
16     Zähler=          1
17     While             (Anfang<Ende+1)
18     While             (Zähler < 11)
19     Ergebnis=        (Zähler*Anfang)
20     Tab              ()
21     Type              (Zähler)
22     Tab              ()
23     Type              ("*")
24     Tab              ()
25     Type              (Anfang)
26     Tab              ()
27     Type              ("=")
28     Tab              ()
29     Type              (Ergebnis)
30     Zähler=          (Zähler+1)
31     Hard Return      ()
32     EndWhile
33     HardReturn        ()
34     Anfang=          (Anfang+1)
35     Zähler=          1
36     EndWhile
37     HardReturn        ()

```

```

38  Type          ("Ende der Berechnung")
39  Quit
40  Label         (Abbruch)
41  Prompt        ("Abbruch"; "Sie haben die Berechnung abgebrochen.";3;;)
42  Wait          (30)
43  EndPrompt

```

## Erklärung

- 1 Produktkennung
- 2 Beim Klicken auf [*Abbrechen*] wird das Makro abgebrochen.
- 3 Öffnen eines leeren Dokumentfensters.
- 4 Makroablauf sichtbar machen.
- 5 Tabstopps setzen, um die Zahlenreihen auszurichten.
- 6 Beginn einer Schleife.
- 7-8 Anzeigen einer Meldung und Eingabe von Unter- und Obergrenze. Die Untergrenze (= Multiplikator) wird in der Variablen *Anfang* gespeichert, die Obergrenze in der Variablen *Ende*.
- 9 Prüfen, ob die beiden Eingaben formal richtig sind. Die Obergrenze darf nicht kleiner als die Untergrenze sein.
- 10–14 Anzeigen einer Fehlermeldung (Untergrenze ist größer als Obergrenze) und Rücksprung zum Beginn der Schleife. Dort erfolgt wieder die Aufforderung zur Eingabe von Unter- und Obergrenze.
- 15 Ende der IF-Bedingung.
- 16 Der Variablen *Zähler* wird der Wert 1 zugeordnet.
- 17 Beginn der äußeren *WHILE*-Schleife. Diese wird so lange abgearbeitet, bis das Einmaleins des gewünschten Zahlenraums gedruckt wurde. Die Schleife endet bei Zeile 39.
- 18 Beginn der inneren *WHILE*-Schleife, die bei Zeile 35 endet. Solange der Wert in der Variablen *Zähler* kleiner als 11 ist, werden die Befehle ausgeführt, die zwischen *WHILE* und dem nächsten *ENDWHILE* definiert wurden (Zeilen 19 – 34).
- 19 Errechnen einer Zeile. Die Variable *Ergebnis* speichert das Ergebnis, die Variable *Zähler* dient hier als Multiplikand, die Variable *Anfang* als Multiplikator:
 

Zähler	*	Anfang	=	Ergebnis
1	*	5	=	5
- 20 Einrücken auf den nächsten Tabstopp.
- 21 Drucken der Variablen *Zähler*.
- 22 Einrücken auf den nächsten Tabstopp.
- 23 Drucken einer Textkonstanten.
- 24 Einrücken auf den nächsten Tabstopp.

- 25 Drucken der Variablen *Anfang*.
- 26 Einrücken auf den nächsten Tabstopp.
- 27 Drucken einer Textkonstanten.
- 28 Einrücken auf den nächsten Tabstopp.
- 29 Drucken der Variablen *Ergebnis*.
- 30 Die Variable *Zähler* (= Multiplikand) wird um 1 erhöht.
- 31 Gedruckte Zeile beenden, Cursor an den Beginn der nächsten Zeile bringen.
- 32 Ende der inneren *WHILE*-Schleife. Eine Zahlenkolonne, z.B. das Einmaleins von fünf wurde beendet.
- 33 Einfügen einer Leerzeile, um die einzelnen Zahlenkolonnen voneinander zu trennen.
- 34 Die Variable *Anfang* (= Multiplikator) wird um 1 erhöht.
- 35 Zurücksetzen der Variablen *Zähler* (= Multiplikand) auf 1.
- 36 Ende der äußeren *WHILE*-Schleife. Der Zahlenraum zwischen Unter- und Obergrenze (einschließlich) wurde komplett abgearbeitet.
- 37 Drucken einer Leerzeile hinter der letzten Zahlenkolonne.
- 38 Schreiben des Textes »Ende der Berechnung«.
- 39 Makroablauf beenden (normales Ende).
- 40 Beginn einer Befehlsfolge, zu der dann verzweigt wird, wenn während des Makroablaufs auf [Abbrechen] geklickt wird.
- 41 Anzeigen einer Meldung.
- 42 Meldung für 30-zehntel Sekunden anzeigen.
- 43 Meldung wieder ausblenden. Danach endet das Makro automatisch, weil keine weiteren Befehle mehr folgen.

## 9.2 Erraten einer eingegebenen Zahl

**Aufgabenstellung:** Ratespiel für zwei Personen. Während sich die erste Person umdreht, gibt die zweite Person eine Zahl ein, die die erste Person anschließend durch Eingabe von Zahlen erraten muß. Beenden Sie das Makro bei Bedarf während der Ausführung durch Klicken auf [Abbrechen].

**Schwerpunkte:** Vergleiche, Bedingungen, Schleifen, Zuordnen.

### Makrodefinition: ZAHLRATE.WCM

```

1 Application      (A1; "WordPerfect"; Default; "DE")
2 Textgroß=       "Zahl ist zu groß: "
  Textklein=      "Zahl ist zu klein: "
  Erraten1=       "Sie haben die Zahl mit "
  Erraten2=       " Versuchen erraten."
```

```

3      OnCancel      (Abbruch)
4      Prompt        ("Zahlen erraten";"Ratespiel für zwei Personen. Während sich die
                       erste Person umdreht, gibt die zweite Person eine Zahl ein, die die
                       erste Person durch Eintippen von Zahlen erraten muß. Max. 10 Versuche
                       sind möglich";;)
5      Wait          (90)
6      EndPrompt
7      Label         (Anfang)
8      Versuche=     0
4      Prompt        ("Zahlen erraten";"Zweite Person bitte umdrehen.";;;)
5      Wait          (30)
6      EndPrompt
9      GetNumber     (Eingabe;"Bitte die zu erratende Zahl eingeben."; "Zahlen erraten")
4      Prompt        ("Zahlen erraten";"Erraten Sie die eingegebene Zahl.";;;)
5      Wait          (30)
6      EndPrompt
10     Label         (Raten)
11     GetNumber     (Zahl;"Erraten Sie die eingegebene Zahl.."; "Zahlen erraten")
12     Versuche=     (Versuche+1)
13     If            (Versuche>10)
4      Prompt        ("Pech gehabt!";"10 Versuche sind um, ohne die Zahl zuerraten.";;;)
5      Wait          (30)
6      EndPrompt
14     Go            (Weiterspielen)
15     EndIf
16     If            (Zahl>Eingabe)
17     Anzeige=     NumStr(Zahl)
18     Meldung=     (Textgroß+Anzeige)
19     Prompt        ("Falsch";Meldung;;;)
20     Wait          (20)
21     EndPrompt
22     Go            (Raten)
15     EndIf
23     If            (Zahl<Eingabe)
17     Anzeige=     NumStr(Zahl)
18     Meldung=     (Textklein+Anzeige)
19     Prompt        ("Falsch";Meldung;;;)
20     Wait          (20)
21     EndPrompt
22     Go            (Raten)
15     EndIf
24     Anzeige=     NumStr(Versuche)
25     Meldung=     (Erraten1+Anzeige+Erraten2)
19     Prompt        ("Volltreffer";Meldung;;;)
20     Wait          (30)
21     EndPrompt
26     Label         (Weiterspielen)
27     GetString     (Eingabe;"Wollen Sie weiterspielen? (j/n)"; "Zahlen erraten")
28     IF            (Eingabe="j")
29     GO            (Anfang)
15     EndIf
4      Prompt        ("Ende des Spieles";"Vielen Dank für das Spiel.";1;;)

```

```

5      Wait                      (30)
6      EndPrompt
30     Quit
31     Label                    (Abbruch)
      Prompt                    ("Abbruch"; "Sie haben das Spiel abgebrochen."; 3;;)
      Wait                      (30)
      EndPrompt

```

## Erklärung

- 1 Produktkennung.
- 2 Über diese vier Befehle werden Texte zugeordnet, die bei späteren *Prompt*-Befehlen benötigt werden.
- 3 Beim Klicken auf [*Abbrechen*] verzweigt das Makro zu *Abbruch* und wird beendet.
- 4 Anzeigen einer Meldung am Bildschirm (Hinweis: über den Befehl *DialogAddEditBox* können Sie ein Dialogfeld definieren, in dem Sie die Eingabe »dunkel« vornehmen können, d.h. anstelle der eingetippten Zeichen werden \* angezeigt).
- 5 Die Prompt-Meldung wird für drei Sekunden angezeigt.
- 6 Jeder *Prompt*-Befehl muß mit dem Befehl *EndPrompt* abgeschlossen werden. Die angezeigte Meldung wird dadurch wieder ausgeblendet.
- 7 Zu diesem Label wird verzweigt, wenn unter Punkt 29 »j« eingegeben wird.
- 8 Jeder Versuch, eine eingegebene Zahl zu erraten, wird gezählt. An dieser Stelle wird der Zähler, also die Variable *Versuche*, auf Null gesetzt.
- 9 Die erste Person muß die zu erratende Zahl eingeben.
- 10 Zu diesem Label wird verzweigt, wenn eine falsche Zahl eingegeben wurde.
- 11 Anzeige eines Dialogfeldes mit der Aufforderung zur Eingabe einer Zahl.
- 12 Nach der Eingabe einer Zahl wird die Variable *Versuche* um 1 erhöht.
- 13 Kontrolle der Variablen *Versuche*. Bei mehr als 10 eingegebenen Zahlen wird das Raten mit einer entsprechenden Meldung abgebrochen.
- 14 Verzweigung zu dem Label (*Weiterspielen*).
- 15 Jede If-Befehlsfolge muß mit einem *EndIf*-Befehl enden.
- 16 Vergleichen der zu erratenden und der eingegebenen Zahl. Ist die aktuell eingegebene Zahl, die in der Variablen *Zahl* gespeichert ist, größer als die zu erratende Zahl, werden alle nachfolgenden Befehle bis zum nächsten *EndIf*-Befehl ausgeführt. Trifft diese Bedingung nicht zu, wird mit dem hinter *EndIf* definierten Befehl fortgefahren.
- 17 Für die unter Punkt 18 anzuzeigende Meldung muß die eingegebene Zahl in einen String umgewandelt werden, sonst kann sie nicht in die Meldung integriert werden. Beide Variablen (*Anzeige* und *Zahl*) enthalten danach dieselbe Zeichenfolge. Der Unterschied besteht darin, daß der Inhalt von *Zahl* als numerisch und der von *Anzeige* als alphanumerisch interpretiert wird.

- 18 Der Text für die unter Punkt 18 anzuzeigende Meldung wird in der Variablen *Meldung* zusammengesetzt. Die Variablen *Textgroß/Textklein* wurde am Beginn des Makros mit »Zahl ist zu groß:« bzw. »Zahl zu klein:« initialisiert. Diesem Text soll die eingegebene Zahl folgen. Wurde z.B. die Zahl 9 eingegeben, würde die Variable *Meldung* folgenden Inhalt haben: »Zahl ist zu groß: 9.«
- 19 Anzeige der Meldung, die unter Punkt 18 generiert wurde.
- 20 Die Meldung wird für 20/10tel Sekunden angezeigt, damit sie der Benutzer in Ruhe lesen kann. Danach wird sie wieder automatisch ausgeblendet.
- 21 Jeder *Prompt*-Befehl muß mit *EndPrompt* enden.
- 22 Da die eingegebene Zahl falsch war, wird wieder an den Beginn der Schleife, also zu dem Label (*Raten*) verzweigt.
- 23 Dieser Befehl entspricht sinngemäß dem unter Punkt 16 beschriebenen Befehl, wobei hier auf kleiner abgeprüft wird.
- 24 Dieser Befehl und alle folgenden bis einschließlich zum nächsten *EndPrompt* werden nur dann ausgeführt, wenn die Bedingungen unter Punkt 16 und 23 nicht zutreffen. In diesem Fall wurde die richtige Zahl eingegeben. Die Beschreibung zu NumStr entnehmen Sie bitte Punkt 17.
- 25 Die anzuzeigende Meldung wird aus drei Variablen generiert. Den Variablen *Erraten1* und *Erraten2* wurden zu Beginn des Makros (siehe Punkt 2) Texte zugeordnet. Wurden z.B. sechs Versuche zum Erraten der Zahl benötigt, wird folgende Meldung angezeigt: »Sie haben die Zahl mit 6 Versuchen erraten«.
- 26 Zu diesem Label wird dann verzweigt, wenn die richtige Zahl eingegeben wurde, d.h. wenn die unmittelbar zuvor definierten Befehle (ab Punkt 24) ausgeführt wurden oder wenn bei Punkt 13 die maximale Anzahl Versuche erreicht wurde.
- 27 Anzeige einer Meldung und Eingabe eines Zeichens (j oder n).
- 28 Prüfung des eingegebenen Zeichens (j oder n).
- 29 Bei der Eingabe von j beginnt ein neues Spiel, ansonsten wird das Makro beendet.
- 30 Normales Makroende, wenn nicht auf [*Abbrechen*] geklickt wurde.
- 31 Zu dieser Befehlsfolge wird beim Klicken auf [*Abbrechen*] verzweigt.

## 9.3 Kalender

**Aufgabenstellung:** Über dieses Makro soll u.a. der Tagesname eines bestimmten Tages ermittelt werden. Jedes beliebige Datum ab 1753 ist erlaubt. Das eingegebene Datum wird formal überprüft, d. h. für die Monatsangaben sind nur die Werte von 1–12 und für die Tagesangaben nur die Werte 1–30 bzw. 31 (in Abhängigkeit der Monate) möglich. Für den Februar sind eigene Routinen erforderlich, da dieser Monat bei Schaltjahren 29, ansonsten nur 28 Tage hat. Der ermittelte Tagesname wird am Bildschirm angezeigt.

Bei dieser sehr komplexen Anwendung werden über *Run*-Befehle weitere Makros aufgerufen. Diese erstellen einen/eine

Terminkalender	pro Tag des Jahres wird ein Kalenderblatt erstellt.
Jahreskalender	das komplette Jahr auf einer A4-Seite.
Jahreskalender	mit einer fortlaufenden Numerierung der Tage.
Feiertagsliste	mit den Feiertagen des betreffenden Jahres.
Jahresplaner	mit gekennzeichneten Feiertagen.
Monatsplaner	mit den Feiertagen des betreffenden Jahres.
Jahresübersicht	mit gekennzeichneten Feiertagen.

Eine genaue Bedienungsanleitung für die Kalender-Makros und Musterabbildungen der einzelnen Kalendertypen finden Sie in Kapitel »10 Makro-Sammlung«.

### Schwerpunkte:

Aufruf selbständiger Makros über *Run*, Übergabe von Daten von dem aufrufenden Makro an das aufgerufene Makro, Schleifenverarbeitung, Definition von Dialogfeldern, mathematische Formeln.

### Allgemeine Hinweise:

Die Erstellung des jeweiligen Kalenders wird am Bildschirm nicht angezeigt, weil die Erstellung sonst zu lange dauert. Das gilt insbesondere für Computertypen unterhalb der Leistungsklasse 80486.

Die für die Erstellung vordefinierten Formulare müssen von Ihnen ggf. geändert werden, da die hier verwendeten Formulgargößen und Schriften speziell für Drucker HP-LaserJet 4SI zutreffen. Wählen Sie für Ihren Drucker Schriften derselben Größe aus (damit das Format erhalten bleibt), sofern sie von WordPerfect nicht ordnungsgemäß konvertiert werden.

Bei der Kompilierung der aufgerufenen Makros (z.B. KALELAUF, KALEJAHR usw.) erfolgt bei der anschließenden Ausführung eine falsche Berechnung des Kalenders, weil die Übergabvariablen nicht vorhanden sind. Dadurch können die definierten Arrays nicht ordnungsgemäß abgearbeitet werden. Brechen Sie in diesem Fall das Makro ab, sofern es nicht bereits mit einer Fehlermeldung abgebrochen wurde, und starten Sie es erneut über das Makro KALENDER.WCM. Schließen Sie in diesem Fall geöffnete Dateien, ohne diese zu speichern! Das gilt auch für alle anderen Abbrüche bei der Kalendererstellung.

### Makrodefinition: KALENDER.WCM

```

Application      (A1;"WordPerfect";Default;"DE")
1 PersistAll
2 TagTab:=        "Sonntag Montag Dienstag Mittwoch Donnerstag Freitag Samstag"
3+4 Text1:=        "Der "      Text2:=      " ist ein "
5 //*****
6 Kalebere:=      "C:\WPWIN7\MAKROS\KALEBERE.WCM"
Kaleterm:=       "C:\WPWIN7\MAKROS\KALETERM.WCM"
```



```

Kalejahr:= "C:\WPWIN7\MAKROS\KALEJAHR.WCM"
Kalelauf:= "C:\WPWIN7\MAKROS\KALELAUF.WCM"
Kalefeie:= "C:\WPWIN7\MAKROS\KALEFEIE.WCM"
Kalemopl:= "C:\WPWIN7\MAKROS\KALEMOPL.WCM"
Kalejapl:= "C:\WPWIN7\MAKROS\KALEJAPL.WCM"
Kalejaub:= "C:\WPWIN7\MAKROS\KALEJAUB.WCM"
7 Call (U13MenüAufbau)
8 Label (H01Anfang)
9 Call (U12MenüAnzeigen)
10 Switch (1)
11 CaseOf Feld1: Call (U01Tagesname)
    CaseOf Feld2: Call (U02KalenBerechnen)
    CaseOf Feld3: Call (U03TerminKalender)
    CaseOf Feld4: Call (U04Jahreskalender)
    CaseOf Feld5: Call (U05JahresTagNum)
    CaseOf Feld6: Call (U14Feiertage)
    CaseOf Feld7: Call (U15Jahresplaner)
    CaseOf Feld8: Call (U16Monatsplaner)
    CaseOf Feld9: Call (U17Jahresübersicht)
    CaseOf Feld10: Go (U99Ende)
12 EndSwitch
13 Go (H01Anfang)
14 Label (U01Tagesname)
15 Call (U06DatumPrüfen)
16 IF (DatumOK="N")
17 Return
    EndIF
18 Call (U09Berechnen)
19 Tag:= NumStr(Tag)
19 Monat:= NumStr(Monat)19Jahr:=NumStr(Jahr)
20 Meldung:= Text1+Tag+"."+Monat+"."+Jahr+Text2+Tagname
21 Prompt ("Tagesname ermitteln";Meldung;;;)
    Wait (30)
    EndPrompt
22 Return
23 Label (U02KalenBerechnen)
24 Call (U11Übergabe)
25 Run (KALEBERE)
26 Quit
23 Label (U03TerminKalender)
24 Call (U11Übergabe)
25 Run (KALETERM)
26 Quit
23 Label (U04JahresKalender)
24 Call (U11Übergabe)
25 Run (KALEJAHR)
26 Quit
23 Label (U05JahresTagNum)
24 Call (U11Übergabe)
25 Run (KALELAUF)
26 Quit
27 Label (U06DatumPrüfen)

```

```

28 DatumOK:= "J"
29 //NumStr (JahrZehnt (;0;Jahr) aus WPin 6.1
30 JahrZehnt:= SubStr (Jahr;3;2)
31 JahrZehnt:= StrNum (JahrZehnt)
32 If ((Tag <1) Or (Tag>31))
    Prompt ("Fehler";"Tagesangabe falsch.";;;)
    Wait (20)
EndPrompt
Go (U06DatumEnde)
EndIf
33 If ((Monat<01) Or (Monat>12))
    Prompt ("Fehler";"Monatsangabe falsch.";;;)
    Wait (20)
EndPrompt
Go (U06DatumEnde)
EndIf
34 If (Jahr<1753)
    Prompt ("Fehler";"Jahresangabe bitte ab 1753.";;;)
    Wait (20)
EndPrompt
Go (U06DatumEnde)
EndIf
35 If ((Monat=01) or (Monat=03) or (Monat=05) or (Monat=07) or
    (Monat=08) or (Monat=10) or (Monat=12))
36 Return
EndIf
37 If ((Monat=04)or(Monat=06)or(Monat=09)or(Monat=11))
    Go (U08Tage30)
EndIf
38 If (Tag<29)
39 Return
EndIf
40 If (Tag>29)
    Prompt ("Fehler";"Tagesangabe für Februar falsch.";;;)
    Wait (20)
EndPrompt
Go (U06DatumEnde)
EndIf
41 If (JahrZehnt=00)
42 SchJ2:= Jahr MOD 400
43 If (SchJ2=0)
44 Return
Else
45 Go (U07Tage28)
EndIf
EndIf
46 SchJ2:= Jahr MOD 4
47 If (SchJ2=0)
    Return
EndIf
48 Label (U07Tage28)
    If (Tag>28)

```

```

Prompt      ("Fehler";"Monat hat nur 28 Tage, da kein Schaltjahr. Volle
Jahrhunderte, wie z. B. 1900, müssen durch 400 teilbar
sein.;;;)
Wait        (60)
EndPrompt
Go          (U06DatumEnde)
Else
Return
EndIf
49 Label    (U08Tage30)
If          (Tag>30)
Prompt     ("Fehler";"Monat hat nur 30 Tage.;;;)
Wait      (20)
EndPrompt
Go        (U06DatumEnde)
EndIf
Return
50 Label  (U06DatumEnde)
DatumOK:= "N"
Return
51 Label  (U09Berechnen)
Integer  (RF1;0.6+(1/Monat))
RF2:=    Jahr-RF1
RF3:=    Monat+(12*RF1)
RF4:=    RF2/100
Integer  (RF5;RF4/4)
Integer  (RF6;RF4)
Integer  (RF7;(5*RF2)/4)
Integer  (RF8;(13*(RF3+1)/5))
RF9:=    RF8+RF7-RF6+RF5+Tag-1
Integer  (RF10;RF9/7)
RF10:=   RF10*7
52 RF9:=  RF9-RF10+1
53 Integer (TabPos;RF9*10-9)
54 Tagname:= SubStr (TagTab;TabPos;10)
55 Return
56 Label    (U10Schaltjahr)
57 If      (JahrZehnt=00)
SchJ2:=    Jahr MOD 400
58 If      (SchJ2=0)
SchJahr:=  "J"
Return
Else
SchJahr:=  "N"
Return
EndIf
EndIf
59 SchJ2:=  Jahr MOD 4
60 If      (SchJ2=0)
SchJahr:=  "J"
Else
SchJahr:=  "N"
EndIf

```

```

61 Return
62 Label (U11Übergabe)
63 //NumStr (JahrZehnt;0;Jahr) aus WPWIN 6.1
   JahrZehnt:= SubStr (Jahr;3;2)
   JahrZehnt:= StrNum (JahrZehnt)
64 If (Jahr<1753)
   Prompt ("Fehler";"Jahresangabe bitte ab 1753.";;;)
   Wait (20)
   EndPrompt
   Return
   EndIf
65 Tag:= 1
   Monat:= 1
66 Call (U09Berechnen)
67 Call (U10Schaltjahr)
68 TagPos:= NumStr (RF9)
69 Jahr:= NumStr (Jahr)
70 //JaPosSch:=(Jahr+RF9+SchJahr) Nicht mehr erforderlich wegen
71 //MischenVariableDefinieren PERSISTALL (konvertiertes Makro).
   //(MischenVariable:VarJPS;
   //StringWert:JaPosSch)
72 Anzeige (Aus!)
73 Return
74 Label (U12MenüAnzeigen)
75 Tag:= 1
75 Monat:= 1
75 Jahr:= 1994
76 Feld1:= 1
77 Feld2:= 0
77 Feld3:= 0
77 Feld4:= 0
77 Feld5:= 0
77 Feld6:= 0
77 Feld7:= 0
77 Feld8:= 0
77 Feld9:= 0
78 DialogDisplay ("Kal";2)
79 If (Feld9=1)
   Go (U99Ende)
   EndIf
80 Return
81 Label (U13MenüAufbau)
82 DialogDefine ("Kal"; 50; 50; 260; 240; OK! | Percent!; "Kalender")
83 DialogAddGroupBox ("Kal"; 1000; 6; 4; 240; 125;
   "Kalender erstellen")
84 DialogAddRadioButton("Kal"; 1001; 10; 80; 15;"Tages&name ermitteln"; Feld1)
84 DialogAddRadioButton("Kal"; 1002; 10; 23; 150; 15;"Kalender &berechnen"; Feld2)
84 DialogAddRadioButton("Kal"; 1003; 10; 36; 150; 15;"Terminkalender erstellen"; Feld3)
84 DialogAddRadioButton("Kal"; 1004; 10; 49; 150; 15;"&Jahreskalender erstellen"; Feld4)
84 DialogAddRadioButton("Kal"; 1005; 10; 62; 150; 15;"Jahres&kalender erstellen
   (Tagesnummer)"; Feld5)
84 DialogAddRadioButton("Kal"; 1006; 10; 75; 150; 15;"&Feiertagsliste erstellen"; Feld6)

```

```

84 DialogAddRadioButton("Kal"; 1007; 10; 88; 150; 15;"Jahres&planer erstellen"; Feld7)
84 DialogAddRadioButton("Kal"; 1008; 10; 101; 150; 15;"&Monatsplaner erstellen"; Feld8)
84 DialogAddRadioButton("Kal"; 1008; 10; 114; 150; 15;"Jahres&übersicht erstellen";
    Feld9)
84 DialogAddRadioButton("Kal"; 1007; 175; 150; 65; 15;"Makro b&eenden"; Feld10)
83 DialogAddGroupBox      ("Kal"; 1008; 6; 135; 120; 75; "Datum eingeben")
83 DialogAddGroupBox      ("Kal"; 1009; 166; 135; 80; 40; "Ende")
85 DialogAddText          ("Kal"; 1010; 30; 150; 30; 15; 1; "Tag")
85 DialogAddText          ("Kal"; 1011; 30; 170; 30; 15; 1; "Monat")
85 DialogAddText          ("Kal"; 1012; 30; 190; 30; 15; 1; "Jahr")
86 DialogAddCounter       ("Kal"; 1013; 70; 150; 37; 15; 0; Tag; 1; 31; 1)
86 DialogAddCounter       ("Kal"; 1014; 70; 170; 37; 15; 0; Monat; 1; 12; 1)
86 DialogAddCounter       ("Kal"; 1015; 70; 190; 37; 15; 0; Jahr; 1753; 9999; 1)
87 Return
88 Label                  (U14Feiertage)
    Run                    (KALEFEIE)
    Quit
89 Label                  (U15Jahresplaner)
    Run                    (KALEJAPL)
    Quit
90 Label                  (U16Monatsplaner)
    Run                    (KALEMOPL)
    Quit
91 Label                  (U17Jahresübersicht)
    Run                    (KALEJAUB)
    Quit
92 Label                  (U99Ende)
    DialogDestroy        ("Kal")
    Quit

```

## Erklärung

- 1 Übernahme aller Variablen in die Persist-Tabelle, um verschiedene der Variablen an die aufgerufenen Makros zu übergeben.
- 2 In diesem Datenfeld werden die Tagesnamen in der vorgegebenen Reihenfolge definiert, die später aufgrund der Tagesberechnung aus diesem Feld selektiert werden. Dieses Feld müssen Sie sich als eine Art Tabelle vorstellen, die aus sieben Feldern (= sieben Tage) mit derselben Länge besteht. Die Länge richtet sich nach dem längsten Tagesnamen (= Donnerstag). Tagesnamen, die kürzer sind, müssen am Ende mit Leerstellen aufgefüllt werden! Dieser Befehl dient hier nur zu Übungszwecken. Einfacher geht es mit *Declare*.
- 3+4 Definition von Textfeldern, die für die Anzeige des Ergebnisses benötigt werden (siehe unten).
- 5 Kommentar.
- 6 Definition von Makro-Dateinamen, die weiter unten über *Run* aufgerufen werden. Dateinamen sollten Sie immer am Beginn eines Makros in Variablen definieren. Sie brauchen dann bei Änderungen dieser Namen nicht das ganze Makro abzusuchen, wo diese Namen überall vorkommen.

- 7 Aufruf der Unterroutine *U13MenüAufbau*. In dieser Unterroutine wird das angezeigte Dialogfeld definiert (Beschreibung siehe weiter unten).
- 8 Beginn der Haupt-Steuerungsroutine.
- 9 Aufruf der Unterroutine *U12MenüAnzeigen*. In dieser Unterroutine wird das Dialogfeld angezeigt (Beschreibung siehe weiter unten).
- 10 Beginn einer *Switch*-Struktur. Eine unter *CaseOf* definierte Bedingung trifft dann zu, wenn das zugehörige Feld eine 1 enthält.
- 11 Aufruf der betreffenden Unterroutine aufgrund der getroffenen Menüauswahl. Eine Bedingung trifft dann zu, wenn eines der Felder *Feld1* – *Feld10* eine 1 enthält.
- 12 Ende der *Switch*-Struktur.
- 13 Sprung zum Schleifenanfang.
- 14 Beginn der Unterroutine *U01Tagesname*. Diese Routine dient der Ermittlung des Tagesnamens.
- 15 Aufruf der Unterroutine *U06DatumPrüfen* (Beschreibung siehe unten).
- 16 Prüfen, ob das Datum in der Unterroutine *U06DatumPrüfen* als korrekt erkannt wurde.
- 17 Wenn das Datum falsch war, wird die Unterroutine *U01TagesName* beendet, es erfolgt ein Rücksprung zum aufrufenden Befehl.
- 18 Aufruf der Unterroutine *U09Berechnen* (Beschreibung siehe unten).
- 19 Die *NumStr*-Befehle konvertieren die numerischen Datenfelder für die Anzeige in Strings.
- 20 Generierung eines Strings, der über den *Prompt*-Befehl am Bildschirm angezeigt wird. Der String setzt sich aus folgenden Einzelkomponenten zusammen, wenn z. B. das Datum 10.4.1996 eingegeben wurde:
 

Text1	=	Der
Tag	=	10
","	=	.
Monat	=	4
","	=	.
Jahr	=	1996
Text2	=	ist ein
Tagname	=	Mittwoch
- 21 Anzeige der zuvor generierten Meldung am Bildschirm für drei Sekunden.
- 22 Beenden der Unterroutine und Rücksprung zum aufrufenden Befehl, sofern die Unterroutine nicht vorher über *Return* beendet wurde bzw. die Makroverarbeitung nicht in dem aufgerufenen Makro beendet wird.
- 23 Beginn der Unter Routinen, die selbständige Makros aufrufen.
- 24 Aufruf der Unterroutine *U11Übergabe* (Beschreibung siehe unten).

- 25 Aufruf eines Makros. Die Makronamen wurden am Beginn dieses Makros definiert (siehe Zeile 6).
- 26 Beenden des Makros, nachdem das aufgerufene Makro beendet wurde.
- 27 Beginn der Unteroutine *U06DatumPrüfen*. Das eingegebene Datum wird formal überprüft. Bei falschen Angaben wird die Eingabe des Datums erneut angefordert.
- 28 Die Variable *DatumOK* wird mit »J« initialisiert. Wird während der Datumsprüfung ein Fehler erkannt, wird diese Variable mit »N« initialisiert. Unterrouتين, die *U06DatumPrüfen* aufrufen, können dann erkennen, ob das Datum richtig war oder nicht.
- 29 Die numerische Variable *Jahr* wird als String in der Variablen *JahrZehnt* gespeichert.
- 30 Aus der vierstelligen Jahreszahl werden die beiden letzten Stellen für die Ermittlung des Schaltjahres selektiert.
- 31 Die eingegebenen Daten werden in numerische Werte konvertiert. Das Makro wird beendet, wenn bei der Eingabe keine Ziffern verwendet wurden.
- 32 Der Inhalt des Feldes *Tag* wird überprüft. Zulässig sind nur die Werte 1–31, andernfalls erfolgt eine Fehlermeldung am Bildschirm. Anschließend verzweigt das Makro wieder zum Schleifenanfang. Achten Sie auf die doppelten Klammern am Beginn und am Ende des Ausdrucks!
- 33 Der Inhalt des Feldes *Monat* wird überprüft. Zulässig sind nur die Werte 1–12, andernfalls erfolgt eine Fehlermeldung am Bildschirm. Anschließend verzweigt das Makro wieder zum Schleifenanfang.
- 34 Der Inhalt des Feldes *Jahr* wird überprüft. Alle Jahresangaben ab 1753 sind erlaubt, andernfalls erfolgt eine Fehlermeldung am Bildschirm. Anschließend verzweigt das Makro wieder zum Schleifenanfang.
- 35 Enthält das Feld *Monat* einen Monat mit 31 Tagen, ist die Datumsprüfung beendet, das eingegebene Datum wird als formal richtig anerkannt.
- 36 Das Datum wurde als richtig anerkannt. Es erfolgt ein Rücksprung zum aufrufenden Befehl.
- 37 Enthält das Feld *Monat* einen Monat mit 30 Tagen, wird zur Unteroutine *U08Tage30* verzweigt (Beschreibung siehe unten).
- 38 Ab hier beginnt die formale Prüfung der Angaben zum Monat Februar. Trafen die beiden vorausgegangenen *If*-Bedingungen nicht zu, handelt es sich um den Monat Februar. Enthält das Feld *Tag* einen Wert, der nicht größer als 28 ist, wird die Datumsprüfung beendet, das eingegebene Datum wird als formal richtig anerkannt.
- 39 Das Datum wurde als richtig anerkannt. Es erfolgt ein Rücksprung zum aufrufenden Befehl.
- 40 Für den Februar wurde ein Tag größer als 29 angegeben, was in jedem Fall falsch ist. Es erfolgt eine Fehlermeldung am Bildschirm. Anschließend verzweigt das Makro wieder zu dem Schleifenanfang.
- 41 Es wird geprüft, ob es sich bei der eingegebenen Jahreszahl um ein volles Jahrhundert handelt, weil hier für die Schaltjahre eine Sonderregelung gilt.

- 42 Ermittlung des Quotienten der Division der Jahreszahl durch 400. Volle Jahrhunderte sind nur dann Schaltjahre, wenn sie durch 400 teilbar sind. Das Jahr 1900 ist darum kein Schaltjahr, obwohl es durch 4 teilbar ist.
- 43 Ist der Quotient Null, war die Jahreszahl durch 400 ohne Rest teilbar, d.h. es handelt sich um ein Schaltjahr (z.B. 2000, 2400).
- 44 Das Datum wurde als richtig anerkannt. Es erfolgt ein Rücksprung zum aufrufenden Befehl.
- 45 Bei einem Wert größer als Null handelt es sich um kein Schaltjahr. Das Makro verzweigt in diesem Fall zu der Unteroutine *U07Tag28* (Beschreibung siehe unten).
- 46 Wenn es sich um kein volles Jahrhundert handelt, wird die Jahreszahl durch vier dividiert und der Rest ermittelt.
- 47 Ist der Quotient Null, war die Jahreszahl ohne Rest durch 4 teilbar, d.h. es handelt sich um ein Schaltjahr (z.B. 1992, 1976). Das Datum wurde als richtig anerkannt. Es erfolgt ein Rücksprung zum aufrufenden Befehl.
- 48 Beginn der Unteroutine *U07Tag28*. Das Makro verzweigt zu dieser Unteroutine, wenn der Monat ein Februar und das Jahr kein Schaltjahr war. Enthält das Feld *Tag* einen Wert größer als 28, erfolgt eine Fehlermeldung und ein Sprung zum Schleifenanfang. Andernfalls wird das Datum als richtig anerkannt. Es erfolgt ein Rücksprung zum aufrufenden Befehl.
- 49 Beginn der Unteroutine *U08Tag30*. Das Makro verzweigt zu dieser Unteroutine, wenn ein Monat mit 30 Tagen erkannt wurde. Enthält das Feld *Tag* einen Wert größer als 30, erfolgt eine Fehlermeldung und ein Sprung zum Schleifenanfang. Andernfalls wird das Datum als richtig anerkannt. Es erfolgt ein Rücksprung zum aufrufenden Befehl.
- 50 Zu dieser Unteroutine wird verzweigt, wenn ein falsches Datum erkannt wurde. Die Variable *DatumOK* wird auf »N« gesetzt, um in aufrufenden UnterROUTINEN einen Fehler zu signalisieren.
- 51 Beginn der Unteroutine *U09Berechnen*. Mit den nachfolgenden Befehlen wird aufgrund des eingegebenen Datums ein Wert von 1–7 ermittelt, der jeweils einem Wochentag entspricht. Die Zuordnung wird mit 1=Sonntag, 2=Montag usw. vorgenommen.
- 52 Nach der Berechnung enthält das Rechenfeld RF9 einen Wert von 1–7. Dieser Wert entspricht dem Tabellenplatz, den der errechnete Tagesname einnimmt. Wurde vier ermittelt, handelt es sich um den vierten Platz, also um einen Mittwoch.
- 53 Ermittlung der Tabellenposition, ab der der errechnete Tagesname in der oben definierten Tabelle beginnt:

Sonntag	=	1	Donnerstag	=	41
Montag	=	11	Freitag	=	51
Dienstag	=	21	Samstag	=	61
Mittwoch	=	31			

Wurde z.B. für den 1. 1. 1992 der Wert 4 ermittelt, wird folgende Rechnung ausgeführt (das Ergebnis wird in *TabPos* gespeichert):



$$4 * 10 - 9 = 31$$

Der zu suchende Tagesname (= Mittwoch) beginnt auf der Stelle 31 und ist zehn Zeichen lang. Dieser Befehl dient hier nur zu Übungszwecken. Eine einfachere Möglichkeit besteht natürlich in der Definition eines Arrays (siehe Befehl *Declare*).

- 54 Aus der Tagestabelle *TagTab* werden aufgrund der ermittelten Tabellenposition (in diesem Beispiel = 31) die nächsten 10 Zeichen (= Mittwoch ) selektiert und in der Variablen *Tagname* gespeichert.
- 55 Rücksprung zum aufrufenden Befehl.
- 56 Beginn der Unteroutine *U10Schaltjahr*. Diese Unteroutine wird dann aufgerufen, wenn ein Kalender gedruckt werden soll. Es wird hier geprüft, ob es sich um ein Schaltjahr handelt oder nicht. Wenn ja, wird in der Variablen *SchJahr* ein *J* gespeichert, andernfalls ein *N*. Der Inhalt dieser Variablen wird beim Aufruf der Druckmakros an das jeweilige Makro übergeben.
- 57 Ermittlung des Restes der Division der Jahreszahl durch 400. Volle Jahrhunderte sind nur dann Schaltjahre, wenn sie durch 400 teilbar sind. Das Jahr 1900 ist darum kein Schaltjahr, obwohl es durch 4 teilbar ist.
- 58 Ist der Rest Null, war die Jahreszahl durch 400 ohne Rest teilbar, d.h. es handelt sich um ein Schaltjahr (z.B. 1600, 2000). In diesem Fall wird die Variable *SchJahr* auf *J* gesetzt, andernfalls auf *N*.
- 59 Wenn es sich um kein volles Jahrhundert handelt, wird die Jahreszahl durch vier dividiert und der Rest ermittelt.
- 60 Ist der Rest Null, war die Jahreszahl ohne Rest durch 4 teilbar, d.h. es handelt sich um ein Schaltjahr (z.B. 1992, 1976). In diesem Fall wird die Variable *SchJahr* auf *J* gesetzt, andernfalls auf *N*.
- 61 Rücksprung zum aufrufenden Befehl.
- 62 Beginn der Unteroutine *U11Übergabe*. Diese Unteroutine wird dann aufgerufen, wenn ein Kalender erstellt werden soll. Dem aufgerufenen Makro muß bekanntgegeben werden, um welches Jahr es sich handelt, ob dieses Jahr ein Schaltjahr ist und mit welchem Wochentag das Jahr beginnt.
- 63 Diese drei Befehle bereiten die eingegebene Jahreszahl so auf, daß sie in den Unter-routinen *U09Berechnen* und *U10Schaltjahr*, die weiter unten aufgerufen werden, verarbeitet werden können.
- 64 Prüfung der eingegebenen Jahreszahl. Bei einer Falscheingabe erfolgt eine Fehlermeldung.
- 65 Die Variablen *Tag* und *Monat* werden auf 1 gesetzt. Zum Drucken eines Kalenders muß immer der 1. Januar (z. B. 1 1 1994) des Jahres eingegeben werden, das gedruckt werden soll, andernfalls erfolgt eine Fehlermeldung.
- 66 Aufrufen der Unteroutine *U09Berechnen*, um den Wochentag zu ermitteln (Beschreibung siehe oben).
- 67 Aufrufen der Unteroutine *U10Schaltjahr* (Berechnung eines Schaltjahres, Beschreibung siehe oben).

- 68 Der numerische Inhalt von *RF9* (= Tabellenplatz des errechneten Tages) wird in einen String konvertiert.
- 69 Der numerische Inhalt von *Jahr* (= Jahreszahl) wird in einen String konvertiert.
- 70–71 Diese Befehle entfallen. Sie waren unter WPWin 5.1/5.2 erforderlich, um Variablen von einem Makro an das andere zu übergeben (dieses Makro wurde konvertiert).
- 72 Der Makroablauf soll nicht am Bildschirm angezeigt werden (schnellere Verarbeitung).
- 73 Rücksprung zum aufrufenden Befehl.
- 74 Beginn der Unteroutine *U12MenüAnzeigen*. Hier wird das unter *U13MenüAufbau* definierte Dialogfeld angezeigt.
- 75 Die Variablen *Tag*, *Monat* und *Jahr* werden mit festen Werten vorbesetzt.
- 76 Über die Variablen *Feld...* wird die Markierung der Optionsfelder (Radiobuttons) gesteuert. *Feld1* wird mit 1 initialisiert. Das zugehörige Optionsfeld wird bei der Anzeige des Dialogfeldes aktiviert.
- 77 Es kann in einer Gruppe immer nur ein Optionsfeld aktiviert werden, die restlichen Felder werden darum mit 0 initialisiert.
- 78 Anzeigen des Dialogfeldes.
- 79 Wurde im Dialogfeld *Feld9* aktiviert (Makro beenden), wird zu dem Label *U99Ende* verzweigt.
- 80 Rücksprung zum aufrufenden Befehl.
- 81 Beginn der Unteroutine *U13MenüAufbau*. Hier werden die Definitionen für das anzuzeigende Dialogfeld vorgenommen. Wenn Sie tiefergehende Informationen zu den verwendeten Befehlen benötigen, schlagen Sie bitte in Kapitel »6.37 Dialog« nach. Schneller geht's mit dem Dialog-Editor.
- 82 Definieren des Dialogfeldes (Position, Breite und Höhe).
- 83 Gruppenfeld erstellen, das die Auswahlmöglichkeiten zusammenfaßt.
- 84 Definition der Optionsfelder.
- 85 Definition von Textfeldern.
- 86 Definition von Zählertasten.
- 87 Rücksprung zum aufrufenden Befehl.
- 88 Aufrufen des Makros KALEFEIE zum Drucken einer Liste der Feiertage.
- 89 Aufrufen des Makros KALEJAPL zur Erstellung des Jahresplaners.
- 90 Aufrufen des Makros KALEMOPL zur Erstellung des Monatsplaners.
- 91 Aufrufen des Makros KALEJAUB zur Erstellung der Jahresübersicht.
- 92 Beginn der Unteroutine *U99Ende*. Zu dieser Routine wird verzweigt, wenn das Makro abgebrochen wurde (OnCancel) oder wenn bei der Menüauswahl *Ende* gewählt wurde. Das aufgebaute Dialogfeld *Kal* wird aus dem Speicher gelöscht.

## 9.3.1 Terminkalender berechnen

**Aufgabenstellung:** Über dieses Makro wird ein Terminkalender errechnet. Die Berechnung erfolgt für das Jahr, das in KALENDER.WCM eingegeben wurde. Es wird eine Datendatei erstellt, die für jeden Tag des Jahres einen Datensatz mit den Daten enthält, die zum Erstellen des Terminkalenders erforderlich sind. Nach Beendigung der Berechnung wird die komplette Datendatei gespeichert.

Zum Drucken des Terminkalenders werden aus der kompletten Datendatei vier kleinere Datendateien erstellt, für jedes Quartal des Jahres eine. Dies ist aus Platz- und Zeitgründen erforderlich. Würde die komplette Datendatei gemischt, kann das bei älteren Rechnern sehr lange dauern. Erstellen Sie darum den Terminkalender quartalsweise.

Eine genaue Bedienungsanleitung für die Kalender-Makros und Musterabbildungen der einzelnen Kalendertypen finden Sie in dem Kapitel »10 Makro-Sammlung«.

**Schwerpunkte:** Laden von Dateien, Übernahme von Daten von dem aufrufenden Makro in das aufgerufene Makro, Schleifenverarbeitung, Mischbefehle in Datendatei einfügen.

### Makrodefinition: KALEBERE.WCM

```

Application      (A1; "WordPerfect"; Default; "DE")
1 //             Bitte Dateinamen ggf. ändern (Namen der erzeugten Datendateien)
KalebereQ:=      "C:\WPWIN7\MAKROS\KALEBERE.Q"
KalebereDat:=    "C:\WPWIN7\MAKROS\KALEBERE.DAT"
2 //Folgende Variablen werden aus Makro KALENDER übergeben:
//Jahr           Jahreszahl
//TagPos         Mit welchem Wochentag beginnt das Jahr
//SCHJahr        Schaltjahr, N = kein Schaltjahr, J = Schaltjahr
VarErrChk       (Off!)
3 Declare (TagTab[7])
                TagTab[1]:="Sonntag"
                TagTab[2]:="Montag"
                TagTab[3]:="Dienstag"
                TagTab[4]:="Mittwoch"
                TagTab[5]:="Donnerstag"
                TagTab[6]:="Freitag"
                TagTab[7]:="Samstag"
Declare (Monat[12])
                Monat[1]:="Januar"
                Monat[2]:="Februar"
                Monat[3]:="März"
                Monat[4]:="April"
                Monat[5]:="Mai"
                Monat[6]:="Juni"

```

```

        Monat[7]:="Juli"
        Monat[8]:="August"
        Monat[9]:="September"
        Monat[10]:="Oktober"
        Monat[11]:="November"
        Monat[12]:="Dezember"
Declare(LetzterTag[12])
        LetzterTag[1]:=31
        LetzterTag[2]:=28
        LetzterTag[3]:=31
        LetzterTag[4]:=30
        LetzterTag[5]:=31
        LetzterTag[6]:=30
        LetzterTag[7]:=31
        LetzterTag[8]:=31
        LetzterTag[9]:=30
        LetzterTag[10]:=31
        LetzterTag[11]:=30
        LetzterTag[12]:=31
DruJahr:=      Jahr
4      OnError      (U77OnError)
5      OnNotFound   (U88NichtGefunden)
6      Call          (U01DatenDatei)
        Call          (U02KalenderBerechnen)
        Call          (U03OsternBerechnen)
        Call          (U04VariableFeiertage)
        Call          (U05FixeFeiertage)
        Call          (U10NumZeichen)
        Call          (U12Separieren)
7      Go            (U99Ende)
8      Label         (U01DatenDatei)
9      Display       (Off!)
10     FileNew        ( )
11     InsertTypeOver (On!)
12     DraftMode      (On!)
13     FontFine        (On!)
14     MergeInsertCode (MergeCode:FieldNames!;
        ParameterString:"SuchDatum;Tag;Monat;Jahr;
        FeierKennz;Woche;Tagesnummer;NochTage;Tagesname
        ;Monatsname;Feiertag")
15     PosDocBottom   ( )
16     FontFine        (Off!)
        Return
17     Label          (U02KalenderBerechnen)
18     If              ((TagPos>5) or (TagPos=1))
        WochZähl:=    0
        Else
        WochZähl:=    1
    EndIf
19     TagZählJa:=     0
20     MonNummer:=     0
21     If              (SchJahr="J")

```

```

    TagGesamt:= 366
    JanFeb:= 60
    JanFebMrz:= 91
    Februar:= 29
    LetzterTag[2]:= 29
Else
    TagGesamt:= 365
    JanFeb:= 59
    JanFebMrz:= 90
    Februar:= 28
    LetzterTag[2]:= 28
EndIf
22 ForNext      ( M;1; 12; 1 )
23 MonNummer:=  (MonNummer+1)
24 TagZählMo:=  0
25 Meldung:=    (DruJahr+", Monat "+Monat[M]+" wird erstellt.Bitte nicht auf
                [OK] klicken.")
                ("Kalender berechnen";Meldung;;;)
26 If           ((Monat[M]="April") Or (Monat[M]="Juli") Or
                (Monat[M]="Oktober"))
    Type        ("separieren")
    HardPageBreakInsert()
EndIf
27 Repeat
28   TagZählMo:= (TagZählMo+1)
29   TagZählJa:= (TagZählJa+1)
30   TagNoch:=   (TagGesamt-TagZählJa)
31   IF          (TagTab[TagPos]="Montag")
       WochZähl:= (WochZähl+1)
   EndIf
32   If          (TagTab[TagPos]="Sonntag")
       FeierKennz:= "F"
   Else
       Feierkennz:= "W"
   EndIf
33   IF          (TagZählMo<=LetzterTag[M])
       Call      (U06Schreiben)
   EndIf
34   If          (TagPos=7)
       TagPos:=  1
   Else
       TagPos:=  (TagPos+1)
   EndIf
35   Until       (TagZählMo>LetzterTag[M])
36   If          (TagPos=1)
       TagPos:=  7
   Else
       TagPos:=  (TagPos-1)
   EndIf
37   IF          (TagTab[TagPos]="Montag")
       WochZähl:= (WochZähl-1)
   EndIf

```

```

38   TagZählJa:=      (TagZählJa-1)
39   EndFor
   Type              ("separieren")
40   Return
41   Label              (U030sternBerechnen)
   Jahr:=              StrNum(DruJahr)
   R19:=               (Jahr MOD 19)
   R07:=               (Jahr MOD 7)
   Quot:=              (Jahr DIV 100)
   R100:=              (Jahr MOD 100)
   R04A:=              (R100 MOD 4)
   Q04:=               (Quot DIV 4)
   R04B:=              (Quot MOD 4)
   Q03:=               (Quot DIV 3)
   J:=                 (Quot-Q03-Q04+15)
   K:=                 (Quot-Q04+4)
   J:=                 (J MOD 30)
   k:=                 (K MOD 7)
   R30:=               19*R19+J
   R30:=               (R30 MOD 30)
   R07:=               4*R07+2*R04A+6*R30+K
   R07:=               (R07 MOD 7)
   Ostern:=            R07+R30
   If                  (Ostern<10)
       Ostern:=        (JanFeb+Ostern+22)
       Return
   EndIf
   Ostern:=            (Ostern-9)
   If                  (Ostern=26)
       Ostern:=        (Ostern-7)
   Else
       If              ((Ostern=25) And (R19>10) And (R30=28))
           Ostern:=    (Ostern-7)
       EndIf EndIf
   Ostern:=            JanFebMrz + Ostern
   Return
42   Label              (U04VariableFeiertage)
43   PosDokTop          ( )
44   FTag:=              (Ostern-48)
45   Call                (U09FTag)
46   Feiertag:=          "Rosenmontag "
47   FeierKennz:=        "N"
48   Call                (U07ErsetzenVariabel)
   FTag:=              (Ostern-47)
   Call                (U09FTag)
   Feiertag:=          "Fastnacht "
   FeierKennz:=        "N"
   Call                (U07ErsetzenVariabel)
   FTag:=              (Ostern-46)
   Call                (U09FTag)
   Feiertag:=          "Aschermittwoch "
   FeierKennz:=        "N"

```

```

Call      (U07ErsetzenVariabel)
Ftag:=    (Ostern-7)
Call      (U09Ftag)
Feiertag:= "Palmsontag "
FeierKennz:= "F"
Call      (U07ErsetzenVariabel)
Ftag:=    Ostern-3
Call      (U09Ftag)
Feiertag:= "Gründonnerstag "
FeierKennz:= "N"
Call      (U07ErsetzenVariabel)
Ftag:=    (Ostern-2)
Call      (U09Ftag)
Feiertag:= "Karfreitag "
FeierKennz:= "F"
Call      (U07ErsetzenVariabel)
Ftag:=    Ostern
Call      (U09Ftag)
Feiertag:= "Ostersonntag "
FeierKennz:= "F"
Call      (U07ErsetzenVariabel)
Ftag:=    (Ostern+1)
Call      (U09Ftag)
Feiertag:= "Ostermontag "
FeierKennz:= "F"
Call      (U07ErsetzenVariabel)
Ftag:=    (Ostern+7)
Call      (U09Ftag)
Feiertag:= "Weißer Sonntag "
FeierKennz:= "F"
Call      (U07ErsetzenVariabel)
Ftag:=    (Ostern+39)
Call      (U09Ftag)
Feiertag:= "Christi Himmelfahrt "
FeierKennz:= "F"
Call      (U07ErsetzenVariabel)
Ftag:=    (Ostern+49)
Call      (U09Ftag)
Feiertag:= "Pfingsten "
FeierKennz:= "F"
Call      (U07ErsetzenVariabel)
Ftag:=    (Ostern+50)
Call      (U09Ftag)
Feiertag:= "Pfingstmontag "
FeierKennz:= "F"
Call      (U07ErsetzenVariabel)
49 SuchText:= (DruJahr+"0105")
SearchString (SuchText)
SearchPrevious (Extended!)
SearchString ("Sonntag")
SearchNext (Extended!)
SearchString ("Sonntag")
SearchNext (Extended!)

```

```

50  PosZeileVor      ( )
    PosZeileVor      ( )
    PosZeileAnfang    ( )
    Schreiben         ("Mutterttag ")
    FTag:=            (Ostern+60)
    Call              (U09FTag)
    Feiertag:=        "Fronleichnam "
    FeierKennz:=      "F"
    Call              (U07ErsetzenVariabel)
51  SuchText:=       (DruJahr+"0110")
    SuchString         (GesuchterString:SuchText)
    SuchenVorwärts     (SuchModus:Erweitert!)
    SuchString         (GesuchterString:"Sonntag")
    SuchenVorwärts     (SuchModus:Erweitert!)
52  PosLineDown      ( )
    PosLineDwon        ( )
    PosLineBegin       ( )
    Type               ("Erntedankfest ")
53  If                (SchJahr="J")
    SuchTag:=          359
    Else
    SuchTag:=          358
    EndIf
54  SuchText:=        NumStr(SuchTag)
    SuchText:=         (SuchText+"#")
    SearchString       (SuchText)
    SearchNext         (Extended!)
55  PosLineDown      ( )
    PosLineDown        ( )
    PosLineBegin       ( )
56  SelectLineEnd     ( )
57  SuchText:=        ?SelectedText
    PosLineBegin       ( )
58  If                (SuchText="Sonntag")
    VierterAdvent:=    SuchTag
    Go                  (U04Volkstrauertag)
    EndIf
    If                 (SuchText="Montag")
    VierterAdvent:=    (SuchTag-1)
    Go                  (U04Volkstrauertag)
    EndIf
    If                 (SuchText="Dienstag")
    VierterAdvent:=    (SuchTag-2)
    Go                  (U04Volkstrauertag)
    EndIf
    If                 (SuchText="Mittwoch")
    VierterAdvent:=    (SuchTag-3)
    Go                  (U04Volkstrauertag)
    EndIf
    If                 (SuchText="Donnerstag")
    VierterAdvent:=    (SuchTag-4)
    Go                  (U04Volkstrauertag)

```



```

EndIf
If          (SuchText="Freitag")
  VierterAdvent:= (SuchTag-5)
  Go        (U04Volkstrauertag)
EndIf
If          (SuchText="Samstag")
  VierterAdvent:= (SuchTag-6)
  Go        (U04Volkstrauertag)
EndIf
59 Label    (U04Volkstrauertag)
  SuchText:= (DruJahr+"0111")
  SearchString (SuchText)
  SearchPrevious (Extended!)
60 FTag:=    (VierterAdvent-35)
  Call        (U09FTag)
  Feiertag:=  "Volkstrauertag"
  FeierKennz:= "F"
  Call        (U07ErsetzenVariabel)
  FTag:=      (VierterAdvent-32)
  Call        (U09FTag)
  Feiertag:=  "Buß- und Betttag"
  FeierKennz:= "F"
  Call        (U07ErsetzenVariabel)
  FTag:=      (VierterAdvent-28)
  Call        (U09FTag)
  Feiertag:=  "Totensonntag"
  FeierKennz:= "F"
  Call        (U07ErsetzenVariabel)
  FTag:=      (VierterAdvent-21)
  Call        (U09FTag)
  Feiertag:=  "1. Advent"
  FeierKennz:= "F"
  Call        (U07ErsetzenVariabel)
  FTag:=      (VierterAdvent-14)
  Call        (U09FTag)
  Feiertag:=  "2. Advent"
  FeierKennz:= "F"
  Call        (U07ErsetzenVariabel)
  FTag:=      (VierterAdvent-7)
  Call        (U09FTag)
  Feiertag:=  "3. Advent"
  FeierKennz:= "F"
  Call        (U07ErsetzenVariabel)
  FTag:=      (VierterAdvent)
  Call        (U09FTag)
  Feiertag:=  "4. Advent"
  FeierKennz:= "F"
  Call        (U07ErsetzenVariabel)
Return
61 Label    (U05FixeFeiertage)
62 PosDokTop ( )
63 InsertTypeOver (Off!)
//Die Feiertage müssen nach Datum sortiert sein!!!

```

```

64   Feiertag:=      "Neujahr"
65   FeierKennz:=    "F"
66   Feierdatum:=    (DruJahr+"01"+"01")
67   Call            (U08ErsetzenFix)
    Feiertag:=      "Hl. drei Könige"
    FeierKennz:=    "N"
    Feierdatum:=    (DruJahr+"06"+"01")
    Call            (U08ErsetzenFix)
    Feiertag:=      "Mariä Lichtmess "
    FeierKennz:=    "N"
    Feierdatum:=    (DruJahr+"02"+"02")
    Call            (U08ErsetzenFix)
    Feiertag:=      "Valentinstag  "
    FeierKennz:=    "N"
    Feierdatum:=    (DruJahr+"14"+"02")
    Call            (U08ErsetzenFix)
    Feiertag:=      "Josefstag  "
    FeierKennz:=    "N"
    Feierdatum:=    (DruJahr+"19"+"03")
    Call            (U08ErsetzenFix)
    Feiertag:=      "Frühlingsanfang "
    FeierKennz:=    "N"
    If              (SchJahr="J")
        Feierdatum:= (DruJahr+"20"+"03")
    Else
        Feierdatum:= (DruJahr+"21"+"03")
    EndIf
    Call            (U08ErsetzenFix)
    Feiertag:=      "Mariä Verkündigung "
    FeierKennz:=    "N"
    Feierdatum:=    (DruJahr+"25"+"03")
    Call            (U08ErsetzenFix)
    Feiertag:=      "Mai-Feiertag  "
    FeierKennz:=    "F"
    Feierdatum:=    (DruJahr+"01"+"05")
    Call            (U08ErsetzenFix)
    Feiertag:=      "Eisheilige (11.-13.) "
    FeierKennz:=    "N"
    Feierdatum:=    (DruJahr+"11"+"05")
    Call            (U08ErsetzenFix)
    Feiertag:=      "Kalte Sophie "
    FeierKennz:=    "N"
    Feierdatum:=    (DruJahr+"15"+"05")
    Call            (U08ErsetzenFix)
    If              ((Jahr>1953) And (Jahr<1990))
        Feiertag:=   "Tag der Deutschen Einheit "
        FeierKennz:= "F"
        Feierdatum:= (DruJahr+"17"+"06")
        Call         (U08ErsetzenFix)
    EndIf
    Feiertag:=      "Sommeranfang "
    FeierKennz:=    "N"

```

```

Feierdatum:= (DruJahr+"21"+"06")
Call (U08ErsetzenFix)
Feiertag:= "Johannistag "
FeierKennz:= "N"
Feierdatum:= (DruJahr+"24"+"06")
Call (U08ErsetzenFix)
Feiertag:= "Siebenschläfer "
FeierKennz:= "N"
Feierdatum:= (DruJahr+"27"+"06")
Call (U08ErsetzenFix)
Feiertag:= "Peter und Paul "
FeierKennz:= "N"
Feierdatum:= (DruJahr+"29"+"06")
Call (U08ErsetzenFix)
Feiertag:= "Mariä Himmelfahrt "
FeierKennz:= "N"
Feierdatum:= (DruJahr+"15"+"08")
Call (U08ErsetzenFix)
Feiertag:= "Mariä Geburt "
FeierKennz:= "N"
Feierdatum:= (DruJahr+"08"+"09")
Call (U08ErsetzenFix)
Feiertag:= "Herbstanfang "
FeierKennz:= "N"
If (SchJahr="J")
    Feierdatum:= (DruJahr+"22"+"09")
Else
    Feierdatum:= (DruJahr+"23"+"09")
EndIf
Call (U08ErsetzenFix)
If (Jahr>1989)
    Feiertag:= "Tag der Dt.Einheit"
    FeierKennz:= "F"
    Feierdatum:= (DruJahr+"03"+"10")
    Call (U08ErsetzenFix)
EndIf
Feiertag:= "Reformationstag "
FeierKennz:= "N"
Feierdatum:= (DruJahr+"31"+"10")
Call (U08ErsetzenFix)
Feiertag:= "Allerheiligen "
FeierKennz:= "N"
Feierdatum:= (DruJahr+"01"+"11")
Call (U08ErsetzenFix)
Feiertag:= "Allerseelen "
FeierKennz:= "N"
Feierdatum:= (DruJahr+"02"+"11")
Call (U08ErsetzenFix)
Feiertag:= "Martinstag "
FeierKennz:= "N"
Feierdatum:= (DruJahr+"11"+"11")
Call (U08ErsetzenFix)

```

```

Feiertag:=      "Nikolaustag "
FeierKennz:=    "N"
Feierdatum:=    (DruJahr+"06"+"12")
Call            (U08ErsetzenFix)
Feiertag:=      "Mariä Empfängnis "
FeierKennz:=    "N"
Feierdatum:=    (DruJahr+"08"+"12")
Call            (U08ErsetzenFix)
Feiertag:=      "Winteranfang "
FeierKennz:=    "N"
If              (SchJahr="J")
    Feierdatum:= (DruJahr+"21"+"12")
Else
    Feierdatum:= (DruJahr+"22"+"12")
EndIf
Call            (U08ErsetzenFix)
Feiertag:=      "Heiligabend "
FeierKennz:=    "N"
Feierdatum:=    (DruJahr+"24"+"12")
Call            (U08ErsetzenFix)
Feiertag:=      "1. Weihnachtstag "
FeierKennz:=    "F"
Feierdatum:=    (DruJahr+"25"+"12")
Call            (U08ErsetzenFix)
Feiertag:=      "2. Weihnachtstag "
FeierKennz:=    "F"
Feierdatum:=    (DruJahr+"26"+"12")
Call            (U08ErsetzenFix)
Feiertag:=      "Silvester "
FeierKennz:=    "N"
Feierdatum:=    (DruJahr+"31"+"12")
Call            (U08ErsetzenFix)
Return
68 Label        (U06Schreiben)
69 DruTag:=      NumStr(TagZählMo)
    DruTagJahr:= NumStr(TagZählJa)
    TageNoch:=   (TagGesamt-TagZählJa)
    DruTageNoch:= NumStr(TageNoch)
    DruWoch:=    NumStr(WochZähl)
    DruMonNummer:= NumStr(MonNummer)
70 Datum:=      (DruJahr)
71 If           (TagZählMo<10)
    Datum:=     (Datum+"0")
EndIf
72 Datum:=      (Datum+DruTag)
73 If           (MonNummer<10)
    Datum:=     (Datum+"0")
EndIf
74 Datum:=      (Datum+DruMonNummer)
75 Type         (Datum) MergeEndField
    Type        (DruTag) MergeEndField
    Type        (DruMonNummer) MergeEndField

```

```

Type                (DruJahr) MergeEndField
Type                (FeierKennz) MergeEndField
76  If              (WochZähl=0)
    Type            ("(Vorjahreswoche)")
    Else
        Type        (DruWoch)
    EndIf
MischenEndeFeld
DruTagJahr:=        (DrutagJahr+"#")
Type                (DruTagJahr) MergeEndField
Type                (DruTageNoch) MergeEndField
Type                (TagTab[TagPos]) MergeEndField
Type                (Monat[M]) MergeEndField
MergeEndField
77  MergeEndRecord
78  Return
79  Label            (U07ErsetzenVariabel)
80  SearchString     (FeierDatum)
    SerchNext        (Extended!)
81  PosLineDown      ()
    PosLineDown      ()
    PosLineDown      ()
    PosLineDown      ()
    PosLineBegin      ()
    Type              (FeierTag)
82  Meldung:=        (Feiertag+"           Bitte nicht auf [OK] klicken.")
    Prompt            ("Variable Feier- und Gedenktage eintragen";Meldung;;;)
83  If              (FeierKennz="N")
    Return
    EndIf
84  PosLineBegin      ()
    PosLineUp         ()
    PosLineUp         ()
    PosLineUp         ()
    PosLineUp         ()
    PosLineUp         ()
    PosLineUp         ()
    PosLineUp         ()
    PosLineBegin      ()
85  DeleteCharNext    ()
    Type              (FeierKennz)
86  Return
87  Label            (U08ErsetzenFix)
88  SearchString     (Feierdatum)
    SerachNext        (Extended!)
89  PosLineDown      ()
    PosLineDown      ()
    PosLineDown      ()
    PosLineDown      ()
    PosLineDown      ()
    PosLineDown      ()
    PosLineDown      ()
    PosLineDown      ()

```

```

PosLineDown      ( )
PosLineDown      ( )
PosLineBegin     ( )
Type             (FeierTag)
90 Meldung:=      (Feiertag+"      Bitte nicht auf [OK] klicken.")
Prompt           ("Fixe Feier- und Gedenktage eintragen";Meldung;;;)
91 If             (FeierKennz="N")
    Return
EndIf
92 PosLineBegin   ( )
PosLineUp        ( )
PosLineUp        ( )
PosLineUp        ( )
PosLineUp        ( )
PosLineUp        ( )
PosLineUp        ( )
PosLineUp        ( )
PosLineBegin     ( )
93 DeleteCharNext ( )
Type             (FeierKennz)
94 Return
95 Label          (U09FTag)
LfdTag:=          NumStr(FTag)
Feierdatum:=      (LfdTag+"#")
Return
96 Label          (U10NumZeichen)
PosDokTop        ( )
Prompt           ("Kalender berechnen";"Nummernzeichen # löschen. Bitte nicht
auf [OK] klicken.";;;)
SearchString     ("#")
ReplaceString     ("" )
ReplaceForward    (Extended!)
EndPrompt
PosDokTop        ( )
Return
97 Label          (U12Separieren)
98 Prompt         ("Kalender berechnen";
"Datendatei (komplett) speichern. Bitte nicht auf [OK] klicken.";;;)
Wait             (30)
99 FileSave       (FileName:KalebereDat;
Overwrite:Yes!)
100 ForNext       (Q;1;4;1)
101   PosDokTop    ( )
102   SearchString ("separieren")
SearchNext       (Extended!)
PosLineUp        ( )
PosLineEnd       ( )
103   SelectMode   (On!)
PosDokTop        ( )
104   Meldung:=    ("Datendatei (" +Q+" . Quartal) speichern. Bitte nicht auf [OK]
klicken.")
Prompt           ("Kalender berechnen";Meldung;;;)
105   SelectSave   (FileName: KalebereQ+Q)

```

```

106      SelectMode      (Off!)
107      PosDocTop        ( )
      PosPageNext      ( )
108      SearchString    ("separieren")
      MatchExtendSelection()
      SearchNext        (Extended!)
109      DeleteCharNext   ( )
      DeleteCharNext   ( )
110      EndFor
      Close              ( )
      Return
111      Label              (U77OnError)
      Beep Beep Beep
      Prompt              ("Fehler bei Kalenderberechnung";"Es trat eine externe
                          Fehlerbedingung auf.";;;)

      Pause
      EndPrompt
      Quit
112      Label              (U88NichtGefunden)
      Beep Beep Beep
      Meldung:=           ("Datum für Zuordnung eines Feier- oder Gedenktages wurde
                          nicht gefunden. Kalender wurde evtl. falsch berechnet, die
                          Logik des Makros wurde geändert oder es wurde ein neuer Tag
                          nicht in der richtigen Reihenfolge eingefügt. Bitte Makro
                          prüfen. Suchstring:= "+Feierdatum+"/"+Suchtext")
      Prompt              ("Fehler bei Kalenderberechnung";Meldung;;;)
      Pause
      EndPrompt
      Quit
113      Label              (U99Ende)
      Meldung:=           ("Ende der Kalenderberechnung für "+DruJahr+". Rufen Sie das Makro
                          KALENDER auf. Drucken Sie den Terminkalender quartalsweise über den
                          Menüpunkt '"Terminkalender erstellen"'")

      Beep Beep Beep
      Prompt              ("Kalender berechnen";Meldung;;;)
      Wait                (50)
      EndPrompt
      Quit

```

## Erklärung

- 1 Definition von Dateinamen (= Datendateien), die nach Ende der Berechnung erstellt werden. Dateinamen sollten Sie immer am Beginn eines Makros in Variablen definieren. Sie brauchen dann bei Änderungen dieser Namen nicht das ganze Makro abzusuchen.
- 2 Die hier genannten Variablen werden von dem aufrufenden Makro KALENDER.WCM zur Kalenderberechnung übergeben.

Bei der Kompilierung dieses Makros erfolgt bei der anschließenden Ausführung eine falsche Berechnung des Kalenders, weil die Übergabeveriablen nicht vorhanden sind. Dadurch können die definierten Arrays nicht ordnungsgemäß abgearbeitet

werden. Brechen Sie in diesem Fall das Makro ab, sofern es nicht bereits mit einer Fehlermeldung abgebrochen wurde, und starten Sie es über das Makro KALENDER.WCM. Schließen Sie geöffnete Dateien, ohne diese zu speichern!.

Über *VarErrChk* werden Fehlermeldungen unterdrückt, wenn Variablen nicht vorhanden sind (Übergabevariable aus KALENDER.WCM).

- 3 Definition von Arrays zum Schreiben von Textkonstanten (Tages- und Monatsnamen) und zur Berechnung eines Monats (Anzahl der Tage eines Monats).
- 4 Beim Auftreten eines Fehlers soll das Makro nach *U77OnError* verzweigen.
- 5 Beim Auftreten einer »Nicht gefunden«-Bedingung soll das Makro nach *U88NichtGefunden* verzweigen.
- 6 Hauptsteuerung des Makros. Die definierten Unterrouтины werden in der angegebenen Reihenfolge aufgerufen.
- 7 Nach dem Abarbeiten aller Unterrouтины verzweigt das Makro zum Ende.
- 8 Beginn des Labels *U01Datendatei*. In dieser Unterroutine werden die Felddefinitionen für die Erstellung der Datendatei vorgenommen.
- 9 Das Anzeigen der Erstellung wird unterdrückt, um Zeit zu sparen.
- 10 Öffnen eines neuen (leeren) Dokumentfensters.
- 11 *Überschreiben* wird eingeschaltet.
- 12 Entwurfmodus wird eingeschaltet.
- 13 Auswählen einer kleinen Schrift, sonst erscheinen beim Einfügen der Mischbefehle Trennvorschläge.
- 14 Definition von Feldnamen für die zu erstellende Misch-Datendatei.
- 15 Cursor am Dokumentende positionieren.
- 16 Kleine Schrift wieder ausschalten.
- 17 Beginn der Unterroutine *U02KalenderBerechnen*.
- 18 Ermittlung der Woche 1 des Jahres. Als erste Woche wird die Woche festgelegt, die den ersten Donnerstag des Jahres enthält.
- 19 Die Variable zum Zählen der Tage pro Jahr wird am Jahresanfang auf Null gesetzt.
- 20 Variable zum Zählen der Monate auf 0 setzen.
- 21 Festlegung der Gesamttage eines Jahres: Bei Schaltjahren 366, ansonsten 365. Für die Berechnung von Ostern wird die Summe der Tage von Januar und Februar sowie von Januar, Februar und März in Abhängigkeit eines Schaltjahres benötigt.
- 22 Über *ForNext* wird die gesamte Kalenderberechnung gesteuert. Für jeden einzelnen Monat werden die nachfolgenden Befehle bis zum letzten *EndFor* ausgeführt.
- 23 Erhöhen der Variablen *MonNummer* (zählen der Monate) um 1.
- 24 Die Variable zum Zählen der Tage pro Monat wird am Monatsanfang auf Null gesetzt.



- 25 Generieren einer Meldung, die über den nachfolgenden *Prompt*-Befehl den Fortgang der Erstellung des Kalenders anzeigt. Klicken Sie nicht auf [OK], weil die Meldung bei der Erstellung des nächsten Monats automatisch überschrieben wird.
- 26 Am Quartalsende wird eine Kennung in die Datendatei eingefügt, damit am Ende der Berechnung die Datendatei in vier Teile (Quartale) gegliedert werden kann.
- 27 Beginn einer *Repeat*-Schleife. Sie wird so lange wiederholt, bis ein kompletter Monat erstellt wurde.
- 28 Der Tageszähler pro Monat wird um 1 erhöht.
- 29 Der Tageszähler pro Jahr wird um 1 erhöht.
- 30 Die ab dem errechneten Datum noch verbleibenden Tage werden ermittelt.
- 31 Beim Wochenwechsel (immer am Montag) wird der Wochenzähler um 1 erhöht.
- 32 Sonntage werden als Feiertage (F) gekennzeichnet, die restlichen Tage der Woche als Werkzeuge (W).
- 33 Aufruf der Unteroutine *U06Schreiben*, wenn der Monat noch nicht komplett abgearbeitet wurde.
- 34 Wurde der letzte Tag aus der Tagedstabelle selektiert, muß wieder mit dem ersten Tag begonnen werden. Die Tabellenposition wird neu ermittelt.
- 35 Ende der *Repeat*-Schleife.
- 36 Bestimmen der Tabellenposition, da in der *Repeat*-Schleife einmal zuviel addiert wurde (kommt dadurch, daß bei *Repeat* die Endbedingung am Ende der Schleife definiert ist).
- 37 Neubestimmen des Wochenzählers, wenn der Monatsanfang ein Montag war, da in der *Repeat*-Schleife einmal zuviel addiert wurde (kommt dadurch, daß bei *Repeat* die Endbedingung am Ende der Schleife definiert ist).
- 38 Neubestimmen des Tageszählers pro Jahr, da in der *Repeat*-Schleife einmal zuviel addiert wurde (kommt dadurch, daß bei *Repeat* die Endbedingung am Ende der Schleife definiert ist).
- 39 Ende der *ForNext*-Schleife.
- 40 Rücksprung zum aufrufenden Befehl.
- 41 Beginn der Unteroutine *U03OsternBerechnen*. In dieser Unteroutine wird aufgrund der Gauß'schen Formel der Ostersonntag berechnet. Basis für die Berechnung ist das eingegebene Jahr, für das ein Kalender berechnet werden soll. Die Erklärung der einzelnen Befehle ist hier nicht erforderlich, da sie selbsterklärend sind.
- 42 Beginn der Unteroutine *U04VariableFeiertage*. Aufgrund der Berechnung des Ostersonntags in *U30OsternBerechnen* werden alle beweglichen kirchlichen Feiertage des ersten Halbjahres berechnet. Die Berechnung wird aufgrund der laufenden Tagesnummer durchgeführt.
- 43 Cursor am Dokumentanfang positionieren, um über »Suchen und Ersetzen« die errechneten Feiertage einzutragen.

- 44 Errechnen der Tagesnummer, für einen Feiertag, immer von Ostersonntag ausgehend. In dieser Zeile wird der Rosenmontag ermittelt, der 48 Tage vor Ostern liegt.
- 45 Aufruf der Unteroutine *U09Ftag* (Beschreibung siehe unten).
- 46 Übertragen des Namens des Feiertags/Gedenktags in die Variable *Feiertag*.
- 47 Festlegen, ob der betreffende Tag ein Feiertag (F) oder ein Werktag (W) ist.
- 48 Aufrufen der Unteroutine *U07ErsetzenVariabel*. Über diese Unteroutine wird der zugehörige Satz in der Datendatei gesucht und entsprechend geändert (Eintragung der beweglichen Feiertage).
- Die nachfolgenden Blöcke der einzelnen Feiertage entsprechen in der Funktion dem zuvor beschriebenen Block. Einziger Unterschied ist das ermittelte Datum und der Name des Feiertags/Gedenktags. Auf eine Beschreibung jedes einzelnen Blocks wird darum verzichtet.
- 49 Ermitteln des Muttertages. Der Muttertag ist immer der zweite Sonntag im Mai. Die Suche erfolgt in der Datendatei, ab dem 1. 5. beginnend. Sie wird zweimal gestartet.
- 50 Cursor in der Datendatei auf dem Feiertagsfeld plazieren und Eintragen des Feiertags.
- 51 Ermitteln des Entedankfestes. Das Erntedankfest ist immer der erste Sonntag im Oktober. Die Suche erfolgt in der Datendatei, ab dem 1. 10. beginnend.
- 52 Cursor in der Datendatei auf dem Feiertagsfeld plazieren und Eintragen des Feiertages.
- 53 Die beweglichen kirchlichen Feiertage des letzten Quartals richten sich immer nach Heiligabend (24.12.). Je nachdem, ob es ein Schaltjahr ist oder nicht, ist das der 358. oder der 359. Tag.
- 54 Heiligabend suchen. Die laufende Tagesnummer wird in der Datendatei durch ein # gefolgt, um bei der Zuordnung Fehler zu vermeiden. Dieses Zeichen wird hier der Tagesnummer angehängt, um einen eindeutigen Suchbegriff zu erzeugen.
- 55 Nachdem aufgrund der laufenden Tagesnummer Heiligabend gefunden wurde, muß ermittelt werden, auf welchen Wochentag Heiligabend fällt. Der Cursor wird in der Datendatei auf dem Feld *Tagesname* plaziert.
- 56 Markieren des Wochentages.
- 57 Der Wochentag wird in der Variablen *Suchtext* gespeichert.
- 58 Aufgrund des Inhaltes von *Suchtext* wird der vierte Advent ermittelt. Von diesem Wert ausgehend, errechnen sich die restlichen beweglichen kirchlichen Feiertage des letzten Quartals.
- 59 Beginn des Labels *U04Volkstrauertag*. Der Cursor wird in der Datendatei auf den 1.11. zurückgesetzt, um von hier aus die noch ausstehenden variablen Feiertage einzutragen.
- 60 Siehe Zeilennummer 44–48.
- 61 Beginn der Unteroutine *U05FixeFeiertage*. Hier werden alle festen Feier- und Gedenktage, d. h. Tage, die immer auf dasselbe Datum fallen, eingetragen. Hier können Sie eigene Feiertage wie Geburtstag, Hochzeitstag usw. eintragen.

Halten Sie bitte die Sortierfolge nach dem Datum ein, sonst wird die Berechnung abgebrochen!!

Die eingetragenen Feiertage gelten in Deutschland für das Bundesland Hessen. Löschen Sie solche Feiertage, die in Ihrem Bundesland nicht zutreffen bzw. ergänzen Sie fehlende.

- 62 Cursor am Beginn des Dokuments positionieren.
- 63 Überschreiben ausschalten. Wenn ein fixer Feiertag auf dasselbe Datum wie ein variabler Feiertag fällt (z.B. Heiligabend und 4. Advent), darf der bereits eingetragene Feiertag nicht überschrieben werden.
- 64 Der Variablen *Feiertag* wird der einzutragende Feiertag zugeordnet.
- 65 Festlegen, ob der betreffende Tag ein Feiertag (F) oder ein Werktag (W) ist.
- 66 Datum generieren, bei dem in der Datendatei der Feiertag einzutragen ist. Dieses generierte Datum dient als Suchkriterium. Es wird immer im ersten Feld eines Satzes der Datendatei gefunden. Das Datum setzt sich zusammen aus Jahr, Tag und Monat (z.B. 19941411, siehe unten).
- 67 Aufruf der Unteroutine *U08ErsetzenFix*. Über diese Unteroutine wird der zugehörige Satz in der Datendatei gesucht und entsprechend geändert.  
Die nachfolgenden Blöcke der einzelnen Feiertage entsprechen in der Funktion dem zuvor beschriebenen Block. Einziger Unterschied ist das ermittelte Datum und der Name des Feiertages. Auf eine Beschreibung jedes einzelnen Blockes wird darum verzichtet.
- 68 Beginn der Unteroutine *U06Schreiben*. Für jeden Tag des Jahres wird ein Datensatz im Datendatei-Format erstellt.
- 69 Zum Schreiben müssen die numerischen Variablen in Strings konvertiert werden.
- 70 Damit das Suchen und Ersetzen in Verbindung mit den Feiertagen funktioniert, muß ein eindeutiger Suchbegriff definiert werden. Dieser setzt sich aus der Jahreszahl, der Tageszahl und der Monatszahl zusammen. Der 1. 3. 1993 würde darum wie folgt generiert: 19930103. In diesem Befehl wird in *Datum* das Jahr übernommen.
- 71 Ist eine Tagesnummer kleiner als 10, wird eine Null vorangestellt.
- 72 Das Datum besteht jetzt aus Jahr und Tag (6 Stellen).
- 73 Ist eine Monatsnummer kleiner als 10, wird eine Null vorangestellt.
- 74 Das Datum besteht jetzt aus Jahr, Tag und Monat (8 Stellen).
- 75 Das Datum wird in die Datei geschrieben. Am Ende des Datums wird ein Mischbefehl *MischenEndeFeld* angehängt.  
Die nachfolgenden Befehle sind nahezu identisch. Es wird immer abwechselnd eine Variable in die Datendatei geschrieben und ein Mischbefehl angehängt. Auf eine weitere Beschreibung wird darum verzichtet.
- 76 Wurde eine Woche mit 0 angegeben, hat das neue Jahr mit einem Freitag begonnen. Die ersten Tage des Jahres bis einschließlich zum nächsten Sonntag zählen darum zur letzten Woche des vorausgegangenen Jahres.

- 77 Am Ende eines Satzes der Datendatei wird der Mischbefehl *EndeDatensatz* angehängt. Er erzeugt gleichzeitig in der Datendatei einen festen Seitenumbruch.
- 78 Rücksprung zum aufrufenden Befehl.
- 79 Beginn der Unterroutine *U07ErsetzenVariabel*. Die in *U04VariableFeiertage* initialisierten Variablen werden in den ausgewählten Satz der Datendatei übertragen.
- 80 Die Variable *Feierdatum* enthält das zu suchende Datum in der Form *TTT#* (laufende Tagesnummer, ergänzt um #, z.B. 165#). Über *SuchenText* wird der zugehörige Datensatz gesucht.
- 81 Zum Eintragen des Feiertags muß der Cursor um vier Zeilen nach unten gesetzt werden und dann an den Zeilenanfang.
- 82 Der gerade eingetragene Feiertag wird über *Prompt* am Bildschirm angezeigt.
- 83 Enthält die Variable *FeierKennz* ein N, wird diese Kennzeichnung nicht übernommen, um ein evtl. in diesem Datensatz gesetztes F (= Sonn- oder Feiertag) nicht zu überschreiben. Die Unterroutine wird in diesem Fall beendet.
- 84 Um das Feiertagskennzeichen einzutragen, muß der Cursor um sechs Zeilen nach oben versetzt werden und dann an den Zeilenanfang.
- 85 Das bisherige Feiertagskennzeichen wird gelöscht und das neue übertragen.
- 86 Rücksprung zum aufrufenden Befehl.
- 87 Beginn der Unterroutine *U08ErsetzenFix*. Die in *U05FixeFeiertage* initialisierten Variablen werden in dieser Routine in den ausgewählten Satz der Datendatei übertragen.
- 88 Die Variable *Feierdatum* enthält das zu suchende Datum in der Form *JJJJTTMM* (z. B. 19962005). Über *SuchenText* wird der zugehörige Datensatz gesucht.
- 89 Zum Eintragen des Feiertags muß der Cursor um zehn Zeilen nach unten gesetzt werden und dann an den Zeilenanfang.
- 90 Der gerade eingetragene Feiertag wird über *Prompt* am Bildschirm angezeigt.
- 91 Enthält die Variabel *FeierKennz* ein N, wird das nicht übernommen, um ein evtl. in diesem Datensatz gesetztes F (= Sonn- oder Feiertag) nicht zu überschreiben. Die Unterroutine wird in diesem Fall beendet.
- 92 Um das Feiertagskennzeichen einzutragen, muß der Cursor um sechs Zeilen nach oben versetzt werden und dann an den Zeilenanfang.
- 93 Das bisherige Feiertagskennzeichen wird gelöscht und das neue übertragen.
- 94 Rücksprung zum aufrufenden Befehl.
- 95 Beginn der Unterroutine *U09FTage*. Diese Routine wird zur Umwandlung des laufenden Tages in einen String, dem ein # angehängt wird, benötigt. Der String wird in der Variablen *FeierDatum* gespeichert, die beim Suchen und Ersetzen als Suchbegriff dient.
- 96 Beginn der Unterroutine *U10NumZeichen*. Das Nummernzeichen diene in Verbindung mit der laufenden Tagesnummer als eindeutiger Suchbegriff. Nach dem Erstellen der Datendatei und dem Eintragen der Feiertage wird es nicht mehr benötigt und wird darum gelöscht, damit es im Terminkalender nicht gedruckt wird.

- 97 Beginn der Unteroutine *U12Separieren*. Hier wird die Datendatei jeweils am Quartalsende getrennt und jedes Quartal für sich gespeichert. Die Trennung erfolgt bei dem Suchbegriff *Separieren*, der beim Erstellen der Datei eingefügt wurde (siehe oben).
- 98 Generieren und Anzeigen einer Meldung (Status der Erstellung).
- 99 Speichern der kompletten Datendatei. Ist noch eine Datei aus einer früheren Kalenderberechnung vorhanden, muß diese überschrieben werden.
- 100 Beginn einer *ForNext*-Schleife, in der für jedes Quartal des Jahres eine eigene Datendatei erstellt wird.
- 101 Cursor am Dateianfang vor allen Steuerzeichen positionieren.
- 102 Jedes Quartal wird durch den Text *separieren* begrenzt. Dieser Text ist zu suchen. Der Cursor wird am Zeilenende der vorausgegangenen Zeile positioniert.
- 103 Blockfunktion einschalten und Rücksprung zum Dateianfang, um ein Quartal als Block zu markieren.
- 104 Hinweis am Bildschirm anzeigen.
- 105 Speichern des markierten Blocks. Ist noch eine Datei aus einer früheren Kalenderberechnung vorhanden, wird diese überschrieben, um den Terminkalender des neu berechneten Kalenders erstellen zu können.
- 106 Blockmarkierung ausschalten.
- 107 Cursor an den Dateianfang bringen und eine Seite vorwärts blättern.
- 108 Den Text *separieren* suchen.
- 109 Den Text *separieren* löschen.
- 110 Ende der *ForNext*-Schleife. Nach dem Speichern der vier Quartalsdateien wird die Datendatei geschlossen, ohne zu speichern.
- 111 Beginn der Unteroutine *U77OnError* (Aufruf externe Fehler erkannt werden).
- 112 Beginn der Unteroutine *U88NichtGefunden*. Diese wird dann aufgerufen, wenn z.B. ein zu suchendes Datum (siehe oben) nicht gefunden wird. Das wird dann vorkommen, wenn Sie das Makro vielleicht geändert haben, und dann die Logik nicht mehr stimmt, oder wenn Sie weitere Feiertage in der falschen Reihenfolge eingefügt haben.
- 113 Beginn des Labels *U99Ende*. Hierher wird dann verzweigt, wenn das Makro komplett abgearbeitet wurde.

## 9.3.2 Terminkalender erstellen

**Aufgabenstellung:** Über dieses Makro wird ein Terminkalender erstellt. Voraussetzung ist, daß zuvor über *KALENDER.WCM* das Makro *KALEBERE.WCM* aufgerufen wurde (Menüpunkt »Kalender berechnen«), das die erforderlichen Datendateien erstellt. Dieses Makro mischt die Datendateien (je Quartal des berechneten Jahres eine Datendatei) mit zwei verschiedenen Formulardateien:

eine Formulardatei für Werktage

eine Formulardatei für Feiertage

Wurden die Datendateien nicht erstellt, erfolgt eine Fehlermeldung. Eine genaue Bedienungsanleitung für die Kalender-Makros und Musterabbildungen der einzelnen Kalendertypen finden Sie in dem Kapitel »10 Makro-Sammlung«.

**Schwerpunkte:** Definition von Dialogfeldern, Mischen verschiedener Formular- und Datendateien.

### Makrodefinition: KALETERM.WCM

```

1      Application      (A1; "WordPerfect"; Default; "DE")
2      OnCancel         (Abbruch)
3      OnNotFound       (NichtGefunden)
      OnError           (Fehler)
      Display           (On!)
                        // Bitte Dateinamen ggf. ändern (Name der
                        Kalenderdateien).
4      KalebereQ:=      "C:\WPWIN7\MAKROS\KALEBERE.Q"4
      KaletermFrm:=     "C:\WPWIN7\MAKROS\KALETERM.FRM"
                        // Bitte ggf. auch Dateinamen in der Formulardatei
                        KALETERM.FRM ändern!
5      Label            (Anfang)6 Anzeige:= "1. Quartal "
7      DialogDefine     (100; 50; 50; 200; 130; OK! | Cancel! | Percent!;
                        "Terminkalender erstellen")
8      DialogAddListBox (100; 1000; 10; 10; 100; 38; 0; Anzeige)
9      DialogAddListItem (100; 1000; "1. Quartal")
      DialogAddListItem (100; 1000; "2. Quartal")
      DialogAddListItem (100; 1000; "3. Quartal")
      DialogAddListItem (100; 1000; "4. Quartal")
10     DialogAddText    (100; 1001; 10; 60; 180; 32; 1;
                        "Bevor Sie den Terminkalender erstellen können,
                        müssen Sie ihn über das Makro KALENDER berechnen
                        lassen. Markieren Sie den gewünschten Eintrag, und
                        klicken Sie auf OK.")
11     DialogDisplay    (100; 1000)
12     DialogDestroy    (100)
13     IF               (Anzeige=="")
        Beep
        Prompt          ("Terminkalender erstellen";
                        "Sie haben keinen Eintrag markiert. Klicken Sie bitte
                        auf das zu erstellende Kalender-Quartal.;;;")
        Wait            (40)
        EndPrompt
        Go              (Anfang)
    EndIf
14     IF               (Anzeige="1. Quartal")
        Quartal:=       "1"
    Else

```

```

IF (Anzeige="2. Quartal")
    Quartal:= "2"
Else
IF (Anzeige="3. Quartal")
    Quartal:= "3"
Else
    Quartal:= "4"
EndIf EndIf EndIf
Meldung:= ("Terminkalender für das "+Quartal+". Quartal wird
erstellt.")
Prompt ("Terminkalender erstellen";Meldung;;;)
15 MischDatei:= KalebereQ+Quartal
TabeleGridLinesToggle()
MergeFileAssociate (FormFile: KaletermFrm;
DataFile: MischDatei)
16 MergeSelect (SelectType: Alle!)
MergeRun (FormFileType: FormFile!;
FormFile: KaletermFrm;
DataFileType: DaaFile!;
DataFile: MischDatei;
OutputFileType: ToCurrentDoc!; )
17 PosDocTop ( )
Quit
18 Label (Abbruch)
Prompt ("Terminkalender erstellen";
"Sie haben die Verarbeitung abgebrochen.";;;)
Wait (30)
EndPrompt
Quit
19 Label (NichtGefunden)
Prompt ("Terminkalender erstellen";
"Erstellen Sie erst die Datendatei!";;;)
Wait (30)
EndPrompt
Quit
20 Label (Fehler)
Prompt ("Terminkalender erstellen";
"Es ist ein externer Fehler aufgetreten. Prüfen Sie
bitte die Dateinamen der Daten- und der
Formulardatei.";;;)
Wait (30)
EndPrompt
Quit

```

## Erklärung

- 1 Produktkennung
- 2 Beim Abbruch des Makros wird zu dem Label *Abbruch* verzweigt.
- 3 Sind die zum Erstellen des Terminkalenders erforderlichen Dateien nicht vorhanden, verzweigt das Makro zu dem Label *NichtGefunden*.

- 4 Definition von Dateinamen (= Formular- und Datendateien), die weiter unten über *MischenAusführen* aufgerufen werden. Dateinamen sollten Sie immer am Beginn eines Makros als Variable definieren. Sie brauchen dann bei Änderungen dieser Namen nicht das ganze Makro abzusuchen, wo diese Namen überall vorkommen.
- 5 Beginn der Routine *Anfang*.
- 6 Die Variable *Anzeige* wird mit einer Leerstelle initialisiert. In dieser Variablen wird der ausgewählte Menüpunkt des angezeigten Dialogfeldes gespeichert.
- 7 Definieren des Dialogfelds (Position, Breite und Höhe).
- 8 Listenfeld definieren, das die Auswahlmöglichkeiten beinhaltet.
- 9 Listeneinträge definieren, die in dem Listenfeld angezeigt werden sollen.
- 10 Hilfetext definieren.
- 11 Dialogfeld anzeigen.
- 12 Dialogfeld aus dem Speicher löschen.
- 13 Überprüfen, ob in dem Listenfeld eine Auswahl getroffen wurde. Wenn nicht, erfolgt eine Fehlermeldung und ein Rücksprung zum Beginn des Makros.
- 14 Überprüfen, welcher Eintrag ausgewählt wurde (welches Quartal soll erstellt werden).
- 15 Dateiname für Datendatei generieren.
- 16 Formular- und Datendatei mischen.
- 17 Makro nach Beendigung des Mischvorgangs beenden.
- 18 Zu diesem Label wird verzweigt, wenn in dem angezeigten Dialogfeld [*Abbrechen*] gewählt wurde.
- 19 Zu diesem Label wird verzweigt, wenn eine der benötigten Dateien nicht vorhanden ist.
- 20 Zu diesem Label wird verzweigt, wenn ein Fehler aufgetreten ist (Datei nicht gefunden).

### 9.3.3 Jahreskalender

**Aufgabenstellung:** Über dieses Makro soll ein Jahreskalender gedruckt werden. Das Makro darf nur über KALENDER.WCM aufgerufen werden, sonst wird ein falscher Kalender gedruckt. Die Eingabe der Jahreszahl des zu druckenden Kalenders sowie die Ermittlung des ersten Wochentags, mit dem das zu druckende Jahr beginnt, erfolgt auch über das Makro KALENDER.WCM.

Eine genaue Bedienungsanleitung für die Kalender-Makros und Musterabbildungen der einzelnen Kalendertypen finden Sie in dem Kapitel »10 Makro-Sammlung«.



**Schwerpunkte:** Laden von Dateien, Übernahme von Daten von dem aufrufenden Makro in das aufgerufene Makro, Schleifenverarbeitung, Drucken.

### Makrodefinition: KALEJAHR.WCM

```

1      Application      (A1; "WordPerfect"; Default; "DE")
2      OnError          (U99Ende)
                        //Display (On!)
3      DECLARE (Monat[12])
                        Monat[1]:="Januar"
                        Monat[2]:="Februar"
                        Monat[3]:="März"
                        Monat[4]:="April"
                        Monat[5]:="Mai"
                        Monat[6]:="Juni"
                        Monat[7]:="Juli"
                        Monat[8]:="August"
                        Monat[9]:="September"
                        Monat[10]:="Oktober"
                        Monat[11]:="November"
                        Monat[12]:="Dezember"
                        //Bitte Dateinamen ggf. ändern
4      KaleJahr:=       "C:\KALEJAHR"
5      FileOpen         (Filename:"C:\WPWIN7\MAKROS\KALEJAHR.FRM")
6      //Folgende Variablen werden aus Makro KALENDER übergeben:
                        //Jahr      Jahreszahl
                        //TagPos    Mit welchem Wochentag beginnt das Jahr
                        //SCHJahr    Schaltjahr, N = kein Schaltjahr, J = Schaltjahr
7      VarErrChk        (Off!)
8      SearchchString   ("#Jahr#")
                        MatchSelection
                        SerachNext    (Normal!)
                        ReplaceString (Jahr)
                        ReplaceCurrent
9      PosDocBottom     ()
10     HardReturn       ()
11     If               (TagPos=1)
                        TagPos:=     7
                        Else
                        TagPos:=     TagPos-1
                        EndIf
12     IF               (SchJahr="J")
                        Februar:=    29
                        Else
                        Februar:=    28
                        EndIf
13     ForNext          ( M;1;12;1)
14     IF (Monat[M]="Januar")LetzterTag:=31 Else
                        IF (Monat[M]="Februar")LetzterTag:=Februar Else
                        IF (Monat[M]="März")LetzterTag:=31 Else
                        IF (Monat[M]="April")LetzterTag:=30 Else

```

```

    IF (Monat[M]="Mai")LetzterTag:=31 Else
    IF (Monat[M]="Juni")LetzterTag:=30 Else
    IF (Monat[M]="Juli")LetzterTag:=31 Else
    IF (Monat[M]="August")LetzterTag:=31 Else
    IF (Monat[M]="September")LetzterTag:=30 Else
    IF (Monat[M]="Oktober")LetzterTag:=31 Else
    IF (Monat[M]="November")LetzterTag:=30 Else
    IF (Monat[M]="Dezember")LetzterTag:=31 Else
        EndIf EndIF EndIf EndIF EndIF EndIf EndIF
    EndIf EndIf EndIf EndIf EndIf
15  Meldung:=(Jahr+", Monat "+Monat[M]+" wird erstellt. Bitte
    nicht auf [OK] klicken.")
    Prompt      ("Kalender berechnen";Meldung;;;)
16  LineCenter  ()
17  FontBold    (On!)
18  Type        (Monat[M])
19  FontBold    (Off!)
20  HardReturn  ()
20  HardReturn  ()
21  Call        (U01Überschrift)
22  Call        (U02DruckPos)
23  For         (TagZähler;1;TagZähler<=LetzterTag;TagZähler+1)
24  DruTag:=    NumStr(TagZähler)
25      If      (TagPos=7)
            FontBold    (On!)
            Type        (DruTag)
            FontBold    (Off!)
        Else
            Type        (DruTag)
        EndIf
26  TagPos:=    TagPos+1
27  IF          (TagPos>7)
        HardReturn  ()
        Tab        ()
        TagPos:=    1
    Else
        Tab        ()
    EndIf
28  EndFor
29  HardPageBreak  ()
30  HardReturn    ()
31  EndFor
32  Go            (U99Ende)
33  Label        (U01Überschrift)
34  ForEach      (TagName;{"Mo";"Di";"Mi";"Do";"Fr";"Sa";"So"})
35      Tab      ()
36      If      (TagName="So")
            FontBold    (On!)
            Type        (TagName)
            FontBold    (Off!)
        Else
            Type        (TagName)

```

```

        EndIf
    EndFor
37    HardReturn      ( )
37    HardReturn      ( )
38    Return
39    Label            (U02DruckPos)
40    For              (Anzahl;1;Anzahl<=TagPos;Anzahl+1)
41        Tab          ( )
42    EndFor
43    Return
44    Label            (U99Ende)
45    DraftMode        (Off!)
46    PosDokTop        ( )
47    FileSave         (FileName:KaleJahr;
                       OverWrite:Yes!)
48    Display          (On!)
49    Meldung:=        ("Kalender für "+Jahr+" wurde erstellt.")
50    Beep Beep Beep
51    Prompt           ("Jahreskalender drucken";Meldung;;)
    Wait              (30)
    EndPrompt
52    Quit

```

## Erklärung

- 1 Produktkennung
  - 2 Beim Auftreten von Fehlern soll das Programm zu *U99Ende* verzweigen.
  - 3 Definition der Textkonstanten für die Monatsnamen.
  - 4 Definition von Dateinamen (= Speicherung des errechneten Kalenders). Dateinamen sollten Sie immer am Beginn eines Makros als Variable definieren. Sie brauchen dann bei Änderungen dieser Namen nicht das ganze Makro abzusuchen, wo diese Namen überall vorkommen.
  - 5 Zum Drucken des Kalenders wird die angegebene Datei geöffnet. In dieser Datei sind bereits zwei Spalten definiert, der Spaltenmodus eingeschaltet, Tabulatoren definiert und eine kleine Schrift ausgewählt worden, damit das ganze Jahr auf eine A4-Seite paßt (siehe *Allgemeine Hinweise* am Beginn des Abschnittes »9.3 Kalender«).
- Sie müssen diese Datei hinsichtlich der verwendeten Schriftgrößen Ihrem Drucker anpassen, sonst erfolgen Verschiebungen.
- 6 In diese Variable werden die aus dem aufrufenden Makro übergebenen Daten gespeichert.
- Bei der Kompilierung dieses Makros erfolgt bei der anschließenden Ausführung eine falsche Berechnung des Kalenders, weil die Übergabevariablen nicht vorhanden sind. Dadurch können die definierten Arrays nicht ordnungsgemäß abgearbeitet werden. Brechen Sie in diesem Fall das Makro ab, sofern es nicht bereits mit einer Fehlermeldung abgebrochen wurde, und starten Sie es über das Makro KALENDER.WCM. Schließen Sie geöffnete Dateien, ohne diese zu speichern!

- 7 Keine Fehlermeldung bei nicht definierten Variablen (Übergabevariable aus dem aufrufenden Makro).
- 8 In der geladenen Datei muß in der Überschrift die Jahreszahl des zu druckenden Kalenders ergänzt werden. Der Text #Jahr# wird durch die übergebene Jahreszahl ersetzt.
- 9 Positionierung des Cursors am Dokumentende, hinter dem Steuerzeichen [Spal.Ein].
- 10 Einfügen einer Leerzeile.
- 11 Tabellenposition für die Ermittlung des ersten Wochentags des Jahres bestimmen.
- 12 Festlegung der Anzahl der Tage des Monats Februar.
- 13 Über *ForNext* wird der gesamte Kalenderdruck gesteuert. Für jeden einzelnen Monat werden die nachfolgenden Befehle bis zum zugehörigen *EndFor* ausgeführt.
- 14 Für die Berechnung eines Monats wird die Gesamtzahl der Tage des jeweiligen Monats festgelegt (beachten Sie bitte auch die Festlegung der Tage über einen Array in Makro KALELAUF.WC(M).
- 15 Generieren einer Meldung, die über den folgenden *Prompt*-Befehl auf den Fortgang der Kalendererstellung hinweist. Die Meldung wird bei jedem Monatswechsel am Bildschirm angezeigt.
- 16 Cursor erg in der Spaltenmitte zentriert.
- 17 Einschalten von Fettdruck, damit die Monatsnamen fett gedruckt werden.
- 18 Einfügen des Monatsnamens.
- 19 Fettdruck ausschalten.
- 20 Zeile beenden und eine Leerzeile einfügen.
- 21 Aufruf der Unterroutine *U01Überschrift* (Beschreibung siehe unten).
- 22 Aufruf der Unterroutine *U03DruckPos* (Beschreibung siehe unten).
- 23 Beginn einer *For*-Schleife, die den Druck eines Monats steuert.
- 24 Konvertieren der Variablen in einen String, damit die Tagesnummer gedruckt werden kann.
- 25 Wenn die Variable *TagPos* den Wert 7 enthält, handelt es sich um einen Sonntag. Der Tag wird fett gedruckt.
- 26 Erhöhen der Variablen *TagPos* um 1 zur Ermittlung des nächsten Wochentages.
- 27 Wenn die Variable *TagPos* einen Wert größer als 7 enthält, wurde eine komplette Woche gedruckt. Die Zeile wird beendet und die Variable *TagPos* auf 1 zurückgesetzt. Andernfalls wird der Cursor um eine Tabulatorposition nach rechts versetzt.
- 28 Ende der *For*-Schleife.
- 29 Durch *Seitenumbruch()* wird eine Spalte beendet. Der Cursor springt an den Beginn der nächsten Spalte (= neuer Monat).
- 30 Einfügen einer Leerzeile.
- 31 Ende der *ForNext*-Schleife.

- 32 Verzweigen zum Makroende.
- 33 Beginn der Unteroutine *U01Überschrift*. Erzeugen einer Überschrift für die sieben Wochentage. Nach Ausführung dieser Unteroutine wurde folgende Zeile gedruckt:  
Mo Di Mi Do Fr Sa So
- 34 Beginn einer *ForEach*-Schleife, die den Druck der Monatsüberschrift steuert.
- 35 Cursor um eine Tabulatorposition nach rechts verschieben.
- 36 Schreiben der Abkürzungen der Tagesnamen. Die Abkürzung für Sonntag (So) wird fett gedruckt.
- 37 Beenden der Zeile und Einfügen einer Leerzeile.
- 38 Rücksprung zum aufrufenden Befehl.
- 39 Beginn der Unteroutine *U02DruckPos*. Bei jedem Monatswechsel muß die Position des ersten Tages des Folgemonats innerhalb der Woche festgelegt werden.
- 40 *For*-Schleife zur Ermittlung der richtigen Tabulatorposition.
- 41 Cursor um eine Tabulatorposition nach rechts verschieben.
- 42 Ende der *For*-Schleife.
- 43 Rücksprung zum aufrufenden Befehl.
- 44 Beginn der Unteroutine *U99Ende*. Makro beenden und anzeigen einer Meldung.
- 45 Entwurfsmodus wieder ausschalten.
- 46 Cursor am Anfang des Dokuments positionieren.
- 47 Errechneten Kalender speichern.
- 48 Makroablauf wieder anzeigen.
- 49 Meldung für den folgenden *Prompt*-Befehl generieren.
- 50 Akustisches Signal ertönen lassen.
- 51 Anzeigen einer Meldung am Bildschirm.
- 52 Beenden des Makros. Hierdurch wird auch das aufrufende Makro beendet, d.h. es erfolgt kein Rücksprung in das aufrufende Makro KALENDER.WCM.

### 9.3.4 Jahreskalender (laufende Tagesnummer)

**Aufgabenstellung:** Über dieses Makro soll ein Jahreskalender mit der laufenden Tagesnummer gedruckt werden. Das Makro darf nur über KALENDER.WCM aufgerufen werden, sonst wird ein falscher Kalender gedruckt. Die Eingabe der Jahreszahl des zu druckenden Kalenders sowie die Ermittlung des ersten Wochentags, mit dem das zu druckende Jahr beginnt, erfolgt über das Makro KALENDER.WCM.

Eine genaue Bedienungsanleitung für die Kalender-Makros und Musterabbildungen der einzelnen Kalendertypen finden Sie in dem Kapitel »10 Makro-Sammlung«.

**Schwerpunkte:** Laden von Dateien, Übernahme von Daten von dem aufrufenden Makro in das aufgerufene Makro, Schleifenverarbeitung, Drucken.

### Makrodefinition: KALELAUF.WCM

```

1      Application      (A1; "WordPerfect"; Default; "DE")
2      Declare (Tagesname[7])
           Tagesname[1]:="So"
           Tagesname[2]:="Mo"
           Tagesname[3]:="Di"
           Tagesname[4]:="Mi"
           Tagesname[5]:="Do"
           Tagesname[6]:="Fr"
           Tagesname[7]:="Sa"
2      Declare (Monat[12])
           Monat[1]:="Januar"
           Monat[2]:="Februar"
           Monat[3]:="März"
           Monat[4]:="April"
           Monat[5]:="Mai"
           Monat[6]:="Juni"
           Monat[7]:="Juli"
           Monat[8]:="August"
           Monat[9]:="September"
           Monat[10]:="Oktober"
           Monat[11]:="November"
           Monat[12]:="Dezember"
2      Declare (LetzterTag[12])
           LetzterTag[1]:=31
           LetzterTag[2]:=28
           LetzterTag[3]:=31
           LetzterTag[4]:=30
           LetzterTag[5]:=31
           LetzterTag[6]:=30
           LetzterTag[7]:=31
           LetzterTag[8]:=31
           LetzterTag[9]:=30
           LetzterTag[10]:=31
           LetzterTag[11]:=30
           LetzterTag[12]:=31
3      OnError          (U99Ende)
4      DraftMode        (On!)
           //      Bitte Dateinamen ggf. ändern
5      KaleLauf:=        "C:\WPWIN7\MAKROS\KALELAUF"
6      FileOpen          ("C:\WPWIN7\MAKROS\KALELAUF.FRM")
7      //Folgende Variablen werden aus Makro KALENDER übergeben:
           //Jahr      Jahreszahl
           //TagPos    Mit welchem Wochentag beginnt das Jahr
           //SCHJahr    Schaltjahr, N = kein Schaltjahr, J = Schaltjahr
8      VarErrChk         (Off!)
9      IF                (SchJahr="J")

```

```

    LetzterTag[2]:= 29
Else
    LetzterTag[2]:= 28
EndIf
TagNumber:=      0
SearchString      ("#Jahr#")
ReplaceString      (Jahr)
ReplaceForward      (Normal!)
PosDocBottom      ()
HardReturn         ()
ForNext            (M;1;12;1)
Meldung:=          (Jahr+", Monat "+Monat[M]+" wird erstellt. Bitte
                    nicht auf [OK] klicken.")

Prompt            ("Kalender berechnen";Meldung;;;)
LineCenter         ()
FontBold           (On!)
Type              (Monat[M])
FontBold           (Off!)
HardReturn         ()
HardReturn         ()
For                (TagZähler;1;TagZähler<=LetzterTag[M];TagZähler+1)
    TagNumber:=    TagNumber+1
    DruNumber:=    NumStr(TagNumber)
    DruTag:=       NumStr(TagZähler)
    FontBold       (On!)
    Type           (Tagesname[TagPos])
    FontBold       (Off!)
    Tab            ()
    Type           (DruTag)
    LineFlushRight ()
    Type           (DruNumber)
    HardReturn     ()
    If             (Tagesname[TagPos]="So")
        HardReturn ()
    EndIf
    If             (TagPos=7)
        TagPos:=   1
    Else
        TagPos:=   (TagPos+1)
    EndIf
EndFor
HardPageBreak      ()
HardReturn         ()
EndFor
Label              (U99Ende)
PosDocTop          ()
FileSave           (FileName:KaleLauf;
                    Typeover:Yes!)

Display            (On!)
Meldung:=          ("Kalender für "+Jahr+" wurde erstellt.")
Beep Beep Beep

```

```
Prompt      ("Jahreskalender drucken";Meldung;;)
Wait        (30)
EndPrompt
Quit
```

## Erklärung

- 1 Produktkennung
- 2 Definition von Arrays zum Schreiben von Textkonstanten (Tages- und Monatsnamen) und zur Berechnung eines Monats (Anzahl der Tage eines Monats).
- 3 Beim Auftreten von Fehlern soll das Programm zu *U99Ende* verzweigen.
- 4 Während der Kalendererstellung wird der Entwurfsmodus eingeschaltet, weil die Erstellung, sofern sie am Bildschirm angezeigt wird, dann schneller erfolgt.
- 5 Definition eines Dateinamens (= Speicherung des errechneten Kalenders). Dateinamen sollten Sie immer am Beginn eines Makros definieren. Sie brauchen dann bei Änderungen dieser Namen nicht das ganze Makro abzusuchen, wo diese Namen überall vorkommen.
- 6 Zum Drucken des Kalenders wird die angegebene Datei geöffnet. In dieser Datei sind bereits zwölf Spalten definiert, der Spaltenmodus eingeschaltet, Tabulatoren definiert und eine kleine Schrift ausgewählt worden, damit das ganze Jahr auf eine A4-Seite im Querformat paßt.
- 7 In diese Variable werden die aus dem aufrufenden Makro übergebenen Daten gespeichert.
- 8 Keine Fehlermeldung bei nicht definierten Variablen (Übergabe von Variablen aus einem aufrufenden Makro).  
  
Wird das Makro kompiliert, erfolgt bei der Ausführung eine falsche Berechnung des Kalenders, weil die Übergabevariablen nicht vorhanden sind. Dadurch können die definierten Arrays nicht ordnungsgemäß abgearbeitet werden. Brechen Sie in diesem Fall das Makro ab, sofern es nicht bereits mit einer Fehlermeldung abgebrochen wurde, und starten Sie es über das Makro KALENDER.WCM.
- 9 Festlegen der Anzahl der Tage für den Monat Februar.
- 10 Definition einer Variablen für die laufende Tagesnummer, sie wird mit Null initialisiert.
- 11 In der geladenen Datei muß in der Überschrift die Jahreszahl des zu druckenden Kalenders ergänzt werden. Der Text #Jahr# wird durch die übergebene Jahreszahl ersetzt.
- 12 Cursor am Dokumentende hinter dem Steuerzeichen [Spal.Ein] positionieren.
- 13 Eine Leerzeile einfügen.
- 14 Über *ForNext* wird der gesamte Kalenderdruck gesteuert. Für jeden einzelnen Monat werden die nachfolgenden Befehle bis zum zugehörigen *EndFor* ausgeführt.
- 15 Generieren einer Meldung, die über den folgenden *Prompt*-Befehl auf den Fortgang der Kalendererstellung hinweist.



- 16 Die Meldung wird bei jedem Monatswechsel am Bildschirm angezeigt.
- 17 Cursor in der Mitte einer Spalte positionieren.
- 18 Monatsname in Fettdruck einfügen.
- 19 Zeile beenden und eine Leerzeile einfügen.
- 20 Beginn einer *For*-Schleife. Sie wird so lange wiederholt, bis ein kompletter Monat gedruckt wurde.
- 21 Laufende Tagesnummer um 1 erhöhen.
- 22 Tagesnummer und Tageszähler zum Drucken in einen String konvertieren.
- 23 Abkürzung des Tagesnamens fett drucken.
- 24 Cursor um eine Tabulatorposition nach rechts verschieben.
- 25 Schreiben der Tagesnummer innerhalb des Monats.
- 26 Cursor an den rechten Spaltenrand verschieben.
- 27 Schreiben der laufenden Tagesnummer innerhalb eines Jahres.
- 28 Zeile beenden (Zeilenvorschub).
- 29 Am Ende einer Woche (nach einem Sonntag) wird eine Leerzeile gedruckt. Danach wird der Positionszähler auf 1 zurückgesetzt, um wieder mit einem Montag zu beginnen.
- 30 Ende der *For*-Schleife.
- 31 Spaltenwechsel bei Monatsende.
- 32 Einfügen einer Leerzeile.
- 33 Ende der *ForNext*-Schleife.
- 34 Beginn des Labels *U99Ende*.
- 35 Cursor am Dokumentanfang positionieren.
- 36 Errechneten Kalender speichern.
- 37 Makroablauf wieder anzeigen.
- 38 Text generieren für den folgenden *Prompt*-Befehl.
- 39 Akustisches Signal ertönen lassen.
- 40 Anzeigen einer Meldung am Bildschirm.
- 41 Beenden des Makros. Hierdurch wird sicherheitshalber auch das aufrufende Makro beendet, d.h. es erfolgt kein Rücksprung in das aufrufende Makro KALENDER.WCM.

## 9.4 Adressenverwaltung

**Aufgabenstellung:** Über dieses Makro wird eine Adressendatei verwaltet. Je nach ausgewählter Verarbeitungsart können Sie Adressen ergänzen, ändern oder löschen. Die eingegebenen Daten werden im Format einer Datendatei gespeichert, die Sie zum Drucken von Adressenaufklebern, Kundenlisten oder Serienbriefen verwenden können. Die eingegebenen Daten werden auf Vollständigkeit überprüft. Wurde eine Eingabe vergessen, erfolgt eine entsprechende Fehlermeldung.

Eine genaue Bedienungsanleitung für dieses Makro finden Sie in dem Kapitel »10 Makro-Sammlung«.

**Schwerpunkte:** Erstellen von Dialogfeldern, Unterroutrinen, Einlesen von Daten in eine Variable, Bearbeiten einer Datendatei.

### Makrodefinition: ADRESSEN.WCM

```

1      Application      (A1;"WordPerfect";Default;"DE")
      //*****
      Name der Adressendatei ggf. ändern *****
2      AdressenDat:=    "C:\WPWIN7\MAKROS\ADRESSEN.DAT"
      //*****
      *****
3      OnCancel        (U99Ende)
4      Call            (U01Init)
5      Meldung:=       ""
6      FileOpen        (FileName:AdressenDat)
7      PosDokBottom    ()
8      Label           (H01Anfang)
9      Call            (U02MenüAufbau)
10     Switch          (VKennz)
      CaseOf           "Hinzufügen": Call (U03Ergänzen)
      CaseOf           "Löschen": Call (U04Löschen)
      CaseOf           "Ändern": Call (U05Ändern)
11     Default:        Meldung:= "Falsche Verarbeitungsart."
      EndSwitch
      Go               (H01Anfang)
12     Label           (U01Init)
      KdNr:=           ""
13     AnredeA:=        "Herrn"
      Vorname:=        ""
      Nachname:=        ""
      Titel:=          ""
      Firma:=          ""
      Abteilung:=       ""
      Strasse:=         ""
14     Land:=           "D"
      Plz:=             ""
      Wohnort:=         ""
15     AnredeB:=        "r Herr"
16     VKennz:=         "Ergänzen"
17     CurPos:=         1000

```

```

Return
18 Label (U02MenüAufbau)
19 DialogDefine (100; 50; 30; 250; 270; OK! | Percent!;
  "Adressenverwaltung")
20 DialogAddEditBox (100; 1000; 90; 10; 30; 12; 1; KdNr; 6)
21 DialogAddComboBox (100; 1010; 90; 25; 50; 50; 16; AnredeA; 30)
22 DialogAddListItem (100; 1010; "Herrn")
  DialogAddListItem (100; 1010; "Frau")
  DialogAddListItem (100; 1010; "Fräulein")
  DialogAddListItem (100; 1010; "Firma")
20 DialogAddEditBox (100; 1020; 90; 40; 150; 12; 1; Vorname; 30)
  DialogAddEditBox (100; 1030; 90; 55; 150; 12; 1; Nachname; 30)
  DialogAddEditBox (100; 1040; 90; 70; 150; 12; 1; Titel; 30)
  DialogAddEditBox (100; 1050; 90; 85; 150; 12; 1; Firma; 30)
  DialogAddEditBox (100; 1060; 90; 100; 150; 12; 1; Abteilung; 30)
  DialogAddEditBox (100; 1070; 90; 115; 150; 12; 1; Strasse; 30)
  DialogAddEditBox (100; 1080; 90; 130; 20; 12; 1; Land; 5)
  DialogAddEditBox (100; 1090; 90; 145; 20; 12; 1; Plz; 5)
  DialogAddEditBox (100; 1100; 90; 160; 150; 12; 1; Wohnort; 30)
21 DialogAddComboBox (100; 1110; 90; 175; 80; 50; 16; AnredeB; 30)
22 DialogAddListItem (100; 1110; "r Herr")
  DialogAddListItem (100; 1110; " Frau")
  DialogAddListItem (100; 1110; "s Fräulein")
  DialogAddListItem (100; 1110; " Damen und Herren")
21 DialogAddComboBox (100; 1120; 90; 195; 80; 40; 16; VKennz; 10)
22 DialogAddListItem (100; 1120; "Ergänzen")
  DialogAddListItem (100; 1120; "Löschen")
  DialogAddListItem (100; 1120; "Ändern")
23 DialogAddText (100; 1200; 10; 10; 75; 12; 1; "Kundennummer")
  DialogAddText (100; 1210; 10; 25; 75; 12; 1; "Anrede")
  DialogAddText (100; 1220; 10; 40; 75; 12; 1; "Vorname")
  DialogAddText (100; 1230; 10; 55; 75; 12; 1; "Nachname")
  DialogAddText (100; 1240; 10; 70; 75; 12; 1; "Titel")
  DialogAddText (100; 1250; 10; 85; 75; 12; 1; "Firma")
  DialogAddText (100; 1260; 10; 100; 75; 12; 1; "Abteilung")
  DialogAddText (100; 1270; 10; 115; 75; 12; 1; "Straße/Postfach")
  DialogAddText (100; 1280; 10; 130; 75; 12; 1; "Landes-
    Kennzeichen")
  DialogAddText (100; 1290; 10; 145; 75; 12; 1; "Postleitzahl")
  DialogAddText (100; 1300; 10; 160; 75; 12; 1; "Wohnort")
  DialogAddText (100; 1310; 10; 175; 75; 17; 1;
    "Anrede im Brief (Sehr geehrte...)")
  DialogAddText (100; 1320; 10; 195; 75; 12; 1; "Verarbeitungsart")
  DialogAddText (100; 1430; 10; 220; 150; 15; 1; Meldung)
24 DialogAddPushButton(100; 1325; 206;200; 32; 13; 0; "Ende")
  Ende:= MacroDialogResult
25 DialogDisplay (100; CurPos)
26 If (MacroDialogResult=Ende)
  Close ( )
  Quit
EndIf
27 DialogDestroy (100)

```

```

Return
28 Label          (U03Ergänzen)
29 If              ((KdNr="" ) or (KdNr=" "))
30   Meldung:=     "Kundennummer fehlt."
31   CurPos:=      1000
32   Go            (H01Anfang)
EndIf
If                ((AnredeA="" ) or (AnredeA=" "))
  Meldung:=       "Anrede-A fehlt."
  CurPos:=        1010
  Go              (H01Anfang)
EndIf
If                ((Vorname="" ) or (Vorname=" "))
  Meldung:=       "Vorname fehlt."
  CurPos:=        1020
  Go              (H01Anfang)
EndIf
If                ((Nachname="" ) or (Nachname=" "))
  Meldung:=       "Nachname fehlt."
  CurPos:=        1030
  Go              (H01Anfang)
EndIf
If                ((Strasse="" ) or (Strasse=" "))
  Meldung:=       "Straße fehlt."
  CurPos:=        1070
  Go              (H01Anfang)
EndIf
If                ((Plz="" ) or (Plz=" "))
  Meldung:=       "Postleitzahl fehlt."
  CurPos:=        1090
  Go              (H01Anfang)
EndIf
If                ((Wohnort="" ) or (Wohnort=" "))
  Meldung:=       "Wohnort fehlt."
  CurPos:=        1100
  Go              (H01Anfang)
EndIf
If                ((AnredeB="" ) or (AnredeB=" "))
  Meldung:=       "Anrede-B fehlt."
  CurPos:=        1110
  Go              (H01Anfang)
EndIf
If                ((AnredeA="Herrn") and (AnredeB<>"r Herr"))
  Meldung:=       "Anredefelder stimmen nicht überein"
  CurPos:=        1010
  Go              (H01Anfang)
EndIf
If                ((AnredeA="Frau") and (AnredeB<>" Frau"))
  Meldung:=       "Anredefelder stimmen nicht überein"
  CurPos:=        1010
  Go              (H01Anfang)
EndIf

```

```

If ((AnredeA="Fräulein") and (AnredeB<>"s Fräulein"))
  Meldung:= "Anredefelder stimmen nicht überein"
  CurPos:= 1010
  Go (H01Anfang)
EndIf
If ((AnredeA="Firma") and (AnredeB<>" Damen und Herren"))
  Meldung:= "Anredefelder stimmen nicht überein"
  CurPos:= 1010
  Go (H01Anfang)
EndIf
33 Call (U06Schreiben)
Return
34 Label (U04Löschen)
35 PosDokTop ( )
36 If ((KdNr="" ) odr (KdNr=" "))
  Meldung:= "Kundennummer fehlt, Satz kann nicht gelöscht werden."
  CurPos:= 1000
  Return
EndIf
37 OnNotFound (U88NichtGefunden)
38 SuchText:= ("#" + KdNr + "#")
39 SearchString (SuchText)
SearchNext (Normal)
40 PosLineBegin ( )
41 DeleteEOP ( )
42 DeleteCharNext ( )
43 FileSave (FileName:AdressenDat;
Typeover:Yes!)
44 Meldung:= ("Satz " + KdNr + " wurde gelöscht.")
45 Call (U01Init)
Return
46 Label (U05Ändern)
35 PosDocTop ( )
36 If ((KdNr="" ) or (KdNr=" "))
  Meldung:= "Kundennummer fehlt, Satz kann nicht geändert werden."
  CurPos:= 1000
  Return
EndIf
37 OnNotFound (U88NichtGefunden)
38 SuchText:= ("#" + KdNr + "#")
39 SearchString (GesuchterString:SuchText)
SearchNext (Normal)
47 PosLineBegin ( )
48 SelectLineEnd ( )
49 KdNr:= ?SelectedText
50 KdNr SubStr (Suchtext ;2;6)
51 PosZeichenRechts ( )
48 MarkierenZeileEnde( )
49 AnredeA:= ?SelectedText

```

```

51  PosCharNext      ()
    SelectLineEnd   ()
    Vorname:=       ?SelectedText
    PosCharNext      ()
    SelectLineEnd   ()
    Nachname:=      ?SelectedText
    PosCharNext      ()
    SelectLineEnd   ()
    Titel:=         ?SelectedText
    PosCharNext      ()
    SelectLineEnd   ()
    Firma:=         ?SelectedText
    PosCharNext      ()
    SelectLineEnd   ()
    Abteilung:=     ?SelectedText
    PosCharNext      ()
    SelectLineEnd   ()
    Strasse:=       ?SelectedText
    PosCharNext      ()
    SelectLineEnd   ()
    Land:=          ?SelectedText
    PosCharNext      ()
    SelectLineEnd   ()
    Plz:=           ?SelectedText
    PosCharNext      ()
    SelectLineEnd   ()
    Wohnort:=       ?SelectedText
    PosCharNext      ()
    SelectLineEnd   ()
    AnredeB:=       ?SelectedText
    PosDocTop        ()
52  SearchString     (SuchText)
    SearchNext       (Normal!)
40  PosLineBegin     ()
41  DeleteEOP        ()
42  DeleteCharNext   ()
43  FileSave         (FileName:AdressenDat;
                    Typeover:Yes!)
44  Meldung:=        ("Satz "+KdNr+" wurde gelöscht, bitte Daten
                    ändern.")
53  VKennz:=         "Ergänzen"
    Return
54  Label            (U06Schreiben)
55  OnNotFound        (U06Schreiben1)
56  SchKdNr:=        ("#" + KdNr + "#")
    PosDocTop        ()
57  SearchString     (SchKdNr)
    SearchPrevious    (Normal)
58  Prompt           ("Doppelter Ordnungsbegriff";
                    "Angabe Kundenummer bereits vorhanden.;;;")
59  Wait             (30)
    Endprompt

```

```

60  Meldung:=      ("Angegebene Kundennummer "+KdNr+" bereits
                    vorhanden.")
61  Return
    Label          (U06Schreiben1)
    PosDocBottom   ()
62  SchKdNr:=      ("#" + KdNr + "#")
63  Type           (SchKdNr) MergeEndFiled
    Type           (AnredeA) MergeEndFiled
    Type           (Vorname) MergeEndFiled
    Type           (Nachname) MergeEndFiled
    Type           (Titel) MergeEndFiled
    Type           (Firma) MergeEndFiled
    Type           (Abteilung) MergeEndFiled
    Type           (Strasse) MergeEndFiled
    Type           (Land) MergeEndFiled
    Type           (Plz) MergeEndFiled
    Type           (Wohnort) MergeEndFiled
    Type           (AnredeB) MergeEndRecord
43  FileSave       (FileName:AdressenDat;
                    Overtime:Yes!)
44  Meldung:=      ("Satz "+KdNr+" wurde ergänzt.")
64  Call           (U01Init)
    Return
65  Label          (U88NichtGefunden)
    Meldung:=      ("Satz "+KdNr+" wurde nicht gefunden.")
    VKennz:=       "Ergänzen"
    aGo            (H01Anfang)
66  Label          (U99Ende)
    Quit

```

## Erklärung

- 1 Produktkennung
- 2 Öffnen der Datendatei. Geben Sie hier einen anderen Namen ein, wenn Sie diese Datei in einem anderen Ordner speichern möchten.
- 3 Bei einem Abbruch soll das Makro zu dem Label *U99Ende* verzweigen.
- 4 Aufruf der Unteroutine *U01Init*. In dieser Unteroutine werden die Textfelder des Dialogfelds mit bestimmten Werten vorbesetzt (Beschreibung siehe unten).
- 5 Der Inhalt der Variablen *Meldung* wird gelöscht, damit beim ersten Anzeigen des Dialogfelds keine Fehlermeldung angezeigt wird.
- 6 Öffnen der Adressendatei.
- 7 Cursor am Ende der Adressendatei zum Einfügen neuer Datensätze positionieren.
- 8 Beginn des Labels *H01Anfang*. Hier handelt es sich um die Hauptsteuerung des Makros.
- 9 Aufruf der Unteroutine *U02MenüAufbau*, die das Dialogfeld definiert und am Bildschirm anzeigt.

- 10 Prüfung des Veränderungskennzeichens. Je nach Auswahl wird über *Call* die jeweils zutreffende Unteroutine aufgerufen. Danach verzweigt das Makro wieder zu *H01Anfang*.
- 11 Trafen die vorausgegangenen *If*-Bedingungen nicht zu, wurde eine falsche Veränderungsart eingegeben. Es erfolgt das Generieren einer Fehlermeldung und der Rücksprung zu *H01Anfang*, um das Dialogfeld (mit Fehlermeldung) erneut anzuzeigen.
- 12 Beginn der Unteroutine *U01Init*. Hier werden die Textfelder des Dialogfelds mit bestimmten Werten vorbesetzt, wenn das Dialogfeld zur Erfassung eines neuen Datensatzes angezeigt wird.
- 13 Die Variable *AnredeA* wird mit »Herrn« vorbesetzt.
- 14 Die Variable *Land* wird mit »D« vorbesetzt.
- 15 Die Variable *AnredeB* wird mit »r Herr« vorbesetzt.
- 16 Die Variable *VKennz* wird mit »Ergänzen« vorbesetzt.
- 17 Die Variable *CurPos* wird mit »1000« vorbesetzt. Über diese Variable wird im Dialogfeld der Cursor positioniert. *CurPos* steht hier für die Variable *Controlld*, die in dem Befehl *DialogDisplay* verwendet wird. Mit dem hier angegebenen Wert wird der Cursor in dem Textfeld der Kundennummer positioniert.
- 18 Beginn der Unteroutine *U02MenüAufbau*. Diese Routine definiert das Dialogfeld, zeigt es am Bildschirm an, und übernimmt die von Ihnen eingegebenen Daten in die jeweils zugehörige Variable.
- 19 Definieren der Höhe und Breite des Dialogfeldes.
- 20 Erstellen eines Textfeldes zur Eingabe von Daten (hier: Kundennummer).
- 21 Erstellen eines Listenfeldes, mit den unter Zeile 22 und den folgenden definierten Einträgen.
- 22 Definieren von Einträgen, die in dem unter Zeile 21 definierten Listinfeld erscheinen sollen.
- 23 Definieren von Textkonstanten (hier: Feldbezeichnungen), die den Textfeldern als Beschreibung vorangestellt werden.
- 24 Definition einer Taste »Ende«. Nach dem Klicken auf dieser Taste wird das Makro beendet.
- 25 Dialogfeld anzeigen. Der Cursor wird in dem Feld (Objekt) plaziert, das unter *CurPos* angegeben wurde.
- 26 Abfragen, ob auf der Taste »Ende« geklickt wurde. Wenn ja, wird die Adressendatei geschlossen und das Makro beendet. Diese Abfrage muß unbedingt *vor* dem Befehl *DialogDestroy* erfolgen.
- 27 Dialogfeld aus dem Speicher löschen.
- 28 Beginn der Unteroutine *U03Ergänzen*. Diese Routine wird aufgerufen, wenn ein neuer Datensatz aufgenommen werden soll.
- 29 Überprüfung der Kundennummer. Ist keine Kundennummer vorhanden, kann der Datensatz nicht aufgenommen werden.



- 30 Erstellen einer Fehlermeldung, die im unteren Teil des Dialogfeldes angezeigt wird.
- 31 Cursor in dem Feld positionieren, das den Fehler verursachte.
- 32 Rücksprung zu dem Label *H01Anfang*. Dort wird das Dialogfeld mit der Fehlermeldung angezeigt. Die nachfolgenden Befehle sind jeweils mit den Zeilennummer 29 bis 32 sinngemäß identisch. Auf eine Beschreibung wird darum verzichtet.
- 33 Aufrufen der Unteroutine *U06Schreiben*. Diese Routine wird dann aufgerufen, wenn alle Muß-Felder mindestens ein Zeichen enthalten (Beschreibung siehe unten).
- 34 Beginn der Unteroutine *U04Löschen*. Diese Routine wird dann aufgerufen, wenn ein Datensatz aus der Adressendatei gelöscht werden soll.
- 35 Sprung zum Anfang der Adressendatei.
- 36 Prüfen, ob eine Kundennummer eingegeben wurde. Wenn nicht, wird eine Fehlermeldung erzeugt, das falsche Textfeld in der Variablen *CurPos* festgelegt und zum aufrufenden Befehl zurückgesprungen.
- 37 Wenn die Kundennummer nicht gefunden wird, soll zu *U88NichtGefunden* verzweigt werden (Beschreibung siehe unten).
- 38 Die eingegebene Kundennummer wird mit dem Zeichen # begrenzt, um eine eindeutige Kundennummer zu erzwingen (z.B. #666666#). Würde die Kundennummer nicht mit diesem Zeichen begrenzt, kann der Suchbegriff auch in einem anderen Feld eines Datensatzes gefunden werden.

**Beispiel:** Sie möchten die Kundennummer 600 verändern. Da die Datensätze der Datendatei unsortiert gespeichert werden, trifft die Suche auch bei der Kundennummer 60000 oder bei der Postleitzahl 6000 zu, was zu Fehlern führen würde.

- 39 Starten des Suchvorgangs.
- 40 Cursor am Zeilenanfang des Feldes Kundennummer im gefundenen Datensatz positionieren.
- 41 Löschen des Datensatzes. Der feste Seitenumbruch bleibt jedoch erhalten.
- 42 Löschen des übriggebliebenen festen Seitenumbruchs, damit kein Leersatz bestehen bleibt.
- 43 Speichern der Adressendatei, der bisherige Inhalt der Datei wird überschrieben.
- 44 Meldung erstellen, die das Löschen des betreffenden Satzes bestätigt.
- 45 Inhalt der Textfelder des Dialogfeldes vor der Anzeige löschen bzw. vorbesetzen.
- 46 Beginn der Unteroutine *U05Ändern*. In dieser Unteroutine wird ein zu ändernder Datensatz gesucht, und die bisherigen Daten im Dialogfeld werden zum Ändern angezeigt.
- 47 Cursor am Beginn der ersten Zeile des gefundenen Datensatzes positionieren.
- 48 Komplette Zeile als Block markieren.
- 49 Der markierte Textblock wird in einer Variablen gespeichert (hier: *KdNr*).
- 50 Die Zeichen # werden von der Kundennummer getrennt.

- 51 Der Cursor wird an den Beginn der nächsten Zeilen bewegt. Die Blockfunktion wird dadurch ausgeschaltet. Die Befehle der Zeilen 48, 49 und 51 wiederholen sich jetzt für jedes Datenfeld des ausgewählten Datensatzes. Auf eine Beschreibung der nachfolgenden Befehle wird darum verzichtet.
- 52 Nachdem alle Felder in den betreffenden Variablen des Dialogfeldes gespeichert wurden, wird an den Beginn des Datensatzes gesprungen, indem wieder die zu löschende Kundennummer gesucht wird. Danach wird der Datensatz gelöscht.
- 53 Das Veränderungskennzeichen wird auf »Ergänzen« gesetzt, damit nach der Eingabe der Änderungen der Datensatz wieder neu aufgenommen werden kann.
- 54 Beginn der Unterroutine *U06Schreiben*. Diese Unterroutine fügt der Adressendatei einen neuen Datensatz aufgrund der eingegebenen Daten zu. Der Cursor wird am Dateiende positioniert.
- 55 Setzen einer »Nicht gefunden«-Bedingung.
- 56 Die Kundennummer wird jeweils am Anfang und am Ende durch das Zeichen # begrenzt (siehe oben). Der Cursor wird am Dokumentanfang positioniert.
- 57 Die Datendatei wird nach der gerade eingegebenen Kundennummer durchsucht. Ist diese bereits vorhanden, erfolgt eine Fehlermeldung, ansonsten wird der Datensatz ergänzt.
- 58 Anzeigen einer Fehlermeldung, wenn ein Satz mit derselben Kundennummer vorhanden ist.
- 59 Meldung für eine bestimmte Zeit am Bildschirm anzeigen.
- 60 Meldung für Dialogfeld generieren.
- 61 Rücksprung zum aufrufenden Befehl. Der Datensatz wird nicht ergänzt.
- 62 Die Kundennummer wird jeweils am Anfang und am Ende durch das Zeichen # begrenzt (siehe oben).
- 63 Schreiben der Daten eines Textfeldes (hier: Kundennummer). Jedes Feld muß mit einem Mischbefehl *MischenEndeFeld* enden.  
Die Befehle der Zeilen 56 und 57 wiederholen sich anschließend für jedes einzelne Datenfeld. Auf eine Beschreibung wird darum verzichtet.
- 64 Aufruf der Unterroutine *U01Init*. Nach dem Speichern des Datensatzes werden alle Variablen des Dialogfeldes wieder gelöscht bzw. mit bestimmten Angaben vorbelegt.
- 65 Beginn der Unterroutine *U88NichtGefunden*. Zu dieser Routine wird dann verzweigt, wenn eine zu suchende Kundennummer nicht gefunden wurde.
- 66 Zu dieser Unterroutine wird verzweigt, wenn das Makro abgebrochen wurde.















