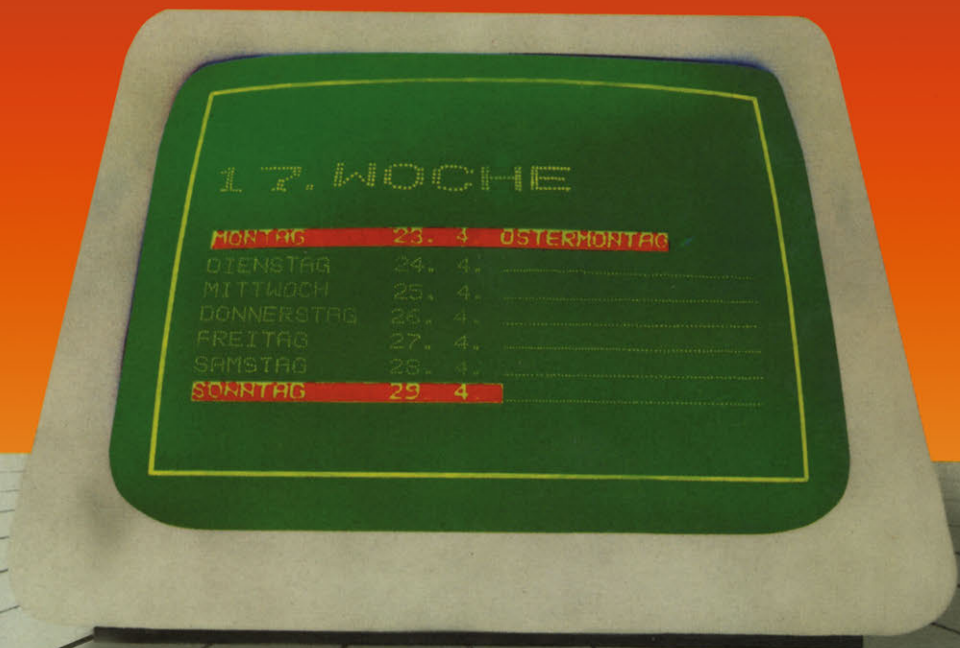


Kastner

Terminplanung mit BASIC auf Commodore

2000/3000
4000/8000



SPRINGER FACHMEDIEN
WIESBADEN GMBH

Business-Software

Kastner

Terminplanung mit BASIC auf Commodore 2000/3000/4000/8000

Prof. Gustav Kastner

Terminplanung
mit BASIC
auf Commodore ^{2000/3000}_{4000/8000}

– Kalenderalgorithmen –

SPRINGER FACHMEDIEN WIESBADEN GMBH

Kastner, Gustav:

Terminplanung mit BASIC auf Commodore 2000/3000,

4000/8000 : Kalenderalgorithmen / Gustav

Kastner.

ISBN 978-3-409-19204-0 ISBN 978-3-663-12990-5 (eBook)

DOI 10.1007/978-3-663-12990-5

Das in diesem Buch vorgestellte Programm wurde intensiv in der praktischen Anwendung getestet. Das Buch wurde mit der größten Sorgfalt hergestellt. Wegen der technologisch empfindlichen Struktur und komplizierten logischen Abhängigkeiten können Verlag und Verfasser jedoch keinerlei Haftung oder juristische Verantwortung für Folgen übernehmen, die sich aus fehlerhaften Angaben ergeben.

© Springer Fachmedien Wiesbaden 1984

Ursprünglich erschienen bei Betriebswirtschaftlicher Verlag Dr. Th. Gabler GmbH, Wiesbaden 1984

Satz: Elsner & Behrens, Oftersheim

Umschlag: Studio für Werbung und Kommunikation, Wiesbaden

Alle Rechte vorbehalten. Auch die fotomechanischen Vervielfältigung des Werkes (Fotokopie, Mikrokopie) oder von Teilen daraus bedarf vorheriger Zustimmung des Verlages.

ISBN 978-3-409-19204-0

Vorwort

Bei der Termin-Planung muß immer der Kalender bereitliegen. Er bestimmt und regelt die Zeitrechnung. Die größte Zeiteinheit ist das Jahr, das in Monate, Wochen und Tage eingeteilt ist. Die Zeiteinteilung des Kalenders richtet sich nach astronomischen Gegebenheiten: die Drehung der Erde um ihre eigene Achse ergibt das Wechselspiel von Tag und Nacht, der Lauf der Erde um die Sonne bestimmt die Jahreszeiten und damit die Monate. Es gibt etliche Kalender. Seit dem 15. 10. 1582 richten wir uns nach dem Gregorianischen Kalender (siehe 1. Kapitel).

Ein Termin wird durch ein Kalender-Datum festgelegt, das durch eine Tageszahl, eine Monatszahl und eine Jahreszahl angegeben wird. Zu jedem Datum gehört ein bestimmter Wochentag. Dieser ist für die Termin-Planung sehr wichtig. Eine Arbeit z. B., die nicht unterbrochen werden darf und für die man drei Werktage braucht, kann nicht an einem Freitag begonnen werden. – Damit hat man das klassische Kalenderproblem (Wochentagsbestimmung zu einem vorgegebenem Datum, siehe 6. Kapitel), das bei fast jeder Termin-Planung benötigt wird. Für die Bestimmung der Werktage müssen in jedem Jahr die Festtage jedesmal neu errechnet werden (siehe 9. Kapitel), da entweder das Datum (z. B. Ostern) oder der Wochentag (z. B. Weihnachten, Neujahr) sich ändert. Wegen der nicht einheitlichen Feiertagsregelung muß der Anwender selbst bestimmen können, welche Festtage Feiertag sein sollen.

Kennt man das Anfangsdatum und die Dauer der beabsichtigten Tätigkeit oder des Vorganges, so ist das zugehörige Enddatum (nach Kalendertagen oder Werktagen mit und ohne Samstag) zu berechnen (siehe Kapitel 16 und 17). Das umgekehrte Problem stellt sich, wenn Anfangs- und Enddatum vorgegeben sind, und die Anzahl der verfügbaren Kalender- oder Werktage (mit oder ohne Samstag) festgestellt werden muß (siehe Kapitel 12 bis 14).

Ein ähnlich gelagertes Problem ergibt sich z. B. bei Krankenhaus- oder Hotel-Aufenthalt: Die Anzahl der Tage bzw. Nächte zwischen den Kalenderdaten ist für die Berechnung nötig.

Viele Planungen oder betriebswirtschaftlichen Vorgänge (z. B. Produktion oder Lieferung) orientieren sich an der Wochen-Nummer eines Jahres. Es ist also einerseits nötig, ein Kalenderdatum umzurechnen in die Wochen-Nummer und umgekehrt, eine nach Wochen vorgenommene Termin-Planung wieder in ein Kalenderdatum zu überführen (siehe Kapitel 7 und 8).

Man könnte noch eine beliebig lange Liste von Problemen anführen, bei denen Termin und Kalender maßgeblich beteiligt sind.

Das vorliegende Buch bietet Lösungen in Form von fertigen BASIC-Programmen an. Es ist so konzipiert, daß die Kapitel systematisch aufeinander aufbauen. Die Lösungen zu den Teilproblemen sind in Unterprogrammen codiert, damit sie leicht bausteinartig (als Module) in andere Programme übernommen werden können. Das Programm-Verzeichnis zeigt, wo in der vorliegenden Programm-Sammlung die Unterprogramme in den nachfolgenden Programmen eingebaut worden sind. Die Unterprogramme sind mit vierstelligen Anweisungs-Nummern von 5000 bis 7430 codiert. Die meisten Microcomputer (abgesehen von kapazitätsmäßig kleinen) gestatten fünfstellige Anweisungs-Nummern. Man kann also bei Erstellung eigener Programme durch Voranstellen einer fünften Ziffer vor die Anweisungs-Nummer die hier angegebenen Unterprogramme an die individuellen Programme anpassen. Die Unterprogramme sind relativ kurz und leicht überschaubar, so daß bei einer eventuellen Änderung der Anweisungs-Nummern auch die Änderung der Sprunganweisungen und Verzweigungen innerhalb des Unterprogrammes leicht und selbst von nicht erfahrenen Programm-Erstellern ohne Komplikationen durchgeführt werden können.

Abgesehen von der Gauß'schen Berechnungsformel für die Oster-Termine (im 9. Kapitel) sind alle anderen Kalender-Algorithmen vom Verfasser erstellt. Sie werden im jeweiligen Kapitel hergeleitet und erläutert. Ziel des Buches ist es, nicht nur fertige, codierte Programme bereitzustellen, sondern auch auf die Probleme der Programm-Erstellung einzugehen. Insbesondere werden die Phasen „Problem-Analyse“, „Algorithmus“ und „Programm-Logik“ (siehe Anhang 1) in allen Kapiteln zumindest exemplarisch angedeutet. Wiederholungen werden selbstredend und viele Details notgedrungen unterdrückt. Die Programm-Logik wird durchgehend in Logik-Diagrammen dargestellt (siehe Anhang 2). Sie sind modifizierte Struktogramme.

Die Programme sind auf den Microcomputern von Commodore der Serien 2000, 3000 und 4000 erstellt und ohne jede Änderung ablauffähig. Nur für das 11. Programm (Monatskalender mit Wochen-Nummer, Hervorhebung der Sonn- und Feiertage, Anzahl der Sonn- und Feiertage, Anzahl der Werkstage mit und ohne Samstage) und für das 12. Programm (quartalsweise strukturierter Jahres-Übersichts-Kalender) ist ein Drucker erforderlich. Sonst wird keine Peripherie benötigt. Da nur die in Anhang 3 angegebenen Sonderzeichen zur Bildschirmsteuerung verwendet wurden, ist eine Übertragung der Programme auf andere Microcomputer mit BASIC-Interpreter ohne besondere Schwierigkeiten möglich. Durch die in allen Kapiteln angegebene Programm-Logik ist selbst eine Übertragung in andere Programmiersprachen realisierbar.

Mainz, im September 1984

Gustav Kastner

Inhaltsverzeichnis

Vorwort	5
1. Der Gregorianische Kalender	9
2. Gültigkeitskontrolle eines Datums	13
3. Tagesnummer eines Datums	25
4. Datum zur Tagesnummer	31
5. Wochentagsberechnung für den Neujahrstag	37
6. Wochentag eines beliebigen Datums	43
7. Wochen-Nummer eines Datums	47
8. Datum zu einer Wochen-Nummer	53
9. Datumsfixe und bewegliche Feste	61
10. Monatskalender mit Wochen-Nummer und Feiertagen	73
11. Jahres-Übersichtskalender	85
12. Anzahl der Kalendertage zwischen Anfangs- und Enddatum	93
13. Anzahl der Werktage zwischen zwei Kalenderdaten	99
14. Anzahl der Werktage ohne Samstage	111
15. Datum-Fortschreibung	115
16. Enddatum nach Kalendertagen	121
17. Enddatum nach Werktagen	129
Programm- und Literaturverzeichnis	141
Anhang 1: Die Phasen der Programm-Erstellung	145
Anhang 2: Logik-Diagramme	149
Anhang 3: BASIC-Sonderzeichen	155
Anhang 4: Liste der verwendeten Variablen-Namen	159

1 Der Gregorianische Kalender

Unser heutiger Kalender, der sogenannte Gregorianische Kalender, gilt seit dem 15. 10. 1582. Bis dahin galt der Julianische Kalender, der aber zu diesem Zeitpunkt um zehn Tage „falsch“ ging. Wie kam das?

Das Kalenderjahr wird bestimmt vom Umlauf der Erde um die Sonne. Ein voller Umlauf dauert 365,2422 Tage. Das bedeutet, daß bei einem Kalenderjahr von 365 Tagen nach vier Jahren ein Unterschied von fast einem (genau $4 \cdot 0,2422 = 0,9688$) Tag zwischen dem Kalenderjahr und der astronomischen Konstellation auftritt. Der Julianische Kalender (nach Julius Cäsar benannt), glich diese Abweichung durch einen Schalttag alle vier Jahre aus. Das war etwas (nämlich $1 - 0,9688 = 0,0312$ Tage) zuviel und so ging bis zum 16. Jahrhundert der Julianische Kalender um zehn Tage nach. Anlaß für den Gregorianischen Kalender (nach Gregor XIII benannt), bessere *Schaltjahresregeln* festzulegen:

1. Die durch 4 teilbaren Jahre sind Schaltjahre.
2. Die durch 100 teilbaren Jahre sind *keine* Schaltjahre.
3. Die durch 400 teilbaren Jahre *sind* Schaltjahre.

Die Regel 1 oder 2 gilt nur, wenn sie nicht durch die nachfolgende außer Kraft gesetzt wird. In einem Schaltjahr wird als Schalttag der 29. Februar eingefügt, sonst hat der Februar nur 28 Tage.

Nach der ersten Schaltjahresregel entstehen durch den eingefügten Schalttag ein Fehler von 0,0312 Tagen (wie oben schon berechnet) alle vier Jahre. In einem Jahrhundert sind 25 Schaltperioden, so daß dieser Fehler auf $25 \cdot 0,0312 = 0,78$ Tage angewachsen ist. Durch die zweite Schaltjahresregel fällt nach hundert Jahren der Schalttag weg, so daß der Fehler nun $0,78 - 1 = -0,22$ Tage beträgt. Nach vier Jahrhunderten beträgt dieser Fehler dann $4 \cdot (-0,22) = -0,88$ Tage, der durch die dritte Schaltjahresregel mit dem Schalttag auf $1 - 0,88 = 0,12$ Tage korrigiert wird.

Für diesen letzten Fehler von 0,12 Tagen, der in einem Zeitraum von vierhundert Jahren anfällt, sieht der Gregorianische Kalender keine Regelung mehr vor. Erst nach 3200 Jahren (8 Zeiträume zu je 400 Jahren) braucht dieser Fehler berücksichtigt zu werden, da er dann auf $8 \cdot 0,12 = 0,96$ Tage angewachsen sein wird. Wie man sieht, genügen also die drei Schaltjahresregeln „vorläufig“.

Wichtig ist für uns die dritte Schaltjahresregel, denn wegen ihr wird das Jahr 2000 wie jedes durch vier teilbare Jahr (entgegen der 2. Regel) ein normales Schaltjahr sein. Das bedeutet, daß in dem Zeitraum von 1901 bis zum Jahre 2099 nur eine einzige (nämlich die erste) Schaltjahresregel zu beachten ist.

2 Gültigkeits- kontrolle eines Datums

„Ein gutes Programm soll sich gegen falsche Daten-Eingabe wehren können“ (siehe [4], Seite 11 und 125). Ein Programm muß korrekte Ergebnisse liefern. Gibt man schon unkorrekte Daten ein, kann man nicht erwarten, daß man richtige Ergebnisse erhält: „Garbage in – garbage out“. Daher sollte jede Daten-Eingabe überprüft werden.

Das Kalenderjahr ist eingeteilt in zwölf Monate mit unterschiedlich vielen Tagen. Die Monate April (4. Monat des Jahres), Juni (6.), September (9.) und November (11.) haben 30 Tage, der Februar hat 28 bzw. in einem Schaltjahr 29 Tage und die übrigen sieben Monate haben 31 Tage. Nicht jedes Datum ist also ein gültiges Datum. So ist beispielsweise der 31. 4. eines Jahres ein ungültiges, der 31. 5. aber ein gültiges Datum und der 29. 2. 1984 ist gültig, nicht aber der 29. 2. 1985. Sollen Fehler vermieden werden, muß also vor der Weiterverarbeitung des Datums eine Gültigkeitskontrolle durchgeführt werden, das Datum muß auf „Plausibilität“ geprüft werden.

Bei der Gültigkeitskontrolle eines Kalender-Datums ist zunächst eine *Bereichskontrolle* durchzuführen, danach eine *Verträglichkeitskontrolle*.

In der *Bereichskontrolle* muß geprüft werden, ob Tag, Monat und Jahr des zu prüfenden Datums Werte im zulässigen Bereich haben: Der Tag darf nur ganzzahlige Werte zwischen 1 und 31, der Monat nur ganzzahlige Werte zwischen 1 und 12 und das Jahr nur ganzzahlige Werte ab 1582 annehmen. Da der Gregorianische Kalender erst ab 15. 10. 1582 gilt, ist für dieses Jahr auch Monat und gegebenenfalls auch der Tag zu überprüfen.

In der *Verträglichkeitskontrolle* muß geprüft werden, ob Tag und Monat eines Datums miteinander verträglich sind, da die Monate unterschiedlich viele Tage haben. So können nur in den Monaten Januar (1), März (3), Mai (5), Juli (7), August (8), Oktober (10), Dezember (12) 31 Tage, in den Monaten April (4), Juni (6), September (9), November (11) nur 30 Tage und im Februar (2) 28 Tage bzw. in einem Schaltjahr 29 Tage vorkommen.

In der nachfolgenden Programmlogik wird die Bereichskontrolle sofort nach Eingabe des entsprechenden Wertes für Tag bzw. Monat bzw. Jahr durchgeführt. Bei nicht zulässigem Wert wird der eingegebene Wert abgelehnt und sogleich eine neue Eingabe verlangt. Werden die eingegebenen Werte akzeptiert, erfolgt anschließend die Verträglichkeitskontrolle.

Die Eingabe mit Bereichskontrolle ist in den Anweisungen 100 bis 260 codiert, die Verträglichkeitskontrolle (Abb. 2) in den Anweisungen 270 bis 370 und die Prüfung auf Schaltjahr (Abb. 3) in 320 bis 340.

Die Abbildung 4 zeigt ein Beispiel einer Bildschirm-Ausgabe. Man kann sie – bei angeschlossenem Drucker – im wesentlichen erhalten durch das Einfügen der vier Anweisungen

```
5 OPEN 4,4:CMD4
135 PRINT TA
165 PRINT MO
195 PRINT JA
```

Wiederhole, bis Datumseingabe akzeptiert wird			
Wiederhole, bis Tag-Eingabe akzeptiert wird			
Eingeben: Tag TA			
Ist $1 \leq TA \leq 31$ und ganzzahlig?			
nein	ja		
Fehler!	(TA akzeptiert)		
Wiederhole, bis Monat-Eingabe akzeptiert wird			
Eingeben: Monat MO			
Ist $1 \leq MO \leq 12$ und ganzzahlig?			
nein	ja		
Fehler!	(MO akzeptiert)		
Wiederhole, bis Jahr-Eingabe akzeptiert wird			
Eingeben: Jahr JA			
Ist $JA > 1582$ und ganzzahlig?			
nein	ja		
Ist $JA = 1582$?			
nein	ja		
Fehler!	(weitere Kontrolle nötig:)		
Ist $MO > 10$?			
nein	ja		
Ist $MO < 10$?			
ja	nein	Ist $TA \geq 15$?	
		nein	ja
Fehler: Erst ab 15.10 gültig!		Verträglichkeitskontrolle (siehe Abb. 2)	

Abb. 1: Eingabe mit Bereichs-Kontrolle

Ist Monat MO = 1 oder 3 oder 5 oder 7 oder 8 oder 10 oder 12?					ja	
nein						
Ist Tag TA = 31?					ja	
nein						
Ist Monat MO = 4 oder 6 oder 9 oder 11 (bzw. MO ≠ 2) ?				ja		Fehler: Monat hat nicht 31 Tage
nein						
Ist Tag TA = 30?				ja		
nein						
Ist Tag TA < 29?				Fehler: Fe- bruar hat nicht 30 Tage		
nein		ja		(Da- tum ver- träg- lich)		(Da- tum ver- träg- lich)
Ist JA Schaltjahr? (siehe Abb. 3)		(Da- tum ver- träg- lich)		(Da- tum ver- träg- lich)		
nein		ja				
Fehler: Kein Schaltjahr		(Datum verträglich)				

Abb. 2: Verträglichkeitskontrolle

Ist JA teilbar durch 400?					ja	
nein						
Ist JA teilbar durch 100?					Schaltjahr!	
nein						
Ist JA teilbar durch 4?				Kein Schaltjahr!		
nein				ja		
Kein Schaltjahr		Schaltjahr!				

Abb. 3: Ist JA Schaltjahr?

1. Programm: Gültigkeitskontrolle eines Datums im Gregorianischen Kalender

```
1 REM          1. KALENDER-PROGRAMM
10 PRINT"Q] GUELTIGKEITSKONTROLLE EINES DATUMS
11 PRINT " IN GREGORIANISCHEN KALENDER.
12 PRINT " -----
100 PRINT
110 PRINT"DATUM AB 15.10.1582 EINGEBEN:
120 PRINT
130 INPUT"TAG (1...31)";TA
140 IF 1<=TA AND TA<=31 AND TA=INT(TA) THEN 160
150 PRINT"Q] FEHLER!" :GOTO 130
160 INPUT"MONAT (1...12)";MO
170 IF 1<=MO AND MO<=12 AND MO=INT(MO) THEN 190
180 PRINT"Q] FEHLER!" :GOTO 160
190 INPUT"JAHR (>=1582)";JA
200 IF JA>1582 AND JA=INT(JA) THEN 270
210 IF JA=1582 THEN 230
220 PRINT"Q] FEHLER!" :GOTO 190
230 IF MO<10 THEN 270
240 IF MO<2 THEN 260
250 IF TA<=15 THEN 270
260 PRINT"Q] FEHLER: ERST AB 15.10. GUELTIG!" :GOTO 190
270 IF MO=10ORMO=3ORMO=5ORMO=7ORMO=8ORMO=10ORMO=12 THEN 400
280 IF TA=31 THEN 360
290 IF MO<2 THEN 400
300 IF TA=30 THEN 370
310 IF TA<29 THEN 400
320 IF JA/400=INT(JA/400) THEN 400
330 IF JA/100=INT(JA/100) THEN 350
340 IF JA/4=INT(JA/4) THEN 400
350 PRINT"Q] FEHLER: KEIN SCHALTJAHR!" :GOTO 190
360 PRINT"Q] FEHLER: MONAT HAT KEINE 31 TAGE!" :GOTO 190
370 PRINT"Q] FEHLER: FEBRUAR HAT KEINE 30 TAGE!" :GOTO 190
400 PRINT"DATUM";STR$(TA);".";STR$(MO);".";JA;"GUELTIG!"
410 PRINT"-----" :GOTO 190
420 END
```

Im voranstehenden Programm erfolgt die Eingabe des Datums getrennt nach Tag, Monat und Jahr. Es ist aber üblich, das Datum auf einmal anzugeben, indem Tag, Monat und Jahr durch Punkt getrennt werden, also beispielsweise 4. 3. 1984 oder 04. 03. 1984 oder 4. 3. 84. In dieser Form ist die Datumseingabe nicht mehr numerisch. Das Programm muß sie daher in numerische Werte für Tag, Monat und Jahr umwandeln, um sie für anschließende Rechenalgorithmen verwenden zu können. Wie am obigen Beispiel erkennbar, soll es gestattet sein das Jahr entweder zweistellig (wie meist üblich) oder vierstellig (was richtiger ist) anzugeben. Bei einer zweistelligen Jahresangabe ist natürlich unser jetziges Jahrhundert gemeint, d. h. das Programm hat von sich aus 1900 zu ergänzen. In diesem Programm soll der Gültigkeits-

GUELTIGKEITSKONTROLLE EINES DATUMS
FUER DEN GREGORIANISCHEN KALENDER.

DATUM AB 15.10.1582 EINGEBEN:

TAG <1...31> 28
MONAT <1...12> 2
JAHR >=1582 1984
DATUM 28. 2. 1984 GUELTIG!

DATUM AB 15.10.1582 EINGEBEN:

TAG <1...31> 29
MONAT <1...12> 2
JAHR >=1582 1984
DATUM 29. 2. 1984 GUELTIG!

DATUM AB 15.10.1582 EINGEBEN:

TAG <1...31> 29
MONAT <1...12> 2
JAHR >=1582 1985
FEHLERMELDUNG: UNMUEGLICH

DATUM AB 15.10.1582 EINGEBEN:

TAG <1...31> 31
MONAT <1...12> 5
JAHR >=1582 1984
DATUM 31. 5. 1984 GUELTIG!

DATUM AB 15.10.1582 EINGEBEN:

TAG <1...31> 31
MONAT <1...12> 6
JAHR >=1582 1985
FEHLERMELDUNG: UNMUEGLICH

Abb. 4: Bildschirmausgabe
zur Gültigkeitskontrolle

bereich auf den Zeitraum der Jahre 1901 bis 2099 beschränkt werden, da einige nachfolgende Rechenalgorithmen (z. B. Berechnung der Schaltjahre und Berechnung der Ostertermine) dann weniger aufwendig werden.

Die Bereichs- und Verträglichkeits-Kontrollen werden in einem Unterprogramm codiert, um sie in nachfolgenden Programmen unverändert übernehmen zu können und bei Bedarf von verschiedenen Stellen eines Hauptprogramms aufrufen zu können. Die Steuerung der Annahme bzw. Ablehnung des eingegebenen Datums erfolgt durch die Fehlermeldung FMS. Ist FMS leer, d. h. FMS = „“, dann liegt kein Fehler

Wiederhole, bis fehlerfreie Eingabe	
Datum (1901 bis 2099) eingeben: z. B. 4. 3. 1984	
Aufspalten der Eingabe in Tag TA, Monat MO, Jahr JA	
Bereichskontrolle mit Fehlermeldung FM\$(Abb. 6)	
Ist Fehlermeldung leer?	
nein	ja
Fehlermeldung ausgeben	Verträglichkeitskontrolle mit Fehlermeldung FM\$(Abb. 2)
	Ist Fehlermeldung leer?
	nein
Fehlermeldung ausgeben	(Gültiges Datum)

Abb. 5: Gültigkeitskontrolle

vor, andernfalls wird im FM\$ der Fehler angegeben. Im Hauptprogramm kann dann entsprechende Vorkehrung beim Auftreten eines Fehlers getroffen werden, indem z. B. eine neue Datumseingabe gefordert wird.

Im nachfolgenden Programm werden in den Anweisungen 5010 bis 5130 die Tages-, Monats- und Jahresangaben abgespalten und in numerische Werte umgewandelt, in den Anweisungen 5140 bis 5180 die Bereichskontrolle und in den Anweisungen 5190 bis 5250 die Verträglichkeitskontrolle durchgeführt. Ab Anweisung 260 kann das eigentliche Hauptprogramm eingefügt werden.

Die meisten der nachfolgenden Programme gelten für den gesamten Zeitraum des Gregorianischen Kalenders, also auch über das Jahr 2099 hinaus, falls dann dieser Kalender überhaupt noch gilt und nicht durch eine (bisher oft versuchte aber nicht durchgeführte) andere Kalender-Reform abgelöst worden ist. Die Datums-Eingabe erfolgt bei diesen Programmen analog zum 1. Programm. Nur dann sind die nachfolgenden Programme auf den Zeitraum von 1901 bis 2099 beschränkt, wenn in ihnen die Berechnung des Oster-Termines (siehe Kapitel 9) benötigt wird. Die Datums-Eingabe erfolgt in diesem Fall analog zum 2. Programm.

Initialisiere Fehlermeldung FM\$ = „ „	
Ist $1 \leq TA \leq 31$?	
nein	ja
FM\$ = „Falscher Tag“	
Ist $1 \leq MO \leq 12$?	
nein	ja
An FM\$ anfügen: „Falscher Monat“	
Ist $1 \leq JA \leq 99$?	
nein	ja
	Ergänze: JA = 1900 + JA
Ist $1901 \leq JA \leq 2099$?	
nein	ja
An FM\$ anfügen: „Falsches Jahr“	

Abb. 6: Bereichskontrolle

2. Programm: Datum-Eingabe und Kontrolle

```
1 REM          2. KALENDER-PROGRAMM
10 PRINT "□ DATUM EINGABE UND KONTROLLE
11 PRINT " VOM 1.1.1901 BIS 31.12.2099
12 PRINT " -----
200 PRINT
210 PRINT "DATUM EINGEBEN IN DER FORM (BEISPIEL):
211 PRINT "4.3.1984 ODER 04.03.1984 ODER 4.3.84
212 PRINT
213 PRINT "GILT NUR VOM 1.1.1901 BIS 31.12.2099 !
214 PRINT
215 PRINT "BET ZWEISTELLIGER JAHRESZAHL WIRD 1900 ADDIERT!
216 PRINT
220 INPUT "DATUM":DA#
230 GOSUB 5000
240 IF FM#="" THEN 260
250 PRINT "J":FM# :PRINT :GOTO 220
260 PRINT "DATUM GUELTIG!"
270 PRINT "-----"
280 PRINT :PRINT :GOTO 220
5000 REM-----
5001 REM          DATUM-KONTROLLE
5002 REM-----
5010 FM#="" :L=LEN(DA#) :IF L<6 THEN 5050
5020 FOR I=1 TO L
5030 IF MID$(DA#,I,1)="#" THEN J=I :I=L :GOTO 5060
5040 NEXT I
5050 FM#="FEHLER!" :RETURN
5060 IF J=1 OR J>L-4 THEN 5050
5070 TA=VAL(LEFT$(DA#,J-1))
5080 FOR I=J+1 TO L
5090 IF MID$(DA#,I,1)="#" THEN K=I :I=L :GOTO 5110
5100 NEXT I :GOTO 5050
5110 IF K=J+1 OR K>L-2 THEN 5050
5120 MO=VAL(MID$(DA#,J+1,K-J-1))
5130 JA=VAL(RIGHT$(DA#,L-K))
5140 IF TA<1 OR TA>31 THEN FM#="FALSCHER TAG! "
5150 IF MO<1 OR MO>12 THEN FM#="FALSCHER MONAT! "
5160 IF 0<JA AND JA<100 THEN JA=JA+1900
5170 IF JA<1901 OR JA>2099 THEN FM#="FALSCHES JAHR! "
5180 IF FM#>"" THEN RETURN
5190 IF MO=10RMO=3ORMO=5ORMO=7ORMO=8ORMO=10RMO=12 THEN RETURN
5200 IF TA=31 THEN FM#="KEINE 31 TAGE!":RETURN
5210 IF MO <> 2 THEN RETURN
5220 IF TA=30 THEN FM#="FEBRUAR HAT NICHT 30 TAGE!":RETURN
5230 IF TA<29 THEN RETURN
5240 IF JA/4=INT(JA/4) THEN RETURN
5250 FM#="KEIN SCHALTJAHR!" :RETURN
6000 END
```

DATUM EINGABE UND KONTROLLE
VOM 1.1.1901 BIS 31.12.2099

DATUM EINGEBEN IN DER FORM (BEISPIEL):
4.3.1984 ODER 04.03.1984 ODER 4.3.84

GILT NUR VOM 1.1.1901 BIS 31.12.2099 !

BEI ZWEISTELLIGER JAHRESZAHL WIRD 1900 ADDIERT!

DATUM? 29.2.1984
DATUM GUELTIG!

DATUM? 29.2.84
DATUM GUELTIG!

DATUM? 29.2.85
~~KEINE DATUMGABE!~~

DATUM? 31.5.85
DATUM GUELTIG!

DATUM? 31.6.85
~~KEINE DATUMGABE!~~

DATUM? 30.6.85
DATUM GUELTIG!

Abb. 7: Datum-Eingabe und Kontrolle

3 Tagesnummer eines Datums