

RECHNEN.BAS	0.45	18. 3.2017
-------------	------	------------

B E S C H R E I B U N G :

- ermittelt die numerischen Bezüge historischer Daten mit 4 Methoden

Q U E L L E N - A N G A B E N :

9. Gedenket des Vorigen von Alters her; denn ich bin Gott, und keiner mehr, ein Gott, desgleichen nirgend ist.

10. Der ich verkündige zuvor, was hernach kommen soll, und vorhin, ehe denn es geschieht, und sage: Mein Anschlag besteht, und ich thue Alles, was mir gefällt.

(Jesaja 46)

18. Denn ich sage euch, wahrlich, bis daß Himmel und Erde zergehe, wird nicht zergehen der kleinste Buchstabe, noch ein Titel vom Gesetz, bis daß es alles geschehe.

(Matthäus 5)

Bibel

"Die Bibel, oder die ganze Heilige Schrift Alten und Neuen Testaments " "nach der deutschen Uebersetzung Dr. Martin Luthers.", Revision durch Dr. J. Ph. Fresenius, (1751); Druck und Verlag von Heinrich Ludwig Brönner, Frankfurt am Main, 40.Auflage, (1841)

B E A R B E I T U N G :

12.11.1994	-	2.12.1994	Norbert Südland
------------	---	-----------	-----------------

12. 2.2000	-	28. 3.2003	Norbert Südland
------------	---	------------	-----------------

18. 4.2007	-	18. 3.2017	Norbert Südland
------------	---	------------	-----------------

VORBEREITUNG:

Erste Befehle:

```
'OPTION EXPLICIT           'Bei Visual Basic hilfreich!
OPTION BASE 1              'Alle Feld-Indices beginnen mit `1`
COMMON Historik%, Arbeitsplatz$, Zeit$, Zaehlweise%
'Die Anweisung ``$DYNAMIC` funktioniert nicht immer zuverlässig!
```

Konstanten:

```
CONST RICHTIG = -1          'Durch BASIC vorgegeben
CONST FALSCH = 0           'Durch BASIC vorgegeben
CONST Konfigurationsdatei$ = "RECHNEN.CFG"
CONST VordatierungFruehestens$ = "1" 'Unvollständige Vordatierung
CONST VordatierungKennzeichen$ = "1" 'Vollständige Vordatierung
```

```
CONST VordatierungSpaetestens$ = "||" 'Unvollständige Vordatierung
```

```
Unterprogramme ankündigen:
```

```
Strukturen behandeln:
```

```
DECLARE FUNCTION STRLEN% (Text$, EndeZeichen$)
DECLARE FUNCTION SIZEOF% (StrukturName$)
DECLARE FUNCTION SchnellePosition% (Liste$(), Name$)

DECLARE SUB Einsortieren (Name$, Position&, NamensTyp%)
```

```
Zeittafel erstellen:
```

```
DECLARE SUB Addieren (Summand1$, Art$, Richtung$, v$, OZahl1%, OZahl2%, w$)
DECLARE SUB Aendern (Daten$, Variable$, Inhalt$)
DECLARE SUB Auswerten (Rest%, RL&, Num&, Was%, Zaehlweise%)
DECLARE SUB BezuegeRueckwaerts (Num&)
DECLARE SUB BezuegeVorwaerts (Num&)
DECLARE SUB DatierungAbschreiben (Num&)
DECLARE SUB Ergebnis (Datei%, DL&, Vorgabe$, Vergleich$, Ziel%, Quelle%, N&)
DECLARE SUB Korrektur (Feld$, Rest%, RL&)
DECLARE SUB Optimieren ()
DECLARE SUB Pause ()
DECLARE SUB Protokollmeldung (Meldetext$, NeueZeile%)
DECLARE SUB Pruefen (Daten$)
DECLARE SUB Restklasse (Vorzeichen%, Jahr%, Monat%, Tag%)
DECLARE SUB Schreiben (Datei%, DL&, Datum$, Ziel%, Art%, Qu%, Ort&, Num&)
DECLARE SUB DatierungSpeichern (Position&, Daten$)
DECLARE SUB StatusNeu (Datum$)
DECLARE SUB TerminZuZahl (Termin$, VZ%, Jahr%, Monat%, Tag%)
DECLARE SUB Umkehren (v1%, j1%, m1%, t1%, v2%, j2%, m2%, t2%, Richtung$)
DECLARE SUB Verschieben (Bezug$(), BL&(), Quelle%, Ziel%)
DECLARE SUB ZeitSchieben (v1%, j1%, m1%, t1%, S$, v2%, j2%, m2%, t2%, R$)
DECLARE SUB ZeitZuZahl (Zeitpunkt$, Jahr%, Monat%, Tag%)
DECLARE SUB Zusammenhang (Num&)

DECLARE FUNCTION Abfrage$ (Frage$, Meldung$)
DECLARE FUNCTION BerechnungStarten$ (Arbeitsplatz$, Neu%, Rest%, RL&, ZW%)
DECLARE FUNCTION Einlesen& (Num&, Daten$)
DECLARE FUNCTION Ereignis% (Symb$)
DECLARE FUNCTION FuerDatei$ (Symbol$)
DECLARE FUNCTION FuerNutzer$ (Symbol$)
DECLARE FUNCTION GleichzeitigTest$ (Text$)
DECLARE FUNCTION Laden$ (Datei%, DatensatzLaenge%, DateiLaenge&, Position&)
DECLARE FUNCTION Namenseintrag$ (Position&)
DECLARE FUNCTION Reihenfolge& (Zaehlweise%, Listenlaenge&, Listenelement&)
DECLARE FUNCTION Schnittmenge$ (Datum1$, Datum2$)
DECLARE FUNCTION Suchen& (Name$, Ort&, NamensTyp%)
DECLARE FUNCTION Summe$ (Zeit$, Tol$, Datum$, Richtung$)
```

```

DECLARE FUNCTION Symbol$ (Ereign%)
DECLARE FUNCTION Teil$ (Daten$, Variable$)
DECLARE FUNCTION Widerspruch% (Datum$)
DECLARE FUNCTION ZahlZuTermin$ (VZ%, Jahr%, Monat%, Tag%)
DECLARE FUNCTION ZahlZuZeit$ (Jahr%, Monat%, Tag%, Wie%)

```

Globale Variablen:

```

DIM SHARED Namendatei$(5)           'AS STRING
DIM SHARED Bezugdatei$(6)
DIM SHARED Protokolldatei$
DIM SHARED GlobVarAnzahl%           'AS INTEGER
DIM SHARED Namen$(5)
DIM SHARED Bezug$(6)
DIM SHARED Eingabe%
DIM SHARED Datenlaenge%
DIM SHARED Feldlaenge%
DIM SHARED GEL&                     'AS LONG
DIM SHARED NL&(5)
DIM SHARED BL&(6)

```

'Für Speicherplatzbilanz Datenfelder anlegen:

'-----'

```

REDIM SHARED GVAufang$(1)
REDIM SHARED GVLaege$(1)
REDIM SHARED GVName$(1)
REDIM SHARED GVTyp$(1)

```

Lokale Variablen:

```

DIM Eingabedatei$      'AS STRING
DIM Puffer$
DIM Programm$
DIM Text$
DIM Name$
DIM Positionsdatei$
DIM Listdatei$
DIM Fertigdatei$
DIM Liste$
DIM Datenrest$
DIM Daten$
DIM Variable$
DIM Feld$
DIM Gleichzeitig$
DIM Unterbrechen$
DIM Frage$
DIM Vergleich$
DIM Dateikopie$
DIM Rest%              'AS INTEGER
DIM Fertig%
DIM Neu%
DIM z%

```

```

DIM Konfiguration%
DIM Position%
DIM L%
DIM Leseversuch%
DIM Gefunden%
DIM Weiter%
DIM Protokoll%
DIM ZW%
DIM Kopie%
DIM Num&           'AS LONG
DIM Ort&
DIM LA&           'Listenanfang
DIM RLStart&
DIM RL&           'Restlistenlänge
DIM N&

```

HAUPTTEIL

```

'Vorbereitung:
'-----'
CLEAR , , 4096
ON ERROR GOTO Fehler
GOSUB ProgrammStart
ZW% = Zaehlweise%
Unterbrechen$ = BerechnungStarten$(Arbeitsplatz$, Neu%, Rest%, RL&, ZW%)
IF Unterbrechen$ = "J" THEN GOTO Programmende

'Bezugsdateien auswerten:
'-----'
L% = SIZEOF$( "Datenrest" )
IF LA& = 0 THEN
    LA& = 1
END IF
RLStart& = RL&
WHILE LA& <= RL&

    'Vorbereitung:
    '-----'
    Datenrest$ = SPACE$(L%)
    GET #Rest%, L% * LA& + 1, Datenrest$
    PUT #Fertig%, L% * LA& + 1, Datenrest$
    IF VAL(Teil$(Datenrest$, "Datenrest.Eintrag")) <> LA& THEN
        Text$ = "Die Berechnungstabelle ist durcheinander geraten!"
        Protokollmeldung Text$, 1
        Pause
        GOTO Programmende
    END IF
    Num& = VAL(Teil$(Datenrest$, "Datenrest.Nummer"))
    z% = Ereignis%(Teil$(Datenrest$, "Datenrest.Z"))
    Protokollmeldung "", 1
    Text$ = LTRIM$(STR$(FRE(0))) + " Bytes noch frei."
    Protokollmeldung Text$, 1

    'Berechnung unterbrechen?
    '-----'

```

```

Frage$ = "Berechnung unterbrechen? "
Text$ = "Die Berechnung wurde unterbrochen. "
Unterbrechen$ = Abfrage$(Frage$, Text$)
IF Unterbrechen$ = "J" THEN GOTO Programmende

'Beginn des nächsten Datensatzes:
'-----'
Text$ = "(" + LTRIM$(STR$(LA&)) + "/" + LTRIM$(STR$(RL&))
Text$ = Text$ + ").Eintrag: "
Protokollmeldung Text$, 0
N& = Einlesen&(Num&, Daten$)
Variable$ = "Daten.Datum[" + LTRIM$(STR$(z%)) + "]"
Text$ = Teil$(Daten$, Variable$) + " " + FuerNutzer$(Symbol$(z%))
Text$ = Text$ + " " + Namenseintrag$(Num&) + STR$(Num&)
Protokollmeldung Text$, 1

'Zusammenhang ermitteln:
'-----'
Zusammenhang Num&

'Gleichzeitige Ereignisse bei Anzeige beachten:
'-----'
Gleichzeitig$ = FuerNutzer$(Teil$(Daten$, "Daten.Gleichzeitig"))
Gefunden% = INSTR(Gleichzeitig$, Symbol$(z%))
IF Gefunden% > 0 THEN
    IF Gefunden% > 3 THEN
        z% = Ereignis%(MID$(Gleichzeitig$, 4, 1))
    ELSE
        z% = Ereignis%(MID$(Gleichzeitig$, 1, 1))
    END IF
END IF
Feld$ = SPACE$(Feldlaenge%)
FOR Ort& = 0 TO BL$(z%)
    GET #Bezug$(z%), Ort& * Feldlaenge% + 1, Feld$
    Text$ = LEFT$(Feld$, Feldlaenge% - 2)
    Protokollmeldung Text$, 1
NEXT Ort&
IF LA& <= 0 THEN 'Debugging-Möglichkeit
    Pause

'Überprüfung der Rechnungen ermöglichen:
'-----'
FOR z% = 1 TO 6
    CLOSE #Bezug$(z%)
NEXT z%
'-----'
'Hier können andere Editoren auf die Ergebnisse zugreifen:
'-----'
SHELL "C:\BIBEL\UED.EXE " + Arbeitsplatz$ + "*.BZ*"
FOR z% = 1 TO 6
    OPEN Bezugdatei$(z%) FOR BINARY ACCESS READ WRITE AS #Bezug$(z%)
NEXT z%
Pause
z% = Ereignis%(Teil$(Datenrest$, "Datenrest.Z"))
END IF

'Rückprüfung zur ermittelten Datierung:
'-----'

```

```

Auswerten Rest%, RL%, Num%, z%, Zaehlweise%
IF LA% <= 0 THEN
    Pause
END IF

'Lesepause im Sekundentakt:
'-----'
'Weiter% = 0
'TIMER ON
'ON TIMER(1) GOSUB Weitermachen
'WHILE Weiter% = 0
'WEND
'TIMER OFF
'Pause

LA% = LA% + 1
WEND

'Eventuell mangelhafte Datierung anprangern:
'-----'
IF Neu% = 1 AND RL% = RLStart% THEN
    Text$ = "Alle Vordatierungen sind zusammenhangslos."
    Protokollmeldung Text$, 1
END IF

'=====
Programmende:
'=====
'Eventuell Fehlernummer angeben:
'-----'
IF ERR > 0 THEN
    Text$ = "Fehler Nr." + STR$(ERR) + " in Zeile" + STR$(ERL)
    Text$ = Text$ + " aufgetreten."
1    Protokollmeldung Text$, 1
    Pause
END IF

'Eventuell Rechenvorgang abschließen:
'-----'
IF Unterbrechen$ <> "J" AND Rest% <> 0 AND Fertig% <> 0 THEN
    L% = SIZEOF("Datenrest")
    Datenrest$ = STRING$(L%, "-")
    Aendern Datenrest$, "Datenrest.Ende", "|" + CHR$(13) + CHR$(10)
    PUT #Rest%, L% * (RL% + 1) + 1, Datenrest$
    PUT #Fertig%, L% * (RL% + 1) + 1, Datenrest$
    CLOSE #Fertig%
    CLOSE #Rest%
    Protokollmeldung "", 1
    Protokollmeldung "Die Berechnung wurde erfolgreich abgeschlossen.", 1

    ZW% = Zaehlweise%
    Zaehlweise% = (Zaehlweise% MOD 4) + 1
    Text$ = "Neue Rechenmethode: Zählweise Nummer" + STR$(Zaehlweise%) + "."
    Protokollmeldung Text$, 1
END IF

'Rechenzeit angeben:
'-----'

```

```

Protokollmeldung "", 1
Zeit$ = "Rechenbeginn: " + Zeit$ + CHR$(13) + CHR$(10)
Text$ = LEFT$(Zeit$, LEN(Zeit$) - 2)
Protokollmeldung Text$, 1
z% = LEN(Zeit$)
Zeit$ = "Rechenende: " + DATE$ + SPACE$(2) + TIME$
Protokollmeldung Zeit$, 1
BEEP: BEEP: BEEP

```

```

'Dateien schließen:
'-----'

```

```

CLOSE
FOR z% = 1 TO 6
  IF Bezugdatei$(z%) <> "" THEN
    KILL Bezugdatei$(z%)
  END IF
NEXT z%
FOR z% = 1 TO 5
  IF Namendatei$(z%) <> "" THEN
    KILL Namendatei$(z%)
  END IF
NEXT z%

```

```

'Datenfelder freigeben:
'-----'

```

```

REDIM SHARED GVAufang$(1)
REDIM SHARED GVLaege$(1)
REDIM SHARED GVName$(1)
REDIM SHARED GVTyp$(1)
Positionsdatei$ = ""
Liste$ = ""
Datenrest$ = ""
Daten$ = ""
Variable$ = ""
Feld$ = ""
Gleichzeitig$ = ""
Frage$ = ""
Vergleich$ = ""

```

```

'Stand der Bearbeitung:
'-----'

```

```

Text$ = LTRIM$(STR$(RL&)) + " Einträge in Listdatei:"
Protokollmeldung Text$, 1
Protokollmeldung Listdatei$, 1
Text$ = LTRIM$(STR$(FRE(0))) + " Bytes noch frei."
Protokollmeldung Text$, 1

```

```

'Eventuell Ergebnis kopieren:
'-----'

```

```

IF Unterbrechen$ <> "J" AND Neu% = 1 THEN
  Protokollmeldung "", 1

```

```

2  MKDIR Arbeitsplatz$ + LTRIM$(STR$(ZW%))
3  Protokollmeldung "Die Ergebnisdateien werden kopiert nach:", 1
   Protokollmeldung Arbeitsplatz$ + LTRIM$(STR$(ZW%)) + "\*.*", 1

```

```

Protokollmeldung Eingabedatei$, 1
Eingabe% = FREEFILE

```

```

OPEN Eingabedatei$ FOR INPUT AS #Eingabe%
    Dateikopie$ = Arbeitsplatz$ + LTRIM$(STR$(ZW%)) + "\" + Name$ + "HQL"
    GOSUB DateiKopieren
CLOSE Eingabe%

Protokollmeldung Listdatei$, 1
Eingabe% = FREEFILE
OPEN Listdatei$ FOR INPUT AS #Eingabe%
    Dateikopie$ = Arbeitsplatz$ + LTRIM$(STR$(ZW%)) + "\" + Name$ + "LST"
    GOSUB DateiKopieren
CLOSE Eingabe%

Protokollmeldung Fertigdatei$, 1
Eingabe% = FREEFILE
OPEN Fertigdatei$ FOR INPUT AS #Eingabe%
    Dateikopie$ = Arbeitsplatz$ + LTRIM$(STR$(ZW%)) + "\" + Name$ + "RDY"
    GOSUB DateiKopieren
CLOSE Eingabe%

Protokollmeldung Protokolldatei$, 1
Eingabe% = FREEFILE
OPEN Protokolldatei$ FOR INPUT AS #Eingabe%
    Dateikopie$ = Arbeitsplatz$ + LTRIM$(STR$(ZW%)) + "\" + Name$ + "LOG"
    GOSUB DateiKopieren
CLOSE Eingabe%
END IF

```

'Restliche Datenfelder löschen:

'-----'

Unterbrechen\$ = ""

Eingabedatei\$ = ""

Listdatei\$ = ""

Fertigdatei\$ = ""

Protokolldatei\$ = ""

Text\$ = ""

Name\$ = ""

Dateikopie\$ = ""

Puffer\$ = ""

'Pause vor Programmverzweigung:

'-----'

Pause

IF Programm\$ <> "" THEN CHAIN Programm\$

SYSTEM

' _____ ENDE DES HAUPTTEILS _____ '

FEHLERBEHANDLUNG

'====='

Fehler:

'====='

SELECT CASE ERR

CASE 5

'Unzulässiger Funktionsaufruf

Pause

'VisualBasic ist nicht kompatibel zur Speicherabfrage!

RESUME NEXT


```
CASE 52                'Unzulässiger Dateiname:
  SELECT CASE ERL
  CASE 501
    PRINT "Protokolldateiname ist nicht vorhanden."
    GOTO Programmende
  CASE ELSE
    Pause
  END SELECT
CASE 53                'Datei nicht gefunden:
  SELECT CASE ERL
  CASE 1
    RESUME NEXT
  CASE 10
    PRINT "Konfigurationsdatei "; Konfigurationsdatei$; " fehlt."
    GOTO Programmende
  CASE 30
    PRINT "Positionsdatei HISTORIK.POS fehlt."
    Leseversuch% = Leseversuch% + 1
    RESUME 21
  CASE 60, 65, 66      'Datei nicht löscherbar.
    RESUME NEXT
  CASE 70
    RESUME 80
  CASE 71
    RESUME 73
  CASE 80
    RESUME NEXT
  CASE ELSE
    Pause
    RESUME NEXT
  END SELECT
CASE 62                'Einlesen nach Dateiende
  SELECT CASE ERL
  CASE 20
    PRINT "Konfigurationsdatei "; Konfigurationsdatei$; " ist leer."
    GOTO Programmende
  CASE 40
    PRINT "Positionsdatei HISTORIK.POS ist leer.";
    GOTO Programmende
  CASE 50
    PRINT "Positionsdatei HISTORIK.POS ist unvollständig."
    GOTO Programmende
  CASE ELSE
    Pause
    RESUME NEXT
  END SELECT
CASE 63                'Unzulässige Datensatznummer
  SELECT CASE ERL
  CASE ELSE
    Pause
    RESUME NEXT
  END SELECT
CASE 70                'Zugriff verweigert
  SELECT CASE ERL
  CASE 1
    'Schreibtests auf `Arbeitsplatz$`
    PRINT "Das Arbeitsplatz-Verzeichnis " + Arbeitsplatz$
    PRINT "ist schreibgeschützt. Es wird über " + CHR$(34);
    PRINT "SET HISTORIKTEMP=.." + CHR$(34) + " auf DOS-Ebene gesetzt."
```

```
        Pause
        SYSTEM          'Harter Programm-Abbruch
CASE ELSE
    Pause
    RESUME NEXT
END SELECT
CASE 75                'Pfad- /Datei-Zugriffsfehler
    SELECT CASE ERL
CASE 2                  'Bestehendes Unterverzeichnis anlegen
    RESUME NEXT
CASE 5                  'Schreibtests auf `Arbeitsplatz$`
    RESUME NEXT
CASE ELSE
    Pause
    RESUME NEXT
END SELECT
CASE 100
    PRINT "Datenstruktur in HISTORIK.POS paßt nicht."
    GOTO Programmende
CASE 101
    Text$ = "Vordatierung fehlt in Datendatei " + Eingabedatei$ + "."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 102
    Text$ = "Name eines Dateieintrags fehlt."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 103
    Text$ = "Angaben zum Vorgänger stimmen nicht."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 104
    Text$ = "Angaben zum Bezugsnamen stimmen nicht."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 105
    Text$ = "Angaben zur Dauer stimmen nicht."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 106
    Text$ = "Zu große Jahreszahl. 4-stellig sollte reichen."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 107
    Text$ = "Quellenangabe zur Vordatierung fehlt."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 108
    Text$ = " in der Datei " + Eingabedatei$ + "."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 109
    Text$ = " in der Datei " + Eingabedatei$ + "."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 110
    Text$ = "Sinnlose Datierung mit Ordnungszahlchaos."
    Protokollmeldung Text$, 1
```

```

    GOTO Programmende
CASE 111
    Text$ = "Unbekannte Struktur kann nicht verändert werden."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 112
    GOTO Programmende
CASE 113
    Text$ = "Widerspruch innerhalb einer Datierung."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 114
    Text$ = "Widerspruch zwischen den Vordatierungen."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 115
    Text$ = "Datierung kann nicht zugeordnet und abgespeichert werden."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 120
    Text$ = "Fehler in Funktion 'Einsortieren':"
    Protokollmeldung Text$, 1
    Text$ = "Doppelter Eintrag in Hilfliste."
    Protokollmeldung Text$, 1
    GOTO Programmende
CASE 190
    'Nichtpositive Strukturlänge abgespeichert:
    Text$ = "Fehler in Funktion 'SIZEOF%':"
    Protokollmeldung Text$, 1
    Text$ = "Nichtpositive Strukturlänge abgespeichert."
    Protokollmeldung Text$, 1
    GOTO Programmende
END SELECT

PRINT "Fehler Nr."; ERR; " in Zeile"; ERL
ON ERROR GOTO 0
GOTO Programmende
' _____ ENDE DER FEHLERBEHANDLUNG _____ '

'
'
' SUBROUTINEN UND FUNKTIONEN
'
'
'=====
ProgrammStart:
'=====
    Zeit$ = DATE$ + SPACE$(2) + TIME$
    CLS

    ' Als `Arbeitsplatz$` wird das Arbeitsverzeichnis verwendet, das
    ' %HISTORIKTEMP%, %TEMP% oder %TMP% angibt.
    ' Falls (bei alten DOS-Versionen) gar kein `Arbeitsplatz$` angegeben ist,
    ' wird versucht, auf dem Datenträger, der auch das Programm enthält, zu
    ' schreiben. Eventuell bricht das Programm ab, wenn der `Arbeitsplatz$`
    ' nicht beschreibbar ist:
    '-----
    Arbeitsplatz$ = ENVIRON$("HISTORIKTEMP")
    IF Arbeitsplatz$ = "" THEN

```

```

    Arbeitsplatz$ = ENVIRON$("QBASICTEMP")
    IF Arbeitsplatz$ <> "" THEN Arbeitsplatz$ = Arbeitsplatz$ + "\"
5    MKDIR Arbeitsplatz$ + "HISTORIK.TMP"
    Arbeitsplatz$ = Arbeitsplatz$ + "HISTORIK.TMP\"
ELSE
    IF Arbeitsplatz$ <> "" THEN Arbeitsplatz$ = Arbeitsplatz$ + "\"
END IF

'Beschreibbarkeit von `Arbeitsplatz$` prüfen:
'-----'
BSAVE Arbeitsplatz$ + "HISTORIK.CHK", 0, 0
KILL Arbeitsplatz$ + "HISTORIK.CHK"

'Konfiguration einlesen:
'-----'
Konfiguration% = FREEFILE
10 OPEN Arbeitsplatz$ + Konfigurationsdatei$ FOR INPUT AS #Konfiguration%
20 LINE INPUT #Konfiguration%, Eingabedatei$
    LINE INPUT #Konfiguration%, Puffer$
    GEL& = VAL(Puffer$)
    LINE INPUT #Konfiguration%, Puffer$
    Neu% = VAL(Puffer$)
    LINE INPUT #Konfiguration%, Puffer$
    IF Zaehlweise% < 1 OR Zaehlweise% > 4 THEN
        Zaehlweise% = VAL(Puffer$)
        IF Zaehlweise% < 1 OR Zaehlweise% > 4 THEN
            Zaehlweise% = 1
        END IF
    END IF
    LINE INPUT #Konfiguration%, Programm$
CLOSE #Konfiguration%

'Rechenprotokoll beginnen:
'-----'
L% = STRLEN$(Eingabedatei$, ".HQL") + 1
Name$ = LEFT$(Eingabedatei$, L%)
Protokolldatei$ = Arbeitsplatz$ + Name$ + "LOG"
Text$ = LTRIM$(STR$(FRE(0))) + " Bytes noch frei."
Protokoll% = FREEFILE
OPEN Protokolldatei$ FOR OUTPUT AS #Protokoll%
    PRINT #Protokoll%, Protokolldatei$
    PRINT #Protokoll%, "Rechenbeginn: "; Zeit$
    PRINT #Protokoll%, Text$
    PRINT #Protokoll%,
CLOSE #Protokoll%

'Strukturpositionen einlesen:
'-----'
Position% = FREEFILE
PRINT "Strukturpositionen von HISTORIK.POS einlesen..."
Leseversuch% = 1
21 SELECT CASE Leseversuch%
CASE 1
    Positionsdatei$ = "HISTORIK.POS"
CASE 2
    Positionsdatei$ = Arbeitsplatz$ + Positionsdatei$
CASE 3
    Konfiguration% = FREEFILE

```

```

        OPEN "HISTORIK.STR" FOR INPUT AS #Konfiguration%
        CLOSE #Konfiguration%
22    OPEN Arbeitsplatz$ + "STRUKTUR.CFG" FOR OUTPUT AS #Konfiguration%
        PRINT #Konfiguration%, "HISTORIK.STR"    'Strukturdatei
        PRINT #Konfiguration%, Positionsdatei$    'tatsächliche Positionsdatei
        PRINT #Konfiguration%, "RECHNEN.BAS"    'Rücksprungprogramm
        CLOSE #Konfiguration%
        CHAIN "STRUKTUR.BAS"
    CASE ELSE
        Pause                                'Programmierfehler!
    END SELECT
30    OPEN Positionsdatei$ FOR INPUT AS #Position%
40    LINE INPUT #Position%, Text$
        L% = STRLEN%(Text$, " ")
        GlobVarAnzahl% = VAL(LEFT$(Text$, L%))
        PRINT LTRIM$(STR$(GlobVarAnzahl%)); " Strukturelemente "
        REDIM SHARED GVAnfang%(GlobVarAnzahl%)    'AS INTEGER
        REDIM SHARED GVLaenge%(GlobVarAnzahl%)    'AS INTEGER
        REDIM SHARED GVName$(GlobVarAnzahl%)    'AS STRING
        REDIM SHARED GVTyp$(GlobVarAnzahl%)    'AS STRING
        FOR z% = 1 TO GlobVarAnzahl%
50        INPUT #Position%, GVAnfang%(z%)
            INPUT #Position%, GVLaenge%(z%)
            INPUT #Position%, GVName$(z%)
            INPUT #Position%, GVTyp$(z%)
        NEXT z%
        CLOSE #Position%
        Datenlaenge% = SIZEOF%("Daten")
        IF Datenlaenge% = 0 THEN ERROR 100
        Feldlaenge% = SIZEOF%("Feld")

        'Datendatei öffnen:
        '-----'
        Eingabe% = FREEFILE
        Eingabedatei$ = Arbeitsplatz$ + Name$ + "HQL"
        OPEN Eingabedatei$ FOR BINARY ACCESS READ WRITE AS #Eingabe%

        'Namenslisten anlegen:
        '-----'
        L% = SIZEOF%("Liste")
        Liste$ = SPACE$(L%)
        Aendern Liste$, "Liste.Ende", "␣" + CHR$(10) + CHR$(13)
        FOR z% = 1 TO 5
            Namendatei$(z%) = Arbeitsplatz$ + Name$ + "NE" + LTRIM$(STR$(z%))
60        KILL Namendatei$(z%)
            Namen%(z%) = FREEFILE
61        OPEN Namendatei$(z%) FOR BINARY ACCESS READ WRITE AS #Namen%(z%)
            SELECT CASE z%
            CASE 1
                Puffer$ = "Name /Ereignis:"
            CASE 2, 3
                Puffer$ = LTRIM$(STR$(z% - 1)) + ".Vorgänger:"
            CASE 4, 5
                Puffer$ = LTRIM$(STR$(z% - 3)) + ".Bezugsname:"
            CASE ELSE
                Pause                                'Programmierfehler!
            END SELECT
            Aendern Liste$, "Liste.Name", Puffer$ + SPACE$(L%)

```

```

    Aendern Liste$, "Liste.Nummer", "Nummer: "
    PUT #Namen%(z%), 1, Liste$
    NL&(z%) = 0
NEXT z%

'Bezugsdateien anlegen:
'-----'
L% = SIZEOF("Feld")
Liste$ = SPACE$(L%)
Aendern Liste$, "Feld.Ziel", " Z"
Aendern Liste$, "Feld.Name", "Art"
Aendern Liste$, "Feld.Quelle", " Q"
Aendern Liste$, "Feld.von", "von:"
Aendern Liste$, "Feld.nach", "nach:"
Aendern Liste$, "Feld.Ende", "||" + CHR$(13) + CHR$(10)
FOR z% = 1 TO 6
    Bezugdatei$(z%) = Arbeitsplatz$ + Name$ + "BZ" + LTRIM$(STR$(z%))
65    KILL Bezugdatei$(z%)
    Bezug%(z%) = FREEFILE
66    OPEN Bezugdatei$(z%) FOR BINARY ACCESS READ WRITE AS #Bezug%(z%)
    PUT #Bezug%(z%), 1, Liste$
    BL&(z%) = 0
NEXT z%

'Berechnungspläne anlegen:
'-----'
Listdatei$ = Arbeitsplatz$ + Name$ + "LST"
Fertigdatei$ = Arbeitsplatz$ + Name$ + "RDY"
L% = SIZEOF("Datenrest")
Rest% = FREEFILE
70 OPEN Listdatei$ FOR BINARY ACCESS READ AS #Rest%
    Fertig% = FREEFILE
71 OPEN Fertigdatei$ FOR BINARY ACCESS READ AS #Fertig%
    LA& = 0
    RL& = 0
    Datenrest$ = SPACE$(L%)
    Vergleich$ = SPACE$(L%)
    DO
        GET #Rest%, L% * RL& + 1, Datenrest$
        IF LA& = RL& THEN
            GET #Fertig%, L% * LA& + 1, Vergleich$
        END IF
        IF Datenrest$ <> STRING$(L%, 0) THEN
            IF Datenrest$ = Vergleich$ THEN
                IF LEFT$(Datenrest$, L% - 3) = STRING$(L% - 3, "-") THEN
                    Datenrest$ = STRING$(L%, 0)
                    LA& = 0
                    RL& = 0
                ELSE
                    LA& = LA& + 1
                    RL& = RL& + 1
                END IF
            ELSE
                RL& = RL& + 1
            END IF
        END IF
        LOOP UNTIL Datenrest$ = STRING$(L%, 0)
72 CLOSE #Fertig%

```

```

73  CLOSE #Rest%
    IF LA& > 0 THEN
        LA& = LA& - 1
    END IF
    IF RL& > 0 THEN
        RL& = RL& - 1
    END IF
    IF RL& > 0 THEN
        Frage$ = "Begonnene Berechnung fortsetzen?"
        Text$ = "Berechnung wird fortgesetzt."
        IF Abfrage$(Frage$, Text$) = "N" THEN
            LA& = 0
            RL& = 0
        END IF
    END IF

80  IF RL& = 0 THEN
        KILL Listdatei$
        KILL Fertigdatei$
    END IF
    Rest% = FREEFILE
    OPEN Listdatei$ FOR BINARY ACCESS READ WRITE AS #Rest%
    IF RL& = 0 THEN
        Datenrest$ = SPACE$(L%)
        Andern Datenrest$, "Datenrest.Eintrag", "Eintrag:" + SPACE$(L%)
        Andern Datenrest$, "Datenrest.Nummer", "Nummer:" + SPACE$(L%)
        Andern Datenrest$, "Datenrest.Z", "Z"
        Andern Datenrest$, "Datenrest.Name", "Name /Ereignis:" + SPACE$(L%)
        Andern Datenrest$, "Datenrest.Datum", "Datum:" + SPACE$(L%)
        Andern Datenrest$, "Datenrest.Ende", "|" + CHR$(13) + CHR$(10)
        PUT #Rest%, 1, Datenrest$
    END IF

    Fertig% = FREEFILE
    OPEN Fertigdatei$ FOR BINARY ACCESS READ WRITE AS #Fertig%
    IF RL& = 0 THEN
        PUT #Fertig%, 1, Datenrest$
    END IF

RETURN 'ProgrammStart _____'

'=====
Weitermachen:
'=====
' Ermöglicht eine Lesepause von 1 Sekunde.
'
' Bearbeitung:
' 31. 1.2003: Norbert Südland
'-----
Weiter% = 1
RETURN 'Weitermachen _____'

'=====
DateiKopieren:
'=====
' Kopiert eine als Eingabe% offene Datei nach Dateikopie$
'
' Bearbeitung:

```

```

' 18. 3.2017 Norbert Südland
'-----'

Kopie% = FREEFILE
OPEN Dateikopie$ FOR OUTPUT AS #Kopie%
  WHILE EOF(Eingabe%) = 0
    LINE INPUT #Eingabe%, Puffer$
    PRINT #Kopie%, Puffer$
  WEND
CLOSE Kopie%

RETURN 'DateiKopieren _____'

'=====
FUNCTION Abfrage$ (Frage$, Meldung$)
'=====
'Stellt nach [ ESC ] die Frage$ und erwartet "J" für Ja oder "N" für Nein.
'Bei [ ESC ] wird "N" zurückgegeben, bei "[ Enter ]" wird "J" zurückgegeben.
'
'Bearbeitung: 4. 3.2017 - 17. 3.2017 Norbert Südland, Aalen
'-----'
DIM Antwort$          'AS STRING

SELECT CASE INKEY$
CASE CHR$(27)
DO
  PRINT Frage$; " [ J / N ] ";
  LOCATE CSRLIN, POS(0) - 2
DO
  Antwort$ = INKEY$
  LOOP UNTIL Antwort$ <> ""
  SELECT CASE LEFT$(Antwort$, 1)
  CASE "J", "j", CHR$(13)
    Antwort$ = "J"
    PRINT Antwort$
    Protokollmeldung Meldung$, 1
  CASE "N", "n", CHR$(27)
    Antwort$ = "N"
    PRINT Antwort$
  CASE ELSE
    Antwort$ = ""
    PRINT
  END SELECT
  LOOP UNTIL Antwort$ <> ""
END SELECT

Abfrage$ = Antwort$
END FUNCTION 'Abfrage$ _____'

'=====
SUB Addieren (Summand1$, Art$, Richtung$, Vorgabe$, OZahl1%, OZahl2%, Weiter$)
'=====
' Addiert `Summand1$` und `Vorgabe$` zusammen.
'
' `Summand1`:  Erste Zeitangabe
' `Art$`:      "U": Vereinigungsmenge
'              ">": übernimmt alle Eigenschaften der `Vorgabe$`

```



```
' `Richtung`:      "+": Addition      "-": Subtraktion
' `Vorgabe$`:      Zweite Zeitangabe, wirkt als Vorgabe der Eigenschaften
' `OZahl1`:        <> 0: `Summand1$` ist eine Ordnungszahl
' `OZahl2`:        <> 0: `Vorgabe$` ist eine Ordnungszahl
' `Weiter$`:
'
' Bearbeitung:
' 27. 6.2001 - 4. 9.2001:      Norbert Südland, München
' -----'
```

```
' Vorbereitung: '
' ====='
DIM Summand2$      'AS STRING
DIM Status1$
DIM Status2$
DIM Status$
DIM Termin1$
DIM Termin2$
DIM Groesse%      'AS INTEGER
DIM v0%, v1%, v2%, v3%, v4%, v5%, v6%
DIM j0%, j1%, j2%, j3%, j4%, j5%, j6%
DIM m0%, m1%, m2%, m3%, m4%, m5%, m6%
DIM t0%, t1%, t2%, t3%, t4%, t5%, t6%
DIM Start%
DIM Ende%
DIM z%
```

```
Summand2$ = Vorgabe$
```

```
' Jeweils aktuellen Status ermitteln:
' -----'
StatusNeu Summand1$
Status1$ = Teil$(Summand1$, "Zeitangabe.Status")
```

```
StatusNeu Summand2$
Status2$ = Teil$(Summand2$, "Zeitangabe.Status")
```

```
' Eventuell ist gar nichts zu addieren:
' -----'
Groesse% = SIZEOF$( "Zeitangabe" )
IF Summand1$ = SPACE$(Groesse%) AND Summand2$ = SPACE$(Groesse%) THEN
    GOTO AddierenEnde
END IF
```

```
' Art der Addition beachten: '
' ====='
SELECT CASE Art$
    CASE "U"      'Vereinigungsmenge:
        Status$ = Status1$
        IF Status$ <> Status2$ THEN
            Status$ = " "
        END IF
    CASE ">>"      'Summand2$ gibt alles vor:
        Status$ = Status2$
        IF Summand2$ = SPACE$(Groesse%) THEN
            Summand1$ = Summand2$
            GOTO AddierenEnde
```

```

    END IF
    CASE ELSE 'nicht vorgesehen:
        GOTO AddierenEnde
    END SELECT

' Eigentliche Addition: '
'=====
TerminZuZahl Teil$(Summand1$, "Zeitangabe.Minimum"), v3%, j3%, m3%, t3%
TerminZuZahl Teil$(Summand2$, "Zeitangabe.Minimum"), v4%, j4%, m4%, t4%
TerminZuZahl Teil$(Summand1$, "Zeitangabe.Maximum"), v5%, j5%, m5%, t5%
TerminZuZahl Teil$(Summand2$, "Zeitangabe.Maximum"), v6%, j6%, m6%, t6%
IF OZahl1% <> 0 THEN
    ZeitSchieben v3%, j3%, m3%, t3%, Status1$, v5%, j5%, m5%, t5%, "-"
END IF
IF OZahl2% <> 0 THEN
    ZeitSchieben v4%, j4%, m4%, t4%, Status2$, v6%, j6%, m6%, t6%, "-"
END IF

Umkehren v4%, j4%, m4%, t4%, v6%, j6%, m6%, t6%, Richtung$

SELECT CASE Status$
CASE ">"
    Start% = 1
    Ende% = 1
CASE "<"
    Start% = 2
    Ende% = 2
CASE " "
    Start% = 1
    Ende% = 2
END SELECT
FOR z% = Start% TO Ende%
    IF z% = 1 THEN
        v1% = v3%: j1% = j3%: m1% = m3%: t1% = t3%
        v2% = v4%: j2% = j4%: m2% = m4%: t2% = t4%
    ELSE
        v1% = v5%: j1% = j5%: m1% = m5%: t1% = t5%
        v2% = v6%: j2% = j6%: m2% = m6%: t2% = t6%
    END IF
    t0% = t1% + t2%
    m0% = m1% + m2%
    j0% = v1% * j1% + v2% * j2%
    v0% = 1
    Restklasse v0%, j0%, m0%, t0%
    IF z% = 1 THEN
        v3% = v0%: j3% = j0%: m3% = m0%: t3% = t0%
    ELSE
        v5% = v0%: j5% = j0%: m5% = m0%: t5% = t0%
    END IF
NEXT z%
IF (OZahl1% <> 0 OR OZahl2% <> 0) AND Weiter$ <> "-" THEN
    ZeitSchieben v3%, j3%, m3%, t3%, Status$, v5%, j5%, m5%, t5%, "+"
END IF

SELECT CASE Status$
CASE ">"
    Termin1$ = ZahlZuTermin$(v3%, j3%, m3%, t3%)
    Termin2$ = SPACE$(SIZEOF("Termin"))

```

```

CASE "<="
    Termin1$ = SPACE$(SIZEOF$( "Termin" ))
    Termin2$ = ZahlZuTermin$(v5%, j5%, m5%, t5%)
CASE " "
    Termin1$ = ZahlZuTermin$(v3%, j3%, m3%, t3%)
    Termin2$ = ZahlZuTermin$(v5%, j5%, m5%, t5%)
END SELECT
Aendern Summand1$, "Zeitangabe.Minimum", Termin1$
Aendern Summand1$, "Zeitangabe.Status", Status$
Aendern Summand1$, "Zeitangabe.Maximum", Termin2$

'=====
AddierenEnde:
'=====
END SUB 'Addieren _____

'=====
SUB Aendern (Daten$, Variable$, Inhalt$)
'=====
' Fügt `Inhalt$` an der richtigen Stelle in `Daten$` ein.
' `l` gibt die Länge des zu betrachtenden Variablen-Strings an.
' `p` gibt die Länge bis zum Auftreten des nächsten Punktes "." an.
'
' Bearbeitung:
' 27. 6.2001 - 12. 2.2003: Norbert Südland
'-----
DIM Suche$      'AS STRING
DIM L%          'AS INTEGER
DIM p%
DIM Anfang%
DIM Wo%
DIM i%

L% = LEN(Variable$)
p% = STRLEN$(Variable$, ".")

IF p% = L% THEN
    Suche$ = Variable$      'kein Strukturpunkt vorhanden
ELSE
    'mindestens ein Strukturpunkt vorhanden:
    p% = p% + 1 + STRLEN$(MID$(Variable$, p% + 2, L% - (p% + 1)), ".")
    IF p% < L% THEN
        'zweiter Strukturpunkt vorhanden:
        Suche$ = LEFT$(Variable$, p%)      'Zuerst nach Grobstruktur suchen!
    ELSE
        Suche$ = Variable$
    END IF
END IF

Anfang% = 1
DO
    Wo% = SchnellePosition$(GVName$(), Suche$)
    IF Wo% = 0 THEN ERROR 111

    Anfang% = Anfang% + GVAnfang%(Wo%) - 1

    IF p% < L% THEN
        'Unterstruktur berücksichtigen:
        Suche$ = GVTyp$(Wo%) + MID$(Variable$, p% + 1, L% - p%)
        p% = p% + 1 + STRLEN$(MID$(Variable$, p% + 2, L% - (p% + 1)), ".")
    
```

```

    ELSE
        p% = L% + 1          'Ausstieg aus der Schleife!
    END IF
LOOP WHILE p% <= L%

i% = LEN(Inhalt$)
IF i% < GVLaeenge%(Wo%) THEN
    Inhalt$ = SPACE$(GVLaeenge%(Wo%) - i%) + Inhalt$
ELSE
    IF i% > GVLaeenge%(Wo%) THEN
        Inhalt$ = LEFT$(Inhalt$, GVLaeenge%(Wo%))
    END IF
END IF

MID$(Daten$, Anfang%, GVLaeenge%(Wo%)) = Inhalt$
END SUB 'Aendern _____

'=====
SUB Auswerten (Rest%, RL&, Num&, Was%, Zaehlweise%)
'=====
' Wertet einen Datei-Eintrag aus.
'
' Bearbeitung:
' 29. 6.2001 - 31. 1.2003:    Norbert Südland
' 25. 3.2009 - 18. 3.2016:    Norbert Südland, Aalen
'-----

DIM Feld$          'AS STRING
DIM Datum$
DIM Daten$
DIM Datenrest$
DIM Variable$
DIM Gleichzeitig$
DIM Text$
DIM Typ%           'AS INTEGER
DIM L%
DIM Auswertung%
DIM Gefunden%
DIM E%(3)
DIM x%
DIM z&            'AS LONG
DIM Zuerst&
DIM Zaehler&

Typ% = Was%
Feld$ = Laden$(Bezug%(Typ%), Feldlaenge%, BL&(Typ%), BL&(Typ%) - 1)
IF BL&(Typ%) = 1 THEN
    Text$ = "Zusammenhanglose Datierung:"
    Protokollmeldung Text$, 1
    Protokollmeldung Feld$, 1
    Pause
    ERROR 112
ELSE
    'Sind Bezüge vorhanden?

    'Auswerte-Differenz feststellen:
    '-----
    L% = SIZEOF$("Termin")
    Auswertung% = 0    'Datierung akzeptabel

```

```

IF Teil$(Feld$, "Feld.Datum.Minimum") <> SPACE$(L%) THEN
    Auswertung% = 1 'Datierung ändert sich
END IF
IF Teil$(Feld$, "Feld.Datum.Maximum") <> SPACE$(L%) THEN
    Auswertung% = Auswertung% + 2
END IF

'Neu errechnetes Datum ermitteln:
'-----'
Feld$ = Laden$(Bezug$(Typ%), Feldlaenge%, BL$(Typ%), BL$(Typ%))
Datum$ = Teil$(Feld$, "Feld.Datum")
StatusNeu Datum$
Zuerst& = Einlesen&(Num&, Daten$)
Gleichzeitig$ = FuerNutzer$(Teil$(Daten$, "Daten.Gleichzeitig"))

'Datum korrigieren:
'-----'
IF Auswertung% >= 1 THEN
    Protokollmeldung Datum$, 0
    IF Widerspruch$(Datum$) = 1 THEN ERROR 113
    IF Typ% = Ereignis$(Teil$(Feld$, "Feld.Ziel")) THEN

        'Neuen Eintrag erstellen:
        '-----'
        L% = SIZEOF("Datenrest")
        Datenrest$ = SPACE$(L%)
        RL& = RL& + 1
        Aendern Datenrest$, "Datenrest.Eintrag", STR$(RL&)
        Aendern Datenrest$, "Datenrest.Nummer", STR$(Num&)
        Aendern Datenrest$, "Datenrest.Z", Symbol$(Typ%)
        Aendern Datenrest$, "Datenrest.Name", Namenseintrag$(Num&)
        Aendern Datenrest$, "Datenrest.Datum", Datum$
        Aendern Datenrest$, "Datenrest.Ende", "|" + CHR$(13) + CHR$(10)
        PUT #Rest%, RL& * L% + 1, Datenrest$

        'Datum ändern, aber nicht bei Vordatierung:
        '-----'
        Variable$ = "Daten.Datum[" + LTRIM$(STR$(Typ%)) + "]"
        SELECT CASE Teil$(Daten$, Variable$ + ".Status")
        CASE VordatierungFruehestens$ "I"
            IF Auswertung% MOD 2 = 1 THEN
                Pause
                ERROR 114
            END IF
            Variable$ = Variable$ + ".Maximum"
            Aendern Daten$, Variable$, Teil$(Datum$, "Zeitangabe.Maximum")
        CASE VordatierungKennzeichen$ "II"
            Pause
            ERROR 114
        CASE VordatierungSpaetestens$ "III"
            IF Auswertung% MOD 2 = 0 THEN
                Pause
                ERROR 114
            END IF
            Variable$ = Variable$ + ".Minimum"
            Aendern Daten$, Variable$, Teil$(Datum$, "Zeitangabe.Minimum")
        CASE ELSE
            Aendern Daten$, Variable$, Datum$

```

```

END SELECT
DatierungSpeichern Num&, Daten$

Text$ = " " + STR$(RL&) + ".Eintrag auf "
Text$ = Text$ + FuerNutzer$(LTRIM$(Teil$(Datenrest$, "Datenrest.Z")))
Text$ = Text$ + " " + Namenseintrag$(Num&) + STR$(Num&)
Protokollmeldung Text$, 1
ELSE
  Pause
  ERROR 115
END IF
END IF

'Gleichzeitige Ereignisse zurückprüfen:
'-----'
Gefunden% = INSTR(Gleichzeitig$, Symbol$(Typ%))
IF Gefunden% > 0 THEN
  IF Gefunden% > 3 THEN
    FOR x% = 4 TO 5
      E%(x%) = Ereignis$(MID$(Gleichzeitig$, x%, 1))
    NEXT x%
    E%(3) = 0
    Gefunden% = Gefunden% - 3
  ELSE
    FOR x% = 1 TO 3
      E%(x%) = Ereignis$(MID$(Gleichzeitig$, x%, 1))
    NEXT x%
  END IF
  FOR x% = 1 TO 3
    IF x% <> Gefunden% AND E%(x%) > 0 THEN
      Aendern Feld$, "Feld.Ziel", Symbol$(E%(x%))
      Aendern Feld$, "Feld.von", Teil$(Feld$, "Feld.nach")
      Korrektur Feld$, Rest%, RL&
    END IF
  NEXT x%
  Typ% = E%(1)      'Dort stehen alle Bezüge!
END IF

'Neu errechnetes Datum zurückprüfen:
'-----'
FOR Zaehler& = 1 TO BL&(Typ%) - 3
  z& = Reihenfolge&(Zaehlweise%, BL&(Typ%) - 3, Zaehler&) + 1
  Feld$ = Laden$(Bezug$(Typ%), Feldlaenge%, BL&(Typ%), z&)
  'Pause
  Aendern Feld$, "Feld.Datum", Datum$
  Korrektur Feld$, Rest%, RL&
NEXT Zaehler&
END IF

END SUB 'Auswerten _____'

'=====
FUNCTION BerechnungStarten$ (Arbeitsplatz$, Neu%, Rest%, RL&, ZW%)
'=====
'-----
' Erstellt 5 Listen mit den Namen der Dateieinträge und ihrem Auftreten.
' Die Listen sind alphabetisch sortiert.
'-----

```

```

' Neuberechnung:
' Alle Datierungen außer Vordatierung löschen.
' Vordatierung wird in `Rest%` eingetragen.
'-----
' keine Neuberechnung:
' Alle Datierungen werden auf Richtigkeit geprüft. Mindestens EINE Vorda=
' tierung muß existieren. Abweichungen werden in `Rest%` eingetragen.
'-----
' Bearbeitung:
' 4. 8.2001 - 4. 9.2001:      Norbert Südland, München
' 18.12.2002 - 18. 3.2003:    Norbert Südland, Aalen
' 7.10.2016 - 18. 3.2017:    Norbert Südland, Aalen
'-----

DIM Daten$          'AS STRING
DIM Name$
DIM Variable$
DIM Datum$
DIM Text$
DIM Frage$
DIM Abbrechen$

DIM Puffer$
DIM Kopie$
DIM Feld$

DIM x%              'AS INTEGER
DIM y%
DIM zwz%
DIM z%
DIM Groesse%

DIM Gefunden%
DIM S%
DIM w%
DIM L%

DIM Ort&            'AS LONG
DIM Position&
DIM Vordatierung&
DIM Zaehler&

'Vorbereitung:
'-----'
CLS
Vordatierung& = 0
Abbrechen$ = "N"
x% = POS(0)
y% = CSRLIN
Groesse% = SIZEOF%("Zeitangabe")
L% = SIZEOF%("Datenrest")
Protokollmeldung "Rechenmethode: Zählweise Nummer" + STR$(ZW%) + ".", 1
Protokollmeldung "", 1

FOR Zaehler& = 1 TO GEL&

    'Eintrag lesen:
    '-----'

```

```

Ort& = Reihenfolge&(ZW%, GEL&, Zaehler&)
Daten$ = Laden$(Eingabe%, Datenlaenge%, GEL&, Ort&)
Name$ = Teil$(Daten$, "Daten.Name")

'Berechnung abbrechen?
'-----'
Frage$ = "Berechnung abbrechen?"
Text$ = "Die Berechnung wurde abgebrochen."
Abbrechen$ = Abfrage$(Frage$, Text$)
IF Abbrechen$ = "J" THEN GOTO EndeBerechnungStarten

'Bildschirm-Anzeige:
'-----'
Text$ = "Eintrag " + Name$ + " ( " + LTRIM$(STR$(Ort&)) + "/"
Text$ = Text$ + LTRIM$(STR$(GEL&)) + ") wird geprüft."
LOCATE y%, x%
Protokollmeldung Text$, 1
Pruefen (Daten$)

'Namen alphabetisch auflisten:
'-----'
Einsortieren Name$, Ort&, 1
Name$ = Teil$(Daten$, "Daten.V[1].Name")
IF LTRIM$(Name$) <> "" THEN
    Einsortieren Name$, Ort&, 2
END IF
Name$ = Teil$(Daten$, "Daten.V[2].Name")
IF LTRIM$(Name$) <> "" THEN
    Einsortieren Name$, Ort&, 3
END IF
Name$ = Teil$(Daten$, "Daten.B[1].Name")
IF LTRIM$(Name$) <> "" THEN
    Einsortieren Name$, Ort&, 4
END IF
Name$ = Teil$(Daten$, "Daten.B[2].Name")
IF LTRIM$(Name$) <> "" THEN
    Einsortieren Name$, Ort&, 5
END IF

'Vordatierungen behandeln:
'-----'
FOR z% = 1 TO 6
    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(z%)) + "]"
    Datum$ = Teil$(Daten$, Variable$)
    IF Neu% = 1 THEN

        'Alles außer Vordatierung löschen:
        '-----'
        SELECT CASE Teil$(Datum$, "Zeitangabe.Status")
        CASE VordatierungFruehestens$      ' "I"
            Gefunden% = RICHTIG
            Variable$ = "Daten.Datum[ " + LTRIM$(STR$(z%)) + "].Maximum"
            Aendern Daten$, Variable$, SPACE$(SIZEOF$( "Termin" ))
        CASE VordatierungKennzeichen$      ' "II"
            Gefunden% = RICHTIG
        CASE VordatierungSpaetestens$      ' "III"
            Gefunden% = RICHTIG
            Variable$ = "Daten.Datum[ " + LTRIM$(STR$(z%)) + "].Minimum"

```



```

    Aendern Daten$, Variable$, SPACE$(SIZEOF%( "Termin"))
CASE ELSE
    Gefunden% = FALSCH
    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(z%)) + "]"
    Aendern Daten$, Variable$, SPACE$(Groesse%)
END SELECT

IF Gefunden% THEN
    '-----'
    ' Hinweis: Die Vordatierung kann sich auch in Dateieinträgen '
    '           befinden, die nicht zuerst vorkommen. '
    '-----'
    Vordatierung& = Ort&

    'Liste der Rechenaufträge ergänzen:
    '-----'
    Puffer$ = SPACE$(L%)
    RL& = RL& + 1
    Aendern Puffer$, "Datenrest.Eintrag", STR$(RL&)
    Aendern Puffer$, "Datenrest.Nummer", STR$(Ort&)
    Aendern Puffer$, "Datenrest.Z", Symbol$(z%)
    Aendern Puffer$, "Datenrest.Name", Namenseintrag$(Ort&)
    Aendern Puffer$, "Datenrest.Datum", Datum$
    Aendern Puffer$, "Datenrest.Ende", "|" + CHR$(13) + CHR$(10)
    PUT #Rest%, RL& * L% + 1, Puffer$
    Text$ = Datum$ + " " + STR$(RL&) + ".Eintrag auf "
    Text$ = Text$ + FuerNutzer$(Symbol$(z%)) + " "
    Text$ = Text$ + Namenseintrag$(Ort&) + STR$(Ort&)
    Protokollmeldung Text$, 1
    '-----'
    ' In der Liste der Rechenaufträge befinden sich die Nummern '
    ' der Ersteinträge einer zu prüfenden Datierung. '
    '-----'
END IF
ELSE
    'Vordatierung suchen:
    '-----'
    SELECT CASE Teil$(Datum$, "Zeitangabe.Status")
    CASE VordatierungFruehestens$
        Gefunden% = RICHTIG
    CASE VordatierungKennzeichen$
        Gefunden% = RICHTIG
    CASE VordatierungSpaetestens$
        Gefunden% = RICHTIG
    CASE ELSE
        Gefunden% = FALSCH
    END SELECT
    IF Gefunden% THEN
        Vordatierung& = Ort&
        z% = 6
    END IF
END IF
NEXT z%
IF Neu% = 1 THEN
    DatierungSpeichern Ort&, Daten$
END IF
NEXT Zaehler&

```

```

IF Vordatierung& = 0 THEN ERROR 101

'Namensdateien schließen und erneut zum Lesen öffnen:
'-----'
L% = SIZEOF%("Liste")
FOR z% = 1 TO 5
    CLOSE #Namen%(z%)
    OPEN Namendatei$(z%) FOR BINARY ACCESS READ AS #Namen%(z%)
NEXT z%

IF Neu% = 0 THEN
    FOR Zaehler& = 1 TO GEL&
        Ort& = Reihenfolge&(ZW%, GEL&, Zaehler&)
        Name$ = Namenseintrag$(Ort&)
        IF Ort& = Suchen&(Name$, 1, 1) THEN

            'Berechnung abbrechen?
            '-----'
            Frage$ = "Berechnung abbrechen?"
            Text$ = "Die Berechnung wurde abgebrochen."
            Abbrechen$ = Abfrage$(Frage$, Text$)
            IF Abbrechen$ = "J" THEN GOTO EndeBerechnungStarten

            'Meldung:
            '-----'
            CLS
            Text$ = "Bezug zu Eintrag " + Name$ + " (" + LTRIM$(STR$(Ort&))
            Text$ = Text$ + "/" + LTRIM$(STR$(GEL&)) + ") wird geprüft."
            LOCATE y%, x%
            Protokollmeldung Text$, 1
            LOCATE y% + 1, x%

            'Bezüge testen:
            '-----'
            Zusammenhang Ort&

            'Überprüfung der Rechnungen ermöglichen:
            '-----'
            'FOR z% = 1 TO 6
            '    CLOSE #Bezug%(z%)
            'NEXT z%
            '-----'
            'Hier können andere Editoren auf die Ergebnisse zugreifen:
            '-----'
            'SHELL "C:\BIBEL\UED.EXE " + Arbeitsplatz$ + "*.BZ*"
            'FOR z% = 1 TO 6
            '    OPEN Bezugdatei$(z%) FOR BINARY ACCESS READ WRITE AS #Bezug%(z%)
            'NEXT z%
            'Pause

            FOR z% = 1 TO 6
                IF BL&(z%) > 1 THEN
                    FOR Position& = 0& TO BL&(z%)
                        GET #Bezug%(z%), Position& * Feldlaenge% + 1, Feld$
                        Text$ = LEFT$(Feld$, Feldlaenge% - 2)
                        Protokollmeldung Text$, 1
                    NEXT Position&
                ELSE

```

```

        Feld$ = Laden$(Bezug%(z%), Feldlaenge%, BL&(z%), BL&(z%))
    END IF
    IF Teil$(Feld$, "Feld.Datum") <> SPACE$(SIZEOF$("Zeitangabe")) THEN
        IF BL&(z%) = 1 THEN
            Text$ = "Zusammenhanglose Datierung:"
            Protokollmeldung Text$, 1
            Protokollmeldung Feld$, 1
            Pause
            ERROR 112
        ELSE
            Auswerten Rest%, RL&, Ort&, z%, ZW%
        END IF
        Protokollmeldung "", 1
    END IF
NEXT z%
END IF
NEXT Zaehler&
IF RL& = 0 THEN
    Text$ = "Alle Daten sind richtig gerechnet."
    Protokollmeldung Text$, 1
END IF
END IF
END IF

'=====
EndeBerechnungStarten:
'=====
    BerechnungStarten$ = Abbrechen$
END FUNCTION 'BerechnungStarten$ _____'

'=====
    SUB BezuegeRueckwaerts (Num&)
'=====
    ' Erstellt die Rückwärtsbezüge vom aktuellen Dateieintrag zu anderen.
    '
    ' Bearbeitung:
    ' 16. 6.2001 - 23.12.2002:    Norbert Südland
    ' 25. 3.2009 -  7. 3.2017:    Norbert Südland, Aalen
    '-----
    DIM Daten$      'AS STRING
    DIM Name$
    DIM Text$
    DIM Vergleich$
    DIM Variable$
    DIM Vorgaenger$
    DIM Datum$
    DIM Bezug$
    DIM Tol$
    DIM Summand$
    DIM Sum$
    DIM ZP1%        'AS INTEGER
    DIM ZP2%
    DIM w%
    DIM x%
    DIM y%
    DIM z%
    DIM Erg&        'AS LONG
    DIM Ort&

```

```

Erg& = Einlesen&(Num&, Daten$)
Name$ = Teil$(Daten$, "Daten.Name ")
x% = POS(0)
y% = CSRLIN
IF y% = 24 THEN
    PRINT
    y% = 23
END IF

'Vorgänger berücksichtigen:
'-----'
FOR z% = 1 TO 2

    Ort& = Suchen&(Name$, 1, z% + 1)

    WHILE (Ort& <> 0)
        Erg& = Einlesen&(Ort&, Vergleich$)

        Text$ = "Rückwärtsbezüge zu Eintrag " + Teil$(Vergleich$, "Daten.Name ")
        Text$ = Text$ + " ( " + LTRIM$(STR$(Ort&)) + "/" + LTRIM$(STR$(GEL&))
        Text$ = Text$ + ") werden erstellt."
        LOCATE y%, x%
        Protokollmeldung Text$, 1

        Variable$ = "Daten.V[ " + LTRIM$(STR$(z%)) + "]"
        Vorgaenger$ = Teil$(Vergleich$, Variable$)

        IF Name$ = Teil$(Vorgaenger$, "Vorgänger.Name ") THEN

            ZP1% = Ereignis%(FuerNutzer$(Teil$(Vorgaenger$, "Vorgänger.ZP1 ")))
            ZP2% = Ereignis%(FuerNutzer$(Teil$(Vorgaenger$, "Vorgänger.ZP2 ")))

            Variable$ = "Daten.Datum[ " + LTRIM$(STR$(ZP2%)) + "]"
            Datum$ = Teil$(Vergleich$, Variable$)

            IF FuerNutzer$(Teil$(Vorgaenger$, "Vorgänger.Direkt ")) = "N" THEN
                Aendern Datum$, "Zeitangabe.Minimum", SPACE$(SIZEOF%("Termin"))
                StatusNeu Datum$
            END IF

            w% = z% + 6
            Schreiben Bezug%(ZP1%), BL&(ZP1%), Datum$, ZP1%, w%, ZP2%, Ort&, Num&
        END IF
        Ort& = Suchen&(Name$, Ort& + 1, z% + 1)
    WEND
NEXT z%

'Bezugsnamen berücksichtigen:
'-----'
FOR z% = 1 TO 2

    Ort& = Suchen&(Name$, 1, z% + 3)

    WHILE (Ort& <> 0)
        Erg& = Einlesen&(Ort&, Vergleich$)

```

```

LOCATE y%, x%
PRINT "Rückwärtsbezüge zu Eintrag "; Teil$(Vergleich$, "Daten.Name");
PRINT " ( "; LTRIM$(STR$(Ort&)); "/" ; LTRIM$(STR$(GEL&)); " ) werden ";
PRINT "erstellt."

Variable$ = "Daten.B[ " + LTRIM$(STR$(z%)) + "]"
Bezug$ = Teil$(Vergleich$, Variable$)

IF Name$ = Teil$(Bezug$, "Bezug.Name") THEN

    ZP1% = Ereignis%(FuerNutzer$(Teil$(Bezug$, "Bezug.ZP1")))

    ZP2% = Ereignis%(FuerNutzer$(Teil$(Bezug$, "Bezug.ZP2")))

    Datum$ = Teil$(Bezug$, "Bezug.Datum")

    Tol$ = Teil$(Bezug$, "Bezug.Tol1")

    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(ZP2%)) + "]"
    Summand$ = Teil$(Vergleich$, Variable$)

    Sum$ = Summe$(Datum$, Tol$, Summand$, "-")

    w% = z% + 8
    Schreiben Bezug%(ZP1%), BL&(ZP1%), Sum$, ZP1%, w%, ZP2%, Ort&, Num&
END IF
Ort& = Suchen&(Name$, Ort& + 1, z% + 3)
WEND
NEXT z%
END SUB 'BezeugeRueckwaerts _____'

'=====
SUB BezuegeVorwaerts (Num&)
'=====
' Erstellt die Vorwärtsbezüge zur Datensatznummer 'Num&' der Datei
' '#Eingabe%' zu allen anderen Datensätzen.
'
' Der Datensatz ist so aufgebaut, daß zuerst die relativen, dann die
' absoluten Datierungen auftreten.
' "Relative Datierungen" lassen lediglich eine Festlegung der Reihenfolge
' zu, "absolute Datierungen" ermöglichen auch die Bestimmung eines Datums.
'
' Bearbeitung:
' 16. 6.2001 - 30. 1.2003:    Norbert Südland
' 25. 3.2009 - 8.10.2016:    Norbert Südland
'-----
DIM Fehler$ 'AS STRING
DIM Daten$
DIM Text$
DIM Variable$
DIM Vorgaenger$
DIM Vergleich$
DIM Datum$
DIM Bezug$
DIM Tol$
DIM Summand$
DIM Sum$
DIM Dauer$

```

```

DIM Richtung$
DIM ZP1%           'Zeitpunkt bei Vorgänger
DIM ZP2%           'Zeitpunkt bei Name/Ereignis
DIM w%
DIM z%
DIM Zuerst&        'Datensatz-Nummer des zugehörigen Ersteintrags
DIM Ort&
DIM Erg&

'Datensatz mit allen Datierungen und Gleichzeitig-Bezügen einlesen:
'-----'
Zuerst& = Einlesen&(Num&, Daten$)
Text$ = "Vorwärtsbezüge für Eintrag " + Teil$(Daten$, "Daten.Name") + " ( "
Text$ = Text$ + LTRIM$(STR$(Num&)) + "/" + LTRIM$(STR$(GEL&))
Text$ = Text$ + ") werden erstellt."
Protokollmeldung Text$, 1

'Vorgänger berücksichtigen:
'-----'
FOR z% = 1 TO 2
  Variable$ = "Daten.V[ " + LTRIM$(STR$(z%)) + "]"

  Vorgaenger$ = Teil$(Daten$, Variable$)

  ZP2% = Ereignis$(FuerNutzer$(Teil$(Vorgaenger$, "Vorgänger.ZP2")))

  IF ZP2% <> 0 THEN
    ZP1% = Ereignis$(FuerNutzer$(Teil$(Vorgaenger$, "Vorgänger.ZP1")))
    Ort& = Suchen&(Teil$(Vorgaenger$, "Vorgänger.Name"), 1, 1)
    IF Ort& = 0 THEN
      Pause
      Text$ = "Angegebener Vorgänger "
      Text$ = Text$ + Teil$(Vorgaenger$, "Vorgänger.Name") + "fehlt "
      Protokollmeldung Text$, 0
      ERROR 108
    ELSE
      Erg& = Einlesen&(Ort&, Vergleich$)
    END IF

    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(ZP1%)) + "]"
    Datum$ = Teil$(Vergleich$, Variable$)
    IF FuerNutzer$(Teil$(Vorgaenger$, "Vorgänger.Direkt")) = "N" THEN
      Aendern Datum$, "Zeitangabe.Maximum", SPACE$(SIZEOF$( "Termin"))
      StatusNeu Datum$
    END IF

    Schreiben Bezug$(ZP2%), BL&(ZP2%), Datum$, ZP2%, z%, ZP1%, Ort&, Num&
  END IF
NEXT z%

'Bezugsnamen berücksichtigen:
'-----'
FOR z% = 1 TO 2
  Variable$ = "Daten.B[ " + LTRIM$(STR$(z%)) + "]"
  Bezug$ = Teil$(Daten$, Variable$)
  ZP2% = Ereignis$(FuerNutzer$(Teil$(Bezug$, "Bezug.ZP2")))
  IF ZP2% <> 0 THEN

```

```

ZP1% = Ereignis%(FuerNutzer$(Teil$(Bezug$, "Bezug.ZP1")))
IF ZP1% <> 0 THEN
    Ort& = Suchen&(Teil$(Bezug$, "Bezug.Name"), 1, 1)
    IF Ort& = 0 THEN
        Fehler$ = Teil$(Bezug$, "Bezug.Name")
        PRINT "Angegebener Bezugsname "; Teil$(Bezug$, "Bezug.Name");
        PRINT " fehlt";
        ERROR 109
    ELSE
        Erg& = Einlesen&(Ort&, Vergleich$)
    END IF
    Datum$ = Teil$(Bezug$, "Bezug.Datum")
    Tol$ = Teil$(Bezug$, "Bezug.Tol1")
    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(ZP1%)) + "]"
    Summand$ = Teil$(Vergleich$, Variable$)
    Sum$ = Summe$(Datum$, Tol$, Summand$, "+")
    w% = z% + 2
    Schreiben Bezug%(ZP2%), BL&(ZP2%), Sum$, ZP2%, w%, ZP1%, Ort&, Num&
END IF

'Dauer:
'-----'
Dauer$ = Teil$(Bezug$, "Bezug.Dauer")
IF Dauer$ <> SPACE$(SIZEOF$( "Zeitpunkt ")) THEN
    Tol$ = Teil$(Bezug$, "Bezug.Tol2")
    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(ZP2%)) + "]"
    Summand$ = Teil$(Daten$, Variable$)
    IF ZP2% > 3 THEN
        Richtung$ = "- "
    ELSE
        Richtung$ = "+ "
    END IF
    Sum$ = Summe$(Dauer$, Tol$, Summand$, Richtung$)

'Hier ist das Ziel `ZP1%` eng mit der Quelle `ZP2%` gekoppelt:
'-----'
ZP1% = 7 - ZP2%
w% = z% + 4
Schreiben Bezug%(ZP1%), BL&(ZP1%), Sum$, ZP1%, w%, ZP2%, Num&, Num&
'-----'
'Gleichzeitige Ereignisse sind hier noch nicht berücksichtigt!
'-----'

'Nun werden Quelle und Ziel vertauscht:
'-----'
Variable$ = "Daten.Datum[ " + LTRIM$(STR$(ZP1%)) + "]"
Summand$ = Teil$(Daten$, Variable$)
IF ZP2% > 3 THEN
    Richtung$ = "+ "
ELSE
    Richtung$ = "- "
END IF
Sum$ = Summe$(Dauer$, Tol$, Summand$, Richtung$)
w% = z% + 4
Schreiben Bezug%(ZP2%), BL&(ZP2%), Sum$, ZP2%, w%, ZP1%, Num&, Num&
END IF
END IF
NEXT z%

```

```

END SUB 'BezeugeVorwaerts _____'

'=====
SUB DatierungAbschreiben (Num&)
'=====
' Schreibt die schon existierende Datierung ab und liefert das erste
' Auftreten des Dateieintrags zurück, das zum selben Namen gehört.
'
' Bearbeitung:
' 4. 8.2001 - 22. 1.2003: Norbert Südland
'-----
DIM Daten$ 'AS STRING
DIM Feld$
DIM Variable$
DIM Datum$
DIM z% 'AS INTEGER

PRINT "Datierung abschreiben..."
Num& = Einlesen&(Num&, Daten$)
Feld$ = SPACE$(Feldlaenge%)
FOR z% = 1 TO 6

    'Zugehörigen Namen abspeichern:
    '-----
    GET #Bezug%(z%), 1, Feld$
    Aendern Feld$, "Feld.Datum", Namenseintrag$(Num&) + SPACE$(Feldlaenge%)
    PUT #Bezug%(z%), 1, Feld$

    'Zugehörige Datierung abspeichern:
    '-----
    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(z%)) + "]"
    Datum$ = Teil$(Daten$, Variable$)
    Schreiben Bezug%(z%), BL&(z%), Datum$, z%, 0, z%, Num&, Num&
NEXT z%
END SUB 'DatierungAbschreiben _____

'=====
SUB DatierungSpeichern (Position&, Daten$)
'=====
' Speichert `Daten$` als `Position&`.Eintrag in #`Eingabe%` und prüft bei
' den Datierungen, daß ein Markierungszeichen für Vordatierungen samt
' Inhalt nicht gelöscht wird.
' Der `Ersteintrag$` zu einem bestimmten Namen erhält alle Daterungen, bei
' den folgenden Einträge zu einem Namen werden die Dateirungen gelöscht.
'
' Bearbeitung:
' 4. 8.2001 - 30. 1.2003: Norbert Südland
'-----
DIM Original$ 'AS STRING
DIM Ersteintrag$
DIM Variable$
DIM Ausschnitt$
DIM z% 'AS INTEGER
DIM Erstposition& 'AS LONG

'Zu ändernde Datensätze laden:

```



```

'-----'
Original$ = Laden$(Eingabe%, Datenlaenge%, GEL&, Position&)
Erstposition& = Suchen&(Teil$(Original$, "Daten.Name"), 1, 1)

'Es ändern sich nur die Datierungen:
'-----'
IF Erstposition& = Position& THEN
  FOR z% = 1 TO 6
    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(z%)) + "]"
    SELECT CASE Teil$(Original$, Variable$ + ".Status")
    CASE VordatierungFruehestens$      ' " "
      Ausschnitt$ = Teil$(Daten$, Variable$ + ".Maximum")
      Aendern Original$, Variable$ + ".Maximum", Ausschnitt$
    CASE VordatierungKennzeichen$      ' " "
    CASE VordatierungSpaetestens$      ' " "
      Ausschnitt$ = Teil$(Original$, Variable$ + ".Minimum")
      Aendern Original$, Variable$ + ".Minimum", Ausschnitt$
    CASE ELSE
      Aendern Original$, Variable$, Teil$(Daten$, Variable$)
    END SELECT
    IF Widerspruch%(Teil$(Original$, Variable$)) = 1 THEN
      Pause
      ERROR 113
    END IF
  NEXT z%
ELSE
  Ersteintrag$ = Laden$(Eingabe%, Datenlaenge%, GEL&, Erstposition&)
  FOR z% = 1 TO 6
    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(z%)) + "]"

    'Datierungen beim `Ersteintrag$` abspeichern:
    '-----'
    SELECT CASE Teil$(Ersteintrag$, Variable$ + ".Status")
    CASE VordatierungFruehestens$      ' " "
      Ausschnitt$ = Teil$(Daten$, Variable$ + ".Maximum")
      Aendern Ersteintrag$, Variable$ + ".Maximum", Ausschnitt$
    CASE VordatierungKennzeichen$      ' " "
    CASE VordatierungSpaetestens$      ' " "
      Ausschnitt$ = Teil$(Original$, Variable$ + ".Minimum")
      Aendern Ersteintrag$, Variable$ + ".Minimum", Ausschnitt$
    CASE ELSE
      Aendern Ersteintrag$, Variable$, Teil$(Daten$, Variable$)
    END SELECT
    IF Widerspruch%(Teil$(Ersteintrag$, Variable$)) = 1 THEN
      Pause
      ERROR 113
    END IF

    'Datierungen beim nachfolgenden Eintrag weitestgehend löschen:
    '-----'
    SELECT CASE Teil$(Original$, Variable$ + ".Status")
    CASE VordatierungFruehestens$      ' " "
      Aendern Original$, Variable$ + ".Maximum", SPACE$(SIZEOF("Termin"))
    CASE VordatierungKennzeichen$      ' " "
    CASE VordatierungSpaetestens$      ' " "
      Aendern Original$, Variable$ + ".Minimum", SPACE$(SIZEOF("Termin"))
    CASE ELSE
      Aendern Original$, Variable$, SPACE$(SIZEOF("Zeitangabe"))

```

```

        END SELECT
    NEXT z%
    PUT #Eingabe%, Erstposition& * Datenlaenge% + 1, Ersteintrag$
END IF

PUT #Eingabe%, Position& * Datenlaenge% + 1, Original$
END SUB 'DatierungSpeichern _____'

'=====
FUNCTION Einlesen& (Ort&, Daten$)
'=====
' Lädt die Angaben vom 'Ort'.Dateieintrag und überschreibt mit den
' Datierungen des Ersteintrags, dessen Position zurückgegeben wird.
' Hier wird die Möglichkeit berücksichtigt, daß beliebig viele Einträge
' für einen Namen vorkommen können. Zuerst werden die gleichzeitigen
' Ereignisse dieses Namens ermittelt, dann damit die bereits vorhandenen
' Datierungen zusammengesetzt. Spätestens hier werden Widersprüche von
' verzwickten strukturierten Vordatierungen erkannt.
'
' Bearbeitung:
' 16. 6.2001 - 24. 1.2003      Norbert Südland
' 25. 3.2009 - 28. 9.2016      Norbert Südland
'-----
DIM Name$                'AS STRING
DIM Variable$
DIM Gleichzeitig$
DIM Zusatz$
DIM Puffer$
DIM Datum$
DIM Var$
DIM DatumNeu$
DIM Termin$
DIM TerminNeu$
DIM z%                   'AS INTEGER
DIM Laenge%
DIM Zusammen%(3)
DIM S&                   'AS LONG
DIM Weiter&

'Gleichzeitige Ereignisse zusammenfassen:
'-----
Daten$ = Laden$(Eingabe%, Datenlaenge%, GEL&, Ort&)
Name$ = Teil$(Daten$, "Daten.Name")
S& = Suchen&(Name$, 1, 1)                'Ersteintrag suchen
Zusatz$ = Laden$(Eingabe%, Datenlaenge%, GEL&, S&)
Puffer$ = FuerNutzer$(Teil$(Zusatz$, "Daten.Gleichzeitig"))
Gleichzeitig$ = GleichzeitigTest$(Puffer$)
Weiter& = Suchen&(Name$, S& + 1, 1)        'weiterer Eintrag?
WHILE Weiter& > 0
    Zusatz$ = Laden$(Eingabe%, Datenlaenge%, GEL&, Weiter&)
    Puffer$ = FuerNutzer$(Teil$(Zusatz$, "Daten.Gleichzeitig"))
    Puffer$ = GleichzeitigTest$(Puffer$)
    SELECT CASE LEFT$(Gleichzeitig$, 3)
    CASE SPACE$(3)
        Gleichzeitig$ = Puffer$
    CASE "*AW", "XE+"                'Alle anderen Einträge müssen leer sein!
    CASE "*A ", "AW ", "WE ", "XE ", "X+ ", "E+ "

```

```

IF RIGHT$(Gleichzeitig$, 2) = SPACE$(2) THEN
    Gleichzeitig$ = LEFT$(Puffer$, 3) + LEFT$(Gleichzeitig$, 2)
    Gleichzeitig$ = GleichzeitigTest$(Gleichzeitig$)
    IF RIGHT$(Gleichzeitig$, 2) = SPACE$(2) THEN
        Gleichzeitig$ = LEFT$(Gleichzeitig$, 3) + RIGHT$(Puffer$, 2)
        Gleichzeitig$ = GleichzeitigTest$(Gleichzeitig$)
    END IF
END IF
END SELECT

'Mit Vordatierung hart überschreiben:
'-----'
FOR z% = 1 TO 6
    Variable$ = "Daten.Datum[ " + LTRIM$(STR$(z%)) + "]"
    Datum$ = Teil$(Zusatz$, Variable$)
    Puffer$ = Teil$(Daten$, Variable$)
    SELECT CASE Teil$(Datum$, "Zeitangabe.Status")
    CASE VordatierungFruehestens$      ' " "
        Var$ = "Zeitangabe.Minimum"
        SELECT CASE Teil$(Puffer$, "Zeitangabe.Status")
        CASE VordatierungFruehestens$, VordatierungKennzeichen$
            IF Teil$(Puffer$, Var$) <> Teil$(Datum$, Var$) THEN
                Pause
                ERROR 114
            END IF
        CASE VordatierungSpaetestens$
            Aendern Daten$, Variable$ + ".Minimum", Teil$(Datum$, Var$)
            Aendern Daten$, Variable$ + ".Status", VordatierungKennzeichen$
        CASE ELSE
            Aendern Daten$, Variable$ + ".Minimum", Teil$(Datum$, Var$)
            Aendern Daten$, Variable$ + ".Status", VordatierungFruehestens$
        END SELECT
    CASE VordatierungKennzeichen$      ' " "
        SELECT CASE Teil$(Puffer$, "Zeitangabe.Status")
        CASE VordatierungFruehestens$
            Var$ = "Zeitangabe.Minimum"
            IF Teil$(Puffer$, Var$) <> Teil$(Datum$, Var$) THEN
                Pause
                ERROR 114
            END IF
            Var$ = "Zeitangabe.Maximum"
            Aendern Daten$, Variable$ + ".Maximum", Teil$(Datum$, Var$)
            Aendern Daten$, Variable$ + ".Status", VordatierungKennzeichen$
        CASE VordatierungKennzeichen$
            Var$ = "Zeitangabe.Minimum"
            IF Teil$(Puffer$, Var$) <> Teil$(Datum$, Var$) THEN
                Pause
                ERROR 114
            END IF
            Var$ = "Zeitangabe.Maximum"
            IF Teil$(Puffer$, Var$) <> Teil$(Datum$, Var$) THEN
                Pause
                ERROR 114
            END IF
        CASE VordatierungSpaetestens$
            Var$ = "Zeitangabe.Maximum"
            IF Teil$(Puffer$, Var$) <> Teil$(Datum$, Var$) THEN
                Pause

```

```

        ERROR 114
    END IF
    Var$ = "Zeitangabe.Minimum"
    Aendern Daten$, Variable$ + ".Minimum", Teil$(Datum$, Var$)
    Aendern Daten$, Variable$ + ".Status", VordatierungKennzeichen$
CASE ELSE
    Aendern Daten$, Variable$, Datum$
END SELECT
CASE VordatierungSpaetestens$      ' " | "'
    Var$ = "Zeitangabe.Maximum"
    SELECT CASE Teil$(Puffer$, "Zeitangabe.Status")
CASE VordatierungFruehestens$
    Aendern Daten$, Variable$ + ".Maximum", Teil$(Datum$, Var$)
    Aendern Daten$, Variable$ + ".Status", VordatierungKennzeichen$
CASE VordatierungKennzeichen$, VordatierungSpaetestens$
    IF Teil$(Puffer$, Var$) <> Teil$(Datum$, Var$) THEN
        Pause
        ERROR 114
    END IF
CASE ELSE
    Aendern Daten$, Variable$ + ".Maximum", Teil$(Datum$, Var$)
    Aendern Daten$, Variable$ + ".Status", VordatierungSpaetestens$
END SELECT
END SELECT
NEXT z%
Weiter& = Suchen&(Name$, Weiter& + 1, 1)
WEND
Aendern Daten$, "Daten.Gleichzeitig", FuerDatei$(Gleichzeitig$)

'Datum aus Ersteintrag übernehmen:
'-----'
IF Ort& <> S& THEN
    Zusatz$ = Laden$(Eingabe%, Datenlaenge%, GEL&, S&)
    FOR z% = 1 TO 6
        Variable$ = "Daten.Datum[" + LTRIM$(STR$(z%)) + "]"
        Datum$ = Teil$(Zusatz$, Variable$)
        SELECT CASE Teil$(Daten$, Variable$ + ".Status")
CASE VordatierungFruehestens$      ' " | "'
        Var$ = Variable$ + ".Minimum"
        Aendern Datum$, "Zeitangabe.Minimum", Teil$(Daten$, Var$)
        Aendern Datum$, "Zeitangabe.Status", "VordatierungFruehestens$"
CASE VordatierungKennzeichen$      ' " | "'
        Datum$ = Teil$(Daten$, Variable$)
CASE VordatierungSpaetestens$      ' " | "'
        Var$ = Variable$ + ".Maximum"
        Aendern Datum$, "Zeitangabe.Maximum", Teil$(Daten$, Var$)
        Aendern Datum$, "Zeitangabe.Status", "VordatierungSpaetestens$"
CASE ELSE
        StatusNeu Datum$
END SELECT
        Aendern Daten$, Variable$, Datum$
    NEXT z%
END IF

'Die Angaben gleichzeitiger Ereignisse müssen jeweils aufgelistet werden:
'-----'
IF LEFT$(Gleichzeitig$, 3) <> SPACE$(3) THEN
    Laenge% = SIZEOF$("Termin")

```

```
Datum$ = SPACE$(SIZEOF%("Zeitangabe"))
DatumNeu$ = Datum$
FOR z% = 1 TO 3
    Zusammen%(z%) = Ereignis%(MID$(Gleichzeitig$, z%, 1))
    IF Zusammen%(z%) > 0 THEN
        Variable$ = "Daten.Datum[" + LTRIM$(STR$(Zusammen%(z%))) + "]"
        Puffer$ = Teil$(Daten$, Variable$)
        Var$ = "Zeitangabe.Minimum"
        IF Teil$(Puffer$, Var$) <> SPACE$(Laenge%) THEN
            Aendern Datum$, Var$, Teil$(Puffer$, Var$)
        END IF
        Var$ = "Zeitangabe.Maximum"
        IF Teil$(Puffer$, Var$) <> SPACE$(Laenge%) THEN
            Aendern Datum$, Var$, Teil$(Puffer$, Var$)
        END IF
        StatusNeu Datum$
    END IF
    DatumNeu$ = Schnittmenge$(DatumNeu$, Datum$)
NEXT z%
FOR z% = 1 TO 3
    IF Zusammen%(z%) > 0 THEN
        Variable$ = "Daten.Datum[" + LTRIM$(STR$(Zusammen%(z%))) + "]"
        SELECT CASE Teil$(Daten$, Variable$ + ".Status")
        CASE VordatierungFruehestens$
            Var$ = Variable$ + ".Minimum"
            Termin$ = Teil$(Daten$, Var$)
            TerminNeu$ = Teil$(DatumNeu$, "Zeitangabe.Minimum")
            IF Termin$ <> TerminNeu$ THEN
                Pause
                ERROR 114
            END IF
            Var$ = Variable$ + ".Maximum"
            Aendern Daten$, Var$, Teil$(DatumNeu$, "Zeitangabe.Maximum")
        CASE VordatierungKennzeichen$
            Var$ = Variable$ + ".Minimum"
            Termin$ = Teil$(Daten$, Var$)
            TerminNeu$ = Teil$(DatumNeu$, "Zeitangabe.Minimum")
            IF Termin$ <> TerminNeu$ THEN
                Pause
                ERROR 114
            END IF
            Var$ = Variable$ + ".Maximum"
            Termin$ = Teil$(Daten$, Var$)
            TerminNeu$ = Teil$(DatumNeu$, "Zeitangabe.Maximum")
            IF Termin$ <> TerminNeu$ THEN
                Pause
                ERROR 114
            END IF
        CASE VordatierungSpaetestens$
            Var$ = Variable$ + ".Minimum"
            Termin$ = Teil$(Daten$, Var$)
            TerminNeu$ = Teil$(DatumNeu$, "Zeitangabe.Minimum")
            IF Termin$ <> TerminNeu$ THEN
                Pause
                ERROR 114
            END IF
            Var$ = Variable$ + ".Maximum"
            Aendern Daten$, Var$, Teil$(DatumNeu$, "Zeitangabe.Minimum")
        END SELECT
    END IF
END FOR
```

```

        CASE ELSE
            Aendern Daten$, Variable$, DatumNeu$
        END SELECT
    END IF
NEXT z%
END IF

IF RIGHT$(Gleichzeitig$, 2) <> SPACE$(2) THEN
    Datum$ = SPACE$(SIZEOF$( "Zeitangabe" ))
    DatumNeu$ = Datum$
    FOR z% = 1 TO 2
        Zusammen%(z%) = Ereignis$(MID$(Gleichzeitig$, z% + 3, 1))
        IF Zusammen%(z%) > 0 THEN
            Variable$ = "Daten.Datum[ " + LTRIM$(STR$(Zusammen%(z%))) + "]"
            Puffer$ = Teil$(Daten$, Variable$)
            Var$ = "Zeitangabe.Minimum"
            IF Teil$(Puffer$, Var$) <> SPACE$(Laenge%) THEN
                Aendern Datum$, Var$, Teil$(Puffer$, Var$)
            END IF
            Var$ = "Zeitangabe.Maximum"
            IF Teil$(Puffer$, Var$) <> SPACE$(Laenge%) THEN
                Aendern Datum$, Var$, Teil$(Puffer$, Var$)
            END IF
            StatusNeu Datum$
        END IF
        DatumNeu$ = Schnittmenge$(DatumNeu$, Datum$)
    NEXT z%
    FOR z% = 1 TO 2
        IF Zusammen%(z%) > 0 THEN
            Variable$ = "Daten.Datum[ " + LTRIM$(STR$(Zusammen%(z%))) + "]"
            SELECT CASE Teil$(Daten$, Variable$ + ".Status")
            CASE VordatierungFruehestens$
                Var$ = Variable$ + ".Minimum"
                Termin$ = Teil$(Daten$, Var$)
                TerminNeu$ = Teil$(DatumNeu$, "Zeitangabe.Minimum")
                IF Termin$ <> TerminNeu$ THEN
                    Pause
                    ERROR 114
                END IF
                Var$ = Variable$ + ".Maximum"
                Aendern Daten$, Var$, Teil$(DatumNeu$, "Zeitangabe.Maximum")
            CASE VordatierungKennzeichen$
                Var$ = Variable$ + ".Minimum"
                Termin$ = Teil$(Daten$, Var$)
                TerminNeu$ = Teil$(DatumNeu$, "Zeitangabe.Minimum")
                IF Termin$ <> TerminNeu$ THEN
                    Pause
                    ERROR 114
                END IF
                Var$ = Variable$ + ".Maximum"
                Termin$ = Teil$(Daten$, Var$)
                TerminNeu$ = Teil$(DatumNeu$, "Zeitangabe.Maximum")
                IF Termin$ <> TerminNeu$ THEN
                    Pause
                    ERROR 114
                END IF
            CASE VordatierungSpaetestens$
                Var$ = Variable$ + ".Minimum"

```

```

        Termin$ = Teil$(Daten$, Var$)
        TerminNeu$ = Teil$(DatumNeu$, "Zeitangabe.Minimum")
        IF Termin$ <> TerminNeu$ THEN
            Pause
            ERROR 114
        END IF
        Var$ = Variable$ + ".Maximum"
        Aendern Daten$, Var$, Teil$(DatumNeu$, "Zeitangabe.Minimum")
    CASE ELSE
        Aendern Daten$, Variable$, DatumNeu$
    END SELECT
END IF
NEXT z%
END IF

'Zusammengeführte Datierungen bei Ersteintrag abspeichern:
'-----'
IF Gleichzeitig$ <> SPACE$(5) THEN
    Zusatz$ = Laden$(Eingabe%, Datenlaenge%, GEL&, S&)
    FOR z% = 1 TO 6
        Variable$ = "Daten.Datum[" + LTRIM$(STR$(z%)) + "]"
        DatumNeu$ = Teil$(Daten$, Variable$)
        Aendern Zusatz$, Variable$, DatumNeu$
    NEXT z%
    PUT #Eingabe%, S& * Datenlaenge% + 1, Zusatz$
END IF

'Nummer des Ersteintrags:
'-----'
Einlesen& = S&
END FUNCTION 'Einlesen& _____'

'=====
SUB Einsortieren (Name$, Position&, NamensTyp%)
'=====
' Baut einen Listen-Eintrag zusammen und erweitert die zum `NamensTyp%`
' zugehörige Namen-Datei um `Name$` und `Position&`.
' Die Namen-Datei wird alphanumerisch sortiert.
'
'
' Bearbeitung:
' 18.12.2002 - 5. 1.2003: Norbert Südland
' Überprüfung:
'
'-----'

DIM Liste$ 'AS STRING
DIM Puffer$
DIM L% 'AS INTEGER
DIM Start& 'AS LONG
DIM Mitte&
DIM Ende&
DIM Ort&

L% = SIZEOF("Liste")
IF LTRIM$(Name$) <> "" THEN

    'Neuen Eintrag erstellen:
    '-----'

```

```

Liste$ = SPACE$(L%)
Aendern Liste$, "Liste.Name", Name$
Aendern Liste$, "Liste.Nummer", STR$(Position&)
Aendern Liste$, "Liste.Ende", "|" + CHR$(10) + CHR$(13)

'Sortier-Position finden:
'-----'
Start& = 1
Mitte& = 1 'Default bei leerer Datei.
Ende& = NL&(NamensTyp%)
WHILE Start& <= Ende&
    Mitte& = CLNG((Start& + Ende&) / 2)
    Puffer$ = Laden$(Namen%(NamensTyp%), L%, NL&(NamensTyp%), Mitte&)
    SELECT CASE Liste$
    CASE IS = Puffer$
        Pause 'Noch nicht aufgetreten!
        ERROR 120 'Doppelter Namens-Eintrag: Programmierfehler?
    CASE IS < Puffer$
        Ende& = Mitte& - 1
    CASE IS > Puffer$
        Start& = Mitte& + 1
    END SELECT
WEND

'Neuen Eintrag einordnen:
'-----'
IF Start& > Mitte& THEN
    Mitte& = Start& 'Neuer Eintrag kommt hinter `Mitte&`
END IF
FOR Ort& = NL&(NamensTyp%) TO Mitte& STEP -1
    Puffer$ = Laden$(Namen%(NamensTyp%), L%, NL&(NamensTyp%), Ort&)
    PUT #Namen%(NamensTyp%), (Ort& + 1) * L% + 1, Puffer$
NEXT Ort&
PUT Namen%(NamensTyp%), Mitte& * L% + 1, Liste$
NL&(NamensTyp%) = NL&(NamensTyp%) + 1
END IF
END SUB 'Einsortieren _____'

'=====
FUNCTION Ereignis% (Symb$)
'=====
'Liefert die Nummer des zu `Symb$` gehörenden Ereignisses.
'
'Bearbeitung:
' 4. 9.2001 - 22. 1.2003: Norbert Südland
'-----'
DIM Erg% 'AS INTEGER

SELECT CASE RTRIM$(LTRIM$(Symb$))
CASE "*"
    Erg% = 1
CASE "A"
    Erg% = 2
CASE "W"
    Erg% = 3
CASE "X"
    Erg% = 4

```



```

CASE "E"
    Erg% = 5
CASE "+"
    Erg% = 6
CASE ELSE
    Erg% = 0
END SELECT

Ereignis% = Erg%
END FUNCTION 'Ereignis% _____'

'=====
SUB Ergebnis (Datei%, DL&, Vorgabe$, Vergleich$, Ziel%, Quelle%, Num&)
'=====
'
'  Bearbeitung:
'  4. 8.2001 - 4. 9.2001: Norbert Südland, München
'-----
DIM Erg$                'AS STRING
DIM Sum$
DIM Termin$
DIM Groesse%            'AS INTEGER
DIM v1%, j1%, m1%, t1%
DIM v2%, j2%, m2%, t2%

Groesse% = SIZEOF%("Zeitangabe")
Erg$ = SPACE$(Groesse%)
IF Teil$(Vergleich$, "Feld.Datum") <> SPACE$(Groesse%) THEN

    'Differenz bilden:
    '-----
    Sum$ = Teil$(Vorgabe$, "Feld.Datum")
    IF Sum$ = SPACE$(Groesse%) THEN
        Termin$ = ZahlZuTermin$(1, 1, 1, 1)          'Dummy bei 1.J 1.M 1.T
        Aendern Sum$, "Zeitangabe.Minimum", Termin$
        Aendern Sum$, "Zeitangabe.Maximum", Termin$
    END IF
    Addieren Sum$, "U", "-", Teil$(Vergleich$, "Feld.Datum"), 1, 1, "-"

    'Betrag der Differenz:
    '-----
    TerminZuZahl Teil$(Sum$, "Zeitangabe.Minimum"), v1%, j1%, m1%, t1%
    TerminZuZahl Teil$(Sum$, "Zeitangabe.Maximum"), v2%, j2%, m2%, t2%
    IF v1% = -1 THEN
        Umkehren v1%, j1%, m1%, t1%, v2%, j2%, m2%, t2%, "-"
    END IF
    Termin$ = ZahlZuTermin$(v1%, j1%, m1%, t1%)
    Aendern Erg$, "Zeitangabe.Minimum", Termin$
    Aendern Erg$, "Zeitangabe.Status", " "
    IF v2% = -1 THEN
        Umkehren v1%, j1%, m1%, t1%, v2%, j2%, m2%, t2%, "-"
    END IF
    Termin$ = ZahlZuTermin$(v2%, j2%, m2%, t2%)
    Aendern Erg$, "Zeitangabe.Maximum", Termin$

    'Ergebnis verdeutlichen:
    '-----

```

```

IF j1% > 0 THEN
    Aendern Erg$, "Zeitangabe.Minimum.JS ", "J"
END IF
IF m1% > 0 THEN
    Aendern Erg$, "Zeitangabe.Minimum.MS ", "M"
END IF
IF t1% > 0 THEN
    Aendern Erg$, "Zeitangabe.Minimum.TS ", "T"
END IF
IF j2% > 0 THEN
    Aendern Erg$, "Zeitangabe.Maximum.JS ", "J"
END IF
IF m2% > 0 THEN
    Aendern Erg$, "Zeitangabe.Maximum.MS ", "M"
END IF
IF t2% > 0 THEN
    Aendern Erg$, "Zeitangabe.Maximum.TS ", "T"
END IF
END IF

'Endergebnis abspeichern:
'-----'
Schreiben Datei%, DL&, Erg$, 0, 0, Quelle%, 0, 0
Erg$ = Teil$(Vergleich$, "Feld.Datum")
Schreiben Datei%, DL&, Erg$, Ziel%, 0, Quelle%, 0, Num&

END SUB 'Ergebnis _____'

'=====
FUNCTION FuerDatei$ (Symbol$)
'=====
' Wandelt 'Symbol$' in einen internationalen Kode für die Daten-Datei.
' Diese Funktion arbeitet auch mit alten Daten-Dateien korrekt.
'
' Bearbeitung:
' 20.11.2007 - 22.11.2007:    Norbert Südland, D-73431 Aalen
'
' Übersetzung:
' 25. 3.2009:                Norbert Südland, D-73431 Aalen
'-----'
DIM Puffer$
DIM Zaehler%

Puffer$ = SPACE$(LEN(Symbol$))
FOR Zaehler% = 1 TO LEN(Symbol$)
    SELECT CASE MID$(Symbol$, Zaehler%, 1)
    CASE "1", "J", "Y"      'Ja
        MID$(Puffer$, Zaehler%, 1) = "1"
    CASE "0", "N"          'Nein
        MID$(Puffer$, Zaehler%, 1) = "0"
    CASE "{", "*"          'Geburt
        MID$(Puffer$, Zaehler%, 1) = "{"
    CASE "[", "A", "B"     '1.Wirkungsanfang
        MID$(Puffer$, Zaehler%, 1) = "["
    CASE "(", "W", "α"     '2.Wirkungsanfang
        MID$(Puffer$, Zaehler%, 1) = "("
    CASE ")", "X", "Ω"     '2.Wirkungsende
        MID$(Puffer$, Zaehler%, 1) = ")"

```

```

CASE "]" , "E"                                '1.Wirkungsende
MID$(Puffer$, Zaehler%, 1) = "]"
CASE "}" , "+"                                'Lebensende
MID$(Puffer$, Zaehler%, 1) = "}"
CASE ELSE
MID$(Puffer$, Zaehler%, 1) = SPACE$(1)
END SELECT
NEXT Zaehler%

FuerDatei$ = Puffer$
END FUNCTION 'FuerDatei$ _____

'=====
FUNCTION FuerNutzer$ (Symbol$)
'=====
' Wandelt die internationale 'Symbol$'-Kodierung in deutschen Nutzer-Kode.
' Diese Funktion arbeitet auch mit allen alten Daten-Dateien korrekt.
'
' Bearbeitung:
' 20.11.2007 - 22.11.2007:      Norbert Südland, D-73431 Aalen
'
' Übersetzung:
' 14. 2.2009 - 25. 3.2009:      Norbert Südland, D-73431 Aalen
'-----
DIM Puffer$
DIM Zaehler%

Puffer$ = SPACE$(LEN(Symbol$))
FOR Zaehler% = 1 TO LEN(Symbol$)
SELECT CASE MID$(Symbol$, Zaehler%, 1)
CASE "1" , "J" , "Y"
MID$(Puffer$, Zaehler%, 1) = "J" 'Ja
CASE "0" , "N"
MID$(Puffer$, Zaehler%, 1) = "N" 'Nein
CASE "{" , "*"
MID$(Puffer$, Zaehler%, 1) = "*" 'Geburt
CASE "[" , "A" , "B"
MID$(Puffer$, Zaehler%, 1) = "A" '1.Wirkungsanfang
CASE "(" , "W" , "α"
MID$(Puffer$, Zaehler%, 1) = "W" '2.Wirkungsanfang
CASE ")" , "X" , "Q"
MID$(Puffer$, Zaehler%, 1) = "X" '2.Wirkungsende
CASE "]" , "E"
MID$(Puffer$, Zaehler%, 1) = "E" '1.Wirkungsende
CASE "}" , "+"
MID$(Puffer$, Zaehler%, 1) = "+" 'Lebensende
CASE ELSE
MID$(Puffer$, Zaehler%, 1) = SPACE$(1)
END SELECT
NEXT Zaehler%

FuerNutzer$ = Puffer$
END FUNCTION 'FuerNutzer$ _____

'=====
FUNCTION GleichzeitigTest$ (Text$)
'=====
' `Text$` muß 5 Zeichen lang sein!

```

```

'
' Bearbeitung:
' 4. 9.2001 - 24.12.2002: Norbert Südland
'-----'
DIM a$(2) 'AS STRING
DIM leer$
DIM z% 'AS INTEGER
DIM y%
DIM x%(2)
DIM E%

FOR z% = 1 TO 2
  a$(z%) = SPACE$(6)
  x%(z%) = 0
  FOR y% = 1 + (z% - 1) * 3 TO 3 + (z% - 1) * 2
    E% = Ereignis$(MID$(Text$, y%, 1))
    IF E% <> 0 THEN
      IF MID$(a$(z%), E%, 1) = " " THEN
        MID$(a$(z%), E%, 1) = MID$(Text$, y%, 1)
        x%(z%) = x%(z%) + 1
      END IF
    END IF
  NEXT y%

  'Wurde nur ein Zeichen eingegeben?
  '-----
  IF x%(z%) < 2 THEN
    a$(z%) = SPACE$(6)
    x%(z%) = 0
  END IF
NEXT z%

'Reduzierung auf passende Größe:
'-----
FOR z% = 1 TO 2
  a$(z%) = RTRIM$(LTRIM$(a$(z%)))
  IF LEN(a$(z%)) > 0 THEN
    IF MID$(a$(z%), 2, 1) = " " THEN 'mind. ein Zeichen weiter rechts!
      a$(z%) = LEFT$(a$(z%), 1) + LTRIM$(RIGHT$(a$(z%), LEN(a$(z%)) - 1))
    END IF
    IF LEN(a$(z%)) > 2 THEN
      IF MID$(a$(z%), 3, 1) = " " THEN
        a$(z%) = LEFT$(a$(z%), 2) + LTRIM$(RIGHT$(a$(z%), LEN(a$(z%)) - 2))
      END IF
    END IF
  END IF
  a$(z%) = a$(z%) + SPACE$((4 - z%) - LEN(a$(z%)))
NEXT z%

'Vergleich der Kombinationen:
'-----
'Es sind 8 Koppelungen von Ereignissen prinzipiell möglich:
'"*AW", "XE+", "*A", "AW", "WE", "XE", "X+", "E+".
'Folgende Paare der Kombination sind zulässig:
'
'      |   *A   AW   WE   XE   X+   E+
'-----+-----

```

'	*A		O	O	X	X	X	-
'	AW		O	O	-	-	X	-
'	WE		X	-	O	-	X	-
'	XE		X	-	-	O	O	O
'	X+		X	X	X	O	O	O
'	E+		-	-	-	O	O	O

'Die eingetragenen Symbole bedeuten:

' O Möglichkeit der Zusammenfassung zu etwas Sinnvollem
 ' X zulässige Kombination
 ' - unzulässige Kombination oder Zusammenfassung

```

leer$ = SPACE$(2)
SELECT CASE a$(1)
CASE "*AW", "XE+"      'Umstellungen der Reihenfolge sind eher verwirrend!
  a$(2) = leer$
CASE "*A "
  SELECT CASE a$(2)
  CASE "AW"
    a$(1) = "*AW": a$(2) = leer$
  CASE "WE", "XE", "X+"
  CASE ELSE
    a$(2) = leer$
  END SELECT
CASE "AW "
  SELECT CASE a$(2)
  CASE "*A "
    a$(1) = "*AW": a$(2) = leer$
  CASE "X+"
  CASE ELSE
    a$(2) = leer$
  END SELECT
CASE "WE "
  SELECT CASE a$(2)
  CASE "*A "
    a$(1) = "*A ": a$(2) = "WE"
  CASE "X+"
  CASE ELSE
    a$(2) = leer$
  END SELECT
CASE "XE "
  SELECT CASE a$(2)
  CASE "*A "
    a$(1) = "*A ": a$(2) = "XE"
  CASE "X+", "E+"
    a$(1) = "XE+": a$(2) = leer$
  CASE ELSE
    a$(2) = leer$
  END SELECT
CASE "X+ "
  SELECT CASE a$(2)
  CASE "*A "
    a$(1) = "*A ": a$(2) = "X+"
  CASE "AW"
    a$(1) = "AW ": a$(2) = "X+"
  CASE "WE"
    a$(1) = "WE ": a$(2) = "X+"
  CASE "XE", "E+"

```

```

        a$(1) = "XE+": a$(2) = leer$
    CASE ELSE
        a$(2) = leer$
    END SELECT
CASE "E+ "
    SELECT CASE a$(2)
    CASE "XE", "X+"
        a$(1) = "XE+": a$(2) = leer$
    CASE ELSE
        a$(2) = leer$
    END SELECT
CASE ELSE
    SELECT CASE a$(2)
    CASE "*A", "AW", "WE", "XE", "X+", "E+"
        a$(1) = a$(2) + " ": a$(2) = leer$
    CASE ELSE
        a$(2) = leer$: a$(1) = a$(2) + " "
    END SELECT
END SELECT

'Ergebnis:
'-----'
GleichzeitigTest$ = a$(1) + a$(2)
END FUNCTION 'GleichzeitigTest$ _____'

'=====
SUB Korrektur (Feld$, Rest%, RL&)
'=====
' Korrigiert die Datierungen wo nötig, wenn ein neues Datum ermittelt wurde.
'
' Bearbeitung:
' 4. 8.2001 - 4. 9.2001:    Norbert Südland, München
' 15. 1.2003 - 24. 1.2003:    Norbert Südland, Aalen
' 25. 3.2009 - 8.10.2016:    Norbert Südland, Aalen
'-----'

DIM Daten$      'AS STRING
DIM Var$
DIM Variable$
DIM Ziel$
DIM Erg$
DIM Vergleich$
DIM Quelle$
DIM Tol$
DIM Richtung$
DIM Status$
DIM Diff$
DIM Drest$
DIM Text$
DIM Ort&        'AS INTEGER
DIM v1%, j1%, m1%, t1%
DIM v2%, j2%, m2%, t2%
DIM Fall%
DIM Korrekt%
DIM L%
DIM Dummy&      'AS LONG

```

```
Dummy& = Einlesen&(VAL(Teil$(Feld$, "Feld.nach")), Daten$)
```

```

Variable$ = LTRIM$(STR$(Ereignis%(Teil$(Feld$, "Feld.Ziel"))))
Variable$ = "Daten.Datum[" + Variable$ + "]"
Ziel$ = Teil$(Daten$, Variable$)
Erg$ = Ziel$
Ort& = VAL(Teil$(Feld$, "Feld.von"))
Dummy& = Einlesen&(Ort&, Vergleich$)
Variable$ = LTRIM$(STR$(Ereignis%(Teil$(Feld$, "Feld.Quelle"))))
Variable$ = "Daten.Datum[" + Variable$ + "]"
Quelle$ = Teil$(Vergleich$, Variable$)
Fall% = VAL(Teil$(Feld$, "Feld.Name"))

SELECT CASE Fall%
CASE 0          'Übertrag durch gleichzeitige Ereignisse
  Erg$ = Teil$(Feld$, "Feld.Datum")
CASE 1, 2
  Variable$ = "Daten.V[" + LTRIM$(STR$(Fall%)) + "].Direkt "
  IF FuerNutzer$(Teil$(Daten$, Variable$)) = "N" THEN
    Aendern Erg$, "Zeitangabe.Minimum", SPACE$(SIZEOF$("Termin"))
    StatusNeu Erg$
  END IF
CASE 3, 4
  Variable$ = "Daten.B[" + LTRIM$(STR$(Fall% - 2)) + "].Tol1 "
  Tol$ = Teil$(Daten$, Variable$)
  Variable$ = "Daten.B[" + LTRIM$(STR$(Fall% - 2)) + "].Datum "
  Erg$ = Summe$(Teil$(Daten$, Variable$), Tol$, Ziel$, "-")
CASE 5, 6
  Variable$ = "Daten.B[" + LTRIM$(STR$(Fall% - 4)) + "].Tol2 "
  Tol$ = Teil$(Daten$, Variable$)
  IF Ereignis%(Teil$(Feld$, "Feld.Ziel")) <= 3 THEN
    Richtung$ = "+"
  ELSE
    Richtung$ = "-"
  END IF
  Variable$ = "Daten.B[" + LTRIM$(STR$(Fall% - 4)) + "].Dauer "
  Erg$ = Summe$(Teil$(Daten$, Variable$), Tol$, Ziel$, Richtung$)
CASE 7, 8
  Variable$ = "Daten.V[" + LTRIM$(STR$(Fall% - 6)) + "].Direkt "
  IF FuerNutzer$(Teil$(Vergleich$, Variable$)) = "N" THEN
    Aendern Erg$, "Zeitangabe.Maximum", SPACE$(SIZEOF$("Termin"))
    StatusNeu Erg$
  END IF
CASE 9, 10
  Variable$ = "Daten.B[" + LTRIM$(STR$(Fall% - 8)) + "].Tol1 "
  Tol$ = Teil$(Vergleich$, Variable$)
  Variable$ = "Daten.B[" + LTRIM$(STR$(Fall% - 8)) + "].Datum "
  Erg$ = Summe$(Teil$(Vergleich$, Variable$), Tol$, Ziel$, "+")
END SELECT

'Prüfen, ob 'Erg$' den Wert von 'Quelle' verändert:
'-----'
Korrekt% = 0
L% = SIZEOF$("Zeitangabe")
IF Erg$ <> SPACE$(L%) THEN
  IF Quelle$ = SPACE$(L%) THEN
    Korrekt% = 1          'noch kein Datum angegeben
  ELSE
    Status$ = Teil$(Quelle$, "Zeitangabe.Status")
    IF Status$ = "≤" THEN

```

```

    Var$ = "Zeitangabe.Minimum"
    Aendern Quelle$, Var$, Teil$(Erg$, Var$)
    StatusNeu Quelle$
    IF Teil$(Quelle$, "Zeitangabe.Status") = " " THEN
        Korrekt% = 1
    END IF
END IF
IF Status$ = "≥" THEN
    Var$ = "Zeitangabe.Maximum"
    Aendern Quelle$, Var$, Teil$(Erg$, Var$)
    StatusNeu Quelle$
    IF Teil$(Quelle$, "Zeitangabe.Status") = " " THEN
        Korrekt% = 1
    END IF
END IF
Diff$ = Erg$
Addieren Diff$, "U", "-", Quelle$, 1, 1, "-"
TerminZuZahl Teil$(Diff$, "Zeitangabe.Minimum"), v1%, j1%, m1%, t1%
TerminZuZahl Teil$(Diff$, "Zeitangabe.Maximum"), v2%, j2%, m2%, t2%
L% = SIZEOF("Termin")
IF v2% = -1 AND Teil$(Erg$, "Zeitangabe.Maximum") <> SPACE$(L%) THEN
    Korrekt% = 1
    Var$ = "Zeitangabe.Maximum"
    Aendern Quelle$, Var$, Teil$(Erg$, Var$)
END IF
Umkehren v1%, j1%, m1%, t1%, v2%, j2%, m2%, t2%, "-"
IF v1% = -1 AND Teil$(Erg$, "Zeitangabe.Minimum") <> SPACE$(L%) THEN
    Korrekt% = 1
    Var$ = "Zeitangabe.Minimum"
    Aendern Quelle$, Var$, Teil$(Erg$, Var$)
END IF
IF Korrekt% = 1 THEN
    Erg$ = Quelle$
    SELECT CASE Teil$(Erg$, "Zeitangabe.Status")
    CASE VordatierungFruehestens$ ' "I"
        ERROR 114
    CASE VordatierungKennzeichen$ ' "II"
        ERROR 114
    CASE VordatierungSpaetestens$ ' "III"
        ERROR 114
    CASE ELSE
        StatusNeu Erg$
    END SELECT
END IF
END IF
END IF
IF Korrekt% = 1 THEN
    Protokollmeldung Erg$, 0
    IF Widerspruch%(Erg$) = 1 THEN
        Pause
        ERROR 113
    END IF
END IF
L% = SIZEOF("Datenrest")
Drest$ = SPACE$(L%)
RL% = RL% + 1
Aendern Drest$, "Datenrest.Eintrag", STR$(RL%)
Aendern Drest$, "Datenrest.Nummer", STR$(Ort%)
Aendern Drest$, "Datenrest.Z", Teil$(Feld$, "Feld.Quelle")

```



```

Aendern Drest$, "Datenrest.Name", Namenseintrag$(Ort&)
Aendern Drest$, "Datenrest.Datum", Erg$
Aendern Drest$, "Datenrest.Ende", "|" + CHR$(13) + CHR$(10)
PUT #Rest%, RL& * L% + 1, Drest$

Variable$ = LTRIM$(STR$(Ereignis%(Teil$(Feld$, "Feld.Quelle"))))
Variable$ = "Daten.Datum[" + Variable$ + "]"
Aendern Vergleich$, Variable$, Erg$
DatierungSpeichern Ort&, Vergleich$

Text$ = " " + STR$(RL&) + ".Eintrag auf " + Teil$(Drest$, "Datenrest.Z")
Text$ = Text$ + " " + Namenseintrag$(Ort&) + STR$(Ort&)
Protokollmeldung Text$, 1
END IF
END SUB 'Korrektur _____'

'=====
FUNCTION Laden$ (Datei%, DatensatzLaenge%, DateiLaenge%, Position&)
'=====
' Liest den `Position&-ten Eintrag einer Binär-`Datei%, die
' `DateiLaenge&` Einträge der Zeichenlänge `DatensatzLaenge%` besitzt.
' Der 0-te Eintrag der Datei wird zur Dokumentation verwendet.
'
' Bearbeitung:
' 4. 8.2001 - 20. 1.2003: Norbert Südland
'-----
DIM Feld$ 'AS STRING
DIM L% 'AS INTEGER

Feld$ = SPACE$(DatensatzLaenge%)
IF Position& = 0 THEN
    Position& = DateiLaenge&
END IF
GET #Datei%, Position& * DatensatzLaenge% + 1, Feld$

Laden$ = Feld$
END FUNCTION 'Laden$ _____'

'=====
FUNCTION Namenseintrag$ (Position&)
'=====
' Gibt den `Position&-ten Namens-Eintrag von #`Eingabe%` zurück.
'
' Bearbeitung:
' 23.12.2002 - 20. 1.2003: Norbert Südland
'-----
DIM Daten$ 'AS STRING

Daten$ = Laden$(Eingabe%, Datenlaenge%, GEL&, Position&)
Namenseintrag$ = Teil$(Daten$, "Daten.Name")
END FUNCTION 'Namenseintrag$ _____'

'=====
SUB Optimieren
'=====
' Bildet die Schnittmenge aus den bislang vorhandenen Datierungen.

```

```

'
' Bearbeitung:
' 27. 6.2001 - 20. 1.2003:    Norbert Südland
' 25. 3.2009:                Norbert Südland, Aalen
' -----
DIM Feld$      'AS STRING
DIM Daten$
DIM Puffer$
DIM Vorgabe$
DIM Vergleich1$
DIM Vergleich2$
DIM Variable$
DIM Sum$
DIM Add$
DIM Status$
DIM Vgl$
DIM E%(5)      'AS INTEGER
DIM v1%, j1%, m1%, t1%
DIM v2%, j2%, m2%, t2%
DIM y%
DIM z%
DIM Groesse%
DIM L%
DIM Gefunden%
DIM Num&       'AS LONG
DIM Erg&
DIM Ort&
DIM Start&

' Gemeinsamkeit der Ereignisse berücksichtigen: '
' =====
Feld$ = Laden$(Bezug%(1), Feldlaenge%, BL&(1), 1)
Num& = VAL(Teil$(Feld$, "Feld.nach"))
Erg& = Einlesen&(Num&, Daten$)
Puffer$ = FuerNutzer$(Teil$(Daten$, "Daten.Gleichzeitig"))
Puffer$ = GleichzeitigTest$(Puffer$)
FOR z% = 1 TO 5
    E%(z%) = Ereignis%(MID$(Puffer$, z%, 1))
NEXT z%
IF Puffer$ <> SPACE$(SIZEOF("Daten.Gleichzeitig")) THEN
    FOR z% = 2 TO 3
        IF E%(z%) <> 0 THEN
            Verschieben Bezug%(), BL&(), E%(z%), E%(1)
        END IF
    NEXT z%
    IF E%(4) <> 0 THEN
        Verschieben Bezug%(), BL&(), E%(5), E%(4)
    END IF
END IF

' Neue Grenzen für Minimal- / Maximal-Datum: '
' =====
FOR z% = 1 TO 6
    Vorgabe$ = Laden$(Bezug%(z%), Feldlaenge%, BL&(z%), 1)
    SELECT CASE Teil$(Vorgabe$, "Feld.Datum.Status")
    CASE VordatierungFruehestens$
        Start& = 1

```

```

CASE VordatierungKennzeichen$      ' "■"
    Start& = 1
CASE VordatierungSpaetestens$      ' "■"
    Start& = 1
CASE ELSE
    Start& = 2
END SELECT
Groesse% = SIZEOF%("Termin")
IF BL&(z%) > 1 THEN
    Vergleich1$ = Laden$(Bezug%(z%), Feldlaenge%, BL&(z%), Start&)
    Vergleich2$ = Vergleich1$
    FOR y% = 1 TO 2      'Minimum und Maximum getrennt behandeln
        FOR Ort& = Start& + 1 TO BL&(z%)
            Vergleich2$ = Laden$(Bezug%(z%), Feldlaenge%, BL&(z%), Ort&)
            L% = SIZEOF%("Zeitangabe")
            IF Teil$(Vergleich1$, "Feld.Datum") = SPACE$(L%) THEN
                Vergleich1$ = Vergleich2$
            ELSE
                IF y% = 1 THEN
                    Variable$ = "Feld.Datum.Minimum"
                ELSE
                    Variable$ = "Feld.Datum.Maximum"
                END IF
                IF Teil$(Vergleich2$, Variable$) = SPACE$(Groesse%) THEN
                    Gefunden% = 0
                ELSE
                    Gefunden% = 1
                END IF
                IF Gefunden% = 1 THEN
                    Sum$ = Teil$(Vergleich1$, "Feld.Datum")
                    Add$ = Teil$(Vergleich2$, "Feld.Datum")
                    Addieren Sum$, "U", "-", Add$, 1, 1, "-"
                    Puffer$ = Teil$(Sum$, "Zeitangabe.Minimum")
                    TerminZuZahl Puffer$, v1%, j1%, m1%, t1%
                    Puffer$ = Teil$(Sum$, "Zeitangabe.Maximum")
                    TerminZuZahl Puffer$, v2%, j2%, m2%, t2%
                    IF y% = 1 THEN
                        Status$ = Teil$(Vergleich1$, "Feld.Datum.Status")
                        IF Status$ = "≤" OR v1% = -1 THEN
                            Add$ = Teil$(Vergleich2$, Variable$)
                            Aendern Vergleich1$, Variable$, Add$
                        END IF
                    ELSE
                        Umkehren v1%, j1%, m1%, t1%, v2%, j2%, m2%, t2%, "-"
                        Status$ = Teil$(Vergleich1$, "Feld.Datum.Status")
                        IF Status$ = "≥" OR v2% = -1 THEN
                            Add$ = Teil$(Vergleich2$, Variable$)
                            Aendern Vergleich1$, Variable$, Add$
                        END IF
                    END IF
                    Add$ = Teil$(Vergleich1$, "Feld.Datum")
                    StatusNeu Add$
                    Aendern Vergleich1$, "Feld.Datum", Add$
                END IF
            END IF
        NEXT Ort&
    NEXT y%

```

```

'Differenz abspeichern:
'-----'
    Ergebnis Bezug%(z%), BL&(z%), Vorgabe$, Vergleich1$, z%, z%, Num&
END IF
NEXT z%

'Gleichzeitige Ereignisse:
'-----'
FOR z% = 2 TO 3
    IF E%(z%) <> 0 THEN
        'Pause
        Puffer$ = Laden$(Bezug%(E%(z%)), Feldlaenge%, BL&(E%(z%)), 0)
        Vgl$ = Laden$(Bezug%(E%(1)), Feldlaenge%, BL&(E%(1)), 0)
        Ergebnis Bezug%(E%(z%)), BL&(E%(z%)), Puffer$, Vgl$, E%(z%), E%(1), Num&
    END IF
NEXT z%
IF E%(5) <> 0 THEN
    Vorgabe$ = Laden$(Bezug%(E%(5)), Feldlaenge%, BL&(E%(5)), 0)
    Vgl$ = Laden$(Bezug%(E%(4)), Feldlaenge%, BL&(E%(4)), 0)
    Ergebnis Bezug%(E%(5)), BL&(E%(5)), Vorgabe$, Vgl$, E%(5), E%(4), Num&
END IF
END SUB 'Optimieren _____'

'=====
SUB Pause
'=====
' Macht eine Pause und gibt die Möglichkeit zur Unterbrechung des Programms
' mit Einsicht in den Programmtext.
' Der Befehl STOP arbeitet unter Windows XP nicht immer zuverlässig.
'
' Bearbeitung:
' 4. 9.2001: Norbert Südland, München
' 10.10.2016: Norbert Südland, Aalen
' 7. 3.2017: Norbert Südland, Aalen
'-----'

DIM Antwort$ 'AS STRING
DIM x%       'AS INTEGER
DIM y%

x% = POS(0)
y% = CSRLIN
COLOR 7, 0
LOCATE 25, 1
PRINT SPACE$(80);
LOCATE 25, 6
COLOR 0, 7
PRINT " Weiter mit Tastendruck ";
COLOR 15, 0
PRINT " Programmcode ansehen: ";
COLOR 0, 7
IF ENVIRON$("COMSPEC") = "Z:\COMMAND.COM" THEN
    PRINT " [ Strg ] - [ Rollen ] ";
ELSE
    PRINT " [ Strg ] - [ Pause ] ";
END IF
COLOR 7, 0

```

```

WHILE INKEY$ <> ""           'Datenmüll raus!
WEND
DO
  Antwort$ = INKEY$
  LOOP WHILE LEN(Antwort$) = 0
  LOCATE 25, 1
  PRINT SPACE$(80);
  LOCATE y%, x%

END SUB 'Pause _____'

'=====
SUB Protokollmeldung (Meldetext$, NeueZeile%)
'=====
' Hängt `Meldetext$` an `Protokolldatei$` an.
'
' Bearbeitung:  7.10.2016 -  8.10.2016 Norbert Südland, Aalen
'-----
DIM Protokoll%

IF NeueZeile% = 0 THEN
  PRINT Meldetext$;
ELSE
  PRINT Meldetext$
END IF
IF Protokolldatei$ <> "" THEN
  Protokoll% = FREEFILE
501 OPEN Protokolldatei$ FOR APPEND AS #Protokoll%
  IF NeueZeile% = 0 THEN
    PRINT #Protokoll%, Meldetext$;
  ELSE
    PRINT #Protokoll%, Meldetext$
  END IF
  CLOSE #Protokoll%
END IF

END SUB 'Protokollmeldung _____'

'=====
SUB Pruefen (Daten$)
'=====
' Prüft, ob der Datei-Eintrag 'Daten$' sinnvoll ist.
'
' Bearbeitung:
'  4. 8.2001 -  4. 9.2001:  Norbert Südland, München
' 25. 3.2009:  Norbert Südland, Aalen
'-----
DIM Puffer$
DIM Variable$
DIM Vorgaenger$
DIM Bezug$
DIM Pruefgroesse%      'AS INTEGER
DIM v%, j%, m%, t%
DIM z%
DIM L%
DIM Gefunden%

```

```

'Name vorhanden?
'-----'
Pruefgroesse% = SIZEOF%( "NAME ")
Puffer$ = Teil$(Daten$, "Daten.Name ")
IF Puffer$ = SPACE$(Pruefgroesse%) THEN ERROR 102

'Vorgänger testen bzw. korrigieren:
'-----'
FOR z% = 1 TO 2
  Variable$ = "Daten.V[ " + LTRIM$(STR$(z%)) + "]"
  Vorgaenger$ = Teil$(Daten$, Variable$)
  Puffer$ = Teil$(Vorgaenger$, "Vorgänger.Name ")
  IF Puffer$ <> SPACE$(Pruefgroesse%) THEN
    Puffer$ = FuerNutzer$(Teil$(Vorgaenger$, "Vorgänger.Direkt "))
    IF Puffer$ <> "J" AND Puffer$ <> "N" THEN
      Pause
      ERROR 103
    END IF
    Puffer$ = FuerNutzer$(Teil$(Vorgaenger$, "Vorgänger.ZP1 "))
    IF Ereignis%(Puffer$) = 0 THEN
      Pause
      ERROR 103
    END IF
    Puffer$ = FuerNutzer$(Teil$(Vorgaenger$, "Vorgänger.ZP2 "))
    IF Ereignis%(Puffer$) = 0 THEN
      Pause
      ERROR 103
    END IF
  ELSE
    'Löschen, falls kein Name angegeben ist:
    Aendern Daten$, Variable$, SPACE$(SIZEOF%( "Vorgänger "))
  END IF
NEXT z%

'Vorgänger eventuell nach vorne schieben:
'-----'
IF Teil$(Daten$, "Daten.V[1] ") = SPACE$(SIZEOF%( "Vorgänger ")) THEN
  Aendern Daten$, "Daten.V[1] ", Teil$(Daten$, "Daten.V[2] ")
  Aendern Daten$, "Daten.V[2] ", SPACE$(SIZEOF%( "Vorgänger "))
END IF

'Quellenangabe eventuell löschen oder einmahnen:
'-----'
L% = SIZEOF%( "Zeitangabe ")
IF Teil$(Daten$, "Daten.V[1].Name ") = SPACE$(Pruefgroesse%) THEN
  Gefunden% = 0
  FOR z% = 1 TO 6
    Puffer$ = "Daten.Datum[ " + LTRIM$(STR$(z%)) + "]"
    IF STRLEN%(Teil$(Daten$, Puffer$), " ") < L% THEN
      Gefunden% = 1      'Vordatierung gefunden
      z% = 6
    END IF
  NEXT z%
  '-----'
  ' Hinweis: Die Quellenangabe muß sich in demjenigen Dateieintrag '
  '           befinden, in dem die Vordatierung vorhanden ist. '
  '-----'
  IF Gefunden% = 0 THEN

```

```

    Aendern Daten$, "Daten.Quelle", SPACE$(Pruefgroesse%)
ELSE
    IF Teil$(Daten$, "Daten.Quelle") = SPACE$(Pruefgroesse%) THEN
        Pause
        ERROR 107
    END IF
END IF
END IF

'Bezüge testen und korrigieren:
'-----'
FOR z% = 1 TO 2
    Variable$ = "Daten.B[" + LTRIM$(STR$(z%)) + "]"
    Bezug$ = Teil$(Daten$, Variable$)
    Puffer$ = Teil$(Bezug$, "Bezug.Name")
    IF Puffer$ <> SPACE$(Pruefgroesse%) THEN
        IF Ereignis%(FuerNutzer$(Teil$(Bezug$, "Bezug.ZP1"))) = 0 THEN
            Pause
            ERROR 104
        END IF
        Puffer$ = Teil$(Bezug$, "Bezug.Toll")
        IF Puffer$ = "±" OR Puffer$ = "=" OR Puffer$ = "<" OR Puffer$ = ">" THEN
            ZeitZuZahl Teil$(Bezug$, "Bezug.Datum"), j%, m%, t%
            v% = 1
            Restklasse v%, j%, m%, t%      'Datum standardisieren
            IF v% <> 1 OR (j% = 0 AND m% = 0 AND t% = 0) THEN ERROR 104
            Aendern Bezug$, "Bezug.Datum", ZahlZuZeit$(j%, m%, t%, 0)
        ELSE
            Pause
            ERROR 104
        END IF
    ELSE
        'Löschen, falls kein Bezugsname angegeben ist:
        Aendern Bezug$, "Bezug.ZP1", SPACE$(SIZEOF$( "Bezug.ZP1 "))
        Aendern Bezug$, "Bezug.Toll", SPACE$(SIZEOF$( "Bezug.Toll "))
        Aendern Bezug$, "Bezug.Datum", SPACE$(SIZEOF$( "Bezug.Datum "))
    END IF

    'Zeitdauer testen und korrigieren:
    '-----'
    Puffer$ = Teil$(Bezug$, "Bezug.Tol2")
    IF Puffer$ = "±" OR Puffer$ = "=" OR Puffer$ = "<" OR Puffer$ = ">" THEN
        IF Ereignis%(FuerNutzer$(Teil$(Bezug$, "Bezug.ZP2"))) <> 0 THEN
            ZeitZuZahl Teil$(Bezug$, "Bezug.Dauer"), j%, m%, t%
            v% = 1
            Restklasse v%, j%, m%, t%      'Dauer standardisieren
            IF v% <> 1 OR (j% = 0 AND m% = 0 AND t% = 0) THEN ERROR 105
            Aendern Bezug$, "Bezug.Dauer", ZahlZuZeit$(j%, m%, t%, 0)
        ELSE
            ERROR 105
        END IF
    ELSE
        'Löschen, wenn keine Toleranz der Dauer existiert:
        Aendern Bezug$, "Bezug.Tol2", SPACE$(SIZEOF$( "Bezug.Tol2 "))
        Aendern Bezug$, "Bezug.Dauer", SPACE$(SIZEOF$( "Bezug.Dauer "))
    END IF

    Aendern Daten$, Variable$, Bezug$
NEXT z%

```

```

'Bezug eventuell nach vorne schieben:
'-----'
L% = SIZEOF%("Bezug")
IF Teil$(Daten$, "Daten.B[1]") = SPACE$(L%) THEN
  Aendern Daten$, "Daten.B[1]", Teil$(Daten$, "Daten.B[2]")
  Aendern Daten$, "Daten.B[2]", SPACE$(SIZEOF%("Bezug"))

'Angabe gleichzeitiger Ereignisse:
'-----'
IF Teil$(Daten$, "Daten.Gleichzeitig") <> SPACE$(5) THEN

  'Quellenangabe kopieren:
  '-----'
  Puffer$ = Teil$(Daten$, "Daten.B[1].Quelle")
  Aendern Daten$, "Daten.B[2].Quelle", Puffer$

  'eventuell doppelte Quellenangabe löschen:
  '-----'
  IF Teil$(Daten$, "Daten.B[1].Name") = SPACE$(Pruefgroesse%) THEN
    IF Teil$(Daten$, "Daten.B[1].Tol2") = " " THEN
      Aendern Daten$, "Daten.B[1].Quelle", SPACE$(Pruefgroesse%)
    END IF
  END IF
END IF
END IF
END IF

'Gleichzeitige Ereignisse:
'-----'
Puffer$ = FuerNutzer$(Teil$(Daten$, "Daten.Gleichzeitig"))
Puffer$ = GleichzeitigTest$(Puffer$)
Aendern Daten$, "Daten.Gleichzeitig", FuerDatei$(Puffer$)
END SUB 'Pruefen _____

'=====
FUNCTION Reihenfolge& (Zaehlweise%, Listenlaenge&, Listenelement&)
'=====
'Gibt in 4 Zaehlweisen eine Reihenfolge von 1 bis Listenlänge an.
'Die Ordnung kann direkt durch Angabe des Listenelementes erzeugt werden.
'
'Bearbeitung: 1.10.2016 - 18.03.2017 Norbert Südland, D-73431 Aalen
'-----'
DIM Wert&

IF Listenelement& < 1 OR Listenelement& > Listenlaenge& THEN
  PRINT "Programmierfehler: Listenelement ="; Listenelement&;
  PRINT "ist nur von 1 bis"; Listenlaenge&; "zulässig."
  Pause
  Wert& = 0
ELSE
  SELECT CASE Zaehlweise%
  CASE 1
    Wert& = Listenelement&
  CASE 2
    Wert& = Listenlaenge& + 1 - Listenelement&
  CASE 3
    Wert& = (-1) ^ (Listenelement& MOD 2)
    Wert& = Wert& * INT(Listenelement& / 2)

```



```

    Wert& = Wert& + INT((Listenlaenge& + 1) / 2)
CASE 4
    Wert& = (-1) ^ ((Listenlaenge& + 1 - Listenelement&) MOD 2)
    Wert& = Wert& * INT((Listenlaenge& + 1 - Listenelement&) / 2)
    Wert& = Wert& + INT((Listenlaenge& + 1) / 2)
CASE ELSE
    PRINT "Programmierfehler: Zaehlweise% = "; Zaehlweise%;
    PRINT "statt 1 bis 4 aufgerufen."
    PRINT "Es wird nun Zaehlweise% = 1 verwendet."
    Zaehlweise% = 1
    Wert& = Listenelement&      'Dies verursacht am wenigsten Schaden.
    Pause
END SELECT
END IF

Reihenfolge& = Wert&
END FUNCTION 'Reihenfolge& _____'

'=====
SUB Restklasse (Vorzeichen%, Jahr%, Monat%, Tag%)
'=====
'
'  Bearbeitung:
'  4. 8.2001 - 27. 3.2003:      Norbert Südland
'-----
DIM Tage&      'AS LONG

Tage& = Vorzeichen% * Jahr% * 360& + Monat% * 30& + Tag%
Jahr% = INT(Tage& / 360&)
Tage& = Tage& - Jahr% * 360&
Monat% = INT(Tage& / 30&)
Tag% = Tage& - Monat% * 30&
IF Jahr% < 0 THEN
    Jahr% = -Jahr%
    Vorzeichen% = -1
ELSE
    Vorzeichen% = 1
END IF
IF ABS(Jahr%) >= 10000 THEN ERROR 106
END SUB 'Restklasse _____'

'=====
FUNCTION SchnellePosition% (Liste$(), Name$)
'=====
'  Sucht in `Liste$()` nach `Name$`
'  - Es wird vorausgesetzt, daß die `Liste$()` alphabetisch sortiert ist,
'    wobei alle Sonderzeichen nach ihren ASCII-Nummern behandelt werden.
'  - Das Ergebnis ist eindeutig, wenn der `Name$` maximal einmal in der
'    `Liste$()` vorkommt.
'  - Die interne Indexierung der `Liste$()` bleibt unberücksichtigt.
'
'  Bearbeitung:
'  4. 8.2001 - 27.12.2002:      Norbert Südland
'  Überprüfung:
'-----
DIM Suchbeginn%      'AS INTEGER
DIM Suchende%

```

```
DIM Gefunden%
DIM Probe%
```

```
Suchbeginn% = LBOUND(Liste$)
Suchende% = UBOUND(Liste$)
Gefunden% = 0
WHILE Gefunden% = 0 AND Suchbeginn% <= Suchende%
    Probe% = INT(Suchbeginn% / 2 + Suchende% / 2)      'Überlauf verhindert!
    SELECT CASE Liste$(Probe%)
        CASE IS < Name$
            Suchbeginn% = Probe% + 1
        CASE IS = Name$
            Gefunden% = Probe%
        CASE IS > Name$
            Suchende% = Probe% - 1
    END SELECT
WEND
```

```
SchnellePosition% = Gefunden%
END FUNCTION 'SchnellePosition% _____'
```

```
'=====
FUNCTION Schnittmenge$ (Datum1$, Datum2$)
'=====
' Bildet die Schnittmenge aus `Datum1$` und `Datum2$`.
' Wenn die Schnittmenge leer ist, wird ein Widerspruch erzeugt.
' Dies ist eine der sinnvollen Anwendung des Befehls GOTO zur Vermeidung
' von geschachtelten IF-Abfragen.
'
' Bearbeitung: 20. 9.2016 - 21.09.2016 Norbert Südland, Aalen
'-----'
```

```
DIM Termin1$
DIM Termin2$
DIM Variable$
DIM Ergebnistermin1$
DIM Ergebnistermin2$
DIM Ergebnisdatum$
```

```
DIM Laenge%
DIM VZ1%, Jahr1%, Monat1%, Tag1%
DIM VZ2%, Jahr2%, Monat2%, Tag2%
```

```
'Vorbereitung:
'-----'
IF Datum1$ = Datum2$ THEN
    Ergebnisdatum$ = Datum1$
    GOTO SchnittmengeAusgeben
END IF
Ergebnisdatum$ = SPACE$(SIZEOF("Zeitangabe"))
Laenge% = SIZEOF("Termin")
```

```
'Neues Minimum finden:
'-----'
Variable$ = "Zeitangabe.Minimum"
Termin1$ = Teil$(Datum1$, Variable$)
Termin2$ = Teil$(Datum2$, Variable$)
```

```
IF Termin1$ = Termin2$ THEN
    Ergebnistermin1$ = Termin1$
    GOTO MinimumEintragen
END IF

IF Termin1$ = SPACE$(Laenge%) THEN
    Ergebnistermin1$ = Termin2$
    GOTO MinimumEintragen
END IF

IF Termin2$ = SPACE$(Laenge%) THEN
    Ergebnistermin1$ = Termin1$
    GOTO MinimumEintragen
END IF

TerminZuZahl Termin1$, VZ1%, Jahr1%, Monat1%, Tag1%
TerminZuZahl Termin2$, VZ2%, Jahr2%, Monat2%, Tag2%

IF VZ1% <> VZ2% THEN
    IF VZ1% > 0 THEN
        Ergebnistermin1$ = Termin1$
    ELSE
        Ergebnistermin1$ = Termin2$
    END IF
    GOTO MinimumEintragen
END IF

IF VZ1% * Jahr1% <> VZ2% * Jahr2% THEN
    IF VZ1% * Jahr1% > VZ2% * Jahr2% THEN
        Ergebnistermin1$ = Termin1$
    ELSE
        Ergebnistermin1$ = Termin2$
    END IF
    GOTO MinimumEintragen
END IF

IF Monat1% <> Monat2% THEN
    IF Monat1% > Monat2% THEN
        Ergebnistermin1$ = Termin1$
    ELSE
        Ergebnistermin1$ = Termin2$
    END IF
    GOTO MinimumEintragen
END IF

IF Tag1% <> Tag2% THEN
    IF Tag1% > Tag2% THEN
        Ergebnistermin1$ = Termin1$
    ELSE
        Ergebnistermin1$ = Termin2$
    END IF
    GOTO MinimumEintragen
END IF
```

Pause 'Wenn dies erreicht wird, wurde ein Fall beim Programmieren übersehen.'

```
'====='
```

MinimumEintragen:

```
'====='  
Aendern Ergebnisdatum$, Variable$, Ergebnistermin1$  
  
'Neues Maximum finden:  
'-----'  
Variable$ = "Zeitangabe.Maximum"  
Termin1$ = Teil$(Datum1$, Variable$)  
Termin2$ = Teil$(Datum2$, Variable$)  
  
IF Termin1$ = Termin2$ THEN  
    Ergebnistermin2$ = Termin1$  
    GOTO MaximumEintragen  
END IF  
  
IF Termin1$ = SPACE$(Laenge%) THEN  
    Ergebnistermin2$ = Termin2$  
    GOTO MaximumEintragen  
END IF  
  
IF Termin2$ = SPACE$(Laenge%) THEN  
    Ergebnistermin2$ = Termin1$  
    GOTO MaximumEintragen  
END IF  
  
TerminZuZahl Termin1$, VZ1%, Jahr1%, Monat1%, Tag1%  
TerminZuZahl Termin2$, VZ2%, Jahr2%, Monat2%, Tag2%  
  
IF VZ1% <> VZ2% THEN  
    IF VZ1% < 0 THEN  
        Ergebnistermin2$ = Termin1$  
    ELSE  
        Ergebnistermin2$ = Termin2$  
    END IF  
    GOTO MaximumEintragen  
END IF  
  
IF VZ1% * Jahr1% <> VZ2% * Jahr2% THEN  
    IF VZ1% * Jahr1% < VZ2% * Jahr2% THEN  
        Ergebnistermin2$ = Termin1$  
    ELSE  
        Ergebnistermin2$ = Termin2$  
    END IF  
    GOTO MaximumEintragen  
END IF  
  
IF Monat1% <> Monat2% THEN  
    IF Monat1% < Monat2% THEN  
        Ergebnistermin2$ = Termin1$  
    ELSE  
        Ergebnistermin2$ = Termin2$  
    END IF  
    GOTO MaximumEintragen  
END IF  
  
IF Tag1% <> Tag2% THEN  
    IF Tag1% < Tag2% THEN  
        Ergebnistermin2$ = Termin1$
```

```

ELSE
    Ergebnistermin2$ = Termin2$
END IF
GOTO MaximumEintragen
END IF

```

Pause 'Wenn dies erreicht wird, wurde ein Fall beim Programmieren übersehen.

```
'====='
```

MaximumEintragen:

```
'====='
```

```
Aendern Ergebnisdatum$, Variable$, Ergebnistermin2$
```

```
'Neuen Status bestimmen:
```

```
'-----'
```

```
StatusNeu Ergebnisdatum$
```

```
'====='
```

SchnittmengeAusgeben:

```
'====='
```

```
Schnittmenge$ = Ergebnisdatum$
```

```
END FUNCTION 'Schnittmenge _____'
```

```
'====='
```

```
SUB Schreiben (Datei%, DL&, Datum$, Ziel%, Art%, Quelle%, Ort&, Num&)
```

```
'====='
```

```
' Protokoll gefundener Datierungsbezüge in die zugehörige Ergebnisdatei  
' schreiben.
```

```
' `DL&`: Anzahl der Datensätze in `Datei`.
```

```
' `Datum$`: Zeitangabe für Datensatz Nr. `Num` mit `Ereignis`() = Ziel`.
```

```
' `Ort&` und `Quelle` geben die Herkunft des Datierungsvorschlags an,
```

```
' `Art` steht für eine von insgesamt 11 verschiedenen Möglichkeiten der
```

```
' Datumsberechnung:
```

```
' `Art` = 0: Vorhandenes abschreiben
```

```
' `Art` = 1 und 2: Vorwärtsbezüge Vorgänger (relativ)
```

```
' `Art` = 3 und 4: Vorwärtsbezüge absoluter Bezug
```

```
' `Art` = 5 und 6: Vorwärtsbezüge Dauer
```

```
' `Art` = 7 und 8: Rückwärtsbezüge Vorgänger (relativ)
```

```
' `Art` = 9 und 10: Rückwärtsbezüge absoluter Bezug
```

```
' Ist `Ort&` und `Num` = `0`, so handelt es sich um das durchschnittliche  
' Endergebnis.
```

```
'
```

```
' Bearbeitung:
```

```
' 4. 8.2001 - 22. 1.2003: Norbert Südland
```

```
' Überprüfung:
```

```
' 10. 9.2001
```

```
'-----'
```

```
DIM Feld$ 'AS STRING
```

```
DIM L% 'AS INTEGER
```

```
'Hinweis: `Ort&` und `Num&` sind Ordnungszahlen für Datensätze.
```

```
' `Ziel` und `Quelle` liegen zwischen `1` und `6`:
```

```
'-----'
```

```
IF Num& = 0 THEN
```

```
IF Ort& = 0 THEN
```

```
IF Quelle% = 0 THEN
```

```
IF Ziel% = 0 THEN
```

```

        Pause
    ELSE
        Pause
    END IF
ELSE
    'Pause
END IF
ELSE
    Pause      'Noch nicht abgefangen!
END IF
ELSE
    IF Quelle% >= 1 OR Ziel% <= 6 THEN
        'Pause
    ELSE
        Pause      'Noch nicht abgefangen!
    END IF
    IF Ziel% >= 1 OR Ziel% <= 6 THEN
        'Pause
    ELSE
        Pause      'Noch nicht abgefangen!
    END IF
END IF
'Als Ausnahme kommt nur vor, daß alle vier Werte `0` sind, wenn der
'-----

Feld$ = SPACE$(Feldlaenge%)
Aendern Feld$, "Feld.Datum", Datum$
Aendern Feld$, "Feld.Ziel", Symbol$(Ziel%)
Aendern Feld$, "Feld.Name", STR$(Art%)
Aendern Feld$, "Feld.Quelle", Symbol$(Quelle%)
Aendern Feld$, "Feld.von", STR$(Ort&)
Aendern Feld$, "Feld.nach", STR$(Num&)
Aendern Feld$, "Feld.Ende", "||" + CHR$(13) + CHR$(10)

DL& = DL& + 1
PUT #Datei%, DL& * Feldlaenge% + 1, Feld$
END SUB 'Schreiben _____

'=====
FUNCTION SIZEOF% (StrukturName$)
'=====
' Gibt die Länge der Datenstruktur namens `StrukturName$` zurück.
'
' Bearbeitung:
' 4. 8.2001 - 12. 2.2003:    Norbert Südland
' Überprüfung:
' 12. 2.2003:                Norbert Südland
'-----

DIM Strukturposition%      'AS INTEGER
DIM Strukturlaenge%        'AS INTEGER

Strukturposition% = SchnellePosition%(GVName$(), StrukturName$)
IF Strukturposition% = 0 THEN
    Pause
    Strukturlaenge% = 0      'Name nicht gefunden: Schreibfehler?
ELSE

```

```

'Gespeicherte Strukturlänge lesen:
'-----'
Strukturlaenge% = GVLaeenge%(Strukturposition%)

IF Strukturlaenge% <= 0 THEN
    Pause
    ERROR 190          'Nichtpositive Strukturlänge abgespeichert!
END IF
END IF

SIZEOF% = Strukturlaenge%
END FUNCTION 'SIZEOF% _____'

'=====
SUB StatusNeu (Datum$)
'=====
'
'  Bearbeitung:
'  4. 9.2001: Norbert Südland, München
'-----'
DIM Status$  'AS STRING
DIM Groesse% 'AS INTEGER

Groesse% = SIZEOF%("Termin")
IF Teil$(Datum$, "Zeitangabe.Minimum") = SPACE$(Groesse%) THEN
    Status$ = "<"
ELSE
    Status$ = " "
END IF
IF Teil$(Datum$, "Zeitangabe.Maximum") = SPACE$(Groesse%) THEN
    IF Status$ = " " THEN
        Status$ = ">"
    ELSE
        Status$ = " "
    END IF
END IF
Aendern Datum$, "Zeitangabe.Status", Status$
END SUB 'StatusNeu _____'

'=====
FUNCTION STRLEN% (Text$, EndeZeichen$)
'=====
' Ermittelt die Stringlänge von `Text$` bis zum ersten Auftreten aller
' `EndeZeichen$`, so wie es in der Programmiersprache C üblich ist.
' Kommt `Ende` nicht vor, wird die Stringlänge `LEN(Text$)` zurückgegeben.
'
'  Bearbeitung:
'  4. 8.2001: Norbert Südland, München
'  Überprüfung:
'  4. 8.2001: Norbert Südland, München
'-----'
DIM Zwischenergebnis%  'AS INTEGER

Zwischenergebnis% = INSTR(Text$, EndeZeichen$)
IF Zwischenergebnis% = 0 THEN
    Zwischenergebnis% = LEN(Text$)
ELSE

```

```

    Zwischenenergebnis% = Zwischenenergebnis% - 1
END IF

    STRLEN% = Zwischenenergebnis%
END FUNCTION 'STRLEN% _____'

'=====
FUNCTION Suchen& (Name$, Ort&, NamensTyp%)
'=====
' Sucht in den Namenslisten nach `Name$` und liefert
' die erste Dateiposition von `Name$` ab `Ort&` <= `GEL&`.
'
' Bearbeitung:
' 27. 6.2001 - 30. 1.2003:    Norbert Südland
'-----
DIM Liste$           'AS STRING
DIM Vergleich$
DIM L%               'AS INTEGER
DIM Weiter%
DIM Start&           'AS LONG
DIM Ende&
DIM Mitte&
DIM Gefunden&
DIM Erg&

'Allgemeine Suche:
'-----
Start& = 1
Ende& = NL&(NamensTyp%)

L% = SIZEOF%("Liste")
Liste$ = SPACE$(L%)
Gefunden& = 0
WHILE Gefunden& = 0 AND Start& <= Ende&
    Mitte& = CLNG((Start& + Ende&) / 2)
    GET #Namen%(NamensTyp%), Mitte& * L% + 1, Liste$
    Vergleich$ = Teil$(Liste$, "Liste.Name")
    SELECT CASE Vergleich$
        CASE IS = Name$
            Gefunden& = Mitte&
        CASE IS < Name$
            Start& = Mitte& + 1
        CASE IS > Name$
            Ende& = Mitte& - 1
    END SELECT
WEND

'Erstes Auftreten ab `Ort&` finden (bei sortierter Reihenfolge):
'-----
Weiter% = RICHTIG
IF Gefunden& > 0 THEN
    Erg& = VAL(Teil$(Liste$, "Liste.Nummer"))
ELSE
    Erg& = 0
END IF
SELECT CASE Erg&
CASE 0

```



```

CASE IS > Ort&
  WHILE Gefunden& > 1 AND Erg& > Ort& AND Weiter% = RICHTIG
    GET #Namen%(NamensTyp%), (Gefunden& - 1) * L% + 1, Liste$
    Vergleich$ = Teil$(Liste$, "Liste.Name")
    IF Name$ = Vergleich$ THEN
      IF VAL(Teil$(Liste$, "Liste.Nummer")) >= Ort& THEN
        Gefunden& = Gefunden& - 1
        Erg& = VAL(Teil$(Liste$, "Liste.Nummer"))
      ELSE
        Weiter% = FALSCH
      END IF
    ELSE
      Weiter% = FALSCH
    END IF
  WEND
CASE IS < Ort&
  WHILE Gefunden& < NL&(NamensTyp%) AND Erg& < Ort& AND Weiter% = RICHTIG
    GET #Namen%(NamensTyp%), (Gefunden& + 1) * L% + 1, Liste$
    Vergleich$ = Teil$(Liste$, "Liste.Name")
    IF Name$ = Vergleich$ THEN
      Gefunden& = Gefunden& + 1
      Erg& = VAL(Teil$(Liste$, "Liste.Nummer"))
    ELSE
      Weiter% = FALSCH
    END IF
  WEND
END SELECT

'Ergebnis:
'-----'
IF Erg& < Ort& THEN
  Erg& = 0
END IF
Suchen& = Erg&
END FUNCTION 'Suchen& _____'

'=====
FUNCTION Summe$ (Zeit$, Tol$, Datum$, Richtung$)
'=====
' Addition zweier Datierungen unter Berücksichtigung der Toleranzen.
'
' Bearbeitung:
' 16. 6.2001 - 4. 9.2001 Norbert Südland, München
'-----'

DIM Toleranz$ 'AS STRING
DIM Erg$
DIM Termin$
DIM Groesse% 'AS INTEGER
DIM v%, j%, m%, t%

'Vorbereitung:
'-----'
Groesse% = SIZEOF("Zeitangabe")
Toleranz$ = SPACE$(Groesse%)
Erg$ = SPACE$(Groesse%)

'Datum existent?

```

```

'-----'
IF Datum$ <> Toleranz$ THEN

  'Toleranz berechnen:
  '-----'
  ZeitZuZahl Zeit$, j%, m%, t%
  IF Tol$ = "<" THEN
    IF m% <> 0 THEN j% = j% - 1
    IF t% <> 0 THEN m% = m% - 1
  END IF

  IF t% <> 0 THEN
    Aendern Toleranz$, "Zeitangabe.Minimum.Tag", "-1 "
    Aendern Toleranz$, "Zeitangabe.Minimum.TS", "."
    Aendern Toleranz$, "Zeitangabe.Maximum.Tag", "1 "
    Aendern Toleranz$, "Zeitangabe.Maximum.TS", "."
  ELSEIF m% <> 0 THEN
    Aendern Toleranz$, "Zeitangabe.Minimum.Monat", "-1 "
    Aendern Toleranz$, "Zeitangabe.Minimum.MS", "."
    Aendern Toleranz$, "Zeitangabe.Maximum.Monat", "1 "
    Aendern Toleranz$, "Zeitangabe.Maximum.MS", "."
  ELSE
    Aendern Toleranz$, "Zeitangabe.Minimum.VZ", "- "
    Aendern Toleranz$, "Zeitangabe.Minimum.Jahr", "1 "
    Aendern Toleranz$, "Zeitangabe.Minimum.JS", "."
    Aendern Toleranz$, "Zeitangabe.Maximum.VZ", " "
    Aendern Toleranz$, "Zeitangabe.Maximum.Jahr", "1 "
    Aendern Toleranz$, "Zeitangabe.Maximum.JS", "."
  END IF

  'Toleranzform berücksichtigen:
  '-----'
  Groesse% = SIZEOF%("Termin")
  SELECT CASE Tol$
    CASE "="
      'keine Toleranz
      Aendern Toleranz$, "Zeitangabe.Minimum", SPACE$(Groesse%)
      Aendern Toleranz$, "Zeitangabe.Maximum", SPACE$(Groesse%)
    CASE "<"
      'Ordnungszahl
      IF Richtung$ = "+" THEN
        Aendern Toleranz$, "Zeitangabe.Maximum", SPACE$(Groesse%)
      ELSE
        Aendern Toleranz$, "Zeitangabe.Minimum", SPACE$(Groesse%)
      END IF
    CASE ">"
      'persische Ordnungszahl "
      IF Richtung$ = "+" THEN
        Aendern Toleranz$, "Zeitangabe.Minimum", SPACE$(Groesse%)
      ELSE
        Aendern Toleranz$, "Zeitangabe.Maximum", SPACE$(Groesse%)
      END IF
  END SELECT

  'Zeit addieren:
  '-----'
  Termin$ = ZahlZuTermin$(1, j%, m%, t%)

  Aendern Erg$, "Zeitangabe.Minimum", Termin$
  Aendern Erg$, "Zeitangabe.Maximum", Termin$

```

```
Addieren Toleranz$, "U", Richtung$, Erg$, 0, 0, " "
```

```
Erg$ = Toleranz$
```

```
Addieren Erg$, ">", "+", Datum$, 0, 1, " "
```

```
END IF
```

```
Summe$ = Erg$
```

```
END FUNCTION 'Summe$ _____'
```

```
'====='
```

```
FUNCTION Symbol$ (Ereign%)
```

```
'====='
```

```
' Liefert das zu `Ereign%` gehörende Symbol für Ausgaben.
```

```
'
```

```
' Bearbeitung:
```

```
' 4. 8.2001 - 4. 9.2001: Norbert Südland, München
```

```
' 15. 1.2003: Norbert Südland, Aalen
```

```
'-----'
```

```
DIM Symb$ 'AS STRING
```

```
SELECT CASE Ereign%
```

```
CASE 1
```

```
    Symb$ = "*" "
```

```
CASE 2
```

```
    Symb$ = "A "
```

```
CASE 3
```

```
    Symb$ = "W "
```

```
CASE 4
```

```
    Symb$ = "X "
```

```
CASE 5
```

```
    Symb$ = "E "
```

```
CASE 6
```

```
    Symb$ = "+ "
```

```
CASE ELSE
```

```
    Symb$ = " "
```

```
END SELECT
```

```
Symbol$ = Symb$
```

```
END FUNCTION 'Symbol$ _____'
```

```
'====='
```

```
FUNCTION Teil$ (Daten$, Variable$)
```

```
'====='
```

```
'
```

```
' Bearbeitung:
```

```
' 4. 8.2001 - 12. 2.2003: Norbert Südland
```

```
'-----'
```

```
DIM Suche$ 'AS STRING
```

```
DIM Erg$
```

```
DIM Weitersuche$
```

```
DIM L% 'AS INTEGER
```

```
DIM p%
```

```
DIM Wo%
```

```
L% = LEN(Variable$)
```

```
p% = STRLEN$(Variable$, ".") + 1
```

```

IF p% = L% THEN
    Suche$ = Variable$
ELSE
    p% = p% + STRLEN%(MID$(Variable$, p% + 1, L% - p%), ".") + 1
    IF p% < L% THEN
        Suche$ = LEFT$(Variable$, p% - 1)
    ELSE
        Suche$ = Variable$
    END IF
END IF
Wo% = SchnellePosition%(GVName$(), Suche$)
IF Wo% = 0 THEN
    Erg$ = ""
ELSE
    Erg$ = MID$(Daten$, GVAnfang%(Wo%), GVLaenge%(Wo%))
    IF p% < L% THEN
        Weitersuche$ = GVTyp$(Wo%) + MID$(Variable$, p%, L% - p% + 1)
        Erg$ = Teil$(Erg$, Weitersuche$)
    END IF
END IF
Teil$ = Erg$
END FUNCTION 'Teil$ _____'

'=====
SUB TerminZuZahl (Termin$, VZ%, Jahr%, Monat%, Tag%)
'=====
'
'  Bearbeitung:
'  4. 8.2001 - 4. 9.2001: Norbert Südland, München
'-----
IF Teil$(Termin$, "Termin.VZ") = "-" THEN
    VZ% = -1
ELSE
    VZ% = 1
END IF
Jahr% = VAL(Teil$(Termin$, "Termin.Jahr"))
Monat% = VAL(Teil$(Termin$, "Termin.Monat"))
Tag% = VAL(Teil$(Termin$, "Termin.Tag"))
END SUB 'TerminZuZahl _____'

'=====
SUB Umkehren (v1%, j1%, m1%, t1%, v2%, j2%, m2%, t2%, Richtung$)
'=====
'  Wenn 'Richtung$' = "-", wird die Zeitangabe 'Datum$' gespiegelt.
'  'Richtung' = "+" hat keine Wirkung.
'
'  Bearbeitung:
'  4. 8.2001 - 4. 9.2001: Norbert Südland, München
'-----
DIM z%          'AS INTEGER
DIM v%, j%, m%, t%

IF Richtung$ = "-" THEN
    FOR z% = 1 TO 2
        IF z% = 1 THEN
            v% = v1%: j% = j1%: m% = m1%: t% = t1%
        ELSE

```

```

        v% = v2%: j% = j2%: m% = m2%: t% = t2%
    END IF
    v% = -v%
    m% = -m%
    t% = -t%
    Restklasse v%, j%, m%, t%
    IF z% = 1 THEN
        v1% = v%: j1% = j%: m1% = m%: t1% = t%
    ELSE
        v2% = v%: j2% = j%: m2% = m%: t2% = t%
    END IF
NEXT z%
END IF
END SUB 'Umkehren _____'

'=====
SUB Verschieben (Bezug%(), BL&(), Quelle%, Ziel%)
'=====
'
'  Bearbeitung:
'  4. 8.2001 - 8. 1.2003:    Norbert Südland
'-----
DIM Feld$      'AS STRING
DIM L%         'AS INTEGER
DIM z&         'AS LONG

L% = SIZEOF%("Feld")
FOR z& = 1 TO BL&(Quelle%)
    Feld$ = Laden$(Bezug%(Quelle%), Feldlaenge%, BL&(Quelle%), z&)
    BL&(Ziel%) = BL&(Ziel%) + 1
    PUT #Bezug%(Ziel%), BL&(Ziel%) * L% + 1, Feld$
NEXT z&
BL&(Quelle%) = 1
END SUB 'Verschieben _____'

'=====
FUNCTION Widerspruch% (Datum$)
'=====
'  Prüft, ob 'Datum$' eine in sich widersprüchliche Datierung darstellt.
'
'  Bearbeitung:
'  4. 8.2001 - 4. 9.2001:    Norbert Südland, München
'-----
DIM Sum1$      'AS STRING
DIM Sum2$
DIM t1$
DIM t2$
DIM L%         'AS INTEGER
DIM Erg%

L% = SIZEOF%("Zeitangabe")
Sum1$ = SPACE$(L%)
Sum2$ = Sum1$
L% = SIZEOF%("Termin")
t1$ = Teil$(Datum$, "Zeitangabe.Minimum")
t2$ = Teil$(Datum$, "Zeitangabe.Maximum")

```

```

IF t1$ <> SPACE$(L%) AND t2$ <> SPACE$(L%) THEN
  Aendern Sum1$, "Zeitangabe.Minimum", t1$
  StatusNeu Sum1$
  Aendern Sum2$, "Zeitangabe.Minimum", t2$
  StatusNeu Sum2$
  Addieren Sum2$, "U", "-", Sum1$, 1, 1, "-" 'Differenz in J M T
  IF Teil$(Sum2$, "Zeitangabe.Minimum.VZ") = "-" THEN
    Erg% = 1 'Maximaldatum < Minimaldatum
  END IF
END IF
Widerspruch% = Erg%
END FUNCTION 'Widerspruch% _____

'=====
FUNCTION ZahlZuTermin$ (VZ%, Jahr%, Monat%, Tag%)
'=====
'
'  Bearbeitung:
'  4. 8.2001 - 4. 9.2001: Norbert Südland
'-----
DIM Termin$ 'AS STRING

Termin$ = SPACE$(SIZEOF("Termin"))
Jahr% = VZ% * Jahr%
IF Jahr% < 0 THEN
  VZ% = -1
  Jahr% = -Jahr%
ELSE
  VZ% = 1
END IF
IF VZ% = 1 THEN
  Aendern Termin$, "Termin.VZ", " "
ELSE
  Aendern Termin$, "Termin.VZ", "-"
END IF
IF Jahr% <> 0 THEN
  Aendern Termin$, "Termin.Jahr", LTRIM$(STR$(Jahr%))
  Aendern Termin$, "Termin.JS", "."
END IF
IF Monat% <> 0 THEN
  Aendern Termin$, "Termin.Monat", LTRIM$(STR$(Monat%))
  Aendern Termin$, "Termin.MS", "."
END IF
IF Tag% <> 0 THEN
  Aendern Termin$, "Termin.Tag", LTRIM$(STR$(Tag%))
  Aendern Termin$, "Termin.TS", "."
END IF
ZahlZuTermin$ = Termin$
END FUNCTION 'ZahlZuTermin$ _____

'=====
FUNCTION ZahlZuZeit$ (Jahr%, Monat%, Tag%, Wie%)
'=====
'
'  Bearbeitung:
'  4. 8.2001 - 4. 9.2001: Norbert Südland, München
'-----

```

```

DIM Zeitpunkt$      'AS STRING

Zeitpunkt$ = SPACE$(SIZEOF%( "Zeitpunkt "))
IF Jahr% > 0 THEN
  Aendern Zeitpunkt$, "Zeitpunkt.Jahr", LTRIM$(STR$(Jahr%))
  IF Wie% = 1 THEN
    Aendern Zeitpunkt$, "Zeitpunkt.JS", " J "
  ELSE
    Aendern Zeitpunkt$, "Zeitpunkt.JS", ".J "
  END IF
END IF
IF Monat% > 0 THEN
  Aendern Zeitpunkt$, "Zeitpunkt.Monat", LTRIM$(STR$(Monat%))
  IF Wie% = 1 THEN
    Aendern Zeitpunkt$, "Zeitpunkt.MS", " M "
  ELSE
    Aendern Zeitpunkt$, "Zeitpunkt.MS", ".M "
  END IF
END IF
IF Tag% > 0 THEN
  Aendern Zeitpunkt$, "Zeitpunkt.Tag", LTRIM$(STR$(Tag%))
  IF Wie% = 1 THEN
    Aendern Zeitpunkt$, "Zeitpunkt.TS", " T "
  ELSE
    Aendern Zeitpunkt$, "Zeitpunkt.TS", ".T "
  END IF
END IF
ZahlZuZeit$ = Zeitpunkt$
END FUNCTION 'ZahlZuZeit$ _____ '

'=====
SUB ZeitSchieben (v1%, j1%, m1%, t1%, Status$, v2%, j2%, m2%, t2%, Richtung$)
'=====
'  Umrechnung von Ordnungszahlen in Zahlen (`Richtung$` = "-" ) und zurück.
'
'  Bearbeitung:
'  4. 8.2001 - 4. 9.2001:  Norbert Südland, München
'-----
DIM Start%      'AS INTEGER
DIM Ende%
DIM v%, j%, m%, t%
DIM z%

SELECT CASE Status$
CASE " "
  Start% = 1: Ende% = 2
CASE "≥"
  Start% = 1: Ende% = 1
CASE "≤"
  Start% = 2: Ende% = 2
END SELECT
FOR z% = Start% TO Ende%
  IF z% = 1 THEN
    v% = v1%: j% = j1%: m% = m1%: t% = t1%
  ELSE
    v% = v2%: j% = j2%: m% = m2%: t% = t2%
  
```

```

END IF
IF Richtung$ = "+" THEN
    Restklasse v%, j%, m%, t%
    IF v% = 1 THEN
        j% = j% + 1
    END IF
    m% = m% + 1
    t% = t% + 1
ELSE
    IF v% = 1 THEN
        IF j% < 1 THEN ERROR 110
        j% = j% - 1
    END IF
    m% = m% - 1
    t% = t% - 1
    Restklasse v%, j%, m%, t%
END IF
IF z% = 1 THEN
    v1% = v%: j1% = j%: m1% = m%: t1% = t%
ELSE
    v2% = v%: j2% = j%: m2% = m%: t2% = t%
END IF
NEXT z%
END SUB 'ZeitSchieben _____'

'=====
SUB ZeitZuZahl (Zeitpunkt$, Jahr%, Monat%, Tag%)
'=====
'
' Bearbeitung:
' 4. 8.2001 - 4. 9.2001: Norbert Südland, München
'-----
Jahr% = VAL(Teil$(Zeitpunkt$, "Zeitpunkt.Jahr"))
Monat% = VAL(Teil$(Zeitpunkt$, "Zeitpunkt.Monat"))
Tag% = VAL(Teil$(Zeitpunkt$, "Zeitpunkt.Tag"))
END SUB 'ZeitZuZahl _____'

'=====
SUB Zusammenhang (Num&)
'=====
' Ermittelt den Zusammenhang zwischen verschiedenen Datierungen.
'
' Bearbeitung:
' 16. 6.2001 - 23.12.2002: Norbert Südland
' 8.10.2016: Norbert Südland
'-----
DIM Feld$ 'AS STRING
DIM Daten$
DIM Name$
DIM Text$
DIM L% 'AS INTEGER
DIM z%
DIM Zuerst& 'AS LONG
DIM Ort&

'Berechnung vorbereiten:
'-----

```



```
Zuerst& = Einlesen&(Num&, Daten$)
Feld$ = SPACE$(Feldlaenge%)
Aendern Feld$, "Feld.Ende", "|" + CHR$(13) + CHR$(10)
FOR z% = 1 TO 6
  FOR Ort& = 1 TO BL&(z%)
    PUT #Bezug%(z%), Ort& * Feldlaenge% + 1, Feld$
  NEXT Ort&
  BL&(z%) = 0          'Neuanfang der Datei
NEXT z%
DatierungAbschreiben Zuerst&

'Bezüge erstellen:
'-----'
Name$ = Teil$(Daten$, "Daten.Name")
Ort& = Zuerst&
WHILE Ort& <> 0
  BezuegeVorwaerts Ort&
  IF Ort& = Zuerst& THEN
    BezuegeRueckwaerts Ort&
  END IF
  Ort& = Suchen&(Name$, Ort& + 1, 1)
WEND

'Prüfen, ob Veränderung nötig:
'-----'
Text$ = "Datierungen optimieren..."
Protokollmeldung Text$, 1
Optimieren
END SUB 'Zusammenhang _____'
```