

STRUKTUR.BAS	2.19	18. 3.2017
--------------	------	------------

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

- Wandelt eine ASCII-Strukturdatei in eine Liste für MID\$-Zugriffe um.
- Das Ergebnis wird alphabetisch sortiert in der Datei "\*.POS" abgelegt.
- Eine Verkettung über 'CHAIN' mit anderen Programmen ist möglich.
- Das Programm ist Bestandteil des HISTORIK-Projektes.

### 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)

### I N H A L T :

- HAUPTTEIL
- FEHLERBEHANDLUNG
- UNTERPROGRAMME VIA GOSUB
  - Strukturen erkennen
  - Felder sortieren
- UNTERPROGRAMME UND FUNKTIONEN
  - Leerzeichenweg\$()
  - Pause()
  - Platz()
  - SPlatz()
  - Taste\$()

### W I C H T I G E   H I N W E I S E :

- Die Zählung der Indizes beginnt immer bei '1'
- Das Programm erwartet eine Strukturdatei mit folgendem Aufbau:
  - Strukturname:           MUSS IN DERSELBEN ZEILE VOR "{" STEHEN
  - Strukturbeginn:       "{"
  - Strukturende:           "}"
  - Feldbeginn:            "["
  - Feldende:              "]"
  - Paarzuordnung zwischen "{" und "}":       |: Länge Name CR/LF :|
  - je Zeile maximal 1 Paarzuordnung
  - Kommentar wird durch Zeilenbeginn "-" begonnen und beendet
- Beispiel einer Struktur:

-----		
Kommentar		
-----		
NAME { 21 }		
Liste {		
NAME	Name	
8	Nummer	
NAME	Text[ 3 ]	
}		
- Zulässig sind alle Text-Schriftzeichen außer "{", "}", "[", "]", "-"		
B E A R B E I T U N G :		
8. 8.1994 - 19.12.1994	Norbert Südland, Aalen	
15. 7.2001 - 11. 9.2001	Norbert Südland, München	
31. 8.2002 - 4. 5.2007	Norbert Südland, Aalen	
8.10.2016 - 18. 3.2017	Norbert Südland, Aalen	

VORBEREITUNG:
---------------

'OPTION EXPLICIT      'Nützlich bei Visual Basic

CONST Konfigurationsdatei\$ = "STRUKTUR.CFG"

DECLARE SUB Pause ( )

DECLARE SUB Platz (AP%, OP%(), LP%(), BP\$(), TP\$(), Erw%)

DECLARE SUB SPlatz (AnzahlS%, Orts%(), BezS\$())

DECLARE FUNCTION Leerzeichenweg\$ (Zeile\$, Leer%, Laenge%)

DECLARE FUNCTION Taste\$ ( )

DIM Arbeitsplatz\$

DIM Strukturdatei\$

DIM Ruecksprungprogramm\$

DIM Ausgabedatei\$

DIM Zeile\$

DIM Name\$

DIM Konfig%

DIM Struktur%

DIM Kommentar%

DIM StrukturStart%

DIM Ausgabe%

DIM z%            'Zähler

DIM p%            'Position

DIM l%            'Länge

DIM sAnzahl%

DIM Anzahl%

DIM Array%

DIM Aktuell%

DIM dl%            'Dummy-Länge

DIM dp%            'Dummy-Position

DIM Gefunden%

DIM Grenze%

```
DIM y%
DIM x%
DIM w%
```

HAUPTTEIL:
------------

```
'Vorbereitung:
```

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

```
COMMON Programmparameter%, Arbeitsverzeichnis$, Zeit$, Zaehlweise%
```

```
ON ERROR GOTO Fehler
```

```
CLS
```

```
IF Arbeitsverzeichnis$ = "" THEN
```

```
    ' 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$ + "\"
```

```
10    MKDIR Arbeitsplatz$ + "HISTORIK.TMP"
```

```
    Arbeitsplatz$ = Arbeitsplatz$ + "HISTORIK.TMP\"
```

```
ELSE
```

```
    IF Arbeitsplatz$ <> "" THEN Arbeitsplatz$ = Arbeitsplatz$ + "\"
```

```
END IF
```

```
'Beschreibbarkeit von `Arbeitsplatz$` prüfen:
```

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

```
BSAVE Arbeitsplatz$ + "STRUKTUR.CHK", 0, 0
```

```
KILL Arbeitsplatz$ + "STRUKTUR.CHK"
```

```
ELSE
```

```
    Arbeitsplatz$ = Arbeitsverzeichnis$
```

```
END IF
```

```
REDIM SHARED sPos%(0)
```

```
REDIM SHARED Ort%(0)
```

```
REDIM SHARED Laenge%(0)
```

```
REDIM SHARED sBez$(0)
```

```
REDIM SHARED Bez$(0)
```

```
REDIM SHARED Typ$(0)
```

```
Konfig% = FREEFILE
```

```
1 OPEN Arbeitsplatz$ + Konfigurationsdatei$ FOR INPUT AS #Konfig%
```

```
2     LINE INPUT #Konfig%, Strukturdatei$
```

```
3     LINE INPUT #Konfig%, Ausgabedatei$
```

```
    LINE INPUT #Konfig%, Ruecksprungprogramm$
```

```
CLOSE #Konfig%
```

```
'Strukturen erfassen:
```

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

```

Struktur% = FREEFILE
4  OPEN Strukturdatei$ FOR INPUT AS #Struktur%
    Kommentar% = 0
    StrukturStart% = 0
    sAnzahl% = 0
    Anzahl% = 0
    WHILE (NOT EOF(Struktur%))
        LINE INPUT #Struktur%, Zeile$
        PRINT Zeile$
        Zeile$ = Leerzeichenweg$(Zeile$, p%, 1%)
        IF LEFT$(Zeile$, 1) = "-" THEN
            IF Kommentar% = 0 THEN
                Kommentar% = 1
            ELSE
                Kommentar% = 0
            END IF
        END IF
        IF Kommentar% = 0 AND LEFT$(Zeile$, 1) <> "-" THEN
            IF LEN(Zeile$) > 0 THEN
                IF RIGHT$(Zeile$, 1) = "{" THEN
                    GOSUB Strukturbeginn
                ELSE
                    IF RIGHT$(Zeile$, 1) = "}" THEN
                        GOSUB Strukturende
                    ELSE
                        GOSUB Strukturinhalt
                    END IF
                END IF
            END IF
        END IF
    WEND
    CLOSE #Struktur%

    REDIM a(Anzahl%)
    GOSUB Sortieren

    PRINT "Daten werden nach "; Ausgabedatei$; " geschrieben."
    Ausgabe% = FREEFILE
    OPEN Ausgabedatei$ FOR OUTPUT AS #Ausgabe%
        PRINT #Ausgabe%, Leerzeichenweg$(STR$(Anzahl%), p%, 1%); " Daten"
        FOR z% = 1 TO Anzahl%
            PRINT #Ausgabe%, Leerzeichenweg$(STR$(Ort%(a(z%))), p%, 1%); ". ", ", ";
            PRINT #Ausgabe%, Leerzeichenweg$(STR$(Laenge%(a(z%))), p%, 1%), ", ";
            PRINT #Ausgabe%, Bez$(a(z%)), ", "; Typ$(a(z%))
        NEXT z%
    CLOSE #Ausgabe%

    'Rücksprung zu aufrufendem Programm:
    '-----'
    IF Ruecksprungprogramm$ <> "" THEN
        REDIM Ort%(0), Laenge%(0), Bez%(0), Typ%(0), sPos%(0), sBez%(0)
5      CHAIN Ruecksprungprogramm$
    END IF

'=====
Programmende:
'=====
SYSTEM

```

\_\_\_\_\_ ENDE DES HAUPTTEILS \_\_\_\_\_

FEHLERBEHANDLUNG:

```
'====='  
Fehler:  
'====='  
SELECT CASE ERR  
CASE 53  
    SELECT CASE ERL  
    CASE 1  
        PRINT "Konfigurationsdatei "; Konfigurationsdatei$; " fehlt."  
        PRINT  
        COLOR 0, 7  
        PRINT " Bitte das Programm über HISTORIK.BAT starten! "  
        COLOR 7, 0  
        Pause  
        GOTO Programmende  
    CASE 4  
        PRINT "Strukturdatei "; Strukturdatei$; " fehlt."  
        PRINT  
        COLOR 0, 7  
        PRINT " Bitte das Programm über HISTORIK.BAT starten! "  
        COLOR 7, 0  
        Pause  
        GOTO Programmende  
    CASE 5  
        PRINT "Rücksprungprogramm "; Ruecksprungprogramm$; " fehlt."  
        PRINT  
        COLOR 0, 7  
        PRINT " Bitte das Programm über HISTORIK.BAT starten! "  
        COLOR 7, 0  
        Pause  
        GOTO Programmende  
    END SELECT  
CASE 62  
    SELECT CASE ERL  
    CASE 2  
        PRINT "Konfigurationsdatei "; Konfigurationsdatei$; " ist leer."  
        PRINT  
        COLOR 0, 7  
        PRINT " Bitte die Originaldaten erneut einlesen! "  
        COLOR 7, 0  
        Pause  
        GOTO Programmende  
    CASE 3  
        'Kein Rücksprungprogramm angegeben.  
        RESUME NEXT  
    CASE ELSE  
        Pause  
    END SELECT  
CASE 70  
    'Zugriff verweigert  
    SELECT CASE ERL  
    CASE 10  
        'Schreibtests auf `Arbeitsplatz$`  
        PRINT "Das Arbeitsplatz-Verzeichnis " + Arbeitsplatz$  
        PRINT "ist schreibgeschützt. Es wird über " + CHR$(34);
```

```
PRINT "SET HISTORIKTEMP=.." + CHR$(34) + " auf DOS-Ebene gesetzt."
Pause
SYSTEM
CASE ELSE
    Pause
END SELECT
CASE 75      'Pfad- /Dateizugriffsfehler
SELECT CASE ERL
CASE 10      'Schreibtests auf `Arbeitsplatz$`
RESUME NEXT
CASE ELSE
    Pause
END SELECT
CASE 100
PRINT "Strukturbeginn ohne Namensangabe in Strukturdatei ";
PRINT Strukturdatei$; "."
GOTO Programmende
CASE 101
PRINT "Verschachtelte Struktur in Strukturdatei "; Strukturdatei$; "."
GOTO Programmende
CASE 102
PRINT "Strukturbeginn fehlt in Strukturdatei "; Strukturdatei$; "."
GOTO Programmende
CASE 103
PRINT "Unbekannte Strukturbezeichnung '" + Name$ + "' ."
GOTO Programmende
END SELECT
ON ERROR GOTO 0
GOTO Programmende
' _____ ENDE DER FEHLERBEHANDLUNG _____

'
'|-----|
'|          UNTERPROGRAMME VIA GOSUB:         |
'|-----|
'

'|-----|
'|              Strukturen erkennen:           |
'|-----|
'

'=====
Strukturbeginn:
'=====
IF StrukturStart% = 0 THEN
    IF l% < 3 THEN ERROR 100
    StrukturStart% = 1
    SPlatz sAnzahl%, sPos(), sBez$()
    Platz Anzahl%, Ort%(), Laenge%(), Bez$(), Typ$(), 1
    sBez$(sAnzahl%) = LEFT$(Zeile$, p% - 1)
    sPos%(sAnzahl%) = Anzahl%
    Ort%(Anzahl%) = 1
    Laenge%(Anzahl%) = 0
    Bez$(Anzahl%) = sBez$(sAnzahl%)
ELSE
    ERROR 101
END IF
RETURN 'Strukturbeginn
```

```

'=====
Strukturende:
'=====
  IF StrukturStart% = 1 THEN
    StrukturStart% = 0
    Laenge%(sPos%(sAnzahl%)) = Ort%(Anzahl%) + Laenge%(Anzahl%) - 1
  ELSE

    'Struktur in 1 Zeile:
    '-----'
    IF MID$(Zeile$, p% + 1, 1) = "{" THEN
      SPlatz sAnzahl%, sPos%(), sBez$()
      Platz Anzahl%, Ort%(), Laenge%(), Bez$(), Typ$(), 1
      sBez$(sAnzahl%) = LEFT$(Zeile$, p% - 1)
      sPos%(sAnzahl%) = Anzahl%
      Ort%(Anzahl%) = 1
      Laenge%(Anzahl%) = VAL(MID$(Zeile$, p% + 2, 1% - p% - 2))
      Bez$(Anzahl%) = sBez$(sAnzahl%)
    ELSE
      ERROR 102
    END IF
  END IF
RETURN 'Strukturende _____'

'=====
Strukturinhalt:
'=====
  Platz Anzahl%, Ort%(), Laenge%(), Bez$(), Typ$(), 1
  Ort%(Anzahl%) = Ort%(Anzahl% - 1) + Laenge%(Anzahl% - 1)

  'Array:
  '-----'
  Array% = 1
  IF RIGHT$(Zeile$, 1) = "]" THEN
    FOR z% = 1% - 1 TO 1 STEP -1
      IF MID$(Zeile$, z%, 1) = "[" THEN
        Array% = VAL(MID$(Zeile$, z% + 1, 1% - z%))
        1% = z% - 1
        z% = 1
      END IF
    NEXT z%
  END IF
  Bez$(Anzahl%) = sBez$(sAnzahl%) + "." + MID$(Zeile$, p% + 1, 1% - p%)
  Aktuell% = Anzahl%
  IF Array% > 1 THEN
    Platz Anzahl%, Ort%(), Laenge%(), Bez$(), Typ$(), Array% - 1
    FOR z% = Array% TO 1 STEP -1
      Name$ = Leerzeichenweg$(STR$(z%), dp%, dl%)
      Bez$(Aktuell% + z% - 1) = Bez$(Aktuell%) + "[" + Name$ + "]"
    NEXT z%
  END IF

  'Datenlänge:
  '-----'
  Name$ = LEFT$(Zeile$, p% - 1)
  Laenge%(Aktuell%) = VAL(Name$)
  IF Laenge%(Aktuell%) = 0 THEN          'verschachtelte Struktur
    Gefunden% = 0

```

```

FOR z% = 1 TO sAnzahl%
  IF sBez$(z%) = Name$ THEN
    Gefunden% = z%
    Laenge%(Aktuell%) = Laenge%(sPos%(Gefunden%))
    z% = sAnzahl%
  END IF
NEXT z%
IF Gefunden% = 0 THEN
  ERROR 103
ELSE
  FOR z% = 0 TO Array% - 1
    Typ$(Aktuell% + z%) = Name$
  NEXT z%
END IF
END IF
FOR z% = 1 TO Array% - 1
  Ort%(Aktuell% + z%) = Ort%(Aktuell%) + z% * Laenge%(Aktuell%)
  Laenge%(Aktuell% + z%) = Laenge%(Aktuell%)
NEXT z%
RETURN 'Strukturinhalt _____'

```

```

' _____ '
' |               Felder sortieren:               | '
' _____ '

```

```

'=====
Sortieren:
'=====

```

```

PRINT
PRINT "Sortieren..."
FOR z% = 1 TO Anzahl%
  a(z%) = z%
NEXT z%
FOR z% = 2 TO Anzahl%
  Grenze% = z%
  FOR y% = 1 TO Grenze% - 1
    SELECT CASE Bez$(z%)
    CASE IS < Bez$(a(y%))
      FOR x% = y% TO 1 STEP -1
        SELECT CASE Bez$(z%)
        CASE IS < Bez$(a(x%))
          FOR w% = Grenze% - 1 TO x% STEP -1
            a(w% + 1) = a(w%)
          NEXT w%
          a(x%) = z%
          x% = 1
          y% = Grenze%
        END SELECT
      NEXT x%
    END SELECT
  NEXT y%
NEXT z%
RETURN 'Sortieren _____'

```

```

' _____ '
' |               UNTERPROGRAMME UND FUNKTIONEN:               | '
' _____ '

```



```

'=====
' FUNCTION Leerzeichenweg$ (Zeile$, LeerPosition%, Laenge%)
'=====
'   Löscht alle Leerzeichen mit Ausnahme des 1. trennenden Leerzeichens in
'   `Zeile$`. Die Position des Leerzeichens wird in `LeerPosition%`, die
'   Stringlänge in `Laenge%` zurückgegeben.
'
'   Bearbeitung:
'   11. 9.2001: Norbert Südland, München
'   23. 4.2007: Norbert Südland, Aalen
'   25. 2.2017: Norbert Südland, Aalen
'-----
DIM Zaehler%

'Führende Leerzeichen und nachfolgende Leerzeichen löschen:
'-----
Zeile$ = LTRIM$(RTRIM$(Zeile$))
Laenge% = LEN(Zeile$)

'Erstes Leerzeichen in der Mitte stehen lassen:
'-----
LeerPosition% = 0
FOR Zaehler% = 2 TO Laenge%
    SELECT CASE MID$(Zeile$, Zaehler%, 1)
    CASE CHR$(8), CHR$(9), CHR$(255)
        MID$(Zeile$, Zaehler%, 1) = SPACE$(1)
    END SELECT
    IF MID$(Zeile$, Zaehler%, 1) = SPACE$(1) THEN
        IF LeerPosition% = 0 THEN
            LeerPosition% = Zaehler%
        ELSE
            Zaehler% = Zaehler% - 1
            Laenge% = Laenge% - 1
            Zeile$ = LEFT$(Zeile$, Zaehler%) + RIGHT$(Zeile$, Laenge% - Zaehler%)
        END IF
    END IF
NEXT Zaehler%

Leerzeichenweg$ = Zeile$
END FUNCTION 'Leerzeichenweg$ _____

'=====
' SUB Pause
'=====
'   Blendet in Zeile 25 eine Ansage ein und wartet auf einen Tastendruck.
'   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:
'   18. 8.2001 Norbert Südland und Eckhard Walter, Adelshofen
'   4. 9.2001: Norbert Südland, München
'   10.10.2016: Norbert Südland, Aalen
'-----
DIM Antwort$ 'AS STRING

```

```

DIM x%          'AS INTEGER
DIM y%

'Aktuelle Cursor-Position ermitteln:
'-----'
x% = CSRLIN
y% = POS(0)

'Zeile 25 löschen und beschriften:
'-----'
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

'Tastendruck abwarten:
'-----'
Antwort$ = Taste$

'Zeile 25 erneut löschen:
'-----'
LOCATE 25, 1, 1
PRINT SPACE$(80);

'Cursor-Position restaurieren:
'-----'
LOCATE x%, y%
END SUB 'Pause _____

'=====
SUB Platz (AP%, OP%(), LP%(), BP$(), TP$(), Erw%)
'=====
' Erweitert den Speicherplatz von `OP%()`, `LP%()`, `BP$()` und `TP$()` um
' `Erw%` Einträge.
' Der Wert von `AP%` wird entsprechend erhöht.
'
' Bearbeitung:
' 11. 9.2001: Norbert Südland, München
' 23. 4.2007: Norbert Südland, Aalen
'-----'
DIM z%

IF Erw% > 0 THEN

    'Bisherige Daten sichern:

```

```

'-----'
REDIM o%(AP%)
REDIM l%(AP%)
REDIM b$(AP%)
REDIM t$(AP%)
FOR z% = 1 TO AP%
    o%(z%) = OP%(z%)
    l%(z%) = LP%(z%)
    b$(z%) = BP$(z%)
    t$(z%) = TP$(z%)
NEXT z%

'Speicherplatz erweitern:
'-----'
AP% = AP% + Erw%
REDIM OP%(AP%)
REDIM LP%(AP%)
REDIM BP$(AP%)
REDIM TP$(AP%)

'bisherige Daten übertragen:
'-----'
FOR z% = 1 TO AP% - Erw%
    OP%(z%) = o%(z%)
    LP%(z%) = l%(z%)
    BP$(z%) = b$(z%)
    TP$(z%) = t$(z%)
NEXT z%

'Zwischenspeicher freigeben:
'-----'
REDIM o%(0)
REDIM l%(0)
REDIM b$(0)
REDIM t$(0)
END IF
END SUB 'Platz _____'

'=====
SUB SPlatz (AnzahlS%, OrtsS(), BezS$())
'=====
' Erweitert den Speicherplatz von `Sort%()` und `SBez$()` um je 1 Eintrag.
' Der Wert von `SAnzahl%` wird entsprechend erhöht.
'
' Bearbeitung:
' 11. 9.2001: Norbert Südland, München
' 23. 4.2007: Norbert Südland, Aalen
'-----'
DIM z%

'Bisherige Daten sichern:
'-----'
REDIM o%(AnzahlS%)
REDIM b$(AnzahlS%)
FOR z% = 1 TO AnzahlS%
    o%(z%) = OrtsS%(z%)
    b$(z%) = BezS$(z%)

```

```

NEXT z%

'Speicherplatz erweitern:
'-----'
AnzahlS% = AnzahlS% + 1
REDIM Orts$(AnzahlS%)
REDIM Bez$(AnzahlS%)

'bisherige Daten übertragen:
'-----'
FOR z% = 1 TO AnzahlS% - 1
    Orts$(z%) = o$(z%)
    Bez$(z%) = b$(z%)
NEXT z%

'Zwischenspeicher freigeben:
'-----'
REDIM o$(0)
REDIM b$(0)

END SUB 'SPlatz _____'

'=====
FUNCTION Taste$
'=====
' Wartet auf einen Tastendruck und liefert das entsprechende ASCII-Zeichen.
'
' Bearbeitung:
' 18. 8.2001: Norbert Südland und Eckhard Walter, Adelshofen
' Überprüfung:
' 18. 8.2001: Norbert Südland und Eckhard Walter, Adelshofen
'-----'
DIM Antwort$ 'AS STRING

'Tastaturpuffer löschen:
'-----'
WHILE INKEY$ <> ""
WEND

'Tastatur erneut abfragen, bis eine Taste gedrückt wurde:
'-----'
Antwort$ = ""
WHILE LEN(Antwort$) = 0
    Antwort$ = INKEY$
WEND

'Ergebnis:
'-----'
Taste$ = Antwort$
END FUNCTION 'Taste$ _____'

```