

Einführung in Programmierung mit ABAP:

Teil 3: Strukturtypen und interne Tabellen

Prof. Dr. Peter Hohmann

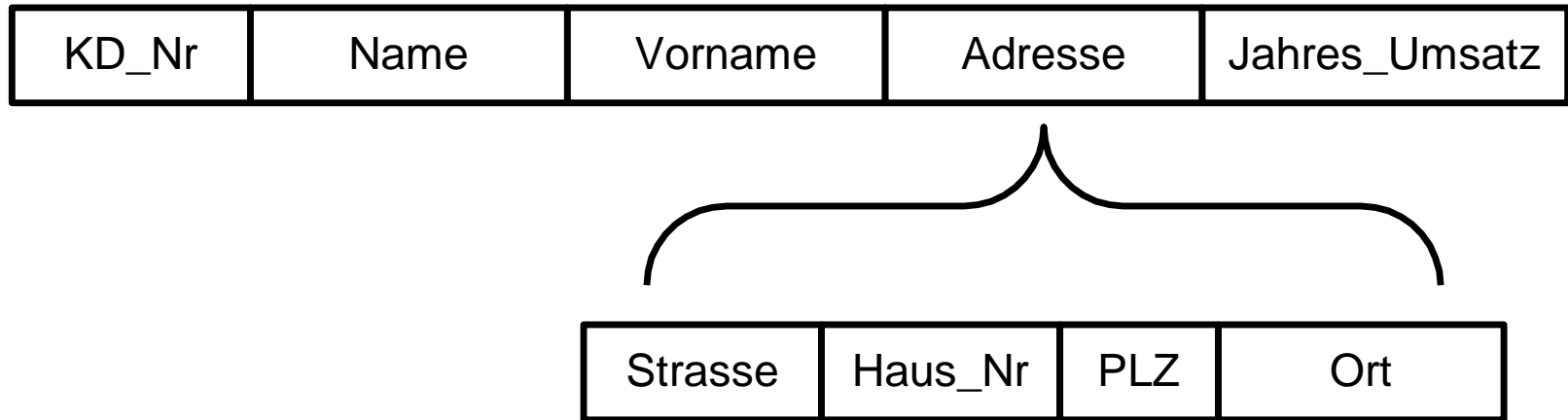
Technische Hochschule Mittelhessen

FB MNI

www.prof-dr-hohmann.de

Version 2017

Strukturtypen



```
TYPES: BEGIN OF adresse,  
        Strasse(30) TYPE c,  
        Haus_Nr(4)  TYPE c,  
        PLZ(5)      TYPE n,  
        Ort(30)     TYPE c,  
END OF adresse.
```

```
TYPES: BEGIN OF KD_Satz,  
        KD_Nr(12)   TYPE c,  
        Name(30)    TYPE c,  
        Vorname(30) TYPE c,  
        KD_Adresse TYPE adresse,  
        Umsatz      TYPE p DECIMALS 2  
END OF KD_Satz.
```

Strukturtypen: Datendefinition und Zugriff

```
DATA:  WA_KD_Satz      TYPE    KD_Satz  .
```

```
WA_KD_Satz-KD_Nr      = '4711' .  
WA_KD_Satz-Name       = 'Meier' .  
WA_KD_Satz-Vorname    = 'Helmut' .  
WA_KD_Satz-KD_Adresse-Strasse = 'Weg' .  
WA_KD_Satz-KD_Adresse-Haus_Nr = 5 .  
WA_KD_Satz-KD_Adresse-PLZ    = '35390' .  
WA_KD_Satz-KD_Adresse-Ort    = 'Giessen' .  
WA_KD_Satz-Umsatz       = '10546.45' .
```

Struktur-Datentypen

```
TYPES typ_name {TYPE typ | LIKE name} .
```

```
{TYPES : | DATA : }  
  BEGIN OF struktur,  
    komp {TYPE typ | LIKE name} ,  
    [komp {TYPE typ | LIKE name} ,] ...  
  END OF struktur .
```

typ / *name* kann sich auf eine beliebige bestehende Typ- oder Datendefinition beziehen (elementarer Typ, Struktur-, Tabellen, Referenztyp, DD,).

Prof. Dr. P. Hohmann,

Interne Tabellen

Index	KD-Nr	Name	Vorname	PLZ	Ort	Strasse
1	4711	Meier	Hubert
2	2345	Müller	Franz
3	1111	Schmitz	Ute
4	2346	Müller	Franz
5	4711	Kern	Manfred
6						

Interne Tabellen

- Tabellenart (wie ist die Tabelle organisiert?)
 - Standardtabelle
 - sortierte Tabelle
 - Hash-Tabelle
- Zeilentyp (wie sind die Spalten der Tabelle strukturiert?)
- Tabellenschlüssel (wie ist der Zugriff?)
 - eindeutig (UNIQUE)
 - nicht eindeutig (NON- UNIQUE)

Tabellen: Index- und Schlüsselzugriff

	Standardtabelle	sortierte Tabelle	Hash-Tabelle
Indexzugriff	<i>ja</i>	<i>ja</i>	<i>nein</i>
Schlüsselzugriff	<i>ja</i>	<i>ja</i>	<i>ja</i>
Schlüsselwerte	<i>nicht eindeutig</i>	<i>eind. o. nicht eind.</i>	<i>eindeutig</i>
bevorzug. Zugriff	<i>vorwieg. Index</i>	<i>vorwieg. Schlüssel</i>	<i>nur Schlüssel</i>

Beispiel

```
TYPES: BEGIN OF t_adresse,  
           strasse(29) TYPE c, ort(20) TYPE c,  
END OF t_adresse .
```

```
TYPES t_adr_tab TYPE  
           STANDARD TABLE OF t_adresse.
```

```
DATA: adr_tab1 TYPE TABLE OF t_adresse ,  
       adr_tab2 TYPE t_adr_tab .
```

```
DATA: BEGIN OF firma,  
           name(20) TYPE c,  
           adr TYPE t_adr_tab,  
END OF firma .
```


Tabellen-Datentypen

```
{TYPES: | DATA:} itab {TYPE | LIKE} tabkind  
    OF {linetype | lineobject}  
    [WITH {UNIQUE | NON-UNIQUE} keys]  
    [INITIAL SIZE n] .
```

```
tabkind: { [STANDARD] TABLE | SORTED TABLE |  
           HASHED TABLE }
```

```
keys: {KEY col1 ... coln | KEY table_line |  
        DEFAULT KEY }
```

Arbeiten mit internen Tabellen I

KD_Nr	Name	Nachname	PLZ	Ort	Strasse
-------	------	----------	-----	-----	---------

```
TYPES: BEGIN OF kd_satz,  
                kd_nr(12)      TYPE n,  
                name(30)      TYPE c,  
                nachname(30)  TYPE c,  
                plz(5)        TYPE n,  
                ort(50)       TYPE c,  
                strasse(30)   TYPE c,  
END OF kd_satz .
```

```
DATA wa_kd_satz                                TYPE kd_satz .
```

— Prof. Dr. P. Hohmann,

Arbeiten mit internen Tabellen II

```
DATA wa_kd_satz TYPE kd_satz .
```

```
DATA itab TYPE STANDARD TABLE OF kd_satz.
```

```
* Kundensatz mit Daten füllen
```

```
wa_kd_satz-kd_nr = 4711 .
```

```
wa_kd_satz-nachname = 'Müller' . " usw.
```

```
APPEND wa_kd_satz TO itab .
```

```
* Kundensatz mit Daten füllen
```

```
wa_kd_satz-kd_nr = 6666 .
```

```
wa_kd_satz-nachname = 'Meier' . " usw.
```

```
APPEND wa_kd_satz TO itab .
```

Prof. Dr. P. Hohmann,

Tabellenzeilen einfügen/ändern (int. Tabelle)

APPEND { *zeile* | **LINES OF** *jtab* } **TO** *itab* .

INSERT { *zeile* | **LINES OF** *jtab* }
INTO TABLE *itab* .

INSERT *zeile* **INTO** *itab* **INDEX** *idx* .

MODIFY TABLE *itab* **FROM** *zeile* .

MODIFY *itab* **FROM** *zeile* **INDEX** *idx* .

Tabellenzeilen lesen/löschen (int. Tabelle)

READ TABLE *itab* **INDEX** *idx* **INTO** *wa* .

READ TABLE *itab* **FROM** *key* **INTO** *wa* .

DELETE *itab* **INDEX** *idx* .

DELETE TABLE *itab* **FROM** *key* .

DELETE *itab* [**FROM** *idx1*] [**TO** *idx2*] .

DELETE *itab* **WHERE** *log_ausdruck* .

Schleife über eine interne Tabelle

```
LOOP AT itab INTO wa  
       $\left[ \left\{ \begin{array}{l} \mathbf{FROM} \ iidx1 \ \mathbf{TO} \ iidx2 \\ \mathbf{WHERE} \ log\_ausdruck \end{array} \right\} \right].$   
  
      <anweisungsblock>  
  
ENDLOOP .
```

2. Beispiel

```
REPORT z_itap_array .
```

```
DATA: zahl TYPE i,  
      etwas TYPE STANDARD TABLE OF i .
```

```
DO 10 TIMES .  
    zahl = sy-index * 2.  
    APPEND zahl TO etwas .  
ENDDO .
```

```
LOOP AT etwas INTO zahl .  
    WRITE: / sy-subrc, sy-tabix, zahl .  
ENDLOOP .
```

Interne Tabelle sortieren / löschen

$$\mathbf{SORT} \, itab \left[\left\{ \begin{array}{l} \mathbf{ASCENDING} \\ \mathbf{DESCENDING} \end{array} \right\} \right]$$
$$\left[\mathbf{BY} \left[comp1 \left[\left\{ \begin{array}{l} \mathbf{ASCENDING} \\ \mathbf{DESCENDING} \end{array} \right\} \right] \right] \dots \right] \cdot$$

REFRESH *itab* .

FREE *itab* .

Information zu internen Tabellen

```
DATA: adr_itab TYPE  
          STANDARD TABLE OF t_adresse ,  
      anzahl    TYPE i ,  
      art(1)    TYPE c.
```

** Anzahl Sätze in der internen Tabelle:*

```
anzahl = LINES( adr_itab ) .
```

```
DESCRIBE TABLE adr_itab LINES anzahl.
```

```
DESCRIBE TABLE { LINES anzahl    |  
                   OCCURS initial |  
                   KIND art } .
```

** art : 'T' , 'S' (sortiert), 'H' (Hash-Tab.)*

Prof. Dr. P. Hohmann,

Noch Fragen?

