

# Betriebliche Anwendungssysteme II / Praktikum

Name: Böttcher Vorname: Felix Fernando MatrNr.: 11147555

SAP-Benutzeraccount: DEVWU-110

**Hinweise:** Die ABAP-Workbench wird mit SE80 gestartet. Die Programmnamen haben den Aufbau

- **ZWU\_IhrSAP\_Account\_SS25\_ProgBeschreibung,**

(ohne Bindestrich), also zum Beispiel **ZWU\_DEVWU000\_SS25\_HELLOWORLD**. Die Programme sind als ausführbares Testprogramm, anwendungsübergreifend und als lokales Objekt anzulegen. Die Editorsperre ist zu setzen, damit Ihr Programm nicht von anderen Nutzern verändert werden kann.

- STRG-F2 prüft die Syntax
- STRG-F3 aktiviert Programme, Textelemente etc.
- F8 ausführen von Programmen

Ihren Programmcode hier bitte als ASCII-Quellcode einfügen (nicht als Snapshot) und bis zum **vorgegebenen Termin** in ILU hochladen. Das Einreichen einer Praktikumsdokumentation stellt eine Prüfungsleistung gemäß der Prüfungsordnung dar. Daher sollen die eingereichten Praktikumslösungen von der obengenannten Person selbst erstellt worden sein und werden im Zweifelsfall einer Plagiatskontrolle unterzogen. Die Programme sind als Quellcode im ASCII-Format in die Dokumentation zu übernehmen, keine Snapshots, jpeg oder andere Bildformate für den Quellcode. Nur für den Nachweis der Testdurchführung sind Snapshots erlaubt.

## Teil 1 ABAP-Basics

**Lernziele:** Operationen auf Strings.

**Voraussetzung:** Sie haben die Vorlesung *Modul 1: ABAP Grundlagen* nachgearbeitet.

**Dokumentation:** Dokumentieren Sie Ihre Lösung in einem Textdokument.

Checkliste Teile 1+2	A1	A2	A3	A4
Erst Programmentwurf, dann Testfälle planen, dann erst Programmierung				
Globale Deklarationen an zentraler Stelle vornehmen				
Lokale Deklarationen am Anfang der Prozedur				
Komplexität von Ausdrücken einschränken und unnötige Programmzeilen vermeiden				
Schachtelungstiefe von Kontrollstrukturen einschränken				
Keine unnötige feste Codierung ( <u>magic numbers</u> vermeiden)				
Implizites Casting vermeiden				
Einheitlicher und lesbarer Quellcode (zB Einrückungen): Pretty Printer				
Programmcode ausreichend dokumentieren				
Dokumentation der Testdurchführung				

- **Wichtige Regeln aus der ABAP-Best Practice**
  - Erst Programmentwurf, dann Testfälle planen, dann erst Programmierung
  - Globale Deklarationen an zentraler Stelle vornehmen
  - Lokale Deklarationen am Anfang der Prozedur
  - Komplexität von Ausdrücken einschränken
  - Schachtelungstiefe von Kontrollstrukturen einschränken
  - Keine unnötige feste Codierung (magic numbers vermeiden)
  - Implizites Casting vermeiden
  - Abfangen von Fehlern, zB Rückgabewert sy-subrc oder sy-tabix auswerten
  - Einheitlicher und lesbarer Quellcode (zB Einrückungen): Pretty Printer
  - Programmcode ausreichend dokumentieren
  - Dokumentation der Testdurchführung

## Aufgabe 1.: Zusammensetzen von Feldern

**Lernziele:** Operationen auf Strings, einfache Logik implementieren

Schreiben Sie ein Programm **ZWU\_IhrSAP\_Account\_SS25...**, das eine Auftragsnummer generiert und diese ausgibt. Die Auftragsnummer wird in einem CHAR-Feld innerhalb des ABAP-Programms zusammengesetzt und aus dem Inhalt von mehreren Parametern gewonnen:

- A: 4-stelliger Code für das Land (z.B. inter-nationale Vorwahl 0049 für Deutschland),
- B: 2-stelliger Alpha-Code für die Region des Landes (z.B. BW: Baden-Württemberg, NW= Nordrhein-Westfalen...),
- C: 1-stelliger Code für Geschäfts- (=G) bzw. Privatkunden (=P),
- D: eine identifizierende Integerzahl (hier beliebige Zahl < 99999).

Der String soll in der Form <ABCD> **physisch gebildet** werden. Die Abbildung zeigt eine exemplarische Ausgabe für K\_NR (s. Codeschnipsel).

EINGABE VON USER <DEVWU-000>:

—> LAND:DEUTSCHLAND  
—> REGION:NORDRHEIN-WESTFALEN  
—> KUNDENTYP:PRIVATKUNDE  
—> ID:00123

DIE GENERIERTE KUNDENNUMMER LAUTET: <0049NWP00123>

Abbildung 1: Exemplarische Ausgabe

Leerzeichen bei den Eingaben sollen eliminiert werden. Die erzeugte Auftragsnummer soll genau **12 Zeichen lang** sein und **keine Leerzeichen** enthalten! Dies soll Ihr Programm garantieren!

Hinweis: Exemplarische Datendefinition und Ausgabezeile:

```
DATA K_Nr(11) TYPE C VALUE '-'.  
  
....  
  
WRITE: / 'DIE GENERIERTE KUNDENNUMMER LAUTET: <' NO-GAP ,K_Nr NO-GAP , '>'.  
  
Lösung bitte hier:==> (Programmcode als ASCII, Testdurchführung als Snapshot)
```

ZWU\_DEVWU110\_SS25\_A1

Als Variante sichern... Mehr

LAND:   
BUNDLAND:   
KUNDE:   
ID:

Die generierte Auftragsnummer lautet: <+357HHP34627>

REPORT ZWU\_DEVWU110\_SS25\_A1.

PARAMETERS: land(20) Type c DEFAULT 'Deutschland',  
bundland(30) Type c DEFAULT 'Berlin',  
kunde(10) Type c DEFAULT 'privat',  
id(5) Type n.

DATA: auftragID Type c LENGTH 12,  
land2(4) TYPE c,  
region(2) Type c.

TRANSLATE land TO UPPER CASE.

*\*Ländercode aus 4 Ziffern, hat die Vorwahl 3 ziffern, wird statt den 2 Nullen das + beibehalten, bei 2 Ziffern wird aus dem + zwei 0 generiert*

CASE land.

WHEN 'ALBANIEN'.

land2 = '+355'.

WHEN 'ANDORRA'.

land2 = '+376'.

WHEN 'BELARUS'.

land2 = '+375'.

WHEN 'BELGIEN'.

land2 = '0032'.

WHEN 'BOSNIEN'.

land2 = '+387'.

WHEN 'BULGARIEN'.

land2 = '+359'.

WHEN 'DEUTSCHLAND'.

land2 = '0049'.

WHEN 'DÄNEMARK'.

land2 = '0045'.

WHEN 'ESTLAND'.

land2 = '+372'.

WHEN 'FINNLAND'.

land2 = '+358'.

WHEN 'FRANKREICH'.

land2 = '0033'.

WHEN 'GEORGIEN'.

land2 = '+995'.

WHEN 'GRIECHENLAND'.

land2 = '0030'.

WHEN 'GROßBRITANNIEN'.

land2 = '0044'.

WHEN 'IRLAND'.

land2 = '+353'.

WHEN 'ISLAND'.

land2 = '+354'.

WHEN 'ITALIEN'.

land2 = '0039'.

WHEN 'KOSOVO'.

land2 = '0038'.

WHEN 'KROATIEN'.

land2 = '+385'.

WHEN 'LETTLAND'.

land2 = '+371'.  
WHEN 'LIECHTENSTEIN'.  
land2 = '+423'.  
WHEN 'LITAUEN'.  
land2 = '+370'.  
WHEN 'MAZEDONIEN'.  
land2 = '+389'.  
WHEN 'MALTA'.  
land2 = '+356'.  
WHEN 'MOLDAVIEN'.  
land2 = '+373'.  
WHEN 'MONACO'.  
land2 = '+377'.  
WHEN 'NIEDERLANDE'.  
land2 = '0031'.  
WHEN 'NORWEGEN'.  
land2 = '0047'.  
WHEN 'ÖSTERREICH'.  
land2 = '0043'.  
WHEN 'POLEN'.  
land2 = '0048'.  
WHEN 'PORTUGAL'.  
land2 = '+351'.  
WHEN 'RUMÄNIEN'.  
land2 = '0040'.  
WHEN 'SAN MARINO'.  
land2 = '+378'.  
WHEN 'SCHWEDEN'.  
land2 = '0046'.  
WHEN 'SCHWEIZ'.  
land2 = '0041'.  
WHEN 'SERBIEN'.  
land2 = '+381'.  
WHEN 'SLOWAKEI'.  
land2 = '+421'.  
WHEN 'SLOWENIEN'.  
land2 = '+386'.  
WHEN 'SPANIEN'.  
land2 = '0034'.  
WHEN 'TSCHECHIEN'.  
land2 = '+420'.  
WHEN 'TÜRKEI'.  
land2 = '0090'.  
WHEN 'UNGARN'.  
land2 = '0036'.  
WHEN 'ZYPERN'.  
land2 = '+357'.  
WHEN OTHERS.  
land2 = 'Ungültiges Land'.

ENDCASE.

*\* deutschen Bundesländer aufgelistet nach ISO 3166-2*

TRANSLATE bundland TO UPPER CASE.

REPLACE ALL OCCURRENCES OF '-' IN bundland WITH ''.

CONDENSE bundland.

CASE bundland.

WHEN 'BADENWÜRTTEMBERG'.

region = 'BW'.

WHEN 'BAYERN'.

region = 'BY'.

WHEN 'BERLIN'.

region = 'BE'.

WHEN 'BRANDENBURG'.

region = 'BB'.

WHEN 'BREMEN'.

region = 'HB'.

WHEN 'HAMBURG'.

region = 'HH'.

WHEN 'HESSEN'.

region = 'HE'.

WHEN 'MECKLENBURGVORPOMMERN'.

region = 'MV'.

WHEN 'NIEDERSACHSEN'.

region = 'NI'.

WHEN 'NORDRHEINWESTFALEN' OR 'NRW'.

region = 'NW'.

WHEN 'RHEINLANDPFALZ'.

region = 'RP'.

WHEN 'SAARLAND'.

region = 'SL'.

WHEN 'SACHSEN'.

region = 'SN'.

WHEN 'SACHSENANHALT'.

region = 'ST'.

WHEN 'SCHLESWIGHOLSTEIN'.

region = 'SH'.

WHEN 'THÜRINGEN'.

region = 'TH'.

WHEN OTHERS.

region = 'Ungültiges Bundesland'.

ENDCASE.

*"Kontrollstrukturen für reibungslosen Durchlauf*

IF kunde+0(1) NE 'G' AND kunde+0(1) NE 'P'.

kunde = 'P'.

ENDIF.

condense land2.

condense region.  
condense kunde.  
condense id.  
auftragID = land2.  
auftragID+4(2) = region.  
auftragID+6(1) = kunde.  
auftragID+7(5) = id.  
condense auftragid.  
WRITE:/ 'Die generierte Auftragsnummer lautet: <' NO-GAP, auftragid NO-GAP,'>'.  
  
(c) Westenberger

## Aufgabe 2.: Datumsberechnungen

**Lernziele:** Operieren mit Datumsvariablen, Extraktion der Jahreszahl aus dem Datum

Schreiben Sie ein Programm, das die Anzahl der Tage seit Ihrem **Geburtstag** berechnet und ausgibt (SY\_DATUM ist eine Systemvariable in ABAP, die das heutige Datum beinhaltet).

Geben Sie aus, in welches Kalenderjahr Ihr **11111. Geburtstag** fallen wird (oder gefallen ist).

```
Nach eigenen Angaben wurden Sie am 01.01.1990 geboren, also vor      12.348  Tagen.  
Ihr      11.111  ter Geburtstag fiel in das Jahr  2020      .  
  
Hoffentlich haben Sie diesen Anlass damals gebührend gefeiert.
```

Abbildung 2: Exemplarisches Ergebnis

Lösung bitte [hier](#) ==> (Programmcode als ASCII, Testdurchführung als Snapshot)

Datumsberechnung

Als Variante sichern... Mehr ▾

GB\_DATE:

Nach eigenen Angaben wurden Sie am 15.02.2001 geboren, also vor 8.894 Tagen.  
Ihr 11.111 ter Geburtstag fiel in das Jahr 2031.  
Hoffentlich feiern sie diesen Anlass gebührend.

REPORT ZWU\_DEVWU110\_SS25\_A2.

PARAMETERS: gb\_date TYPE d DEFAULT '20010215'.

DATA: date\_today type d,  
amount type i,  
date\_11111 type d,  
lived\_days TYPE i,  
year\_11111 type d.

" 1. Anzahl der Tage seit dem Geburtstag

date\_today = sy-datum.

lived\_days = date\_today - gb\_date.

" 2. 11111 Geburtstag

date\_11111 = gb\_date + 11111.

year\_11111 = date\_11111(4) .



```
WRITE: / 'Nach eigenen Angaben wurden Sie am ' NO-  
GAP, gb_date DD/MM/YYYY, ' geboren, also vor ' NO-GAP, lived_days NO-GAP , ' Tagen.',  
    / 'Ihr', 11111, 'ter Lebenstag fiel in das Jahr ' NO-GAP, year_11111 NO-GAP, '!.  
if year_11111 <= date_today.  
WRITE: / 'Hoffentlich haben Sie diesen Anlass damals gebührend gefeiert.'.  
ELSE.  
    WRITE: / 'Hoffentlich feiern sie diesen Anlass gebührend.'.  
ENDIF.
```

### Aufgabe 3.: String-Ersetzungen

**Lernziele:** Operationen auf Strings, Bewertung unterschiedlicher Lösungen

Schreiben Sie ein Programm, das eine beliebige **Zeichenkette** vorgegebener maximaler Länge einliest und verarbeitet. Zur Verarbeitung sind Zeichen einzulesen, die maskiert werden sollen, also durch ein Maskierungszeichen, zB „#“ ersetzt werden sollen. Halten Sie Ihr Programm flexibel erweiterbar. Vermeiden Sie statische Programmierung.

Es gibt zahlreiche Möglichkeiten, diese Aufgabe in ABAP umzusetzen. Eine kompakte, gut verständliche und zugleich flexible Lösung ist hier anzustreben.

```
Ausgangskette: 1234DUBAI4321
zu maskieren: AEIOU
Ergebniskette: 1234D#B##4321
```

Abbildung 3: Exemplarische Lösung

Lösung bitte [hier](#) ==> (Programmcode als ASCII, Testdurchführung als Snapshot)

String Ersetzung

Als Variante sichern... Mehr ▾

INPUT:

MASK:

SYMBOL:

< SAP

String Ersetzung

Mehr ▾

String Ersetzung

Ausgangskette: AAaiiuu00iFFuHFioaF

zu maskieren: AEIOU

Ergebniskette: #####FF#HF###F

REPORT ZWU\_DEVWU110\_SS25\_A3.

PARAMETERS: input TYPE string,

mask TYPE string DEFAULT 'AEIOU',

symbol TYPE c DEFAULT '#'.

DATA:

result TYPE string,

i TYPE i VALUE 0.

result = input.

" Schleife: für jedes Zeichen (das ersetzt)

DO strlen( mask ) TIMES.

" wird für jedes index das zeichen durch ein symbol ersetzt

REPLACE ALL OCCURRENCES OF mask+i(1) in result WITH symbol.

i = i + 1.

ENDDO.

" 3) Ausgabe

WRITE: / 'Ausgangskette: ', input,

      / 'zu maskieren: ', mask,

      / 'Ergebniskette: ', result.

## Teil 2 Interne Tabellen

**Voraussetzung:** Kenntnisse des *Modul 2: ABAP Tabellenverarbeitung*.

**Lernziele:** Arbeiten mit Feldleisten und internen Tabellen, Schreiben in eine interne Tabelle, Collect

**Hinweis:** Spätestens in diesem Teil des Praktikums empfiehlt es sich, erst ein Lösungskonzept zu entwerfen, bevor Sie mit der Implementierung beginnen! Oft gibt es mehrere Ansätze, um eine Aufgabe zu lösen. Überlegen Sie, welche Möglichkeiten es gibt, wägen Sie die Vor-/Nachteile ab und entscheiden Sie sich bewusst für einen Weg.

### Aufgabe 4.: Interne Tabelle

a) Definieren Sie in einem ABAP Report eine interne Tabelle **ITAB1** für Umsatzpositionen wie folgt (Datentyp passend auswählen),

Feldname	Type	Bedeutung
ZNO	?	Laufende Nummer
ZUSER	?	Ersteller der Umsatzposition -> hier Ihr SAP-Account „...“
ZNAME	?	Name des Verkäufers
ZARTICLE	?	Artikel
ZDATE	?	Umsatzdatum
ZREVNET	?	Umsatzbetrag Netto

fügen mehrere Einträge in die Tabelle ein und geben diese Tabelle aus.

Anschließend übertragen Sie alle Datensätze von **ITAB1** in eine weitere interne Tabelle **ITAB2**, wobei Sie den Umsatz gruppiert nach Verkäufer und Artikel aggregieren sollen.

Feldname	Type	Bedeutung
ZNAME	?	Name des Verkäufers
ZARTICLE	?	Artikel
ZREVSUM	?	Umsatzbetrag Netto

**Lösung bitte hier ==>** (Programmcode als ASCII, Testdurchführung als Snapshot)

<

SAP

Basics interne Tabelle

▼

Mehr ▼

Basics interne Tabelle

0000000001DEVWU-110	Felix	Uhr	15062025	12.345,60
0000000002DEVWU-114	Niklas	Handy	17062025	900,00
0000000003DEVWU-198	Furkan	Laptop	11062025	1.200,00
0000000004DEVWU-115	Nils	Motorrad	22022020	3.499,99
0000000005DEVWU-116	VLAD	Bier	30062025	1,09
0000000006DEVWU-110	Felix	Uhr	27062025	50,00

Felix	Uhr	12.395,60
Niklas	Handy	900,00
Furkan	Laptop	1.200,00
Nils	Motorrad	3.499,99
VLAD	Bier	1,09

*"1) 'Erstellung der Struktur, interne Tabelle und Working Area*

```
TYPES: BEGIN OF itab1,  
      zno(10) TYPE n, "Nummerischer Text eignet sich gut für Auftragsnummern  
      zuser(15) TYPE c,  
      zname(15) TYPE c,  
      zarticle(20) TYPE c,  
      zdate TYPE d,  
      zrevnet TYPE p DECIMALS 2,  
End of itab1.
```

```
DATA: z_itab TYPE TABLE OF itab1,  
      wa TYPE itab1,  
      z_itab2 TYPE TABLE OF itab1,  
      wa2 TYPE itab1.
```

*" 2) Schreiben in die interne Tabelle*

```
wa-zno = 1.  
wa-zuser = 'DEVWU-110'.  
wa-zname = 'Felix'.  
wa-zarticle = 'Uhr'.  
wa-zdate = '20250615'.  
wa-zrevnet = '12345.6'.  
APPEND wa TO z_itab.
```

```
wa-zno = 2.  
wa-zuser = 'DEVWU-114'.  
wa-zname = 'Niklas'.  
wa-zarticle = 'Handy'.  
wa-zdate = '20250617'.  
wa-zrevnet = 900.  
APPEND wa TO z_itab.
```

```
wa-zno = 3.  
wa-zuser = 'DEVWU-198'.  
wa-zname = 'Furkan'.  
wa-zarticle = 'Laptop'.  
wa-zdate = '20250611'.  
wa-zrevnet = 1200.  
APPEND wa TO z_itab.
```

```
wa-zno = 4.  
wa-zuser = 'DEVWU-115'.  
wa-zname = 'Nils'.  
wa-zarticle = 'Motorrad'.  
wa-zdate = '20200222'.
```

```
wa-zrevnet = '3499.99'.  
APPEND wa TO z_itab.
```

```
wa-zno = 5.  
wa-zuser = 'DEVWU-116'.  
wa-zname = 'VLAD'.  
wa-zarticle = 'Bier'.  
wa-zdate = '20250630'.  
wa-zrevnet = '1.09'.  
APPEND wa TO z_itab.
```

*"Zum test für Teilaufgabe 2, ein zweiter Eintrag für einen Verkäufer*

```
wa-zno = 6.  
wa-zuser = 'DEVWU-110'.  
wa-zname = 'Felix'.  
wa-zarticle = 'Uhr'.  
wa-zdate = '20250627'.  
wa-zrevnet = '50'.  
APPEND wa TO z_itab.
```

```
LOOP AT z_itab into wa.  
  Write: / wa-zno NO-GAP,  
        wa-zuser,  
        wa-zname,  
        wa-zarticle,  
        wa-zdate ,  
        wa-zrevnet DECIMALS 2.  
ENDLOOP.
```

*\* 3) TODO eine neue Tabelle mit der alten befüllen und nach Verkäufer und artikel aggregieren*

*"Soll die Ausgabe danach sortiert sein oder die tabelle die Struktur haben*

*\*Wenn zweiteres, dann hat die 2te Tabelle ja eine andere Struktur, ansonsten könnte man kein Collect machen und das sollte man für eine Gruppierung machen.*

*" die working class muss die selbe struktur haben wie interne Tabelle, sonst unmöglich collect zu benutzen*

*" COLLECT muss also simuliert werden*

```
LOOP AT z_itab INTO wa.  
  READ TABLE z_itab2 INTO wa2 WITH KEY zname = wa-zname zarticle = wa-zarticle.  
  IF sy-subrc = 0.  
    wa2-zrevnet = wa2-zrevnet + wa-zrevnet.  
    MODIFY z_itab2 FROM wa2 INDEX sy-tabix.  
  ELSE.  
    APPEND wa TO z_itab2.  
  ENDIF.  
ENDLOOP.
```

```
SKIP 3.  
ULINE.
```

LOOP AT z\_itab2 INTO wa2.

WRITE: /

wa2-zname,

wa2-zarticle,

wa2-zrevnet.

ENDLOOP.

### Teil 3 Interne+permanente Tabellen, Daten-Aggregation, komplexe Algorithmen

Checkliste	A5	A6	A7
Erst Programmwurf, dann Testfälle planen, dann erst Programmierung			
Globale Deklarationen an zentraler Stelle vornehmen			
Lokale Deklarationen am Anfang der Prozedur			
Komplexität von Ausdrücken einschränken und unnötige Programmzeilen vermeiden			
Schachtelungstiefe von Kontrollstrukturen einschränken			
Keine unnötige feste Codierung (magic numbers vermeiden)			
Implizites Casting vermeiden			
Abfangen von Fehlern, zB Rückgabewert sy-subrc oder sy-tabix auswerten			
Einheitlicher und lesbarer Quellcode (zB Einrückungen): Pretty Printer			
Programmcode ausreichend dokumentieren			
Dokumentation der Testdurchführung			

Bei Aufgabe 7 sind nur der Programmwurf sowie die Testdurchführung zu dokumentieren.

**Lernziele:** Arbeiten mit Feldleisten und internen Tabellen, Schreiben in eine vorgegebene permanente Tabelle, Collect, Sortieren

**Hinweis:** Auch in diesem Teil des Praktikums empfiehlt es sich, erst ein Lösungskonzept zu entwerfen, bevor Sie mit der Implementierung beginnen! Oft gibt es mehrere Ansätze, um eine Aufgabe zu lösen. Überlegen Sie, welche Möglichkeiten es gibt, wägen Sie die Vor-/Nachteile ab und entscheiden Sie sich bewusst für einen Weg.

## Aufgabe 5.: Interne Tabellen

a) Definieren Sie in einem ABAP Report eine interne Tabelle für Umsatzpositionen (s. folgende Abbildung) und fügen fünf Einträge in die interne Tabelle ein

Feldname	Type	Bedeutung
ZNO	?	Laufende Nummer
ZUSER	?	Ersteller der Umsatzposition -> hier Ihr SAP-Account DEVWU-xyzxyz
ZNAME	?	Name des Verkäufers
ZARTICLE	?	Artikel
ZDATE	?	Umsatzdatum
ZREVENUE	?	Umsatzbetrag Netto

Abbildung 4: Interne Tabelle Umsatz

Anschließend schreiben Sie nun die Umsatzpositionen in die gegebene Data Dictionary-Tabelle ZWU\_SALES (s. folgende Abbildung).

Feld	Key	Initi...	Datenelement	Datentyp	Länge	DezStel...	KoordSystem	Kurzbeschreibung
ZMANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0		0Mandant
ZNO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		INT2	5	0		0
ZCREATEDBY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		CHAR	10	0		0Created by (SAP-USER)
ZMODIFIEDBY	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	10	0		0Modified by (SAP-USER)
ZNAME	<input type="checkbox"/>	<input checked="" type="checkbox"/>		CHAR	20	0		0Name
ZDATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>		DATS	8	0		0Date of last modification
ZARTICLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>		CHAR	20	0		0Article Code
ZREVENUE	<input type="checkbox"/>	<input type="checkbox"/>		DEC	11	2		0Revenue
ZNYEARMONTH	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	20	0		0Year-Month Description
ZNARTICLE	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	30	0		0Article ACII consistent

Abbildung 5: ZXM\_DEV\_SALES Tabellenbeschreibung

Für das Feld „ZCREATEDBY“ weisen Sie die Systemvariable sy-uname zu, für ZMANDT die Systemvariable sy-mandt. Die Felder „ZMODIFIEDBY“, „ZDATE“ und „ZNARTICLE“ können Sie leer lassen. Achten Sie darauf, dass der Primary Key, den Sie definieren, den Datensatz eindeutig macht und noch nicht vergeben wurde. Sinnvollerweise sucht sich IHR Programm einen Wert für „ZNO“ selbständig, der noch nicht vergeben wurde.

Hinweis: Bitte schreiben Sie nur vernünftige Daten in die Tabelle. Wir werden mit den Tabellendaten in folgenden Praktikumsaufgaben weiterarbeiten.

Verwenden Sie Daten mit Verkäufern {MEIER, SCHMITZ} und den Artikeln {BIKINI, BADEHOSE, SONNENCREME, TEICHPUMPE 100} mit Datum aus den Jahren 2021-2023.

Die Tabellenstruktur von **ZWU\_DEV\_SALES** darf nicht verändert werden!

	ZMANDT	ZNO	ZCREATEDBY	ZMODIFIEDBY	ZNAME	ZDATE	ZARTICLE	ZREVENUE
<input type="checkbox"/>	232	7	DEVWU-007		SCHMITZ	10.04.2015	MÖVENSCHIRM	34,00
<input type="checkbox"/>	232	8	DEVWU-007		SCHMITZ	27.11.2014	WÜNSCHELRUTE	92,00
<input type="checkbox"/>	232	9	DEVWU-007		MEIER	06.10.2015	MÖVENSCHIRM	1.018,00
<input type="checkbox"/>	232	10	DEVWU-007		SCHMITZ	26.08.2016	WINDFÄNGER	86,00
<input type="checkbox"/>	232	11	DEVWU-007		SCHMITZ	22.10.2022	WINDFÄNGER	67,00
<input type="checkbox"/>	232	12	DEVWU-007		SCHMITZ	06.01.2024	WÜNSCHELRUTE	19,00
<input type="checkbox"/>	232	13	DEVWU-007		SCHMITZ	15.07.2023	WINDFÄNGER	388,00



Abbildung 6: Exemplarisches Ergebnis

Lösung bitte hier:==>

SAP						DD
						Mehr ▾
DD						
232	2.517	DEVWU-114	SCHMITZ	22.10.2023	Bikini	34,00
232	2.518	DEVWU-114	MEIER	02.05.2022	Bikini	3,00
232	2.519	DEVWU-114	Schmitz	02.04.2022	Badehose	26,00
232	2.520	DEVWU-114	MEIER	03.05.2022	Sonnencreme	13,00
232	2.521	DEVWU-114	Schmitz	04.05.2022	Bikini	33,00
232	2.522	DEVWU-114	MEIER	12.05.2022	Badehose	14,00
232	2.523	DEVWU-114	MEIER	02.05.2022	Bikini	3,00
232	2.524	DEVWU-114	Schmitz	02.04.2022	Badehose	26,00
232	2.525	DEVWU-114	MEIER	03.05.2022	Sonnencreme	13,00
232	2.526	DEVWU-114	Schmitz	04.05.2022	Bikini	33,00
232	2.527	DEVWU-114	MEIER	12.05.2022	Badehose	14,00
232	2.528	DEVWU-114	MEIER	02.05.2022	Bikini	3,00
232	2.529	DEVWU-114	Schmitz	02.04.2022	Badehose	26,00
232	2.530	DEVWU-114	MEIER	03.05.2022	Sonnencreme	13,00
232	2.531	DEVWU-114	Schmitz	04.05.2022	Bikini	33,00
232	2.532	DEVWU-114	MEIER	12.05.2022	Badehose	14,00
232	2.533	DEVWU-219	MEIER	10.11.2023	SONNENCREME	25,50
232	2.534	DEVWU-219	SCHMITZ	10.11.2022	BADEHOSE	89,90
232	2.535	DEVWU-219	MEIER	10.11.2021	BIKINI	115,59
232	2.536	DEVWU-219	SCHMITZ	01.01.2023	BADEHOSE	54,99
232	2.537	DEVWU-219	SCHMITZ	01.01.2021	BIKINI	19,99
232	2.548	DEVWU-113	Meier	15.04.2021	Bikini	34,00
232	2.549	DEVWU-113	Meier	23.09.2021	Badehose	19,99
232	2.550	DEVWU-113	Schmitz	07.06.2022	Sonnencreme	6,99
232	2.551	DEVWU-113	Meier	17.10.2022	Sonnencreme	2,99
232	2.552	DEVWU-113	Schmitz	12.08.2021	Teichpumpe 100	104,99
232	2.553	DEVWU-110	MEIER	18.06.2025	Bikini	100,00
232	2.554	DEVWU-110	SCHMITZ	17.06.2025	Badehose	80,00
232	2.555	DEVWU-110	SCHMITZ	22.06.2025	Sonnencreme	120,00
232	2.556	DEVWU-110	SCHMITZ	18.06.2025	Teichpumpe 100	20,00
232	2.557	DEVWU-110	MEIER	18.06.2025	Windfänger	1,00

REPORT ZWU\_DEVWU110\_SS25\_A5.

TABLES: ZWU\_SALES.

DELETE FROM ZWU\_SALES WHERE ZCREATEDBY = 'DEVWU-110'.

TYPES: BEGIN OF tb\_umsatz,  
 zno TYPE i,  
 zuser(20) TYPE c,  
 zname(20) TYPE c,  
 zarticle(20) TYPE c,  
 zdate TYPE d,  
 zrevenue(15) TYPE p,  
 END OF tb\_umsatz.

DATA: z\_umsatz TYPE TABLE OF tb\_umsatz,

Autor: M. Mustermann

```
wa TYPE tb_umsatz,  
wa_SALES TYPE ZWU_SALES,  
max TYPE i.
```

*" Datensätze hinzufügen*

```
wa-zno = 1.  
wa-zuser = 'DEVWU-110'.  
wa-zname = 'MEIER'.  
wa-zarticle = 'Bikini'.  
wa-zdate = '20250618'.  
wa-zrevenue = 100.  
APPEND wa TO z_umsatz.
```

```
wa-zno = 2.  
wa-zuser = 'DEVWU-036'.  
wa-zname = 'SCHMITZ'.  
wa-zarticle = 'Badehose'.  
wa-zdate = '20250617'.  
wa-zrevenue = 80.  
APPEND wa TO z_umsatz.
```

```
wa-zno = 3.  
wa-zuser = 'DEVWU-114'.  
wa-zname = 'SCHMITZ'.  
wa-zarticle = 'Sonnencreme'.  
wa-zdate = '20250622'.  
wa-zrevenue = 120.  
APPEND wa TO z_umsatz.
```

```
wa-zno = 4.  
wa-zuser = 'DEVWU-115'.  
wa-zname = 'SCHMITZ'.  
wa-zarticle = 'Teichpumpe 100'.  
wa-zdate = '20250618'.  
wa-zrevenue = 20.  
APPEND wa TO z_umsatz.
```

```
wa-zno = 5.  
wa-zuser = 'DEVWU-052'.  
wa-zname = 'MEIER'.  
wa-zarticle = 'Windfänger'.  
wa-zdate = '20250618'.  
wa-zrevenue = 1.  
APPEND wa TO z_umsatz.
```

*" die Datensätze in ZWU\_sales einfügen*

*\* zno soll der Primary Key sein*

```
SELECT MAX( zno ) FROM ZWU_SALES INTO max.
```

```
LOOP AT z_umsatz INTO wa.  
  max = max + 1.  
  wa_SALES-zmandt = sy-mandt.  
  wa_SALES-zno = max.  
  wa_SALES-zcreatedby = sy-uname.  
  wa_SALES-zname = wa-zname.  
  wa_SALES-zdate = wa-zdate.  
  wa_SALES-zarticle = wa-zarticle.  
  wa_SALES-zrevenue = wa-zrevenue.  
  wa_SALES-zyearmonth = ".  
  wa_SALES-znarticle = ".  
  INSERT ZWU_SALES FROM wa_SALES.  
ENDLOOP.
```

```
SKIP.  
Write: / 'UMSATZDATEN'.  
SELECT * FROM ZWU_SALES.  
WRITE: /  
  ZWU_SALES-zmandt,  
  ZWU_SALES-zno,  
  ZWU_SALES-zcreatedby,  
  ZWU_SALES-zmodifiedby,  
  ZWU_SALES-zname,  
  ZWU_SALES-zdate,  
  ZWU_SALES-zarticle,  
  ZWU_SALES-zrevenue,  
  ZWU_SALES-zyearmonth,  
  ZWU_SALES-znarticle.  
ENDSELECT.
```

## Aufgabe 6.: Tabellen lesen und als multi-dimensionalen Report ausgeben

MyDiscounter ist ein Handelsunternehmen, das saisonale Artikel anbietet und das Warenwirtschaftssystem von SAP einsetzt. Im SAP-System des Unternehmens gibt es eine Data Dictionary Tabelle **ZWU\_SALES**, die in knapper Form die Umsatzpositionen vorhält (Verkäufer, Artikel, Datum, Umsatzbetrag, s. unten). Wir wollen nun die Umsätze in aggregierter und dimensionsorientierter Form vorhalten: Dimensionen sind Verkäufer, Artikel und die Zeit als Jahr, und schreiben die aggregierten Daten zunächst in eine berichtsorientierte temporäre Tabelle (**Periodic Snapshot Fact Table**), die als Basistabelle geeignet ist, um die aggregierten Daten in einer „vernünftigen“ Berichtsform auszugeben.

	ZMANDT	ZNO	ZCREATEDBY	ZMODIFIEDBY	ZNAME	ZDATE	ZARTICLE	ZREVENUE
<input type="checkbox"/>	232	7	DEVWU-007		SCHMITZ	10.04.2015	MÖVENSCHIRM	34,00
<input type="checkbox"/>	232	8	DEVWU-007		SCHMITZ	27.11.2014	WÜNSCHELRUTE	92,00
<input type="checkbox"/>	232	9	DEVWU-007		MEIER	06.10.2015	MÖVENSCHIRM	1.018,00
<input type="checkbox"/>	232	10	DEVWU-007		SCHMITZ	26.08.2016	WINDFÄNGER	86,00
<input type="checkbox"/>	232	11	DEVWU-007		SCHMITZ	22.10.2022	WINDFÄNGER	67,00
<input type="checkbox"/>	232	12	DEVWU-007		SCHMITZ	06.01.2024	WÜNSCHELRUTE	19,00
<input type="checkbox"/>	232	13	DEVWU-007		SCHMITZ	15.07.2023	WINDFÄNGER	388,00

Abbildung 7: Exemplarische Datensätze in ZWU\_SALES

Filtern Sie dabei die Daten, sodass Sie nur Datensätze zu den Verkäufern {MEIER, SCHMITZ} und zu den Artikeln {MÖVENSCHIRM, WINDFÄNGER, WÜNSCHELRUTE} aus den Jahren 2021-2023 auswerten. Geben Sie die Faktentabelle aus!

Jahr	NAME	ARTIKEL	Umsatz	ANZAHL VERKAUFE
2021	MEIER	MÖVENSCHIRM	156,00	2
2021	MEIER	WINDFÄNGER	69,00	2
2021	MEIER	WÜNSCHELRUTE	37,00	1
2021	SCHMITZ	MÖVENSCHIRM	61,00	1
2021	SCHMITZ	WINDFÄNGER	403,00	1
2021	SCHMITZ	WÜNSCHELRUTE	42,00	1
2022	MEIER	MÖVENSCHIRM	228,00	4
2022	MEIER	WÜNSCHELRUTE	69,00	1

Abbildung 8: Exemplarisches Ergebnis

Lösung bitte hier.==>

<

SAP

Report

▼

Mehr ▼

Report

Jahr	NAME	ARTIKEL	Umsatz	ANZAHL VERKAUFE
2021	MEIER	WÜNSCHELRUTE	2.174,52	13
2021	Meier	Mövenschirm	3.744,00	7
2021	Meier	Windfänger	3.312,00	10
2021	SCHMITZ	WINDFÄNGER	509,83	6
2021	Schmitz	Windfänger	1.656,00	7
2021	Schmitz	WünscheLrute	2.016,00	7
2022	MEIER	MÖVENSCHIRM	7.701,20	13
2022	Meier	WünscheLrute	1.776,00	7
2022	SCHMITZ	WINDFÄNGER	6.571,20	7
2023	MEIER	MÖVENSCHIRM	6.271,36	8
2023	SCHMITZ	WINDFÄNGER	25.064,00	13
2023	SCHMITZ	WÜNSCHELRUTE	509,83	6

REPORT ZWU\_DEVWU110\_SS25\_A6.

TABLES: ZWU\_SALES.

TYPES: BEGIN OF psft,  
    zdate TYPE zwu\_sales-zdate,  
    zname TYPE zwu\_sales-zname,  
    zarticle TYPE zwu\_sales-zarticle,  
    zrevenue TYPE zwu\_sales-zrevenue,  
    zamount TYPE i,  
    zjahr TYPE c LENGTH 4,  
END OF psft.

DATA: it\_sales TYPE TABLE OF psft,  
      wa\_sales TYPE psft.

*" Selektion mit gewünschten Ergebnissen*

```
SELECT zdate, zname, zarticle, zrevenue
FROM zwu_sales
WHERE UPPER( zname ) IN ('MEIER', 'SCHMITZ')
AND UPPER( zarticle ) IN ('MÖVENSCHIRM', 'WINDFÄNGER', 'WÜNSCHELRUTE')
AND zdate BETWEEN '20210101' AND '20231231'
INTO TABLE @it_sales.
```

*"Das Jahr aus datum extrahieren und pro zeile dann 1 setzen*

```
LOOP AT it_sales INTO wa_sales.
wa_sales-zjahr = wa_sales-zdate+0(4).
wa_sales-zamount = 1.
MODIFY it_sales FROM wa_sales TRANSPORTING zjahr zamount.
ENDLOOP.
```

*\* Aggregiere mit COLLECT nach Jahr, Name, Artikel*

```
DATA: it_agg TYPE TABLE OF psft,
      wa_agg TYPE psft.
" Aggregation wenn jahr name und artikel übereinstimmen
LOOP AT it_sales INTO wa_sales.
CLEAR wa_agg.
wa_agg-zjahr = wa_sales-zjahr.
wa_agg-zname = wa_sales-zname.
wa_agg-zarticle = wa_sales-zarticle.
wa_agg-zrevenue = wa_sales-zrevenue.
wa_agg-zamount = wa_sales-zamount.
COLLECT wa_agg INTO it_agg.
ENDLOOP.
```

*" Sortiere für saubere Ausgabe*

```
SORT it_agg BY zjahr zname zarticle.
```

```
WRITE: / 'Jahr', 6 'NAME', 20 'ARTIKEL', 40 'Umsatz', 60 'ANZAHL VERKAEUFE'.
```

ULINE.

LOOP AT it\_agg INTO wa\_agg.

WRITE: / wa\_agg-zjahr, 6 wa\_agg-zname, 20 wa\_agg-zarticle, 40 wa\_agg-zrevenue, 60 wa\_agg-zamount.

ENDLOOP.

### Aufgabe 7.: Mehrstufige Baukastenstückliste

Gegeben sei die folgende Erzeugnisstruktur **ZWU\_EZS** zur Beschreibung Stücklisten (ZWU\_EZS: ZMANDT, Z\_PARENT, Z\_CHILD, Z\_AMOUNT).

Feld	Key	Initia...	Datenelement	Datentyp	Länge	DezSte...	KoordSystem	Kurzbeschreibung
<input type="checkbox"/> ZMANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0		0 Mandant
<input type="checkbox"/> Z_PARENT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		CHAR	18	0		0 Parent (Oberteil)
<input type="checkbox"/> Z_CHILD	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		CHAR	18	0		0 Child of Parent (Unterteil oder Komponente)
<input type="checkbox"/> Z_AMOUNT	<input type="checkbox"/>	<input type="checkbox"/>		INT4	10	0		0 Number of child instances of a parent

Abbildung 9: Tabelle ZWU\_EZS (Tx-Code: „SE16“)

Mit dem ABAP-Programm **Z\_TBD** und dem ABAP-Funktionsbaustein **Z\_WU\_MYDISPLAYASSEMBLY** sollen die Daten der Tabelle **ZWU\_EZS** verarbeitet werden (Datensätze der Tabelle s.u.).

```
REPORT Z_TBD.
TABLES: zwu_ezs.
PARAMETERS: parent LIKE zwu_ezs-zparent DEFAULT '100-400'.
WRITE: 'zmat', 30 'zmenge'.

CALL FUNCTION 'Z_WU_DISPLAYASSEMBLY'
  EXPORTING
    zmat      = parent
    zdispost  = 0
    zmenge    = 1.
```

Abbildung 10: ABAP-Programm, das einen Funktionsbaustein aufruft

```

1  FUNCTION z_wu_displayassembly.
2  "-----
3  " "Lokale Schnittstelle:
4  " IMPORTING
5  "     VALUE(ZMAT) TYPE CHAR18
6  "     VALUE(ZDISPOST) TYPE INT4
7  "     VALUE(ZMENGE) TYPE INT4
8  "-----
9
10 TABLES: zwu_ezs.
11
12 DATA: ezs      TYPE TABLE OF zwu_ezs,
13        wa_ezs   TYPE zwu_ezs,
14        nstufe   TYPE i.
15
16 DO zdispost TIMES.
17     WRITE: ' '.
18 ENDDO.
19 WRITE: zmat, 30 zmenge.
20
21 nstufe = zdispost + 1.
22 SELECT * FROM zwu_ezs INTO TABLE ezs
23        WHERE zparent = zmat.
24 IF sy-dbcnt > 0.
25     LOOP AT ezs INTO wa_ezs.
26
27         CALL FUNCTION 'Z_WU_DISPLAYASSEMBLY'
28             EXPORTING
29                 zmat      = wa_ezs-zchild
30                 zdispost = nstufe
31                 zmenge   = wa_ezs-zamount.
32
33     ENDLOOP.
34 ENDIF.
35
36 ENDFUNCTION.

```

Tabelle: ZWU\_EZS  
Angezeigte Felder: 4 von 4      Feststehende Führungsspa

	ZMANDT	ZPARENT	ZCHILD	ZAMOUNT
<input type="checkbox"/>	232	100-100	100-110	1
<input type="checkbox"/>	232	100-100	100-120	1
<input type="checkbox"/>	232	100-100	100-130	8
<input type="checkbox"/>	232	100-200	100-210	1
<input type="checkbox"/>	232	100-300	100-310	1
<input type="checkbox"/>	232	100-400	100-410	1
<input type="checkbox"/>	232	100-400	100-420	1
<input type="checkbox"/>	232	100-400	100-430	1
<input type="checkbox"/>	232	100-430	100-431	1
<input type="checkbox"/>	232	100-430	100-432	3
<input type="checkbox"/>	232	100-430	100-433	3
<input type="checkbox"/>	232	100-500	100-510	1
<input type="checkbox"/>	232	P-100	100-100	1
<input type="checkbox"/>	232	P-100	100-130	8
<input type="checkbox"/>	232	P-100	100-200	1
<input type="checkbox"/>	232	P-100	100-300	1
<input type="checkbox"/>	232	P-100	100-400	1
<input type="checkbox"/>	232	P-100	100-500	1
<input type="checkbox"/>	232	P-100	100-600	1

Abbildung 12: Exemplarische Daten in ZWU\_EZS

Beschreiben Sie per **Pseudocode** nachvollziehbar ( → Semantik, also die inhaltliche Bedeutung) den Programmablauf!

Und weisen auch die Testdurchführung aus! Das heißt implementieren Sie das Programm sowie den Funktionsbaustein und führen es in mindestens einem Testfall aus.

**Lösung bitte hier: ==>**

### Programmablauf grob:

Tiefe zuerst. Vom Parent werden die ersten Kinder gesammelt und mit einem Loop über die kinder gegangen und direkt beim ersten kind mit einem Funktionsaufruf der vorgang wiederholt, bis kein Kind mehr vorhanden ist, bevor man zum geschwister Blatt wechselt.

#### 1. Pseudocode Programm

Lese Eingabeparameter parent (hier: Material) Typ ReferenzTabelle ZWU\_EZS ein.  
Rufe Funktion Z\_WU\_DISPLAYASSEMBLY auf Mit Eingabeparemtern materialnummer = parent, zdispost ( Tiefenstufe) = 0 und zmenge(Materialmenge) =1.

Pseudocode Funktion (rekursiv):

FUNCTION Z\_WU\_DISPLAYASSEMBLY (material, stufe, menge):

Create interne Tabelle ezs TYP Tabelle ZWU\_EZS, Array (Working Area) wa\_ezs vom TYP ZWU\_EZS und integer Variable nstufe.

Schleife (für visuelle Einrückung für jede Tiefe ein Punkt mehr), für lenght von zdispost Write „  
.“.



ENDE SCHLEIFE.

Write Materialnummer und Menge in Spalte 30.

Erhöhe variable dispost(Tiefe) um 1 und weise Sie der variable nstufe zu.

(Selektiere alle Kinder und sicher alle Datensätze in interne Tabelle)

Ezs (itab) = Select ALL Records FROM TABELLE ZWU\_EZS

BEDINGUNG parent = material.

Wenn Datensätze(Kinderknoten) gefunden IS not Empty

Dann schleife über ezs (itab) schreibe in wa\_ezs

For each Datensatz (Kind) DO:

(Rekursion)

Rufe Funktion FUNCTION Z\_WU\_DISPLAYASSEMBLY mit param materialnummer = wa\_ezs-child, param Stufe = variable nstufe (wurde um 1 erhöht) und param zmenge= wa\_ezs-amount (Menge des Kindes) für jede Zeile der internen Tabelle ezs.

ENDE SCHLEIFE.

ENDE WENN.

ENDE FUNKTION.

#### Mehrstufige Baukastenstückliste

 Als Variante sichern... Mehr 

PARENT:

### Mehrstufige Baukastenstückliste

zmat	zmenge
100-400	1
. 100-410	1
. 100-420	1
. 100-430	1
. . 100-431	1
. . 100-432	3
. . 100-433	3

zmat                      zmenge

zmat	zmenge
> 100	1
. . 100-100	1
. . . 100-110	1
. . . 100-120	1
. . . 100-130	8
. 100-130	8
. 100-200	1
. . . 100-210	1
. 100-300	1
. . . 100-310	1
. 100-400	1
. . . 100-410	1
. . . 100-420	1
. . . 100-430	1
. . . . 100-431	1