

**Klausur am 6. Februar 2016**  
**Anwendungsentwicklung in ERP-Systemen**  
(*B\_Inf14.0 B098a, IAW6.5 552, ITAS1.0 114, ITAW1.0 211, KAI1.0 213 – 120 Minuten*)

### Aufgabe 1

Forme folgendes Programmfragment in eine syntaktisch korrekte Fassung um:

```
DATA X TYPE F.  
DATA Y TYPE F.  
  
IF ( X + 2 ) = ( Y - 2 ).  
    WRITE 'MOIN MOIN'.  
ENDIF.
```

Welche Werte beinhalten die Variablen *MyString1*, *MyString2*, *MyString3* und *MyString4* nach Ausführung des folgenden Programmfragments (nachfolgende Leerzeichen in den Ergebnissen bleiben unberücksichtigt) ?

```
DATA myString1(42) TYPE C VALUE 'ERP-Software und ERP-Systeme'.  
DATA myString2(42) TYPE C VALUE 'ERP-Software und ERP-Systeme'.  
DATA myString3(4) TYPE C.  
DATA myString4(4) TYPE C.  
  
DATA i TYPE I VALUE 17.  
DATA j TYPE I VALUE 3.  
  
myString1+i(j) = 'SAP'.  
  
myString2+j(i) = '-Systeme und SAP'.  
  
myString3 = i.  
  
j = i + myString3.  
  
myString4 = j.
```

## Aufgabe 2

Entwickle jeweils ein ABAP-Codefragment (beinhaltend Datendeklaration und -verarbeitung) für nachfolgende vier Datentypen. Dabei sind folgende charakteristische Verwendungen zu berücksichtigen: Anzahl Minuten seit Mitternacht, Anzahl Stunden zwischen zwei Kalenderdaten, Summe zweier Geldbeträge und Zählschleife für genau zehn Durchläufe (pro Durchlauf Ausgabe von „Hello World“).

1. D

2. I

3. P

4. T

### **Aufgabe 3**

Ordne nachfolgenden Aussagen bzw. Funktionen den jeweils am ehesten passenden der folgenden achtzehn SAP-Begriffe zu: ABAP Editor, ALV-Grid, BAPI, Berechtigungsprofil, Class Builder, Debugger, Function Builder, Menu Painter, Nummernkreisobjekt, Object Navigator, Paket, Rolle, Screen Painter, Sperrobjekt, Suchhilfe, Transportauftrag, User-Exit und Verbuchungsbaustein:

1. Dient dazu, ein Menü auszuwählen und dazu passend ein Berechtigungsprofil zu erzeugen.
2. Element des Berechtigungssystems, gewährt den Benutzern Zugriff auf das System.
3. Benutzerfreundliche Oberfläche, die alle Entwicklungswerkzeuge intuitiv integriert.
4. Nach betriebswirtschaftlicher Sichtweise gekapselte Funktionsbausteine.
5. Erstellung und Pflege von Dynpros.
6. Erstellung und Pflege von Menüs, Überschriften und Symbolleisten in ABAP-Programmen.
7. Oberflächenelement, mit dem tabellarische Daten in Anwendungen angezeigt werden können.
8. Objekt des ABAP Dictionary, mit dem Eingabehilfen (F4-Hilfen) definiert werden können.
9. Synchronisation des gleichzeitigen Zugriffs zweier Benutzer auf denselben Datenbestand.
10. Zeitpunkt im SAP-Programm, zu dem ein kundeneigener Programmteil aufgerufen werden kann.

## **Aufgabe 4**

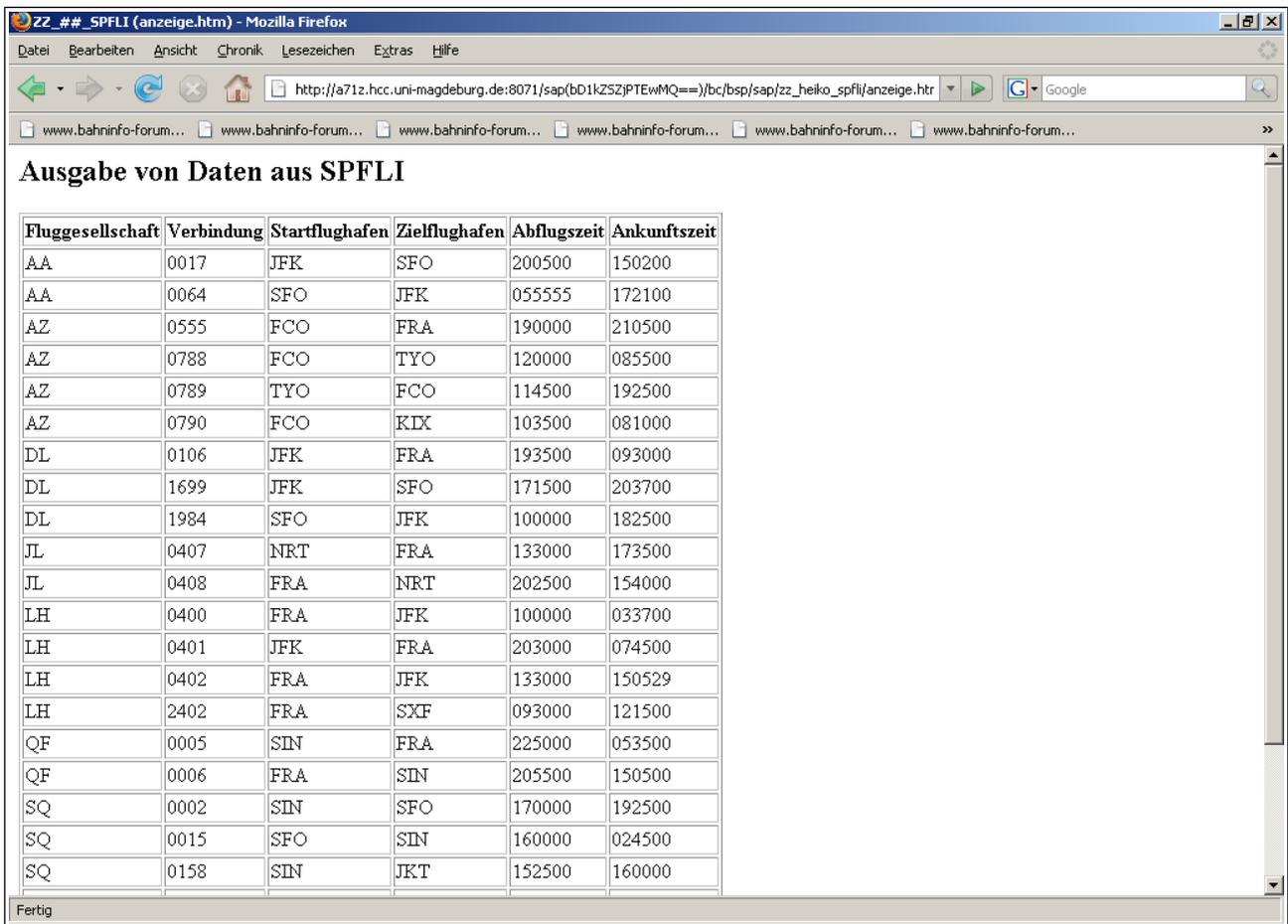
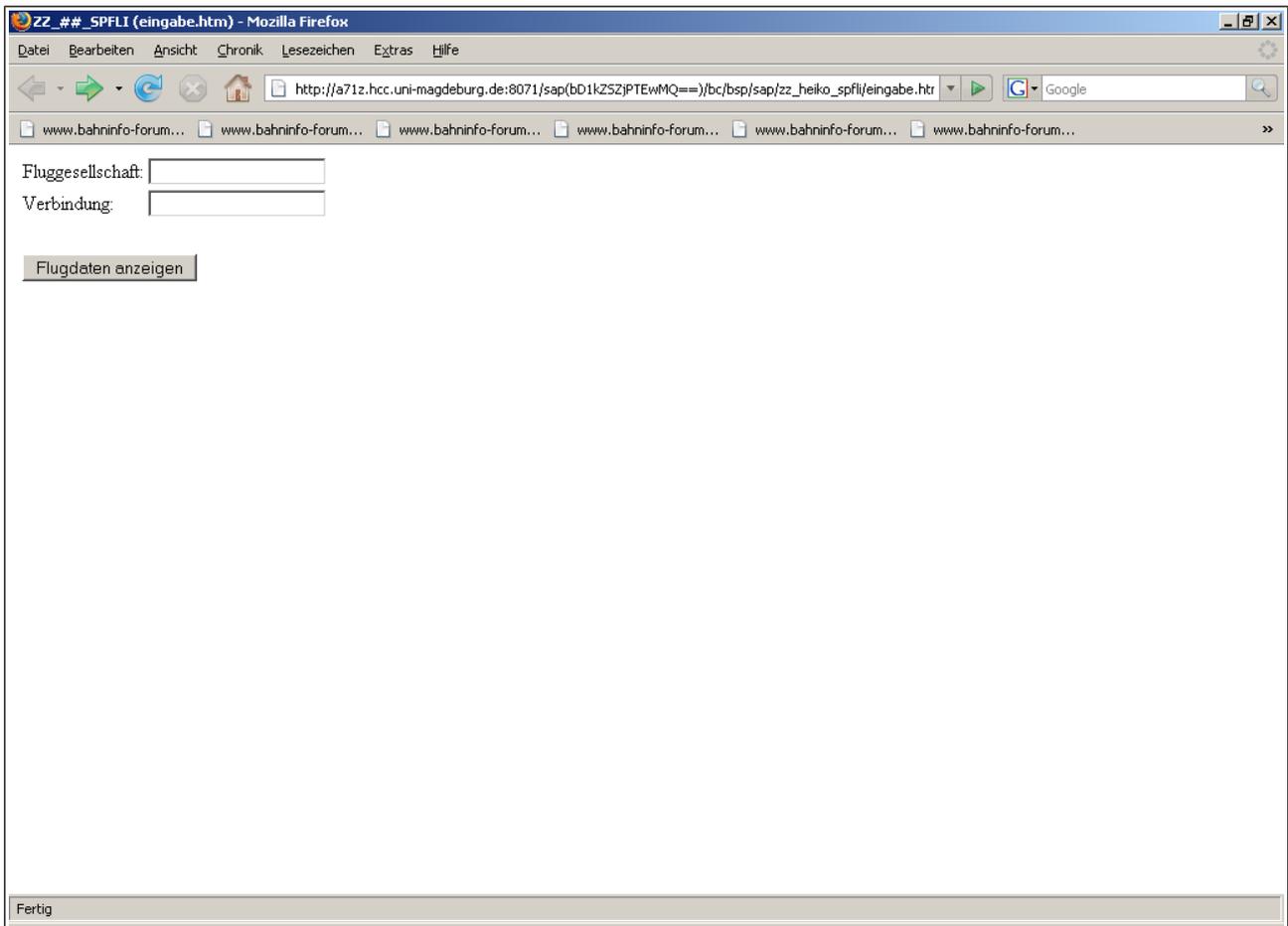
Entwickle zunächst einen sich rekursiv aufrufenden Funktionsbaustein zur Berechnung der Fakultät basierend auf den beiden Grundrechenarten Addition und Multiplikation sowie anschließend einen Report, der eine natürliche Zahl einliest und den mit dem vorstehenden Funktionsbaustein berechneten Fakultätswert ausgibt.

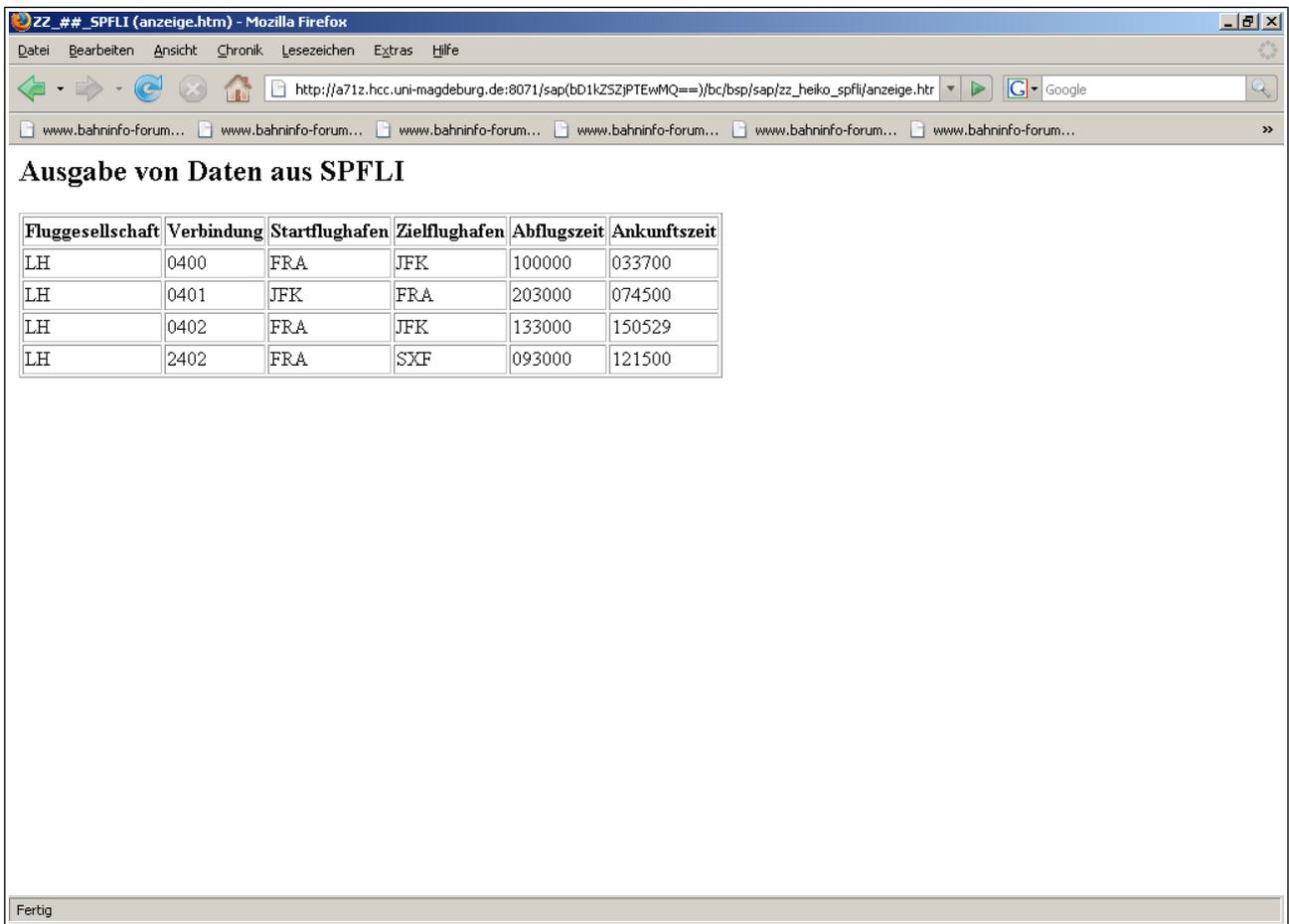
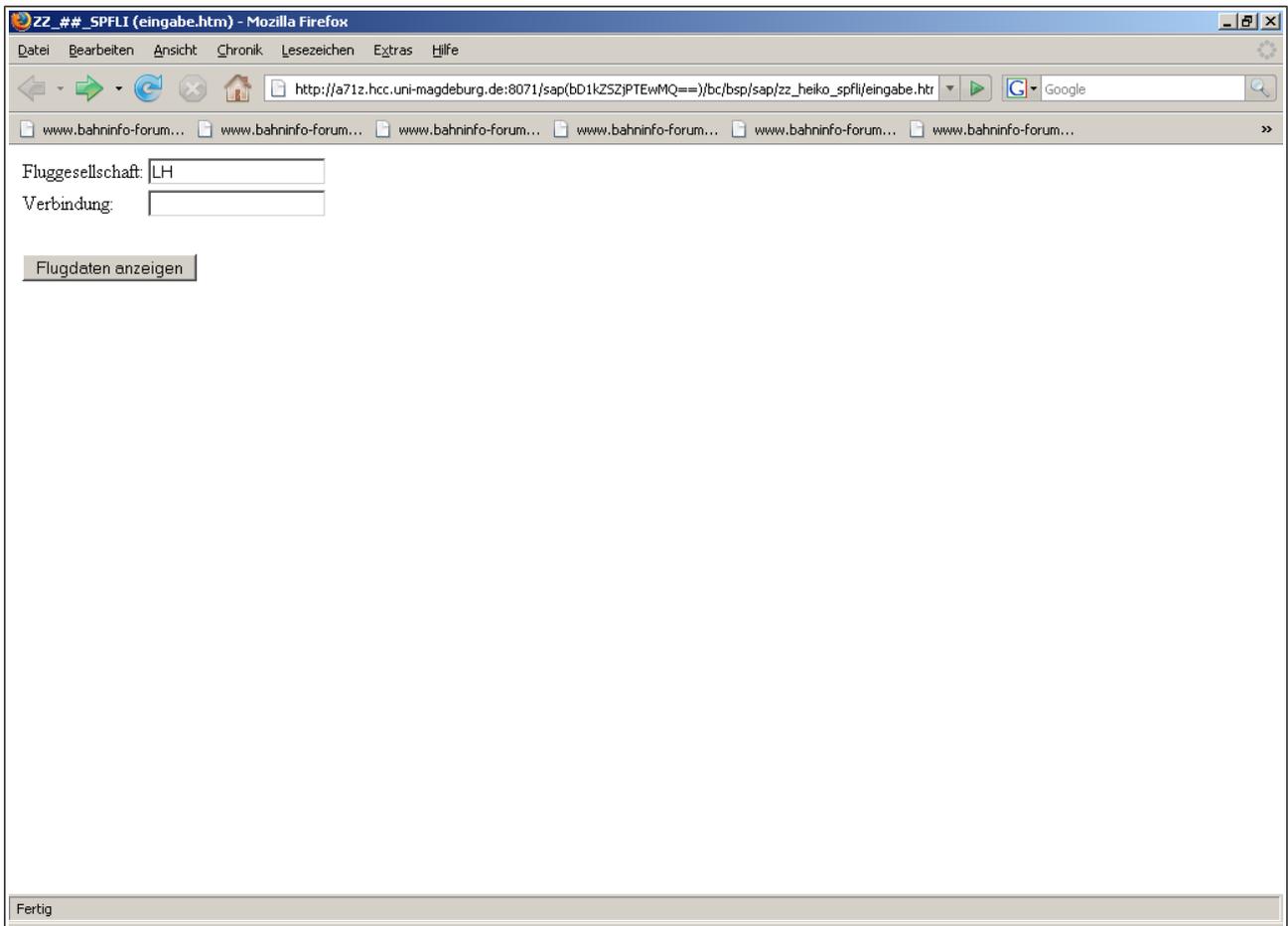
## **Aufgabe 5**

Gegeben ist eine Business Server Pages (BSP) - Applikation zur Auswahl und Anzeige von Flugverbindungen, die durch nachfolgende Bildschirmbilder beschrieben wird.

Gesucht ist das Layout der Seiten „eingabe.htm“ und „ausgabe.htm“ (in den nachfolgenden Bildschirmbildern der Transaktion „SE80“ jeweils als weiße Fläche dargestellt).

Sofern die exakte HTML-Syntax nicht bekannt ist, sind entsprechend gekennzeichnete Annahmen zu treffen.





ZZ\_##\_SPFLI (eingabe.htm) - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://a712.hcc.uni-magdeburg.de:8071/sap(bD1k2SZjPTEwMQ==)/bc/bsp/sap/zz\_heiko\_spfli/eingabe.htm

www.bahninfo-forum... www.bahninfo-forum... www.bahninfo-forum... www.bahninfo-forum... www.bahninfo-forum... www.bahninfo-forum...

Fluggesellschaft:

Verbindung:

Fertig

ZZ\_##\_SPFLI (anzeige.htm) - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://a712.hcc.uni-magdeburg.de:8071/sap(bD1k2SZjPTEwMQ==)/bc/bsp/sap/zz\_heiko\_spfli/anzeige.htm

www.bahninfo-forum... www.bahninfo-forum... www.bahninfo-forum... www.bahninfo-forum... www.bahninfo-forum... www.bahninfo-forum...

### Ausgabe von Daten aus SPFLI

Fluggesellschaft	Verbindung	Startflughafen	Zielflughafen	Abflugszeit	Ankunftszeit
LH	0400	FRA	JFK	100000	033700

Fertig





*Hier ist Platz für das Layout der Seite „eingabe.htm“*

*Hier ist Platz für das Layout der Seite „anzeige.htm“*

## Aufgabe 6

Gegeben sei im Data Dictionary die Datenbanktabelle ZZ\_PERSONEN („Mitarbeiter“):

- PID: Mitarbeiteridentifikation (6-stellig alphanumerisch, Primärschlüssel)
- VORNAME: Vorname des Mitarbeiters (30-stellig alphanumerisch)
- NACHNAME: Nachname des Mitarbeiters (30-stellig alphanumerisch)
- TITEL: Titel des Mitarbeiters (15-stellig alphanumerisch, Fremdschlüssel)

Entwickelt wurde eine Transaktion zur auf Anzeigen und Ändern beschränkte Stammdatenpflege, die einerseits auf der obigen Datenbanktabelle basiert, und die andererseits zwei Dynpros („Auswahl“ über den Primärschlüssel und „Anzeigen/Ändern“ ohne den Primärschlüssel) beinhaltet. Spezifiziert ist der ABAP-Code der einzelnen Module (Include-Dateien) inklusive dem TOP-Include, sowie der Ablauflogik-Code der beiden Dynpros. Auf dem „Auswahl“-Dynpro wird zusätzlich ein Table-Control zum Markieren genau eines Mitarbeiters verwendet, wobei das spezielle Ereignis „Doppelklick in eine Tabellenzeile“ ebenfalls berücksichtigt werden soll.

***Erläutere mit eigenen Worten stichwortartig in den grauen Feldern möglichst exakt die Bedeutung und notwendige Verwendung der jeweils vorstehenden ABAP-Anweisung.***

### **Ablauflogik Dynpro 0100 („Auswahl“)**

PROCESS BEFORE OUTPUT.

MODULE status\_0100.

LOOP WITH CONTROL personen.

MODULE fill\_table\_control.

ENDLOOP.

PROCESS AFTER INPUT.

LOOP WITH CONTROL personen.

MODULE read\_table\_control.

ENDLOOP.

MODULE user\_command\_0100.

### **Ablauflogik Dynpro 0200 („Anzeigen/Ändern“)**

PROCESS BEFORE OUTPUT.

MODULE status\_0200.

PROCESS AFTER INPUT.

MODULE user\_command\_0200.

**TOP-Include**

PROGRAM zz\_dynpro.

CONTROLS personen TYPE TABLEVIEW USING SCREEN 100.

TABLES zz\_personen.

TABLES zz\_personen\_t.

DATA: ok\_code LIKE sy-ucomm.

DATA: wa\_person TYPE zz\_personen.

DATA: itab TYPE TABLE OF zz\_personen\_t.

**Include zu MODULE status\_0100**

```
MODULE status_0100 OUTPUT.
```

```
SET PF-STATUS '100'.
```

```
SET TITLEBAR '100'.
```

```
CLEAR ok_code.
```

```
SELECT * FROM zz_personen INTO CORRESPONDING FIELDS OF TABLE itab.
```

```
DESCRIBE TABLE itab LINES personen-lines.
```

```
ENDMODULE.
```

**Include zu MODULE user\_command\_0100**

MODULE user\_command\_0100 INPUT.

CASE ok\_code.

WHEN 'BACK'.

LEAVE PROGRAM.

WHEN 'SELECT' OR 'PICK'.

SELECT SINGLE \* FROM zz\_personen INTO wa\_person WHERE pid = zz\_personen-pid.

IF sy-dbcnt = 1.

LEAVE TO SCREEN 200.

ENDIF.

ENDCASE.

ENDMODULE.

**Include zu MODULE status\_0200**

MODULE status\_0200 OUTPUT.

SET PF-STATUS '200'.

SET TITLEBAR '200'.

CLEAR ok\_code.

zz\_personen = wa\_person.

ENDMODULE.

**Include zu MODULE user\_command\_0200**

MODULE user\_command\_0200 INPUT.

CASE ok\_code.

WHEN 'LEAVE'.

LEAVE PROGRAM.

WHEN 'BACK'.

LEAVE TO SCREEN 100.

WHEN 'SAVE'.

MODIFY zz\_personen FROM zz\_personen.

ENDCASE.

wa\_person = zz\_personen.

ENDMODULE.

**Include zu MODULE fill\_table\_control**

```
MODULE fill_table_control OUTPUT.
```

```
  READ TABLE itab INTO zz_personen_t INDEX personen-current_line.
```

```
ENDMODULE.
```

**Include zu MODULE read\_table\_control**

```
MODULE read_table_control INPUT.
```

```
  DATA cursorline TYPE i.
```

```
  GET CURSOR LINE cursorline.
```

```
  IF sy-subrc = 0.
```

```
    cursorline = personen-top_line + cursorline - 1.
```

```
  ELSE.
```

```
    cursorline = 0.
```

```
  ENDIF.
```

```
  IF ( zz_personen_t-selection = 'X' ) OR ( personen-current_line = cursorline ).
```

```
    zz_personen-pid = zz_personen_t-pid.
```

```
  ENDIF.
```

```
ENDMODULE.
```

***Das Team der Fachhochschule Wedel und PTL Wedel wünscht viel Erfolg***