

Klausur ERP-Systeme und ERP 2 am 13. Februar 2009

(B_Inf1.0/2.0 v361, B_Winf1.0 v362, IAW3.2 552, WI12.0 h916 – Dauer: 105 Minuten)

Das FH-/PTL-Team wünscht viel Erfolg

Aufgabe 1

Entwickle einen **objektorientierten Report**, der in einer ersten Schleife für jeden Eintrag der Datenbanktabelle SPFLI – beschränkt auf die Felder Fluggesellschaft (CARRID), Flugnummer/Verbindung (CONNID), Abflugstadt (CITYFROM) und Ankunftsstadt (CITYTO) – ein eigenes Objekt erstellt dessen Referenz jeweils an eine interne Tabelle angehängt wird, und der in einer nachfolgenden zweiten Schleife die Inhalte der zuvor erstellten Objekte – ohne aufwändige Formatierung – ausgibt.

Hier ist weiterer Platz zur Bearbeitung der ersten Aufgabe

Aufgabe 2

Entwickle eine **Transaktion** zur Pflege (beschränkt auf die Funktionen Anzeigen und Ändern sowie der Felder Abflugstadt *CITYFROM* und Ankunftsstadt *CITYTO*) von Inhalten der Datenbanktabelle SPFLI, die sich über zwei **Dynpros** erstreckt, wobei das Erste dem Selektieren (über die Felder Fluggesellschaft *CARRID* und Flugnummer/Verbindung *CONNID*), und das Zweite dem „eigentlichen“ Anzeigen und Ändern dient. Zur Komfortsteigerung der BenutzerInnen soll das erste Dynpro auch eine Selektionstabelle (**Table Control**) beinhalten.

Eure konkrete Aufgabe ist es, die nachfolgenden Codefragmente des Programms ZZ_DYNPRO sowie der sechs Module STATUS_0100, USER_COMMAND_0100, STATUS_0200, USER_COMMAND_0200, FILL_TABLE_CONTROL und READ_TABLE_CONTROL zu ergänzen.

Die Selektionstabellenzeile TC_SPFLI entspricht den Feldern CARRID (Fluggesellschaft), CONNID (Flugnummer/Verbindung), CITYFROM (Abflugstadt) und CITYTO (Ankunftsstadt) aus der Datenbanktabelle SPFLI ergänzt um ein vorangestelltes „Ein-Zeichen“-Feld SELECTION.

Das Dynpro 0100 („Selektieren“) soll den GUI-Status 100 verwenden, welcher „BACK“ oder „SELECT“ als „OK-Code“ liefert. Diese beiden Codes entsprechen der Funktionalität des ersten Dynpros. Wird das Bildschirmlayout mit der „Enter-Taste“ verlassen, erfolgt keine Rückgabe eines neuen „OK-Code“, und dementsprechend soll keine Funktionalität ausgelöst werden.

Das Dynpro 0200 („Anzeigen/Ändern“) soll den GUI-Status 200 verwenden, welcher „BACK“, „LEAVE“ oder „SAVE“ als „OK-Code“ liefert. Diese drei Codes entsprechen der Funktionalität des zweiten Dynpros. Wird das Bildschirmlayout mit der „Enter-Taste“ verlassen, erfolgt keine Rückgabe eines neuen „OK-Code“, und dementsprechend soll keine Funktionalität ausgelöst werden.

Ablauflogik Dynpro 100

PROCESS BEFORE OUTPUT.

MODULE **status_0100**.

LOOP WITH CONTROL flights.

MODULE **fill_table_control**.
ENDLOOP.

PROCESS AFTER INPUT.

LOOP WITH CONTROL flights.

MODULE **read_table_control**.
ENDLOOP.

MODULE **user_command_0100**.

Ablauflogik Dynpro 200

PROCESS BEFORE OUTPUT.

MODULE **status_0200**.

PROCESS AFTER INPUT.

MODULE **user_command_0200**.

PROGRAM zz_dynpro MESSAGE-ID zz_dynpro. **# bitte ergänzen**

CONTROLS flights TYPE TABLEVIEW USING SCREEN 100.

TABLES spfli.

TABLES tc_spfli.

DATA: ok_code LIKE sy-ucomm.

DATA: wa_spfli TYPE spfli.

MODULE status_0100 OUTPUT. # *bitte ergänzen*

ENDMODULE .

MODULE user_command_0100 INPUT. *# bitte ergänzen*

CALL FUNCTION 'ENQUEUE_EZ_SPFLI'
EXPORTING
 mode_spfli = 'X'
 carrid = wa_spfli-carrid
 connid = wa_spfli-connid
EXCEPTIONS
 foreign_lock = 1.

ENDMODULE.

MODULE status_0200 OUTPUT. # *bitte ergänzen*

ENDMODULE.

MODULE user_command_0200 INPUT. # bitte ergänzen

CALL FUNCTION 'DEQUEUE_EZ_SPFLI'
EXPORTING
mode_spfli = 'X'
carrid = wa_spfli-carrid
connid = wa_spfli-connid.

ENDMODULE .

MODULE fill_table_control OUTPUT. # bitte ergänzen

ENDMODULE.

MODULE read_table_control INPUT. # bitte ergänzen

ENDMODULE.

Aufgabe 3

Entwickle eine **Business Server Pages (BSP) - Applikation** basierend auf der Datenbanktabelle SPFLI zur Auswahl (Fluggesellschaft *CARRID* und Flugnummer/Verbindung *CONNID*) sowie Anzeige (Fluggesellschaft *CARRID*, Flugnummer/Verbindung *CONNID*, Startflughafen *AIRPFROM*, Zielflughafen *AIRPTO*, Abflugszeit *DEPTIME* und Ankunftszeit *ARRTIME*) von Flugverbindungen.

Die Anzeige der Flugverbindungen soll berücksichtigen, das bei der Auswahl der Flugverbindungen entweder kein Kriterium, oder nur das erste Kriterium (Fluggesellschaft), oder nur das zweite Kriterium (Flugnummer/Verbindung) oder beide Kriterien eingegeben wird bzw. werden.

Eure konkrete Aufgabe ist es, die nachfolgenden drei Fragmente zu ergänzen.

Layout der Seite *auswahl.htm*

```
<%@page language="abap" %>
```

```
<INPUT TYPE="text" NAME="param_carrid">
```

```
<INPUT TYPE="text" NAME="param_connid">
```

```
<INPUT TYPE="submit" NAME="onInputProcessing(myEvent) "  
        VALUE="Flugverbindungen anzeigen">
```

ABAP-Code des Eventhandlers *onInputProcessing*

```
CASE event_id.
```

```
ENDCASE.
```

Layout der Seite *anzeige.htm*

```
<%@page language="abap" %>
```