

Klausur am 15. Februar 2018

Anwendungsentwicklung in ERP-Systemen

(B_Inf14.0 B098a, B_Winf14.0 B098a, ÜB Bß98a, ITAS2.0 209, ITAW2.0 212, KAI2.0 213 – 120 Minuten)

Aufgabe 1

1. Kennzeichne die <i>richtige</i> Behauptung (1 Kreuz von 4 Möglichkeiten)	
ABAP richtet sich weder an die Wiederverwendbarkeit von Code- noch von Datenobjekten aus	<input type="checkbox"/>
ABAP richtet sich stark an die Wiederverwendbarkeit von Codeobjekten aus	<input type="checkbox"/>
ABAP richtet sich stark an die Wiederverwendbarkeit von Datenobjekten aus	<input type="checkbox"/>
ABAP richtet sich stark an die Wiederverwendbarkeit von Code- und Datenobjekten aus	<input type="checkbox"/>

2. Kennzeichne die <i>richtigen</i> Behauptungen (2 Kreuze von 4 Möglichkeiten)	
ABAP-Programme liegen nur in kompilierter Form vor	<input type="checkbox"/>
ABAP-Programme liegen auch als Quelltext vor	<input type="checkbox"/>
ABAP-Programme werden beim ersten Aufruf kompiliert	<input type="checkbox"/>
ABAP-Programme werden bei jedem Aufruf kompiliert	<input type="checkbox"/>

3. Kennzeichne die <i>richtigen</i> Behauptungen (2 Kreuze von 4 Möglichkeiten)	
Das Repository ist mandantenunabhängig	<input type="checkbox"/>
Das Repository ist mandantenabhängig	<input type="checkbox"/>
Die Anwendungsdaten sind mandantenunabhängig	<input type="checkbox"/>
Die Anwendungsdaten sind mandantenabhängig	<input type="checkbox"/>

4. Transporte in einer typischen SAP-Systemlandschaft erfolgen:
(2 Kreuze von 6 Möglichkeiten)

Von Development System nach Production System	<input type="checkbox"/>
Von Development System nach Quality Assurance System	<input type="checkbox"/>
Von Production System nach Development System	<input type="checkbox"/>
Von Production System nach Quality Assurance System	<input type="checkbox"/>
Von Quality Assurance System nach Development System	<input type="checkbox"/>
Von Quality Assurance System nach Production System	<input type="checkbox"/>

5. Kennzeichne die *richtige* Behauptung
(1 Kreuz von 3 Möglichkeiten)

In ABAP ist die Konvertierung von String nach Integer nicht möglich	<input type="checkbox"/>
In ABAP ist die Konvertierung von String nach Integer in Abhängigkeit vom Inhalt möglich	<input type="checkbox"/>
In ABAP ist die Konvertierung von String nach Integer generell möglich	<input type="checkbox"/>

6. Markiere die Datentypen im Sinne des ABAP Dictionarys
(3 Kreuze von 9 Möglichkeiten) (*doppelt gewertet*)

Datenbanktabelle	<input type="checkbox"/>
Datenelement	<input type="checkbox"/>
Domäne	<input type="checkbox"/>
Sperrobjekt	<input type="checkbox"/>
Struktur	<input type="checkbox"/>
Suchhilfe	<input type="checkbox"/>
Tabellentyp	<input type="checkbox"/>
Typgruppe	<input type="checkbox"/>
View	<input type="checkbox"/>

7. Kennzeichne die *richtigen* Behauptungen
(2 Kreuze von 4 Möglichkeiten)

Der Inhalt von Datenbanktabellen steht nicht über die Programmlaufzeit hinaus zur Verfügung	<input type="checkbox"/>
Der Inhalt von Datenbanktabellen steht über die Programmlaufzeit hinaus zur Verfügung	<input type="checkbox"/>
Der Inhalt von internen Tabellen steht nicht über die Programmlaufzeit hinaus zur Verfügung	<input type="checkbox"/>
Der Inhalt von internen Tabellen steht über die Programmlaufzeit hinaus zur Verfügung	<input type="checkbox"/>

8. Kennzeichne die *richtigen* Behauptungen

(2 Kreuze von 4 Möglichkeiten)

Interne Tabellen haben eine feste Länge

Interne Tabellen können prinzipiell beliebig lang sein

Interne Tabellen können nicht aus internen Tabellen bestehen

Interne Tabellen können auch aus internen Tabellen bestehen

9. Kennzeichne die *richtigen* Behauptungen

(3 Kreuze von 6 Möglichkeiten) (*doppelt gewertet*)

Unterprogramme können nicht auf übergeordnete ("globale") Variablen des Programms zugreifen.

Innerhalb von Unterprogrammen können keine lokalen Variablen definiert werden.

Der rekursive Aufruf von Unterprogrammen ist nicht möglich

Unterprogramme können auf übergeordnete ("globale") Variablen des Programms zugreifen.

Innerhalb von Unterprogrammen können lokale Variablen definiert werden.

Der rekursive Aufruf von Unterprogrammen ist möglich.

10. Markiere aus der Sicht des Funktionsbausteins die *richtigen* Aussagen zu Funktionsbausteinen:

(3 Kreuze von 6 Möglichkeiten) (*doppelt gewertet*)

Kapselung von Quellcode.

Funktionsbausteine können auf übergeordnete Variablen zugreifen.

Organisation von Funktionsbausteinen in Funktionsgruppen.

Import-Parameter: Dies sind die Rückgabeparameter des Funktionsbausteins.

Export-Parameter: Es erfolgt die Angabe der Eingabeparameter des Funktionsbausteins.

Changing-Parameter: Es handelt sich um Parameter, die gleichzeitig als Import- und Export-Parameter dienen.

11. Objekt des ABAP Dictionary, mit dem Eingabehilfen (F4-Hilfen) definiert werden können.

(1 Kreuz von 6 Möglichkeiten)

Berechtigungsprofil

Nummernkreisobjekt

Sperrobjekt

Suchhilfe

User-Exit

Verbuchungsbaustein

12. Synchronisation des gleichzeitigen Zugriffs zweier Benutzer auf denselben Datenbestand durch:
(1 Kreuz von 6 Möglichkeiten)

Berechtigungsprofil	<input type="checkbox"/>
Nummernkreisobjekt	<input type="checkbox"/>
Sperrojekt	<input type="checkbox"/>
Suchhilfe	<input type="checkbox"/>
User-Exit	<input type="checkbox"/>
Verbuchungsbaustein	<input type="checkbox"/>

13. Zeitpunkt im SAP-Programm, zu dem ein kundeneigener Programmteil aufgerufen werden kann:
(1 Kreuz von 6 Möglichkeiten)

Berechtigungsprofil	<input type="checkbox"/>
Nummernkreisobjekt	<input type="checkbox"/>
Sperrojekt	<input type="checkbox"/>
Suchhilfe	<input type="checkbox"/>
User-Exit	<input type="checkbox"/>
Verbuchungsbaustein	<input type="checkbox"/>

Aufgabe 2

Entwickle zunächst einen sich rekursiv aufrufenden Funktionsbaustein (inklusive Spezifikation der Schnittstelle) zur Berechnung der Potenz zweier natürlicher Zahlen basierend auf den beiden Grundrechenarten Addition (bzw. Subtraktion) und Multiplikation sowie anschließend einen Report, der zwei natürliche Zahl einliest und den mit dem vorstehenden Funktionsbaustein berechneten Potenzwert ausgibt.

Aufgabe 2 (Fort.)

Aufgabe 3

Erläutere stichwortartig die Funktionsweise - der durch FettKursivDruck markierten – Befehle des nachfolgenden PAI-Moduls:

```
MODULE USER_COMMAND_0100 INPUT.

CASE ok_code.
  WHEN 'BACK'.
    LEAVE PROGRAM.
  WHEN 'SELECT'.
    SELECT SINGLE * FROM spfli INTO wa_flug WHERE carrid = spfli-carrid AND
                                     connid = spfli-connid.
    IF sy-dbcnt = 1.
      CALL FUNCTION 'ENQUEUE_EZ_SPFLI'
        EXPORTING
          mode_spfli = 'E'
          carrid      = spfli-carrid
          connid      = spfli-connid
        EXCEPTIONS
          FOREIGN_LOCK = 1.
      IF sy-subrc = 0.
        MESSAGE S010 WITH wa_flug-carrid wa_flug-connid.
        LEAVE TO SCREEN 200.
      ELSE.
        MESSAGE S011 WITH wa_flug-carrid wa_flug-connid DISPLAY LIKE 'E'.
      ENDIF.
    ENDIF.
  ENDCASE.
  CLEAR ok_code.

ENDMODULE.
```

Aufgabe 3 (Fort.)

Erläutere stichwortartig die Funktionsweise - der durch FettKursivDruck markierten – Befehle des nachfolgenden PAI-Moduls:

```
MODULE USER_COMMAND_0200 INPUT.  
  
CASE ok_code.  
  WHEN 'LEAVE'.  
    LEAVE PROGRAM.  
  WHEN 'BACK'.  
    CALL FUNCTION 'DEQUEUE_EZ_SPFLI'  
      EXPORTING  
        mode_spfli = 'E'  
        carrid      = spfli-carrid  
        connid     = spfli-connid.  
    LEAVE TO SCREEN 100.  
  WHEN 'SAVE'.  
    AUTHORITY-CHECK OBJECT 'ZZ00SPFLI'  
      ID 'ACTVT' FIELD '03'.  
    IF SY-SUBRC = 0.  
      MODIFY spfli FROM spfli.  
    ELSE.  
      MESSAGE S013 DISPLAY LIKE 'E'.  
    ENDIF.  
ENDCASE.  
CLEAR ok_code.  
  
wa_flug = spfli.  
  
ENDMODULE.
```


Aufgabe 4

Skizziere und Beschrifte (sofern erforderlich Erläutere zusätzlich stichwortartig) den Zusammenhang zwischen folgenden Komponenten einer Transaktion zur Auswahl eines Eintrages einer Datenbanktabelle mittels Table-Control. Die Betrachtung möge enden bei dem Sprung zu einem zweiten Dynpro, das den ausgewählten Datenbanktabelleneintrag anzeigt.

- Datenbanktabelle
- Dynpro Ablauflogik
- Dynpro Layout (insbesondere Table-Control)
- Programm PAI-Module
- Programm PBO-Module
- Programm TOP-Include (insbesondere interne Tabelle)
- Transaktionscode

Aufgabe 4 (Fort.)

Aufgabe 5

Gebe die exakte Ausgabe des nachfolgenden Reports an:

```
REPORT zz_objects.

CLASS train DEFINITION.
  PUBLIC SECTION.
    METHODS: setter IMPORTING
              imp_name TYPE string,
              getter.
  PRIVATE SECTION.
    DATA: name TYPE string.
ENDCLASS.

CLASS train IMPLEMENTATION.
  METHOD setter.
    name = imp_name.
  ENDMETHOD.
  METHOD getter.
    WRITE: / name.
  ENDMETHOD.
ENDCLASS.

DATA: name    TYPE string VALUE '0',
      train   TYPE REF TO train,
      trains  TYPE TABLE OF REF TO train.

START-OF-SELECTION.

  WHILE strlen( name ) < 10.
    CREATE OBJECT train.
    CONCATENATE 'X' name INTO name SEPARATED BY '-'.
    train->setter( imp_name = name ).
    INSERT train INTO trains INDEX 1.
  ENDWHILE.
  LOOP AT trains INTO train.
    train->getter( ).
  ENDLOOP.
```

Aufgabe 6

Skizziere und Beschrifte (sofern erforderlich Erläutere zusätzlich stichwortartig) den Zusammenhang zwischen folgenden Komponenten in Seiten mit Ablauflogik einer BSP-Applikation, die über eine erste HTML-Seite Selektionskriterien einliest, und über eine zweite HTML-Seite die per Funktionsbaustein selektierten Zeilen einer internen Tabelle ausgibt:

- Eventhandler: OnInputProcessing
- Layout: ABAP-Code
- Layout: HTML-Drucktaste („Button“)
- Layout: HTML-Formular
- Layout: HTML-InputField („Eingabefeld“)
- Layout: HTML-Table („Tabelle“)
- Seitenattribute

Aufgabe 6 (Fort.)

Das Team der Berufsfachschule und Fachhochschule Wedel wünscht viel Erfolg