

Klausur am 11. Februar 2019

Anwendungsentwicklung in ERP-Systemen

(B_IMCA16.0 B098a, B_Inf14.0 B098a, B_Winf14.0 B098a, ITAS2.0 209, KAI2.0 213 – 120 Minuten)

Aufgabe 1

1. Kennzeichne die <i>richtigen</i> Behauptungen (2 Kreuze von 4 Möglichkeiten)	
ABAP richtet sich an die Wiederverwendbarkeit von Codeobjekten aus	
ABAP richtet sich an die Wiederverwendbarkeit von Datenobjekten aus	
ABAP richtet sich nicht an die Wiederverwendbarkeit von Codeobjekten aus	
ABAP richtet sich nicht an die Wiederverwendbarkeit von Datenobjekten aus	

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

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

4. Kennzeichne die <i>richtigen</i> Behauptungen (2 Kreuze von 6 Möglichkeiten)	
In ABAP ist die Konvertierung von String nach Integer nicht möglich	
In ABAP ist die Konvertierung von String nach Integer generell möglich	
In ABAP ist die Konvertierung von String nach Integer in Abhängigkeit vom Inhalt möglich	
In ABAP ist die Konvertierung von Gleitpunktzahl nach Integer nicht möglich	
In ABAP ist die Konvertierung von Gleitpunktzahl nach Integer generell möglich	
In ABAP ist die Konvertierung von Gleitpunktzahl nach Integer in Abhängigkeit vom Inhalt möglich	

5. 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"/>

6. Markiere die Datentypen im Sinne des ABAP Dictionarys (3 Kreuze von 9 Möglichkeiten)	
Datenbanktabelle	<input type="checkbox"/>
Datenelement	<input type="checkbox"/>
Domäne	<input type="checkbox"/>
Sperrojekt	<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 <i>richtigen</i> Behauptungen (2 Kreuze von 4 Möglichkeiten)	
Der Inhalt von Datenbanktabellen steht nicht 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 Datenbanktabellen steht 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 nicht aus internen Tabellen bestehen

Interne Tabellen können prinzipiell beliebig lang sein

Interne Tabellen können auch aus internen Tabellen bestehen

9. Kennzeichne die *richtigen* Behauptungen

(3 Kreuze von 6 Möglichkeiten)

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

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

Innerhalb von Unterprogrammen können keine lokalen Variablen definiert werden.

Innerhalb von Unterprogrammen können lokale Variablen definiert werden.

Der rekursive Aufruf von Unterprogrammen ist nicht möglich

Der rekursive Aufruf von Unterprogrammen ist möglich.

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

(1 Kreuz von 6 Möglichkeiten)

BAPI

Berechtigungsprofil

Rolle

Sperrojekt

Suchhilfe

User-Exit

11. Synchronisation des gleichzeitigen Zugriffs zweier Benutzer auf denselben Datenbestand durch:

(1 Kreuz von 6 Möglichkeiten)

BAPI

Berechtigungsprofil

Rolle

Sperrojekt

Suchhilfe

User-Exit

12. Nach betriebswirtschaftlicher Sichtweise gekapselte Funktionsbausteine: (1 Kreuz von 6 Möglichkeiten)	
BAPI	<input type="checkbox"/>
Berechtigungsprofil	<input type="checkbox"/>
Rolle	<input type="checkbox"/>
Sperrojekt	<input type="checkbox"/>
Suchhilfe	<input type="checkbox"/>
User-Exit	<input type="checkbox"/>

13. Dient dazu, ein Menü auszuwählen und dazu passend ein Berechtigungsprofil zu erzeugen: (1 Kreuz von 6 Möglichkeiten)	
BAPI	<input type="checkbox"/>
Berechtigungsprofil	<input type="checkbox"/>
Rolle	<input type="checkbox"/>
Sperrojekt	<input type="checkbox"/>
Suchhilfe	<input type="checkbox"/>
User-Exit	<input type="checkbox"/>

14. Element des Berechtigungssystems, gewährt den Benutzern Zugriff auf das System: (1 Kreuz von 6 Möglichkeiten)	
BAPI	<input type="checkbox"/>
Berechtigungsprofil	<input type="checkbox"/>
Rolle	<input type="checkbox"/>
Sperrojekt	<input type="checkbox"/>
Suchhilfe	<input type="checkbox"/>
User-Exit	<input type="checkbox"/>

15. Zeitpunkt im SAP-Programm, zu dem ein kundeneigener Programmteil aufgerufen werden kann: (1 Kreuz von 6 Möglichkeiten)	
BAPI	<input type="checkbox"/>
Berechtigungsprofil	<input type="checkbox"/>
Rolle	<input type="checkbox"/>
Sperrojekt	<input type="checkbox"/>
Suchhilfe	<input type="checkbox"/>
User-Exit	<input type="checkbox"/>

Aufgabe 2

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

Aufgabe 2 (Fort.)

Aufgabe 3

Erläutere mit eigenen Worten stichwortartig möglichst exakt die Bedeutung und Funktionsweise - der durch **FettKursiv**Druck markierten – ABAP-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_SPLI'  
        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 mit eigenen Worten stichwortartig möglichst exakt die Bedeutung und Funktionsweise - der durch FettKursivDruck markierten – ABAP-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 '02'.
```

```
    IF SY-SUBRC = 0.  
      MODIFY spfli FROM spfli.
```

```
    ELSE.  
      MESSAGE S013 DISPLAY LIKE 'E'.
```

```
  ENDF.  
  ENDCASE.  
  CLEAR ok_code.  
  
  wa_flug = spfli.
```

```
ENDMODULE.
```

Aufgabe 4

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

- Berechtigungsobjekt
- Datenbanktabelle
- Dynpro Ablauflogik
- Dynpro Layout (insbesondere Table-Control)
- Profil
- 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
              value(imp_name) TYPE string,
              getter RETURNING
              value(exp_name) type string.
  PRIVATE SECTION.
    DATA: name TYPE string.
ENDCLASS.

CLASS train IMPLEMENTATION.
  METHOD setter.
    CONCATENATE '2-' imp_name INTO name.
  ENDMETHOD.
  METHOD getter.
    exp_name = name.
  ENDMETHOD.
ENDCLASS.

DATA: name    TYPE string VALUE '-1',
      train   TYPE REF TO train,
      trains  TYPE TABLE OF REF TO train.

START-OF-SELECTION.

  WHILE strlen( name ) < 7.
    CREATE OBJECT train.
    CONCATENATE 'X' name INTO name.
    train->setter( imp_name = name ).
    INSERT train INTO trains INDEX 1.
  ENDWHILE.
  LOOP AT trains INTO train.
    name = train->getter( ).
    WRITE / name.
  ENDLOOP.
```

Das Team der Berufsfachschule und Fachhochschule Wedel wünscht viel Erfolg