

**Klausur am 13. Februar 2014**  
**Anwendungsentwicklung/Geschäftsprozesse in/mit ERP-Systemen**  
*(B\_ECom1.0/2.0 26, B\_Inf11.0 26, B\_Winf11.0 26 – 150 Minuten)*

**Aufgabe 1 (20% Bewertungsanteil)**

Antworten bitte jeweils im Kontext eines SAP ERP Systems, dabei gelten folgende Spielregeln: Pro richtig bearbeitetem Aufgabenteil ein Punkt, pro falsch bearbeitetem Aufgabenteil kein Punkt.

|  |  |
|--|--|
| <b>1. Welche Organisationseinheit dient zur Abbildung eines Konzerns ?</b><br>(1 Kreuz von 12 Möglichkeiten) |  |
| Buchungskreis  |  |
| Einkaufsorganisation   |  |
| Gesellschaft   |  |
| Kontenplan   |  |
| Kostenrechnungskreis   |  |
| Lagerort   |  |
| Mandant  |  |
| Sparte   |  |
| Verkaufsorganisation   |  |
| Vertriebsbereich   |  |
| Vertriebsweg   |  |
| Werk   |  |

|  |  |
|--|--|
| <b>2. Durch welche Organisationseinheit wird das externe Rechnungswesen der Tochtergesellschaften eines Konzerns unterschieden ?</b><br>(1 Kreuz von 12 Möglichkeiten) |  |
| Buchungskreis  |  |
| Einkaufsorganisation   |  |
| Gesellschaft   |  |
| Kontenplan   |  |
| Kostenrechnungskreis   |  |
| Lagerort   |  |
| Mandant  |  |
| Sparte   |  |
| Verkaufsorganisation   |  |
| Vertriebsbereich   |  |
| Vertriebsweg   |  |
| Werk   |  |

**3. Durch welche Organisationseinheit wird das interne Rechnungswesen der Tochtergesellschaften eines Konzerns unterschieden ?**

(1 Kreuz von 12 Möglichkeiten)

|                      |  |
|----------------------|--|
| Buchungskreis        |  |
| Einkaufsorganisation |  |
| Gesellschaft         |  |
| Kontenplan           |  |
| Kostenrechnungskreis |  |
| Lagerort             |  |
| Mandant              |  |
| Sparte               |  |
| Verkaufsorganisation |  |
| Vertriebsbereich     |  |
| Vertriebsweg         |  |
| Werk                 |  |

**4. Welche wesentliche Organisationseinheit definiert ein Auslieferungslager ?**

(1 Kreuz von 12 Möglichkeiten)

|                      |  |
|----------------------|--|
| Buchungskreis        |  |
| Einkaufsorganisation |  |
| Gesellschaft         |  |
| Kontenplan           |  |
| Kostenrechnungskreis |  |
| Lagerort             |  |
| Mandant              |  |
| Sparte               |  |
| Verkaufsorganisation |  |
| Vertriebsbereich     |  |
| Vertriebsweg         |  |
| Werk                 |  |

**5. Aus welchen Organisationseinheiten wird die Organisationseinheit Vertriebsbereich gebildet ?**  
(3 Kreuze von 12 Möglichkeiten)

|                      |  |
|----------------------|--|
| Buchungskreis        |  |
| Einkaufsorganisation |  |
| Gesellschaft         |  |
| Kontenplan           |  |
| Kostenrechnungskreis |  |
| Lagerort             |  |
| Mandant              |  |
| Sparte               |  |
| Verkaufsorganisation |  |
| Vertriebsbereich     |  |
| Vertriebsweg         |  |
| Werk                 |  |

**6. Von welchen Organisationseinheiten sind die drei Vertriebsstufen des Materialstamms (insgesamt) abhängig ?**  
(3 Kreuze von 12 Möglichkeiten)

|                      |  |
|----------------------|--|
| Buchungskreis        |  |
| Einkaufsorganisation |  |
| Gesellschaft         |  |
| Kontenplan           |  |
| Kostenrechnungskreis |  |
| Lagerort             |  |
| Mandant              |  |
| Sparte               |  |
| Verkaufsorganisation |  |
| Vertriebsbereich     |  |
| Vertriebsweg         |  |
| Werk                 |  |

**7. Von welchen Organisationseinheiten sind die vier Dispositionssichten des Materialstamms (insgesamt) abhängig ?**  
(2 Kreuze von 12 Möglichkeiten)

|                      |                          |
|----------------------|--------------------------|
| Buchungskreis        | <input type="checkbox"/> |
| Einkaufsorganisation | <input type="checkbox"/> |
| Gesellschaft         | <input type="checkbox"/> |
| Kontenplan           | <input type="checkbox"/> |
| Kostenrechnungskreis | <input type="checkbox"/> |
| Lagerort             | <input type="checkbox"/> |
| Mandant              | <input type="checkbox"/> |
| Sparte               | <input type="checkbox"/> |
| Verkaufsorganisation | <input type="checkbox"/> |
| Vertriebsbereich     | <input type="checkbox"/> |
| Vertriebsweg         | <input type="checkbox"/> |
| Werk                 | <input type="checkbox"/> |

**8. Von welchen Organisationseinheiten können die beiden Buchhaltungssichten des Materialstamms abhängig sein ?**  
(2 Kreuze von 12 Möglichkeiten)

|                      |                          |
|----------------------|--------------------------|
| Buchungskreis        | <input type="checkbox"/> |
| Einkaufsorganisation | <input type="checkbox"/> |
| Gesellschaft         | <input type="checkbox"/> |
| Kontenplan           | <input type="checkbox"/> |
| Kostenrechnungskreis | <input type="checkbox"/> |
| Lagerort             | <input type="checkbox"/> |
| Mandant              | <input type="checkbox"/> |
| Sparte               | <input type="checkbox"/> |
| Verkaufsorganisation | <input type="checkbox"/> |
| Vertriebsbereich     | <input type="checkbox"/> |
| Vertriebsweg         | <input type="checkbox"/> |
| Werk                 | <input type="checkbox"/> |

**9. Durch das Feld Beschaffungsart im Materialstamm wird unterschieden zwischen:**  
(2 Kreuze von 4 Möglichkeiten)

|                     |                          |
|---------------------|--------------------------|
| Arbeitsplatzauswahl | <input type="checkbox"/> |
| Eigenfertigung      | <input type="checkbox"/> |
| Fremdbeschaffung    | <input type="checkbox"/> |
| Lieferantenauswahl  | <input type="checkbox"/> |

**10. Einkaufsinfosätze bilden eine m:n-Relation zwischen:**  
(2 Kreuze von 5 Möglichkeiten)

|                  |                          |
|------------------|--------------------------|
| Arbeitsplan      | <input type="checkbox"/> |
| Kundenstamm      | <input type="checkbox"/> |
| Lieferantenstamm | <input type="checkbox"/> |
| Materialstamm    | <input type="checkbox"/> |
| Stückliste       | <input type="checkbox"/> |

**11. Primärbedarfe werden ausgelöst durch:**  
(1 Kreuz von 5 Möglichkeiten)

|                      |                          |
|----------------------|--------------------------|
| Bestellanforderungen | <input type="checkbox"/> |
| Bestellungen         | <input type="checkbox"/> |
| Fertigungsaufträge   | <input type="checkbox"/> |
| Kundenaufträge       | <input type="checkbox"/> |
| Planaufträge         | <input type="checkbox"/> |

**12. Sekundärbedarfe werden ausgelöst durch:**  
(1 Kreuz von 5 Möglichkeiten)

|                      |                          |
|----------------------|--------------------------|
| Bestellanforderungen | <input type="checkbox"/> |
| Bestellungen         | <input type="checkbox"/> |
| Fertigungsaufträge   | <input type="checkbox"/> |
| Kundenaufträge       | <input type="checkbox"/> |
| Planaufträge         | <input type="checkbox"/> |

**13. Reservierungen werden ausgelöst durch:**  
(1 Kreuz von 5 Möglichkeiten)

|                      |                          |
|----------------------|--------------------------|
| Bestellanforderungen | <input type="checkbox"/> |
| Bestellungen         | <input type="checkbox"/> |
| Fertigungsaufträge   | <input type="checkbox"/> |
| Kundenaufträge       | <input type="checkbox"/> |
| Planaufträge         | <input type="checkbox"/> |

|  |                          |
|--|--------------------------|
| <b>14. Geplante Lagerabgänge werden ausgelöst durch:</b><br>(3 Kreuze von 5 Möglichkeiten) |                          |
| Bestellanforderungen   | <input type="checkbox"/> |
| Bestellungen   | <input type="checkbox"/> |
| Fertigungsaufträge   | <input type="checkbox"/> |
| Kundenaufträge   | <input type="checkbox"/> |
| Planaufträge   | <input type="checkbox"/> |

|  |                          |
|--|--------------------------|
| <b>15. Geplante Lagerzugänge werden ausgelöst durch:</b><br>(4 Kreuze von 5 Möglichkeiten) |                          |
| Bestellanforderungen   | <input type="checkbox"/> |
| Bestellungen   | <input type="checkbox"/> |
| Fertigungsaufträge   | <input type="checkbox"/> |
| Kundenaufträge   | <input type="checkbox"/> |
| Planaufträge   | <input type="checkbox"/> |

|  |                          |
|--|--------------------------|
| <b>16. Sinnvolle Formel zur Berechnung des Kapazitätsbedarfs:</b><br>(1 Kreuz von 5 Möglichkeiten) |                          |
| $\text{Rüstzeit} + \text{Einsatzzeit} * \text{Vorgangsmenge} / \text{Vorgangssplitts}$             | <input type="checkbox"/> |
| $\text{Rüstzeit} * \text{Vorgangsmenge} + \text{Einsatzzeit} * \text{Vorgangssplitts}$             | <input type="checkbox"/> |
| $\text{Rüstzeit} * \text{Vorgangsmenge} + \text{Einsatzzeit} / \text{Vorgangssplitts}$             | <input type="checkbox"/> |
| $\text{Rüstzeit} * \text{Vorgangssplitts} + \text{Einsatzzeit} * \text{Vorgangsmenge}$             | <input type="checkbox"/> |
| $\text{Rüstzeit} / \text{Vorgangssplitts} + \text{Einsatzzeit} * \text{Vorgangsmenge}$             | <input type="checkbox"/> |

|   |                          |
|---|--------------------------|
| <b>17. Sinnvolle Formel zur Berechnung der Terminierung (Dauer):</b><br>(1 Kreuz von 5 Möglichkeiten) |                          |
| $\text{Rüstzeit} + \text{Einsatzzeit} * \text{Vorgangsmenge} / \text{Vorgangssplitts}$                | <input type="checkbox"/> |
| $\text{Rüstzeit} * \text{Vorgangsmenge} + \text{Einsatzzeit} * \text{Vorgangssplitts}$                | <input type="checkbox"/> |
| $\text{Rüstzeit} * \text{Vorgangsmenge} + \text{Einsatzzeit} / \text{Vorgangssplitts}$                | <input type="checkbox"/> |
| $\text{Rüstzeit} * \text{Vorgangssplitts} + \text{Einsatzzeit} * \text{Vorgangsmenge}$                | <input type="checkbox"/> |
| $\text{Rüstzeit} / \text{Vorgangssplitts} + \text{Einsatzzeit} * \text{Vorgangsmenge}$                | <input type="checkbox"/> |

**18. Können Primärkostenarten ohne Entsprechung zu einem Erfolgskonto des Kontenplans der Finanzbuchhaltung angelegt werden ?**

(1 Kreuz von 2 Möglichkeiten)

Ja

Nein

**19. Können Sekundärkostenarten mit Entsprechung zu einem Erfolgskonto des Kontenplans der Finanzbuchhaltung angelegt werden ?**

(1 Kreuz von 2 Möglichkeiten)

Ja

Nein

**20. Allgemeiner Buchungssatz für den Verbrauch von Rohstoffen aus dem Lager in der Produktion:**

(1 Kreuz von 6 Möglichkeiten)

Per Rohstoffe (*Bestandskonto*) an Rohstoffe (*Aufwandskonto*)

Per Rohstoffe (*Aufwandskonto*) an Rohstoffe (*Bestandskonto*)

Per Rohstoffe (*Bestandskonto*) an Unfertige Erzeugnisse

Per Unfertige Erzeugnisse an Rohstoffe (*Bestandskonto*)

Per Bestandsveränderungen an Unfertige Erzeugnisse

Per Unfertige Erzeugnisse an Bestandsveränderungen

**21. Allgemeiner Buchungssatz für die Einlagerung von produzierten Baugruppen (unfertige Erzeugnisse, Halbfertigfabrikate):**

(1 Kreuz von 6 Möglichkeiten)

Per Rohstoffe (*Bestandskonto*) an Rohstoffe (*Aufwandskonto*)

Per Rohstoffe (*Aufwandskonto*) an Rohstoffe (*Bestandskonto*)

Per Rohstoffe (*Bestandskonto*) an Unfertige Erzeugnisse

Per Unfertige Erzeugnisse an Rohstoffe (*Bestandskonto*)

Per Bestandsveränderungen an Unfertige Erzeugnisse

Per Unfertige Erzeugnisse an Bestandsveränderungen

**22. Allgemeiner Buchungssatz für den Verkaufs-Beleg „Lieferschein“:**

(1 Kreuz von 4 Möglichkeiten)

Per Fertige Erzeugnisse an Bestandsveränderungen

Per Bestandsveränderungen an Fertige Erzeugnisse

Per Forderungen aus Lieferungen und Leistungen an Umsatzerlöse

Per Umsatzerlöse an Forderungen aus Lieferungen und Leistungen

**23. Allgemeiner Buchungssatz für den Verkaufs-Beleg „Rechnung“ (ohne Berücksichtigung von Mehrwertsteuer):**

(1 Kreuz von 4 Möglichkeiten)

|  |                          |
|--|--------------------------|
| Per Fertige Erzeugnisse an Bestandsveränderungen               | <input type="checkbox"/> |
| Per Bestandsveränderungen an Fertige Erzeugnisse               | <input type="checkbox"/> |
| Per Forderungen aus Lieferungen und Leistungen an Umsatzerlöse | <input type="checkbox"/> |
| Per Umsatzerlöse an Forderungen aus Lieferungen und Leistungen | <input type="checkbox"/> |

**24. Felder im Materialstamm zur Hinterlegung von Bewertungspreisen:**

(2 Kreuze von 4 Möglichkeiten)

|                  |                          |
|------------------|--------------------------|
| Bewertungsklasse | <input type="checkbox"/> |
| Gleitender Preis | <input type="checkbox"/> |
| Preissteuerung   | <input type="checkbox"/> |
| Standardpreis    | <input type="checkbox"/> |

**25. Durch welche Beziehung legt man fest, in welchem externen Rechnungswesen eine Materialbewegung parallel wertmässig gebucht wird ? (typische Installation vorausgesetzt)**

(1 Kreuz von 5 Möglichkeiten)

|                                      |                          |
|--------------------------------------|--------------------------|
| Buchungskreis : Einkaufsorganisation | <input type="checkbox"/> |
| Buchungskreis : Kontenplan           | <input type="checkbox"/> |
| Buchungskreis : Kostenrechnungskreis | <input type="checkbox"/> |
| Buchungskreis : Verkaufsorganisation | <input type="checkbox"/> |
| Buchungskreis : Werk                 | <input type="checkbox"/> |

**26. Wodurch wird eine automatische Materialabbuchung bei Rückmeldung von Fertigungsaufträgen ermöglicht ?**

(2 Kreuze von 6 Möglichkeiten)

|   |                          |
|---|--------------------------|
| Feld Dispolosgröße im Materialstamm                           | <input type="checkbox"/> |
| Feld Dispomerkmale im Materialstamm                           | <input type="checkbox"/> |
| Feld Fremdbeschaffungslagerort im Materialstamm               | <input type="checkbox"/> |
| Feld Produktionslagerort im Materialstamm                     | <input type="checkbox"/> |
| Feld Retrograde Entnahme im Materialstamm                     | <input type="checkbox"/> |
| Steuerschlüssel „Meilenstein mit Wareneingang“ im Arbeitsplan | <input type="checkbox"/> |

|  |                          |
|--|--------------------------|
| <b>27. Wodurch wird eine automatische Materialzubuchung bei Rückmeldung von Fertigungsaufträgen ermöglicht ?</b><br>(2 Kreuze von 6 Möglichkeiten) |                          |
| Feld Dispolosgröße im Materialstamm  | <input type="checkbox"/> |
| Feld Dispomerkmale im Materialstamm  | <input type="checkbox"/> |
| Feld Fremdbeschaffungslagerort im Materialstamm  | <input type="checkbox"/> |
| Feld Produktionslagerort im Materialstamm  | <input type="checkbox"/> |
| Feld Retrograde Entnahme im Materialstamm  | <input type="checkbox"/> |
| Steuerschlüssel „Meilenstein mit Wareneingang“ im Arbeitsplan  | <input type="checkbox"/> |

|   |                          |
|---|--------------------------|
| <b>28. Welche Begriffe werden auch für das „externe Rechnungswesen“ verwendet ?</b><br>(2 Kreuze von 7 Möglichkeiten) |                          |
| Controlling   | <input type="checkbox"/> |
| Finanzbuchhaltung   | <input type="checkbox"/> |
| Finanzwesen   | <input type="checkbox"/> |
| Kosten-/Leistungsrechnung   | <input type="checkbox"/> |
| Materialwirtschaft  | <input type="checkbox"/> |
| Produktion  | <input type="checkbox"/> |
| Vertrieb  | <input type="checkbox"/> |

|   |                          |
|---|--------------------------|
| <b>29. Welche Begriffe werden auch für das „interne Rechnungswesen“ verwendet ?</b><br>(2 Kreuze von 7 Möglichkeiten) |                          |
| Controlling   | <input type="checkbox"/> |
| Finanzbuchhaltung   | <input type="checkbox"/> |
| Finanzwesen   | <input type="checkbox"/> |
| Kosten-/Leistungsrechnung   | <input type="checkbox"/> |
| Materialwirtschaft  | <input type="checkbox"/> |
| Produktion  | <input type="checkbox"/> |
| Vertrieb  | <input type="checkbox"/> |

|  |                          |
|--|--------------------------|
| <b>30. Mandantenübergreifende Auswertungen lassen sich realisieren durch:</b><br>(1 Kreuz von 3 Möglichkeiten) |                          |
| Customizing-Einstellungen im Standard  | <input type="checkbox"/> |
| Individualprogramme  | <input type="checkbox"/> |
| Standardprogramme  | <input type="checkbox"/> |

## **Aufgabe 2 (20% Bewertungsanteil)**

**Gegeben seien folgende Stamm- und Bewegungsdaten:**

*Materialstamm:*

Fertigerzeugnis *Bike Standard One*

- Lagerbestand: 100 Stück (frei verfügbar)
- Eigenfertigungszeit: 4 Kalenderwochen
- feste Losgröße: 25 Stück

Fertigerzeugnis *Bike Premium*

- Lagerbestand: 50 Stück (frei verfügbar)
- Eigenfertigungszeit: 2 Kalenderwochen
- exakte Losgröße

Fertigerzeugnis *Bike Standard Two*

- Lagerbestand: 100 Stück (frei verfügbar)
- Eigenfertigungszeit: 4 Kalenderwochen
- feste Losgröße: 25 Stück

Halbfabrikat *Tyre One*

- Lagerbestand: 0 Stück
- Eigenfertigungszeit: 1 Kalenderwoche
- exakte Losgröße

Halbfabrikat *Tyre Two*

- Lagerbestand: 0 Stück
- Eigenfertigungszeit: 1 Kalenderwoche
- exakte Losgröße

Rohstoff *Steel Pipe*

- Lagerbestand: 900 Stück (frei verfügbar)
- Planlieferzeit: 2 Kalenderwoche
- exakte Losgröße

Rohstoff *Spoke*

- Lagerbestand: 3100 Stück (frei verfügbar)
- Planlieferzeit: 4 Kalenderwochen
- feste Losgröße: 250 Stück

Rohstoff *Aluminium Pipe*

- Lagerbestand: 900 Stück (frei verfügbar)
- Planlieferzeit: 2 Kalenderwoche
- exakte Losgröße

## **Aufgabe 2 (Fort.)**

### *Baukastenstücklisten:*

Ein Stück *Bike Standard One* besteht aus zwei Stück *Tyre One* und vier Stück *Steel Pipe*.

Ein Stück *Bike Premium* besteht aus einem Stück *Tyre One* und einem *Tyre Two*.

Ein Stück *Bike Standard Two* besteht aus zwei Stück *Tyre Two* und vier Stück *Aluminium Pipe*.

Ein Stück *Tyre One* besteht aus zwei Stück *Steel Pipe* und vier Stück *Spoke*.

Ein Stück *Tyre Two* besteht aus vier Stück *Spoke* und zwei Stück *Aluminium Pipe*.

### *Kundenaufträge:*

Kunde A bestellt 90 Stück *Bike Standard One* auf die Kalenderwoche 21/2014 und 90 Stück *Bike Standard Two* auf die Kalenderwoche 21/2014.

Kunde B bestellt 90 Stück *Bike Standard One* auf die Kalenderwoche 23/2014, 250 Stück *Bike Premium* auf die Kalenderwoche 21/2014 und 90 Stück *Bike Standard Two* auf die Kalenderwoche 23/2014.

Kunde C bestellt 20 Stück *Bike Standard One* auf die Kalenderwoche 25/2014 und 20 Stück *Bike Standard Two* auf die Kalenderwoche 25/2014.

### *Hinweise:*

Für alle Materialien gilt das Dispomerkmal „PD“ (plangesteuerte Disposition), und die Granularität der Planung beträgt eine Kalenderwoche.

Die Eigenfertigungszeiten gelten jeweils unabhängig von der zu produzierenden Menge und der Kapazitätsauslastung. Für den Versand sind keine Zeiten zu berücksichtigen. Die aktuelle Kalenderwoche lautet 7/2014.

**Berechne – im Kontext eines SAP ERP Systems und im Sinne eines MRP-Laufs – die Mengen und Termine (als Kalenderwochen) für Bestellanforderungen, Planaufträge und Sekundärbedarfe. Für Bestellanforderungen und Planaufträge sind jeweils sowohl der Start- als auch der Endtermin anzugeben.**

## Aufgabe 2 (Fort.)

**Aufgabe 2 (Fort.)**

### Aufgabe 3 (20% Bewertungsanteil)

|  |  |
|--|--|
| <b>1. Kennzeichne die richtige Behauptung</b><br>(1 Kreuz von 4 Möglichkeiten) |  |
| ABAP ist datenbankabhängig und plattformabhängig                               |  |
| ABAP ist datenbankabhängig und plattformunabhängig                             |  |
| ABAP ist datenbankunabhängig und plattformabhängig                             |  |
| ABAP ist datenbankunabhängig und plattformunabhängig                           |  |

|   |  |
|---|--|
| <b>2. Kennzeichne die richtige Behauptung</b><br>(1 Kreuz von 4 Möglichkeiten)          |  |
| ABAP richtet sich weder an die Wiederverwenbarkeit von Code- noch von Datenobjekten aus |  |
| ABAP richtet sich stark an die Wiederverwendbarkeit von Codeobjekten aus                |  |
| ABAP richtet sich stark an die Wiederverwendbarkeit von Datenobjekten aus               |  |
| ABAP richtet sich stark an die Wiederverwendbarkeit von Code- und Datenobjekten aus     |  |

|  |  |
|--|--|
| <b>3. Kennzeichne die richtigen Behauptungen</b><br>(2 Kreuze von 4 Möglichkeiten) |  |
| ABAP-Programme liegen nur in kompilierter Form vor                                 |  |
| ABAP-Programme liegen auch als Quelltext vor                                       |  |
| ABAP-Programme werden beim ersten Aufruf kompiliert                                |  |
| ABAP-Programme werden bei jedem Aufruf kompiliert                                  |  |

|  |  |
|--|--|
| <b>4. Kennzeichne die richtigen Behauptungen</b><br>(2 Kreuze von 4 Möglichkeiten) |  |
| Das Repository ist mandantenunabhängig   |  |
| Das Repository ist mandantenabhängig   |  |
| Die Anwendungsdaten sind mandantenunabhängig                                       |  |
| Die Anwendungsdaten sind mandatenabhängig  |  |

|  |                          |
|--|--------------------------|
| <b>5. Transporte in einer typischen SAP-Systemlandschaft erfolgen:</b><br>(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"/> |

|  |                          |
|--|--------------------------|
| <b>6. Kennzeichne die richtige Behauptung</b><br>(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"/> |

|  |                          |
|--|--------------------------|
| <b>7. Markiere die Datentypen im Sinne des ABAP Dictionary</b><br>(3 Kreuze von 9 Möglichkeiten) ( <i>doppelt gewertet</i> ) |                          |
| Datenbanktabelle   | <input type="checkbox"/> |
| Datenelement   | <input type="checkbox"/> |
| Domäne   | <input type="checkbox"/> |
| Sperrobject  | <input type="checkbox"/> |
| Struktur   | <input type="checkbox"/> |
| Suchhilfe  | <input type="checkbox"/> |
| Tabellentyp  | <input type="checkbox"/> |
| Typgruppe  | <input type="checkbox"/> |
| View   | <input type="checkbox"/> |

|   |                          |
|---|--------------------------|
| <b>8. Kennzeichne die richtigen Behauptungen</b><br>(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"/> |

|  |  |
|--|--|
| <b>9. Kennzeichne die richtigen Behauptungen</b><br>(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                        |  |

|   |  |
|---|--|
| <b>10. Kennzeichne die richtigen Behauptungen</b><br>(3 Kreuze von 6 Möglichkeiten) ( <i>doppelt gewertet</i> ) |  |
| 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.   |  |

|   |  |
|---|--|
| <b>11. Markiere aus der Sicht des Funktionsbausteins die richtigen Aussagen zu Funktionsbausteinen:</b><br>(3 Kreuze von 6 Möglichkeiten) ( <i>doppelt gewertet</i> ) |  |
| 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.   |  |

|  |  |
|--|--|
| <b>12. Das Abfangen von Laufzeitfehlern unter Umgehung der objektorientierten Programmierung ist möglich mit:</b><br>(1 Kreuz von 5 Möglichkeiten) |  |
| Befehl CATCH SYSTEM-EXCEPTIONS   |  |
| Variable Sy-dbent  |  |
| Variable Sy-fdpos  |  |
| Befehl TRY CATCH   |  |
| Variable Sy-subrc  |  |

### **Aufgabe 3 (Fort.)**

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 (20% Bewertungsanteil)**

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.

Doch Halt, der Programmierer scheint beim Spezifizieren ganz gehörig durcheinander geraten zu sein, Helft ihm durch Ankreuzen, welche Codezeilen erforderlich sind, und damit auch, welche Codezeilen nicht erforderlich (bzw. falsch oder überflüssig) sind.

*Dabei gilt folgende „Spielregel“: Die Anzahl der erforderlichen Kreuze steht jeweils in Klammern hinter der Blocknummer, jeder richtig bearbeiteter Block erzielt einen Pluspunkt, jeder falsch bearbeiteter Block bleibt unberücksichtigt.*

| <b>Ablauflogik Dynpro 0100 („Auswahl“)</b> | erforderlich | Block<br>(Kreuzanzahl) |
|--|--------------|------------------------|
| PROCESS BEFORE OUTPUT.                     | X            |                        |
| MODULE status_0100.                        |              | 1 (4)                  |
| MODULE user_command_0100.                  |              | 1 (4)                  |
| LOOP WITH CONTROL personen.                |              | 1 (4)                  |
| LOOP WITH CONTROL mein_alv.                |              | 1 (4)                  |
| MODULE fill_table_control.                 |              | 1 (4)                  |
| MODULE read_table_control.                 |              | 1 (4)                  |
| MODULE status_0100.                        |              | 1 (4)                  |
| MODULE user_command_0100.                  |              | 1 (4)                  |
| ENDLOOP.                                   |              | 1 (4)                  |
| MODULE status_0100.                        |              | 1 (4)                  |
| MODULE user_command_0100.                  |              | 1 (4)                  |
| PROCESS AFTER INPUT.                       | X            |                        |
| MODULE status_0100.                        |              | 2 (4)                  |
| MODULE user_command_0100.                  |              | 2 (4)                  |
| LOOP WITH CONTROL personen.                |              | 2 (4)                  |
| LOOP WITH CONTROL mein_alv.                |              | 2 (4)                  |
| MODULE fill_table_control.                 |              | 2 (4)                  |
| MODULE read_table_control.                 |              | 2 (4)                  |
| MODULE status_0100.                        |              | 2 (4)                  |
| MODULE user_command_0100.                  |              | 2 (4)                  |
| ENDLOOP.                                   |              | 2 (4)                  |
| MODULE status_0100.                        |              | 2 (4)                  |
| MODULE user_command_0100.                  |              | 2 (4)                  |

| <b>Ablauflogik Dynpro 0200 („Anzeigen/Ändern“)</b> | erforderlich | Block<br>(Kreuzanzahl) |
|--|--------------|------------------------|
| PROCESS BEFORE OUTPUT.                             | X            |                        |
| MODULE fill_table_control.                         |              | 3 (1)                  |
| MODULE read_table_control.                         |              | 3 (1)                  |
| MODULE status_0200.                                |              | 3 (1)                  |
| MODULE user_command_0200.                          |              | 3 (1)                  |
| PROCESS AFTER INPUT.                               | X            |                        |
| MODULE fill_table_control.                         |              | 4 (1)                  |
| MODULE read_table_control.                         |              | 4 (1)                  |
| MODULE status_0200.                                |              | 4 (1)                  |
| MODULE user_command_0200.                          |              | 4 (1)                  |

| <b>TOP-Include</b>  | erforderlich | Block<br>(Kreuzanzahl) |
|---|--------------|------------------------|
| PROGRAM zz_dynpro.  |              | 5 (5)                  |
| CONTROLS personen TYPE TABLEVIEW USING SCREEN 100.        |              | 5 (5)                  |
| CONTROLS personen TYPE TABLEVIEW USING SCREEN 200.        |              | 5 (5)                  |
| TABLES zz_personen.                                       |              | 5 (5)                  |
| TABLES zz_personen_t.                                     |              | 5 (5)                  |
| DATA: ok_code LIKE sy-ucomm.                              |              | 5 (5)                  |
|   |              |                        |
| DATA: wa_person TYPE zz_personen.                         |              | 6 (2)                  |
| DATA: wa_person TYPE zz_personen_t.                       |              | 6 (2)                  |
| DATA: itab TYPE TABLE OF zz_personen.                     |              | 6 (2)                  |
| DATA: itab TYPE TABLE OF zz_personen_t.                   |              | 6 (2)                  |
| DATA: mein_container TYPE REF TO c1_gui_custom_container. |              | 6 (2)                  |
| DATA: mein_alv TYPE REF TO c1_gui_alv_grid.               |              | 6 (2)                  |

| <b>Include zu MODULE status_0100</b>                                    | erforderlich | Block<br>(Kreuzanzahl) |
|---|--------------|------------------------|
| MODULE status_0100 OUTPUT.  | X            |                        |
| SET PF-STATUS '100'.  |              | 7 (3)                  |
| SET PF-STATUS '200'.  |              | 7 (3)                  |
| SET TITLEBAR '100'.   |              | 7 (3)                  |
| SET TITLEBAR '200'.   |              | 7 (3)                  |
| CLEAR ok_code.  |              | 7 (3)                  |
|   |              |                        |
| SELECT * FROM zz_personen INTO<br>TABLE itab.                           |              | 8 (2)                  |
| SELECT * FROM zz_personen INTO<br>CORRESPONDING FIELDS OF TABLE itab.   |              | 8 (2)                  |
| SELECT * FROM zz_personen_t INTO<br>TABLE itab.                         |              | 8 (2)                  |
| SELECT * FROM zz_personen_t INTO<br>CORRESPONDING FIELDS OF TABLE itab. |              | 8 (2)                  |
| DESCRIBE TABLE itab LINES personen-lines.                               |              | 8 (2)                  |
| DESCRIBE TABLE itab LINES mein_alv-lines.                               |              | 8 (2)                  |
| ENDMODULE.  | X            |                        |

| <b>Include zu MODULE user_command_0100</b>   | erforderlich | Block<br>(Kreuzanzahl) |
|--|--------------|------------------------|
| MODULE user_command_0100 INPUT.  | X            |                        |
| CASE ok_code.  | X            |                        |
| WHEN 'BACK'.   | X            |                        |
| LEAVE PROGRAM.   | X            |                        |
| WHEN 'SELECT'  |              | 9 (2)                  |
| WHEN 'SELECT' OR 'PICK'.   |              | 9 (2)                  |
| SELECT SINGLE * FROM zz_personen INTO wa_person<br>WHERE pid = zz_personen-pid.  |              | 9 (2)                  |
| SELECT SINGLE * FROM zz_personen_t INTO wa_person<br>WHERE pid = zz_personen-pid.  |              | 9 (2)                  |
| SELECT SINGLE * FROM zz_personen INTO wa_person<br>WHERE pid = zz_personen-pid AND<br>titel = zz_personen-titel.   |              | 9 (2)                  |
| SELECT SINGLE * FROM zz_personen_t INTO wa_person<br>WHERE pid = zz_personen-pid AND<br>titel = zz_personen-titel.   |              | 9 (2)                  |
| SELECT SINGLE * FROM zz_personen INTO wa_person<br>WHERE pid = zz_personen-pid AND<br>titel = zz_personen-titel AND<br>vorname = zz_personen-vorname AND<br>nachname = zz_personen-nachname.   |              | 9 (2)                  |
| SELECT SINGLE * FROM zz_personen_t INTO wa_person<br>WHERE pid = zz_personen-pid AND<br>titel = zz_personen-titel AND<br>vorname = zz_personen-vorname AND<br>nachname = zz_personen-nachname. |              | 9 (2)                  |
|  |              |                        |
| IF sy-dbcnt = 0.   |              | 10 (2)                 |
| IF sy-dbcnt = 1.   |              | 10 (2)                 |
| LEAVE TO SCREEN 100.   |              | 10 (2)                 |
| LEAVE TO SCREEN 200.   |              | 10 (2)                 |
| ENDIF.   | X            |                        |
| ENDCASE.   | X            |                        |
| ENDMODULE.   | X            |                        |

| <b>Include zu MODULE status_0200</b> | erforderlich | Block<br>(Kreuzanzahl) |
|--------------------------------------|--------------|------------------------|
| MODULE status_0200 OUTPUT.           | X            |                        |
| SET PF-STATUS '100'.                 |              | 11 (3)                 |
| SET PF-STATUS '200'.                 |              | 11 (3)                 |
| SET TITLEBAR '100'.                  |              | 11 (3)                 |
| SET TITLEBAR '200'.                  |              | 11 (3)                 |
| CLEAR ok_code.                       |              | 11 (3)                 |
|                                      |              |                        |
| wa_person = zz_personen.             |              | 12 (1)                 |
| zz_personen = wa_person.             |              | 12 (1)                 |
| ENDMODULE.                           | X            |                        |

| <b>Include zu MODULE user_command_0200</b> | erforderlich | Block<br>(Kreuzanzahl) |
|--|--------------|------------------------|
| MODULE user_command_0200 INPUT.            | X            |                        |
| CASE ok_code.                              | X            |                        |
| WHEN 'LEAVE'.                              |              | 13 (4)                 |
| LEAVE PROGRAM.                             |              | 13 (4)                 |
| WHEN 'BACK'.                               |              | 13 (4)                 |
| LEAVE TO SCREEN 100.                       |              | 13 (4)                 |
| LEAVE TO SCREEN 200.                       |              | 13 (4)                 |
| WHEN 'SAVE'.                               | X            |                        |
| MODIFY zz_personen FROM zz_personen.       |              | 14 (1)                 |
| MODIFY zz_personen_t FROM zz_personen.     |              | 14 (1)                 |
| MODIFY zz_personen FROM wa_person.         |              | 14 (1)                 |
| MODIFY zz_personen_t FROM wa_person.       |              | 14 (1)                 |
| ENDCASE.                                   | X            |                        |
| zz_personen = wa_person.                   |              | 15 (1)                 |
| wa_person = zz_personen.                   |              | 15 (1)                 |
| ENDMODULE.                                 | X            |                        |

| <b>Include zu MODULE fill_table_control</b>                                 | erforderlich | Block<br>(Kreuzanzahl) |
|---|--------------|------------------------|
| MODULE fill_table_control OUTPUT.   | X            |                        |
| READ TABLE itab INTO zz_personen_t<br>INDEX personen-current_line.          |              | 16 (1)                 |
| READ TABLE itab INTO zz_personen_t<br>INDEX sy-index.                       |              | 16 (1)                 |
| READ TABLE zz_personen INTO zz_personen_t<br>INDEX personen-current_line.   |              | 16 (1)                 |
| READ TABLE zz_personen INTO zz_personen_t<br>INDEX sy-index.                |              | 16 (1)                 |
| READ TABLE zz_personen_t INTO zz_personen_t<br>INDEX personen-current_line. |              | 16 (1)                 |
| READ TABLE zz_personen_t INTO zz_personen_t<br>INDEX sy-index.              |              | 16 (1)                 |
| ENDMODULE.  | X            |                        |

| <b>Include zu MODULE read_table_control</b>  | erforderlich | Block<br>(Kreuzanzahl) |
|--|--------------|------------------------|
| MODULE read_table_control INPUT.   | X            |                        |
| DATA cursorline TYPE i.  | X            |                        |
| GET CURSOR cursorline.   |              | 17 (1)                 |
| GET CURSOR LINE cursorline.  |              | 17 (1)                 |
| GET LINE cursorline.   |              | 17 (1)                 |
| IF sy-subrc = 0.   | X            |                        |
| cursorline = personen-top_line + cursorline - 1.                                   |              | 18 (1)                 |
| cursorline = personen-top_line + cursorline.                                       |              | 18 (1)                 |
| cursorline = personen-top_line + cursorline + 1.                                   |              | 18 (1)                 |
| ELSE.  | X            |                        |
| cursorline = -1.   |              |                        |
| cursorline = 0.  | X            |                        |
| cursorline = 1.  |              |                        |
| ENDIF.   | X            |                        |
| IF ( zz_personen_t-selection = 'X' ).  |              | 19 (1)                 |
| IF ( zz_personen_t-selection = 'X' ) OR<br>( personen-current_line = cursorline ). |              | 19 (1)                 |
| IF ( zz_personen_t-selection = 'X' ) OR<br>( mein_alv-current_line = cursorline ). |              | 19 (1)                 |
| IF ( personen-current_line = cursorline ).   |              | 19 (1)                 |
| IF ( mein_alv-current_line = cursorline ).   |              | 19 (1)                 |
|  |              |                        |
| zz_personen_t-pid = zz_personen-pid.   |              | 20 (1)                 |
| zz_personen_t-title = zz_personen-title.   |              | 20 (1)                 |
| zz_personen-pid = zz_personen_t-pid.   |              | 20 (1)                 |
| zz_personen-title = zz_personen_t-title.   |              | 20 (1)                 |
| ENDIF.   | X            |                        |
| ENDMODULE.   | X            |                        |

## **Aufgabe 5 (20% Bewertungsanteil)**

Erläutere mit eigenen Worten stichwortartig möglichst exakt entweder die drei Programmfragmente auf dieser Seite oder den Report auf der nächsten Seite.

### ***Top-Include***

```
PROGRAM zz_dynpro.  
  
TABLES spfli.  
DATA ok_code LIKE sy-ucomm.  
DATA wa_flug TYPE spfli.
```

### ***Process After Input (PAI) Dynpro 100***

```
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.  
        LEAVE TO SCREEN 200.  
      ENDIF.  
    ENDIF.  
  ENDCASE.  
  CLEAR ok_code.  
  
ENDMODULE.
```

### ***Process After Input (PAI) Dynpro 200***

```
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'.  
    MODIFY spfli FROM spfli.  
  ENDCASE.  
  CLEAR ok_code.  
  wa_flug = spfli.  
  
ENDMODULE.
```

## Aufgabe 5 (Fort.)

### Report ZZ\_OBJECTS

```
REPORT ZZ_OBJECTS.

CLASS lcl_airplane DEFINITION.

    PUBLIC SECTION.
        METHODS: attribute_setzen IMPORTING
            im_name TYPE string
            im_planetype TYPE string,
            attribute_anzeigen.

    PRIVATE SECTION.
        DATA: name TYPE string,
            planetype TYPE string.

ENDCLASS.

CLASS lcl_airplane IMPLEMENTATION.

    METHOD attribute_setzen.
        name = im_name.
        planetype = im_planetype.
    ENDMETHOD.

    METHOD attribute_anzeigen.
        WRITE: / 'Name des Flugzeugs: ', name,
            / 'Typ des Flugzeugs: ', planetype.
    ENDMETHOD.

ENDCLASS.

DATA: r_plane TYPE REF TO lcl_airplane,
    it_plane_list TYPE TABLE OF REF TO lcl_airplane.

START-OF-SELECTION.

CREATE OBJECT r_plane.
r_plane->attribute_setzen( im_name = 'München'
                        im_planetype = 'Boeing 737' ).
APPEND r_plane TO it_plane_list.

CREATE OBJECT r_plane.
r_plane->attribute_setzen( im_name = 'Hamburg'
                        im_planetype = 'Airbus 380' ).
APPEND r_plane TO it_plane_list.

LOOP AT it_plane_list INTO r_plane.
    r_plane->attribute_anzeigen( ).
ENDLOOP.
```

## **Aufgabe 5 (Fort.)**

## **Aufgabe 5 (Fort.)**

*Das Team der Fachhochschule Wedel wünscht viel Erfolg*