

1. Handlungsschritt (20 Punkte)

- aa) – Komplexe Aufgabe
– Kein Routineprozess
– Konkrete Ziele
– In der Regel Kooperation mehrerer Fachbereiche
– Vorgegebene Ressourcen
– u. a. (2 P.)
- ab) – Kooperativer Führungsstil
– Fachkompetenz (2 P.)
– Konfliktfähigkeit
- ac) – Projektbezeichnung
– Projektleitung
– Projektziel
– Status (Termine, Kosten, Technik, Qualität)
– Projektsteuerung
– Fremdleistung
– Störungen, Probleme
– Abschlussbericht
– Bewertung des Projektes (2 P.)
- ba) – Wasserfallmodell (Klassisches Phasenmodell)
– Spiralmodell (2 P.)
- bb) Wasserfallmodell
– Startphase (Projektanalyse, Projektdefinition)
– Phase 2 (Entwurf)
– Phase 3 (Implementation)
– Phase 4 (Systemtest)
– Abschlussphase (Einsatz, Wartung)

oder:

- Spiralmodell
Segment 1: Festlegen der Ziele und Aufgaben (Soll-Vorgaben)
Segment 2: Bewertung von Entwicklungspfaden und Analyse von Risiken
Segment 3: Erstellung eines Vor- bzw. Zwischenprodukts und Testläufe
Segment 4: Ist-Aufnahme Soll-Ist-Vergleich, Festlegung weiterer Entwicklungs- und Arbeitsschritte (6 P.)

ca)		Vorteile	Nachteile
	Standardsoftware	bewährt	individuelle Anpassung nur bedingt möglich hat nicht benötigte Funktionen
	Individual	Kehrwert	Kehrwert (4 P.)

- cb) Ein Objekt ist die Instanz einer Klasse. (2 P.)

2. Handlungsschritt (20 Punkte)

- Zählschleife für Abfrage benutzt
- Variablen vorhanden und sinnvoll benannt
- Datenverdoppelung vor Sortierung
- Sortieralgorithmus: Schleifenkonstruktion verwandt
- Zwischenwert gespeichert
- Ausgabe über Zählschleife

(6 x 3 P. = 18 P.)

sinnvolle Kommentare

(2 P.)

Lösungsvorschlag in VBA

Hinweis: Eine Lösung über Datenfelder ist sinnvoll, aber nicht Voraussetzung!

```
Sub BubbleSort()
```

```
    'Lösungsvorschlag in VBA
```

```
    Const MAX = 7
```

```
    'Variablen deklarieren
```

```
    Dim datenFeld(MAX), datenFeldEin(MAX) As Integer
```

```
    Dim zw, i, j, k As Integer
```

```
    'Werte von Benutzer erfragen
```

```
    For i = 1 To MAX Step 1
```

```
        datenFeld(i) = CInt(InputBox("Bitte den " & i & ". Wert eingeben:"))
```

```
    Next i
```

```
    'Werte für Ausgabe dublizieren
```

```
    For i = 1 To MAX Step 1
```

```
        datenFeldEin(i) = datenFeld(i)
```

```
    Next i
```

```
    'Sortieren      'Anwendung des "Brute-Force-Verfahrens"
```

```
    For i = MAX To 1 Step -1
```

```
        For j = MAX To 1 Step -1
```

```
            If (datenFeld(i) > datenFeld(j)) Then
```

```
                zw = datenFeld(i)      'alten Wert sichern
```

```
                datenFeld(i) = datenFeld(j)      'kleineren Wert nach vorne
```

```
                datenFeld(j) = zw      'größeren Wert nach hinten
```

```
            End If
```

```
        Next j
```

```
    Next i
```

```
    'Ausgabe
```

```
    For i = 1 To MAX Step 1
```

```
        Debug.Print i & ".Eingabe: " & datenFeldEin(i) & " - sortierter Wert: " & datenFeld(i)
```

```
    Next i
```

```
End Sub
```

3. Handlungsschritt (20 Punkte)

a)

Kunden-Nr
678
131
234
657
342

(5 P.)

b) Datendefinitionen:

- i, j, n Zählvariablen
anzahl Anzahl der gewünschten Werbeflächen des Kunden
a Datenfeld für A-Kunden
b Datenfeld für B-Kunden
anzA Anzahl der im Datenfeld a enthaltenen Elemente
anzB Anzahl der im Datenfeld b enthaltenen Elemente

Die beiden Tabellen A-Kunden und B-Kunden haben die Felder Kundennr und Werbefolien; jeweils Ganzzahlen.

i = 0, j = 0, n = 0

anzA = GET_KUNDEN_NR(a, "A-Kunden")

anzB = GET_KUNDEN_NR(b, "B-Kunden")

solange n < anzahl

 solange a(i).Werbefolien < b(j).Werbefolien und n < anzahl und i < anzA

 Ausgabe a(i).Kundennr

 i = i + 1, n = n + 1

 solange a(i).Werbefolien >= b(j).Werbefolien und n < anzahl und j < anzB

 Ausgabe b(j).Kundennr

 j = j + 1, n = n + 1

Hinweis: Je nach zugrundeliegender Programmiersprache kann die Indizierung der Datenfelder bei 0 oder bei 1 beginnen.

(15 P.)

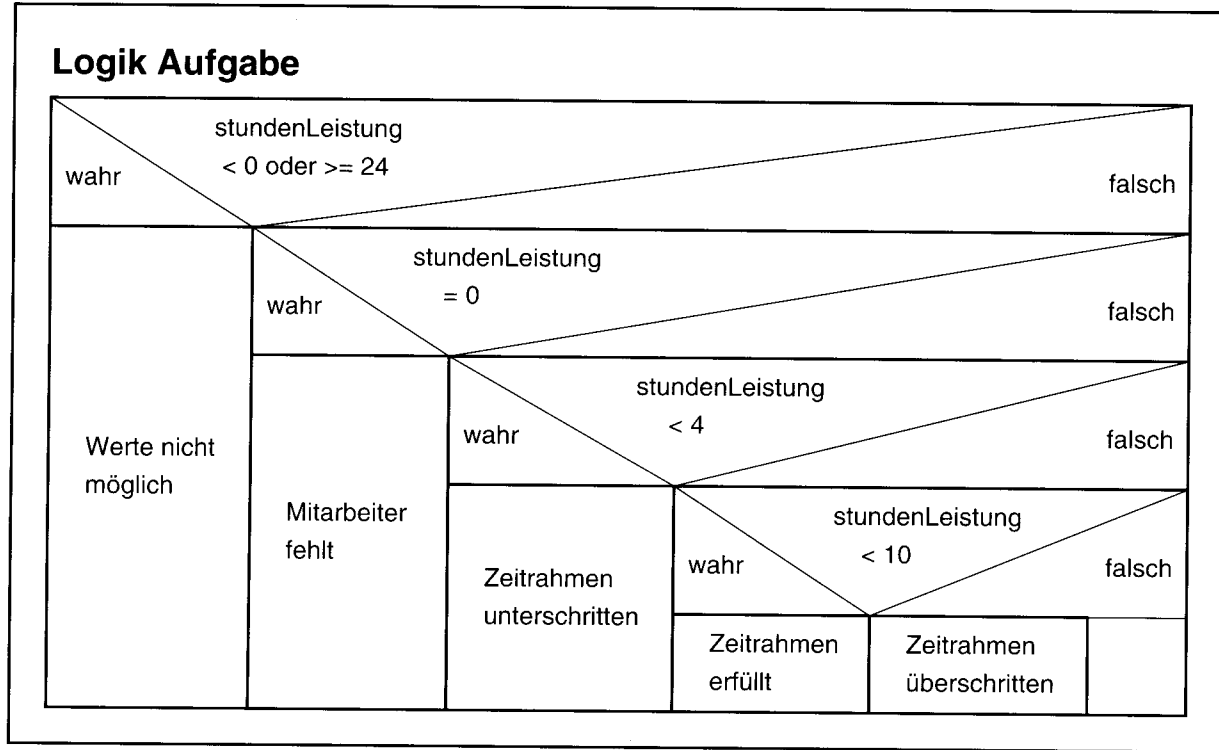
4. Handlungsschritt (20 Punkte)

- aa) Beim BlackBox-Verfahren wird anhand von Testdaten die Reaktion des Programms getestet.
Beim WhiteBox-Verfahren wird die interne Struktur des Programms analysiert.

(2 P.)

- ab) – Pfadabdeckungsgrad
– Namensräume von Variablen und Konstanten
– Einhaltung von Programmierungskonventionen
– Zulässige Schachtelungstiefe
– Kommentierungen

(4 P.)



bb) Schwachpunkte des Programmentwurfs:

- Gültigkeitsbereich kleiner 0 und größer 24 wird nicht überprüft.
- Es gibt keine bereichsabdeckende Prüfung (letzte Prüfung zählt).
- Abfrage < "4.00" ist keine numerische Konstante.
- Falscher Wert 10,1 wird abgefragt.
- Dezimalpunkt und Komma dürfen nicht zusammen verwendet werden.

(6 P.)

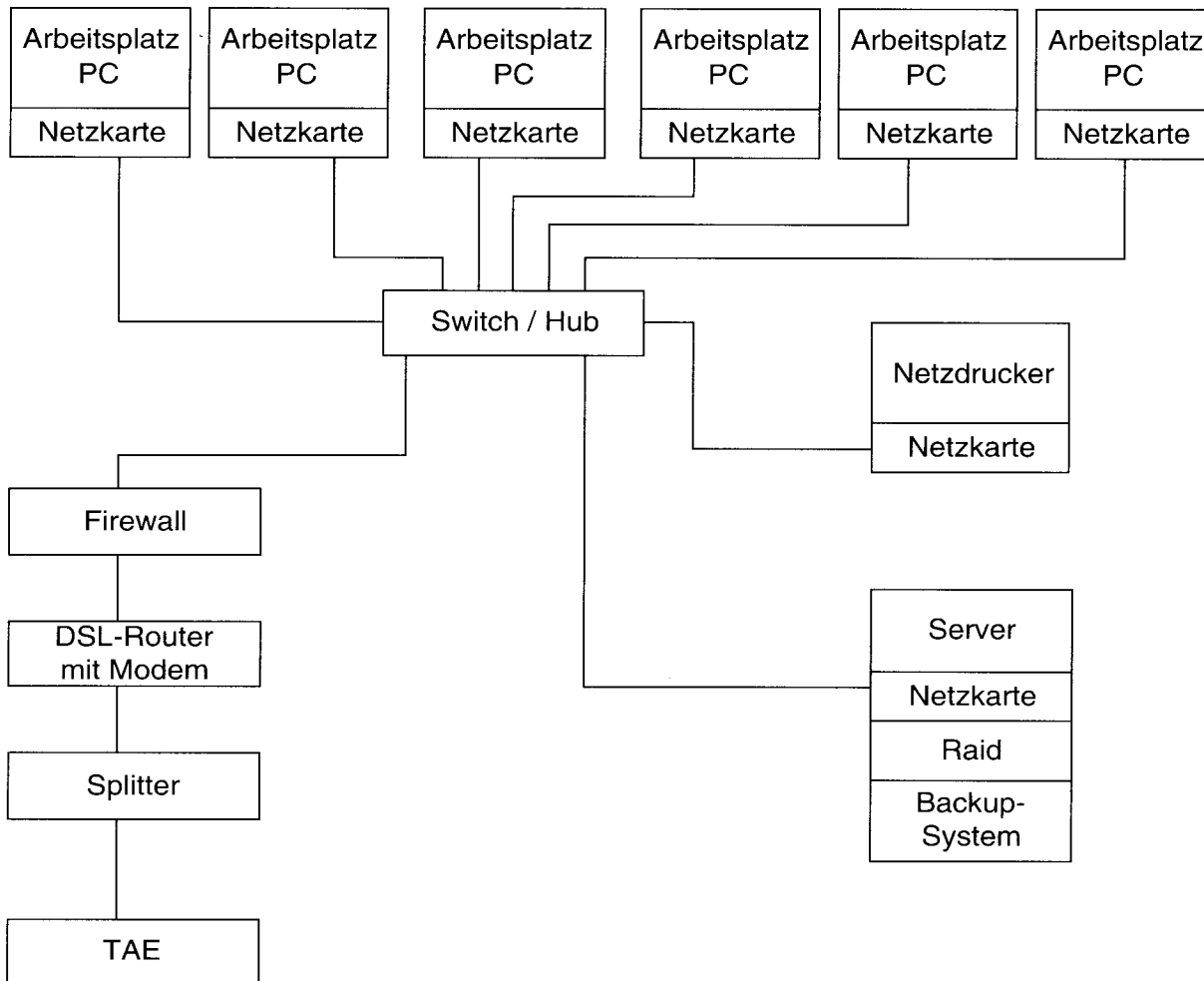
Zur Info: sinnvoller Javacode

```
// sinnvolle Version ==> Lösungshinweis
if (stundenLeistung < 0.0 || stundenLeistung >= 24.0)
{
    wertung = "Wert unmöglich!";
}
else
{
    if (stundenLeistung == 0.0)
    {
        wertung = "Mitarbeiter fehlt";
    }
    else
    {
        if (stundenLeistung <= 4.0)
        {
            wertung = "Zeitraahmen unterschritten";
        }
        else
        {
            if (stundenLeistung <= 10.0)
            {
                wertung = "Zeitraahmen eingehalten";
            }
            else
            {
                wertung = "Zeitraahmen ueberschritten";
            }
        }
    }
}
}
```

5. Handlungsschritt (20 Punkte)

a)

(8 P.)



b) Switch

- ermöglicht Sterntopologie
- hat mehrere Bandbreiten 10, 100, 1000 Mbit/s
- kanalisiert Netzlast
- segmentiert ein Netz in Kollisionsdomänen
- überträgt zwischen zwei Ports Pakete mit max. Geschwindigkeit

(3 P.)

Hub

- ermöglicht sternförmige Verbindung zwischen mehr als 2 Rechnern
- überträgt Daten mit 10 und / oder 100 Mbips (Dualspeed-Hub)
- arbeitet auf der 1. Schicht des OSI-Schichtmodells
- ist aktiv oder passiv
- ermöglicht Kaskadierung

(3 P.)

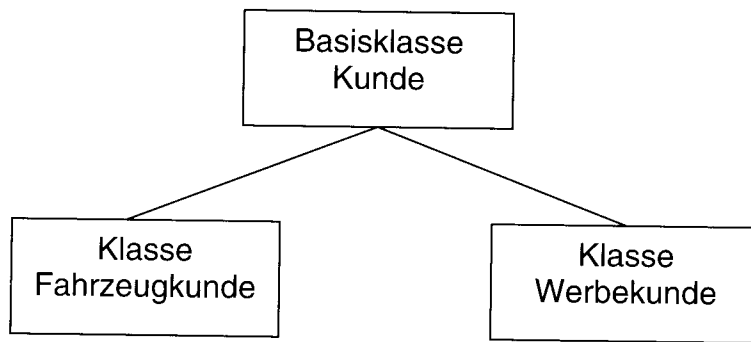
c) Eine Firewall überwacht die Kommunikation zwischen zwei Netzen. Sie wird zum Schutz gegen Angriffe aus einem Netz mit geringerem Schutzbedarf eingesetzt.

(6 P.)

6. Handlungsschritt (20 Punkte)

- a) Erstellen einer Basisklasse von der die beiden Klassen Werbekunde und Fahrzeugkunde abgeleitet werden. Das Datenfeld muss dann vom Typ der Basisklasse sein.

(8 P.)



b)

(12 P.)

- ba) – Die Fehler verursachenden Anweisungen müssen in einem try-Block untergebracht werden
– Ein Laufzeitfehler oder eine vom Entwickler gewollte Bedingung löst eine Ausnahme aus.
– Diese kann nun entweder von dem Programmteil, in dem sie ausgelöst wurde, in einem catch-Block behandelt werden, oder sie kann weitergegeben werden.
– Wird die Ausnahme weitergegeben, so hat der Empfänger der Ausnahme erneut die Möglichkeit, sie entweder zu behandeln oder selbst weiterzugeben.
– Wird die Ausnahme von keinem Programmteil behandelt, so führt sie zum Abbruch des Programms und zur Ausgabe einer Fehlermeldung.
- bb) Durch die Verwendung dieses Konzepts kann die Programmlogik von der Fehlerbehandlung getrennt werden; traditionell wird die Fehlerbehandlung in die Programmlogik integriert.

(8 P.)

(4 P.)