

# 1

Planen eines  
Softwareproduktes

**Teil 2 der Abschlussprüfung**

## Allgemeine Korrekturhinweise

Die Lösungs- und Bewertungshinweise zu den einzelnen Handlungsschritten sind als Korrekturhilfen zu verstehen und erheben nicht in jedem Fall Anspruch auf Vollständigkeit und Ausschließlichkeit. Neben hier beispielhaft angeführten Lösungsmöglichkeiten sind auch andere sach- und fachgerechte Lösungsalternativen bzw. Darstellungsformen mit der vorgesehenen Punktzahl zu bewerten. Der Bewertungsspielraum des Korrektors (z. B. hinsichtlich der Berücksichtigung regionaler oder branchenspezifischer Gegebenheiten) bleibt unberührt.

Zu beachten ist die unterschiedliche Dimension der Aufgabenstellung (nennen – erklären – beschreiben – erläutern usw.).

Für die Bewertung gilt folgender Punkte-Noten-Schlüssel:

Note 1	=	100 – 92 Punkte	Note 2	=	unter	92 – 81 Punkte	
Note 3	=	unter	81 – 67 Punkte	Note 4	=	unter	67 – 50 Punkte
Note 5	=	unter	50 – 30 Punkte	Note 6	=	unter	30 – 0 Punkte

## 1. Aufgabe (24 Punkte)

a) 4 Punkte

Erwartet werden Beschreibungen wie z. B.:

- Das Verkaufspersonal frühzeitig in das Projekt einbinden
- Die Befürchtungen des Verkaufspersonals identifizieren und gemeinsam Lösungen finden, wie z. B. einer Zusicherung des Arbeitsplatzes
- Einbindung der Projektverantwortlichen, z. B. der Geschäftsführung der Parkanlage
- u. a.

b) 8 Punkte

Folgende Ursachen und Auswirkungen sind zum Beispiel denkbar:

Risiko	Ursache	Auswirkung
<b>Unterschätzung des Entwicklungsaufwandes der App</b>	Keine Erfahrung bei der AMAG Soft GmbH über die Prozesse der Parkanlage	Projektende verzögert und das Projekt verteuert sich
<b>Inkompatible Software-Schnittstellen</b>	Fehler in der Anforderungsanalyse	Verzögerung und Mehrkosten
<b>Widerspruch des Personalrates</b>	Der Personalrat befürchtet Arbeitsplatzkontrollen oder sogar Verlust von Arbeitsplätzen	Projektende verzögert sich bzw. Projekt wird abgebrochen

c) 4 Punkte

Mögliche **funktionale Anforderungen** können z. B. sein:

- Lauffähigkeit auf gängigen mobilen Systemen (iOS, Android, etc.)
- Berechnung der Einzelpreise und des Gesamtpreises für die Buchung
- Authentifizierung gewährleisten
- Integrierte Bezahlösung
- Push-Benachrichtigung bei Abweichungen z. B. der Eintrittszeiten

Mögliche **nichtfunktionale Anforderungen** können z. B. sein:

- Verfügbarkeit
- Zuverlässigkeit
- Usability
- Fehlertoleranz
- Sicherstellung der Datenintegrität

d) 4 Punkte

Erwartet wird die Nennung von zwei Kostenarten mit Beispiel, wie z. B.:

- Personalkosten  
Kosten für externes Projektpersonal der AMAG Soft GmbH
- Softwarekosten  
Software für die Erstellung der App  
Softwarelizenzen
- Hardwarekosten  
Kamerasysteme für den Eingangsbereich  
Lesegeräte für die digitalen Tickets
- Kommunikationskosten  
Leitungs-/Übertragungskosten
- Qualifizierungskosten  
Mitarbeiterqualifizierung  
Kosten für Trainer

e) 4 Punkte

Erwartet werden Beschreibungen wie z. B.

- Verwendung zertifizierter Softwareentwicklungswerkzeuge
- Definition und Verwendung standardisierter Testumgebungen, -verfahren und -daten
- Einhaltung von gesetzlichen Vorgaben zum Datenschutz
- Auswahl und Verwendung geeigneter standardisierter Softwareentwicklungsmodelle wie z. B. SCRUM

## 2. Aufgabe (26 Punkte)

a) 4 Punkte, je 1 Punkt pro Merkmal, je 1 Punkt pro Erklärung

### Functionality:

The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.

Die Fähigkeit eines Softwareproduktes Funktionen zur Verfügung zu stellen, welche die vorgegebenen und implizierten Anforderungen erfüllen, wenn es unter den spezifizierten Bedingungen verwendet wird.

### Reliability:

The capability of the software product to maintain a specified level of performance when used under specified conditions.

Die Fähigkeit eines Softwareproduktes ein spezifiziertes Performanz Level aufrecht zu erhalten, wenn es unter den spezifizierten Bedingungen verwendet wird.

### Usability:

The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.

Die Fähigkeit eines Softwareproduktes verständlich, lernbar, benutzbar und attraktiv für die Benutzer zu sein, wenn es unter den spezifizierten Bedingungen verwendet wird.

### Efficiency:

The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions.

Die Fähigkeit eines Softwareproduktes eine bezüglich der aufgewendeten Ressourcen angemessene Performanz zur Verfügung zu stellen, wenn es unter den spezifizierten Bedingungen verwendet wird.

### Portability:

The capability of the software product to be transferred from one environment to another.

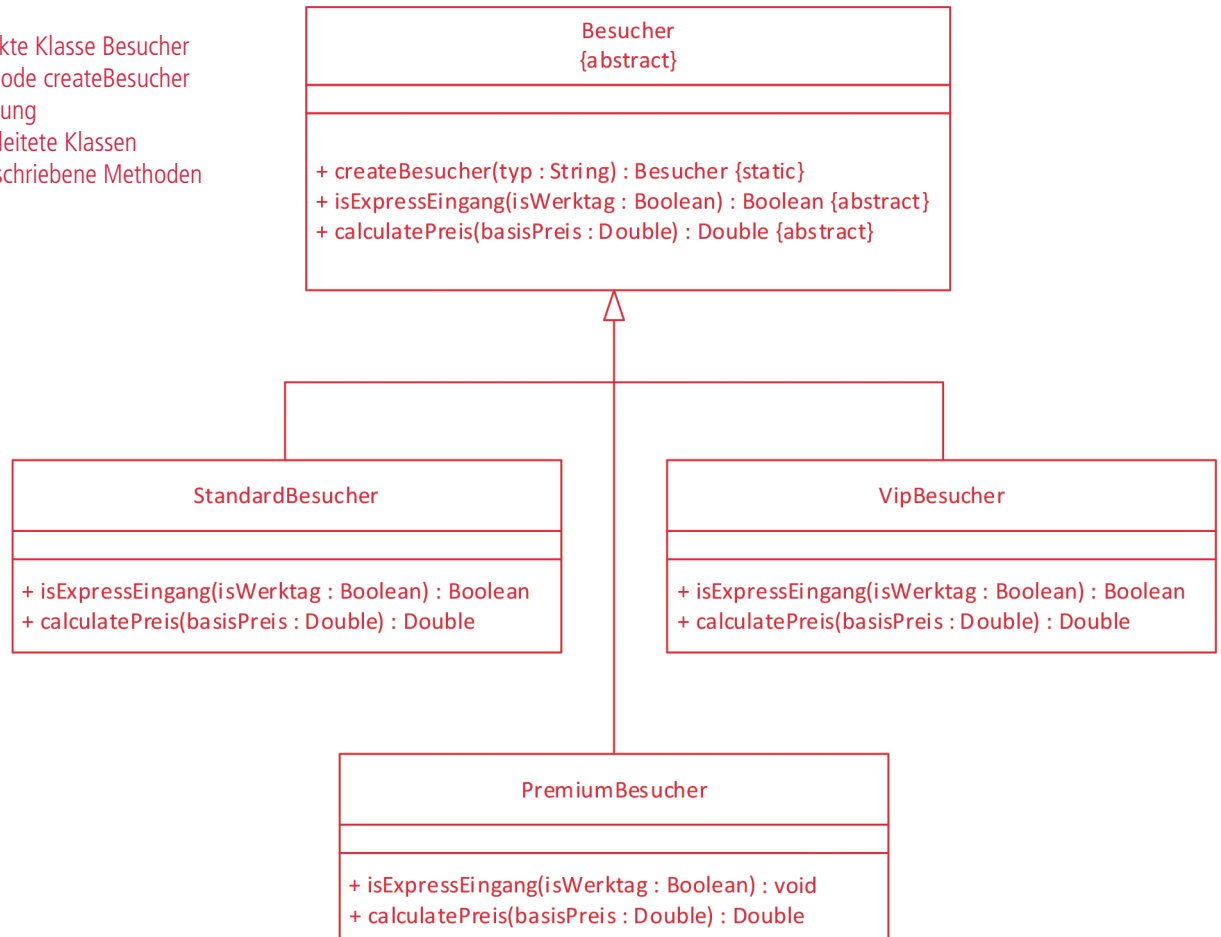
Die Fähigkeit eines Softwareproduktes zur Übertragung auf eine andere Umgebung.

ba) 4 Punkte

Die Polymorphie der objektorientierten Programmierung ist eine Eigenschaft, die häufig im Zusammenhang mit Vererbung und Schnittstellen (Interfaces) auftritt. Eine Methode ist polymorph, wenn sie in verschiedenen Klassen die gleiche Signatur hat, jedoch erneut implementiert oder erweitert wird. Gibt es in einem Vererbungsbaum einer Klassenhierarchie mehrere Methoden auf unterschiedlicher Hierarchieebene, jedoch mit gleicher Signatur, wird erst zur Laufzeit bestimmt, welche der Methoden für ein gegebenes Objekt verwendet wird (Dynamisches Binden).

bb) 9 Punkte

- 1 Punkt abstrakte Klasse Besucher
- 2 Punkte Methode createBesucher
- 1 Punkt Vererbung
- 3 Punkte abgeleitete Klassen
- 2 Punkte überschriebene Methoden



bc) 5 Punkte

1 Punkt Selektor typ, 1 Punkt Mehrfachauswahlstruktur, 3 Punkte Rückgaben

**+ createBesucher(typ : String) : Besucher {static}**

typ		
"STANDARD"	"PREMIUM"	"VIP"
Rückgabe: new StandardBesucher()	Rückgabe: new PremiumBesucher()	Rückgabe: new VipBesucher()

oder

```
public static Besucher createBesucher(String typ) {  
    switch (typ) {  
        case "STANDARD":  
            return new StandardBesucher();  
        case "PREMIUM":  
            return new PremiumBesucher();  
        case "VIP":  
            return new VipBesucher();  
    }  
}
```

bd) 4 Punkte

2 Punkte je Methode

**+ isExpressEingang(isWerktag : Boolean) : Boolean**

Rückgabe: isWerktag

**+ calculatePreis(basisPreis : Double) : Double**

Rückgabe: basisPreis\*0.95

Oder

```
@Override  
public Boolean isExpressEingang(Boolean isWerktag) {  
    return isWerktag;  
}  
  
@Override  
public Double calculatePreis(Double basisPreis) {  
    return basisPreis*0.95;  
}
```

### 3. Aufgabe (26 Punkte)

- a) Die deutschen Funktionsnamen sind ebenfalls als richtig zu werten.

VLOOKUP => SVERWEIS

IF => WENN

WEEKDAY => WOCHENTAG

- aa) 2 Punkte

$C3 = A2 + 1$

- ab) 5 Punkte

$E3 = \text{VLOOKUP}(C3; \text{Wetterdaten!B3:C17}; 2; \text{FALSE})$

Hinweis: Bei Verwendung geschachtelter IF-Funktionen muss die Formel vollständig ausgeschrieben werden.

- ac) 5 Punkte

$F3 = H3 * \text{IF}(\text{WEEKDAY}(C3) \leq 5; 0,9; 1)$

Oder  $=H3 * I3$

Mit der Formel  $= \text{IF}(\text{WEEKDAY}(C3) \leq 5; 0,9; 1)$  in I3

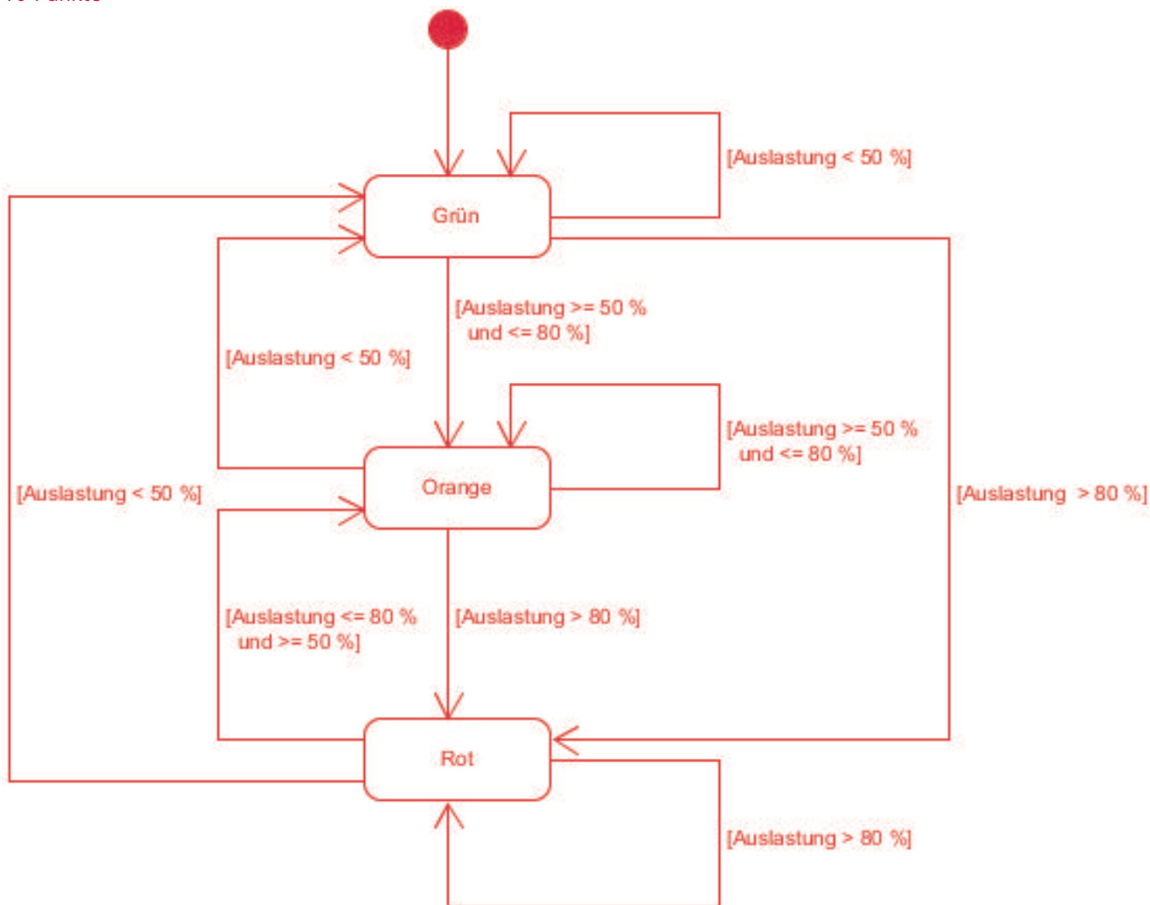
- ba) 2 Punkte

Ein Tabellenkalkulationsprogramm kann z. B. zur Arbeitszeitberechnung eingesetzt werden.

- bb) 2 Punkte

Z. B. werden bei umfangreichem Datenbestand und vielfachen Abhängigkeiten der Berechnungen untereinander die Anwendungen schlecht wartbar.

- c) 10 Punkte



Die Selbstreferenzen sind optional und in der Bewertung nicht zu berücksichtigen.

#### 4. Aufgabe (24 Punkte)

a) 4 Punkte

Erwartet wird, dass der Prüfling die Anwendung einer symmetrischen Verschlüsselung vorschlägt. Diese erfordert, verglichen mit der asymmetrischen Verschlüsselung, nur wenig Rechenleistung. Außerdem ist in diesem Fall jeder Sensor zuvor bekannt, womit die Nutzung eines Pre-Shared-Keys zumutbar ist. Sollte der Prüfling anstelle des Verfahrens eine Implementierung (bspw. AES) benennen, wird abhängig von der Begründung vorgeschlagen, dies mit nur mit geringem Punktabzug zu bewerten.

b) 4 Punkte

Da es sich hier um Daten handelt, bei denen primär die Authentizität und Integrität relevant sind, während die Vertraulichkeit nicht stark zu gewichten ist, dürfte dieser Vorschlag als sinnvoll betrachtet werden, da ein Hash oftmals einen geringeren Rechenaufwand bedeutet als eine effektive Verschlüsselung.

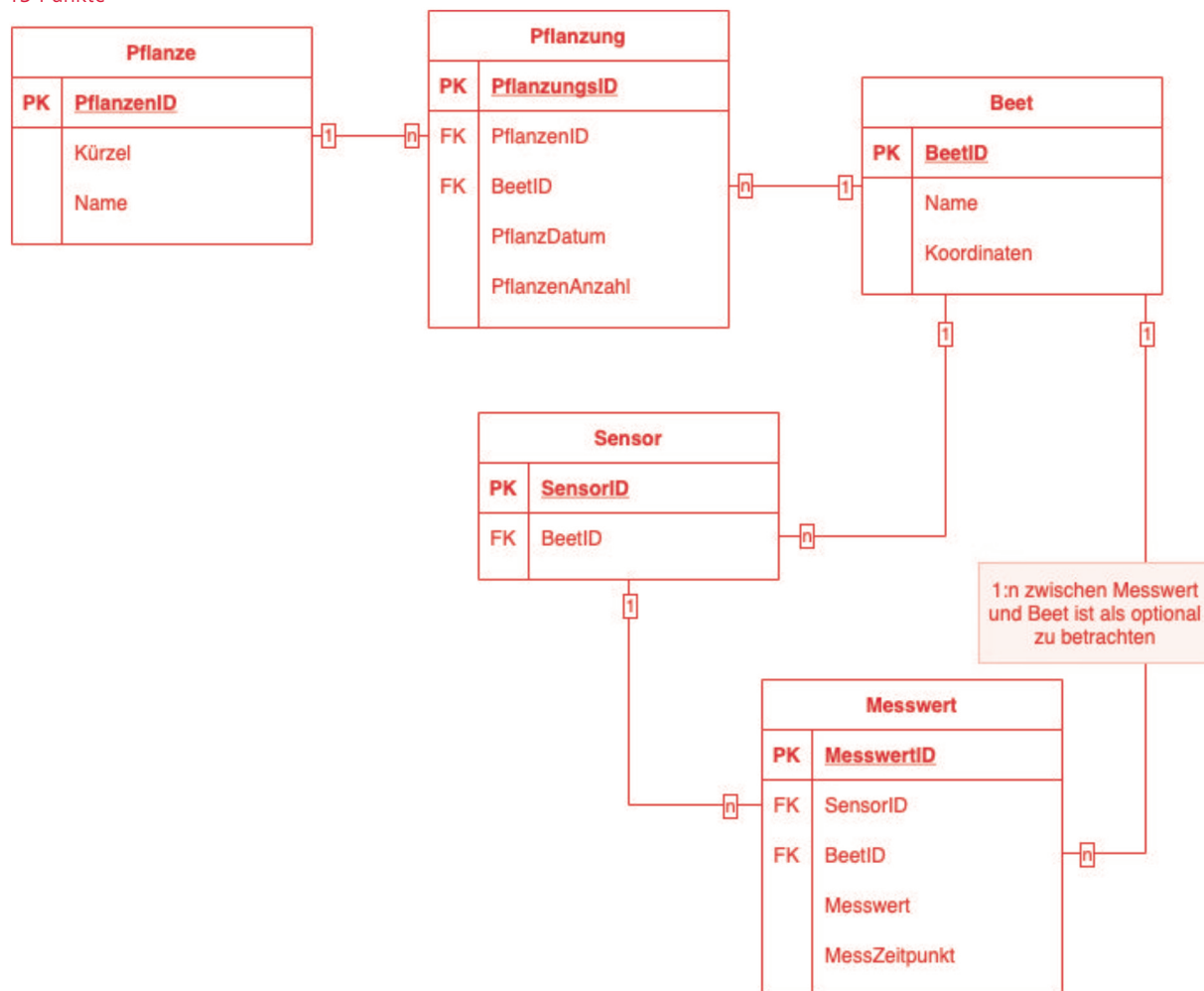
Eine Ablehnung des Vorschlags kann bei entsprechender Begründung auch als richtig gewertet werden. Diese könnte zum Beispiel eine begründete höhere Gewichtung der Vertraulichkeit betreffen.

ca) 3 Punkte

Lösungsvorschlag:

Es können inkonsistente Daten entstehen. Dies ist bspw. der Fall, wenn ein Beet umbenannt wird und Altdaten nicht vollständig aktualisiert werden (Änderungsanomalie).

cb) 13 Punkte



Bewertungsvorschlag:

Je Tabelle 1 Punkt (5 Punkte)

Je Fremdschlüssel 1 Punkt (4 Punkte)

Je Beziehung mit Kardinalität 1 Punkt (4 Punkte)



