

Klausur Softwaretechnik

15.03.2007

Prof. Dr. Walter F. Tichy
Dipl.-Inform. T. Gelhausen
Dipl.-Inform. G. Malpohl

Musterlösung

Punkte	Note
0	5,0
18	4,0
21	3,7
24	3,3
27	3,0
30	2,7
33	2,3
36	2,0
39	1,7
42	1,3
45	1,0

Aufgabe 1: Aufwärmen (18P)

a.) Kreuzen Sie an, ob die Aussage wahr oder falsch ist. (8P)

Hinweis: Jedes korrekte Kreuz zählt 0,5 Punkte, jedes falsche Kreuz bewirkt 0,5 Punkte Abzug! Die Teilaufgabe wird mit mindestens 0 Punkten bewertet.

Wahr	Falsch	Aussage
	<input checked="" type="checkbox"/>	Software ist leichter zu ändern als ein physikalisches Produkt vergleichbarer Komplexität.
<input checked="" type="checkbox"/>		Das Lastenheft beschreibt die Eigenschaften, die das Produkt aus Sicht des Kunden erfüllen soll, während das Pflichtenheft zusätzlich die genauen Vorschriften für den Entwickler enthält.
	<input checked="" type="checkbox"/>	Bei der Anforderungvalidierung in der Planungsphase wird nicht überprüft, ob die Anforderungen erfüllt werden können.
	<input checked="" type="checkbox"/>	Eine vollständige Methodensignatur in UML besteht aus dem Rückgabotyp und der Parameterliste einer Funktion.
<input checked="" type="checkbox"/>		Klassendiagramme sind Multi-Hyper-Graphen.
	<input checked="" type="checkbox"/>	Wird das Attribut „Multiplizität“ einer Assoziation in einem UML-Klassendiagramm nicht angegeben, bedeutet dies „nicht spezifiziert“ und damit spezifikationsgemäß „beliebig viele, aber mindestens 1“.
	<input checked="" type="checkbox"/>	In Java muss eine Klasse, die eine Schnittstelle implementiert, alle in der Schnittstelle vorgegebenen Methoden implementieren.
	<input checked="" type="checkbox"/>	Auf ein private -Attribut eines Exemplars einer Klasse kann nur das Exemplar selber zugreifen.
<input checked="" type="checkbox"/>		In UML-Aktivitätsdiagrammen unterscheidet man zwischen Objekt- und Kontrollflüssen.
	<input checked="" type="checkbox"/>	Die Diagramme „Kollaborationsdiagramm“, „Zeitdiagramm“, „Zustandsdiagramm“ und „Sequenzdiagramm“ zählen zur Gruppe der „Interaktionsdiagramme“.
<input checked="" type="checkbox"/>		Bei Modulen gibt es zu jeder Definitionseinheit nicht mehr als eine Implementierungseinheit.
<input checked="" type="checkbox"/>		Das Entwurfsmuster „Iterator“ ermöglicht „polymorphe Iteration“.
	<input checked="" type="checkbox"/>	Kombiniert man das Entwurfsmuster „Befehl“ mit dem Entwurfsmuster „Stellvertreter“, erhält man einen Makrobefehl.
<input checked="" type="checkbox"/>		Die Hauptaufgaben der Implementierungsphase des Wasserfallmodells sind die Umsetzung der Spezifikation in korrekte, ablauffähige Programme, die Dokumentation und das Testen der Komponenten.
<input checked="" type="checkbox"/>		Bei der Feinoptimierung ändern sich lediglich Konstanten des Laufzeitverhaltens, die asymptotische Laufzeit des Algorithmus bleibt unverändert.
	<input checked="" type="checkbox"/>	Beim Planungsspiel im XP gilt: „Das Entwicklungsteam trifft die geschäftsrelevanten Entscheidungen.“

- b.) Definieren Sie die Begriffe „Signaturvererbung“ und „Implementierungsvererbung“ aus der objektorientierten Programmierung. In welchem Zusammenhang stehen diese beiden Eigenschaften? (3P)

- **Signaturvererbung:** Eine in der Oberklasse definierte und (evtl.) implementierte Methode überträgt nur ihre Signatur auf die Unterklasse. (1 P)
- **Implementierungsvererbung:** Eine in der Oberklasse definierte und implementierte Methode überträgt ihre Signatur und ihre Implementierung auf die Unterklasse. (1 P)
- Implementierungsvererbung geht nicht ohne Signaturvererbung, umgekehrt geht es aber schon. (1 P)

- c.) Was ist der Nachteil von Zyklen in der Benutzt-Relation zwischen Modulen? (1P)

Die einzelnen Module können nicht nacheinander implementiert und getestet werden, weil ihr Funktionieren von einer korrekten Implementierung aller Module des Zyklus abhängt. Oder: „Nothing works until everything works.“

- d.) Definieren Sie die zwei Begriffe „Wartung“ und „Pflege“ von Softwareprodukten. (1P)

- **Wartung:** Behebung von Fehlerursachen von in Betrieb befindlichen Software-Produkten, wenn die Fehlerwirkung bekannt ist (ereignisgesteuert). (0,5 P)
- **Pflege:** Durchführung von Änderungen und Erweiterungen von in Betrieb befindlichen Software-Produkten, wenn die Art der gewünschten Änderungen/Erweiterungen feststeht. (0,5 P)

- e.) Nennen sie jeweils zwei Qualitätssicherungsverfahren aus der Kategorie der dynamischen und der statischen Verfahren. (2P)

Dynamische Verfahren:

1. _____ . z.B. Strukturtests

2. _____ . z.B. Funktionale Tests, Leistungstests

Statische Verfahren:

1. _____ . z.B. Manuelle Prüfmethode

2. _____ . z.B. Prüfprogramme (stat. Analyse)

- f.) In der objektorientierten Analyse und in UML gibt es das Konzept der Assoziation.
(i.) Was bietet dieses Konzept für einen semantischen Mehrwert gegenüber den einfachen Referenzen, wie sie z.B. in Java zu finden sind? (ii.) Welche Probleme hat man ohne dieses Konzept der Assoziation? (iii.) Java kennt keine Assoziationen. Wie setzt man also Assoziationen in Java um? (3P)

- (i.) Transaktion oder ACID (1P)
(ii.) Problem ohne: inkonsistente Zustände möglich (1P)
(iii.) ACID-Idee muss mit geschützten Variablen und Code sichergestellt werden, sämtlicher Code, der auf die Variablen zugreift muss das berücksichtigen (1P)

Aufgabe 2: Anforderungserhebung (12P)

Ihr Chef hat beim Golf spielen einen Auftrag an Land gezogen: Ihre Firma (insges. 15 Mitarbeiter) soll den Webauftritt eines mittelständigen Reiseunternehmens komplett neu gestalten. Das Unternehmen hat bisher nur einige wenige statische Seiten und möchte gerne „einen komfortablen, interaktiven Reisekatalog“ anbieten. Für den Vertragsabschluss sollen Sie nun die Anforderungen erheben. Ihr Chef gibt Ihnen eine Woche Zeit.

a) Welche Erhebungstechniken haben wir in der Vorlesung kennen gelernt? (4P)

- Introspektion (0,5P)
- Dokumenten-/Datenanalyse, Altanwendung (0,5P)
- Interviews (0,5P)
- Umfragen/Fragebögen (0,5P)
- Gruppentechniken, Brainstorming, JAD, RAD, Planspiele (0,5P)
- Prototypen (0,5P)
- Anwendungsfälle (Use Cases) (0,5P)
- Ethnografie (0,5P)

b) Diskutieren Sie für jede einzelne Technik, ob und warum (oder warum nicht) sie sich **im beschriebenen Szenario** anbietet. (8P)

Jeweils 1P bei geeigneter Argumentation mit den aus der Vorlesung bekannten Eigenschaften, Vor- und Nachteilen.

Introspektion: Vorteile: Einfach, schnell, billig, ohne Missverständnisse. Nachteile: Oft nicht anwendbar, kann in die falsche Richtung gehen. Eignung: mittel.

Dokumentenanalyse: Vorteile: Kann schnell und billig sein. Nachteile: Oft nicht anwendbar, Daten könne verzerrt oder veraltet sein, Zu viele Daten, zu viel Detail. Eignung: mittel bis gut.

Interviews: Vorteile: Reiche Quelle kontextbezogener Information. Nachteile: Große Menge qualitativer Daten kann schwierig zu analysieren sein, Vergleich von Interviewpartnern schwierig, Interview-Leitung ist eine spezielle Fähigkeit. Achtung bei: Still-

schweigendem Wissen/Annahmen, Fragen aus dem Zusammenhang, Interviewers Haltung kann Antworten verzerren. Eignung: gut.

Umfragen: Vorteile: rasch Information von vielen Personen, kann auch über das Web durchgeführt werden, kann Meinungen und Haltungen abfragen. Nachteile: Einfache Fragekategorien bieten wenig Kontext, keine Möglichkeit für die Benutzer, ihre wahren Bedürfnisse anzugeben. Achtung bei: Nicht repräsentativer Stichprobe, kleine Stichprobe, Suggestivfragen, mehrdeutige Fragen. Fragebögen müssen getestet werden. Eignung: schlecht.

Gruppentechniken: Vorteile: Natürlichere Interaktion als im Interview, kann Reaktionen auf Vorschläge prüfen. Nachteile: unnatürlichen Gruppen, Gruppenzwang, liefert evtl. nur oberflächliche Information, Moderator muss ausgebildet sein. Achtung bei: Unrepräsentativer Stichprobe, Dominanz und Unterwerfung. Eignung: mittel bis gut.

Anwendungsfälle: Vorteile: Leicht zu schreiben und zu verstehen, hilft bei der Bestimmung der Systemgrenzen. Nachteile: Anwendungsfälle erfassen kaum Domänenwissen, sind selten exakt. Eignung mittel.

Ethnographie: Vorteile: Zuverlässig, reichhaltiger Kontext, liefert Details, die andere Methoden nicht liefern können. Nachteile: Extrem zeitaufwendig, Aufzeichnungen können wegen des Umfangs schwierig zu analysieren sein. Achtung bei: Übernahme der Gruppenhaltung, Konformitätszwang. Eignung: überhaupt nicht.

Aufgabe 3: Kontrollflussorientierter Strukturtest (13P)

Gegeben ist die folgende Java-Funktion:

```
int zaehleZeichen(String text, char zeichen) {
    int zaehler = 0;
    if (text != null) {
        for (int i=text.length()-1; i>=0; i--) {
            if (text.charAt(i) == zeichen) {
                zaehler++;
            }
        }
    }
    return zaehler;
}
```

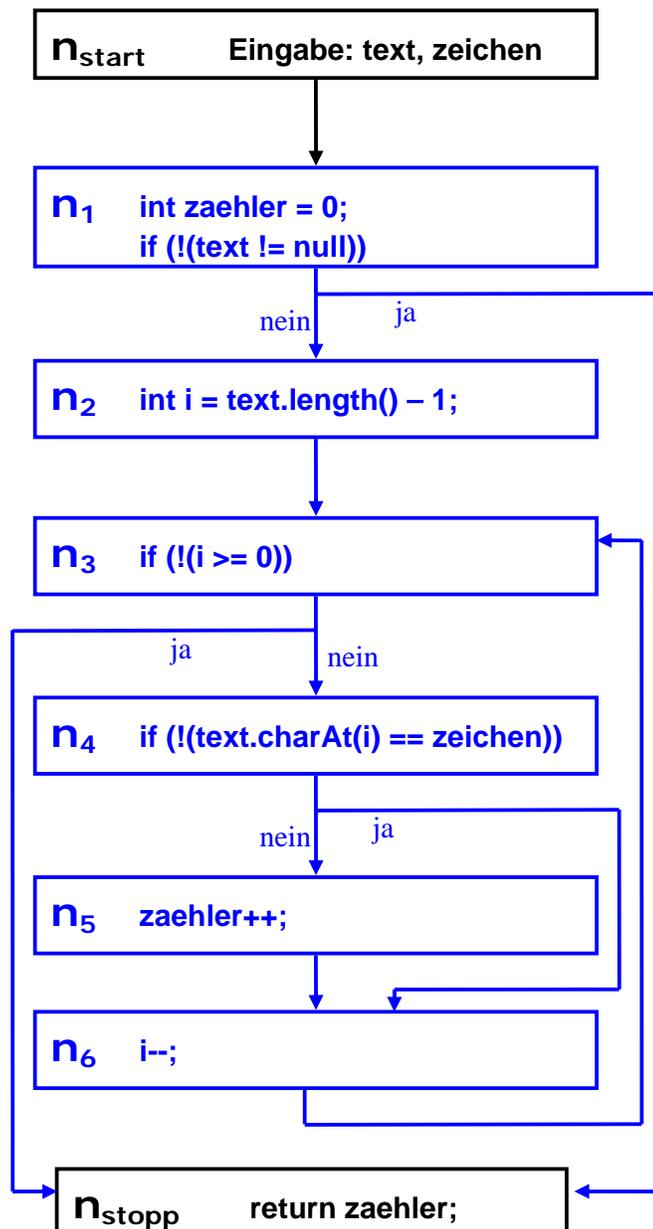
- a.) Wandeln Sie obiges Programm mit einer strukturerhaltenden Transformation in eine der Definition der Vorlesung entsprechenden Zwischensprache um. (3P)

Eingabeparameter: text, zeichen

```
10:    int zaehler = 0;
20:    if (!(text != null)) goto 90;
30:    int i = text.length() - 1;
40:    if (!(i >= 0)) goto 90;
50:    if (!(text.charAt(i) == zeichen)) goto 70;
60:    zaehler++;
70:    i--;
80:    goto 40;
90:    return zaehler;
100:
110:
120:
```

-1 P pro Fehler

- b.) Erstellen Sie den Kontrollflussgraphen der Funktion `zaehleZeichen()`. Wenden Sie dabei das aus der Vorlesung bekannte Verfahren an. (4P)



kleine Fehler (zB. fehlende Bedingung): -0,5 P

- c.) Nennen Sie eine minimale Testfall-Menge für die Anweisungsüberdeckung und geben Sie zu jedem Testfall den durchlaufenen Pfad an. (1P)

("a", 'a') start, 1, 2, 3, 4, 5, 6, 3, stopp
0,5 P 0,5 P

- d.) Nennen Sie eine minimale Testfall-Menge für die Zweigüberdeckung des obigen Programms. Geben Sie zu jedem Testfall den durchlaufenen Pfad an. (2P)

(null, 'a') start, 1, stopp 2x0,5 P
("ab", 'a') start, 1, 2, 3, 4, 6, 3, 4, 5, 6, 3, stopp 2x0,5 P

bei nicht minimaler Testfallmenge: max. 1 P

- e.) Nennen Sie die Teilpfade der Schleifenquerer (Boundary-Interior-Test). (1P)

3, 4, 5, 6 und 3, 4, 6 2x0,5 P
Bei vollständigen Pfaden: 0 P

- f.) Geben Sie eine Testfallmenge für den Grenztest (Boundary-Interior-Test) an. (1P)

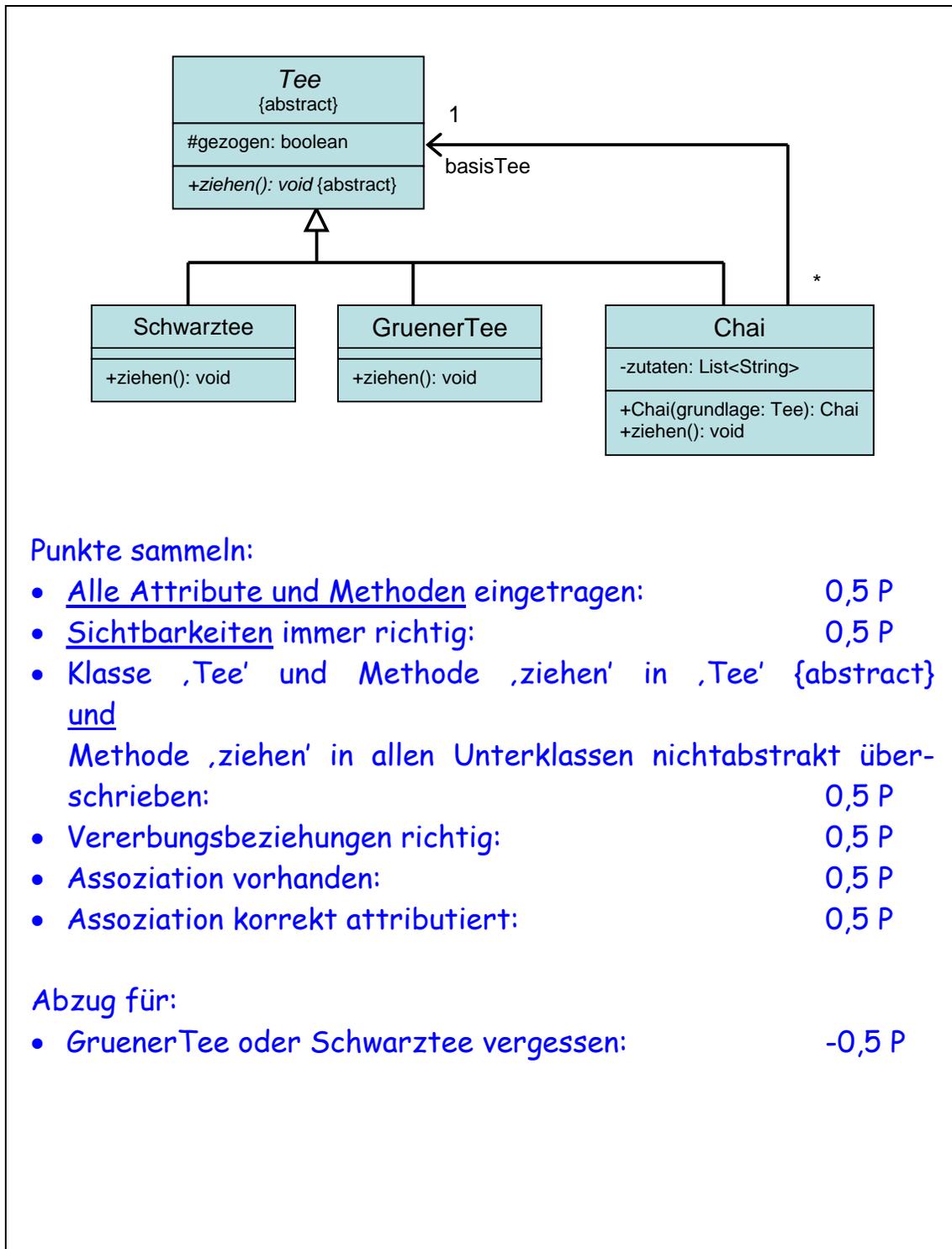
z.B. { ("a", 'a'), ("a", 'b') }. Ein Testfall durchläuft die Schleife genau 1x und deckt genau einen Schleifenquerer ab. Alle Schleifenquerer müssen abgelaufen werden. 2x0,5 P

- g.) Geben Sie eine Testfallmenge für den Interieurtest (Boundary-Interior-Test). (1P)

z.B. { ("aa", 'a'), ("aa", 'b') }. Jeder Testfall durchläuft die Schleife min 2x. Im 2. Schleifendurchlauf müssen sie unterschiedliche Schleifenquerer benutzen. (1P, ganz oder gar nicht)

- c.) Geben Sie das UML-Klassendiagramm des hinter dem Code stehenden Modells an. Ihr Diagramm soll alle Klassen (außer der Klasse „Test“) mit allen Attributen, Methoden (incl. Sichtbarkeit etc.) und Assoziationen enthalten. (3P)

Hinweis: Achten Sie darauf, Attribute und Assoziationen nicht zu verwechseln!



Punkte sammeln:

- Alle Attribute und Methoden eingetragen: 0,5 P
- Sichtbarkeiten immer richtig: 0,5 P
- Klasse ‚Tee‘ und Methode ‚ziehen‘ in ‚Tee‘ {abstract} und Methode ‚ziehen‘ in allen Unterklassen nichtabstrakt überschrieben: 0,5 P
- Vererbungsbeziehungen richtig: 0,5 P
- Assoziation vorhanden: 0,5 P
- Assoziation korrekt attributiert: 0,5 P

Abzug für:

- GruenerTee oder Schwarztee vergessen: -0,5 P

- d.) Wie heißt (i.) das übergeordnete Entwurfsmuster zu dem im angegebenen Programm implementierten? Was ist (ii.) der Unterschied? („Oberklasse“ ist nicht gemeint!) (2P)

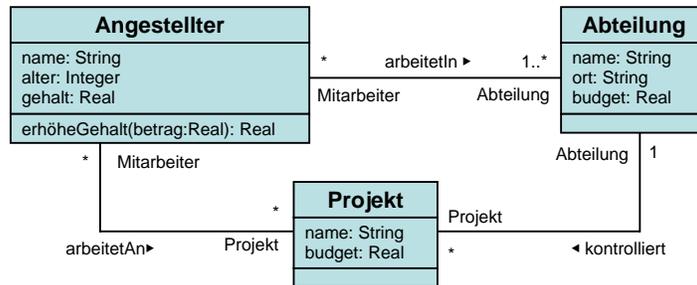
- | | | |
|-------|--|-----|
| (i.) | Stellvertreter | 1 P |
| (ii.) | Dem (allgemeineren) Stellvertreter fehlt gegenüber dem (spezielleren) Dekorierer die Forderung nach der zusätzlichen Zuständigkeit | 1 P |

- e.) Was haben (i.) das im angegebenen Programm implementierte Entwurfsmuster und der „Adapter“ gemeinsam? Und (ii.) wie unterscheiden sich die beiden? (2P)

- | | | |
|-------|---|-----|
| (i.) | Ziel: Transparente Entkopplung einer bestimmten <u>In-</u>
<u>stanz</u> gegenüber dem Rest des Programms | 1 P |
| (ii.) | Der Adapter ändert im Gegensatz zum Dekorierer die Schnittstelle des verborgenen Objektes | 1 P |

Aufgabe 5: OCL (7P)

Gegeben seien das nebenstehende UML-Klassendiagramm und der folgende OCL-Ausdruck. In der Vorlesung wurde gezeigt, wie Klassendiagramme in Java umgesetzt werden. Was ist hingegen bei der Umsetzung des gegebenen OCL-Ausdrucks in Java zu tun?



Hinweise: Nennen Sie alle Stellen, an denen Maßnahmen zu treffen sind. Beachten Sie, dass sich die Zusicherung auch auf etwaige Verwaltungsmethoden der Assoziationen auswirken könnte. Sie brauchen in dieser Aufgabe keinen konkreten Code anzugeben, beschreiben Sie die Maßnahmen abstrakt.

```

context Abteilung inv:
  sel f. Mitarbeiter->forall (a1, a2 |
    a1. Projekt->size() > a2. Projekt->size()
  ) implies a1. gehalt > a2. gehalt
  
```

- Alle Zugriffe auf die Instanzvariable „gehalt“ müssen kontrolliert werden. Daher sollte sie, wie die Instanzvariablen zum Umsetzen der Assoziationen auch, „private“ gemacht werden. (Der Zugriff erfolgt nur noch über öffentliche Zugriffsmethode(n).) (1P)

Was muss in diesen Methoden passieren?

- Bedingung wird durch Vergleich mit Menge aller Mitarbeiter der gleichen Abteilung geprüft, (if-Ausdruck oder Zusicherung), (Ausnahme oder Fehler oder Standardwert) (2 von 3: 1P)
- Liste bekommt man über die Abteilung des Mitarbeiters (1P)

In welche Methoden muss diese Prüfung stattfinden?

- Gehalt festsetzen/ändern („erhöheGehalt“) (1P)
- Projekt (bei einem Mitarbeiter) hinzufügen (1P)
- Projekt (bei einem Mitarbeiter) löschen (1P)
- Hinzufügen eines Mitarbeiters zu einer Abteilung (1P)

1P wenn nur/zu allgemein