

Klausur Softwaretechnik

16.03.2005

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

Hier das Namensschild aufkleben.

Zur Klausur sind keine Hilfsmittel und kein eigenes Papier zugelassen. Die Bearbeitungszeit beträgt 60 Minuten. Die Klausur ist vollständig und geheftet abzugeben.

Aufgabe	1	2	3	4	5	6	Σ
Maximal	8	11	11	13	8	9	60
K1							
K2							
K3							

Punkte:

Note:

Aufgabe 1: Aufwärmen (3+2+2+1=8P)

- a.) Kreuzen Sie an, ob die Aussage Wahr oder Falsch ist. (3P)

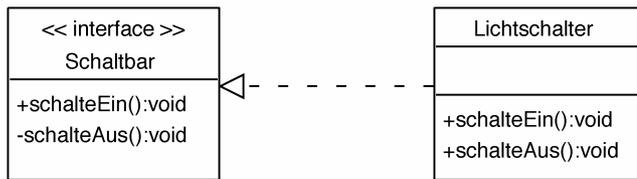
Hinweis: Jedes korrekte Kreuz zählt 0,5 Punkte, jedes falsche Kreuz bewirkt 0,5 Punkte Abzug! Die Teilaufgabe wird mindestens mit 0 Punkten bewertet.

WAHR	FALSCH	Aussage
<input type="checkbox"/>	<input type="checkbox"/>	Der einzige Nachteil eines Parallellaufs in der Einführungsphase sind die erhöhten Kosten für die doppelt benötigten Systemressourcen.
<input type="checkbox"/>	<input type="checkbox"/>	In Java wird das Entwurfsmuster Null-Objekt durch das Konzept <code>null</code> realisiert.
<input type="checkbox"/>	<input type="checkbox"/>	Im Harel-Automaten sind ϵ -Übergänge (Zustandsübergänge ohne auslösendes Ereignis) zulässig.
<input type="checkbox"/>	<input type="checkbox"/>	Den kompletten Lebenszyklus eines Objektes, von der Speicherallokation bis zur -freigabe, nennt man <i>α-λ-Zyklus</i> .
<input type="checkbox"/>	<input type="checkbox"/>	Walkthroughs, zumindest nach ANSI/IEEE 729-1983, sind formaler als Reviews.
<input type="checkbox"/>	<input type="checkbox"/>	Das Verfahren, gegenseitige Benutzung von Modulen durch eine Verfeinerung der Modulstruktur aufzulösen, heißt <i>Stacking</i> .

- b.) Nehmen Sie an, Sie inspizieren das Klassendiagramm eines Kollegen. Auf der Prüfliste steht, Sie sollen für alle Klassen prüfen, ob diese tatsächlich existenzberechtigt sind. Nennen Sie 2 handfeste Indizien dafür, dass **keine** Klasse vorliegt. (2P)

- c.) **Warum** ist es bei der Implementierung des Einzelstücks in Java sinnvoll, das Schlüsselwort **synchronized** zu verwenden? **An welcher Stelle** muss der Code synchronisiert werden? (2P)

d.) Warum ist folgende Modellierung falsch (mit **Begründung**)? (1P)

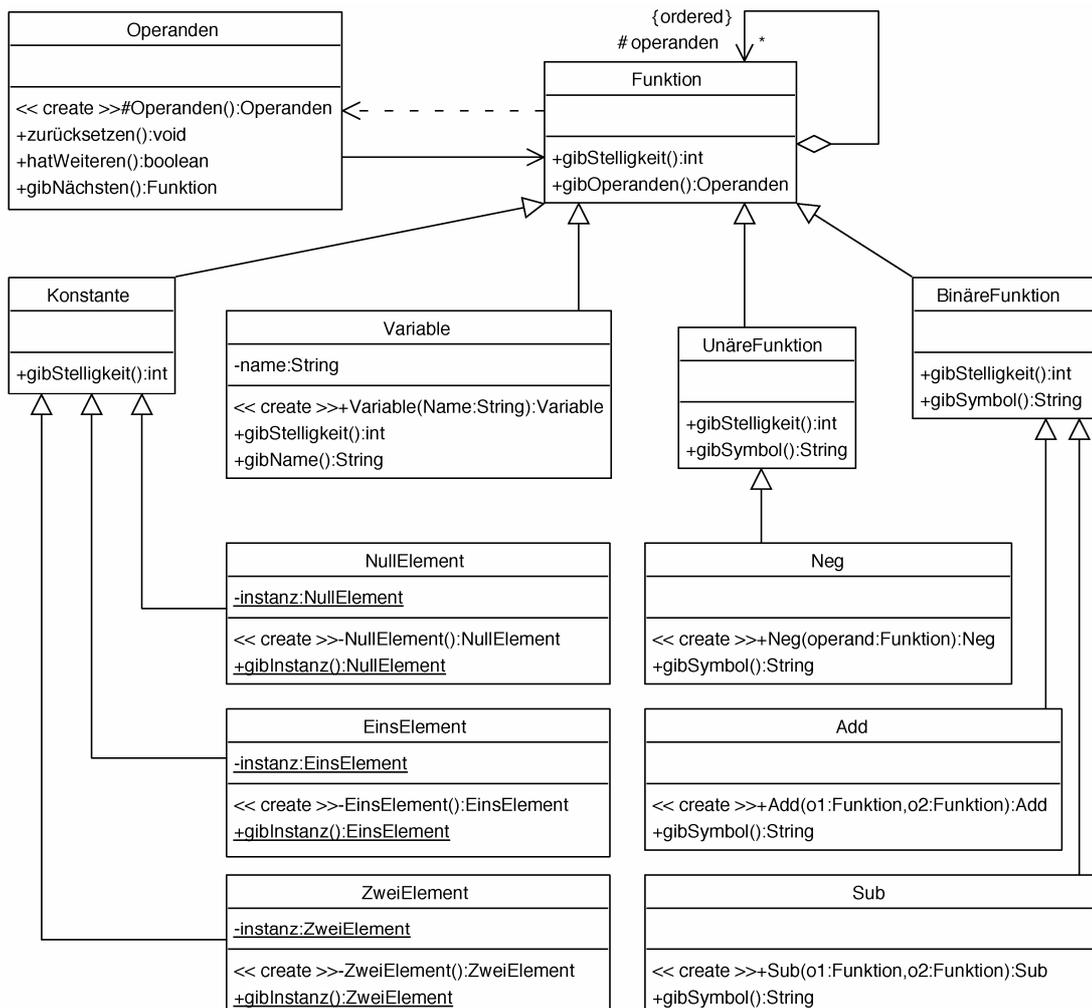


Aufgabe 2: Entwurfsmuster (1+1+6+3=11P)

a.) Welchem **Zweck** dient das Besucher-Muster? (1P)

b.) Welchem **Zweck** dient das Entwurfsmuster „Brücke“? (1P)

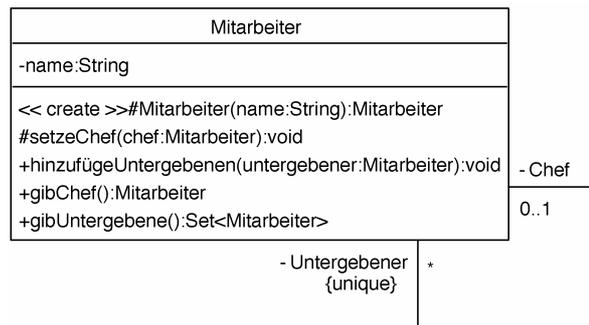
c.) Gegeben sei folgender Entwurf. Einer Ihrer Kommilitonen behauptet, er habe 3 Entwurfsmuster identifiziert: Iterator, Einzelstück und Kompositum. Hat er Recht? Begründen Sie Ihre Aussage, indem Sie jeweils **die Klassen** (3P), die er wohl gemeint hat, und **die Indizien** (3P), die dafür (oder dagegen) sprechen, angeben. Bedenken Sie, dass konkrete Implementierungen von den idealisierten Mustern abweichen können! (6P)

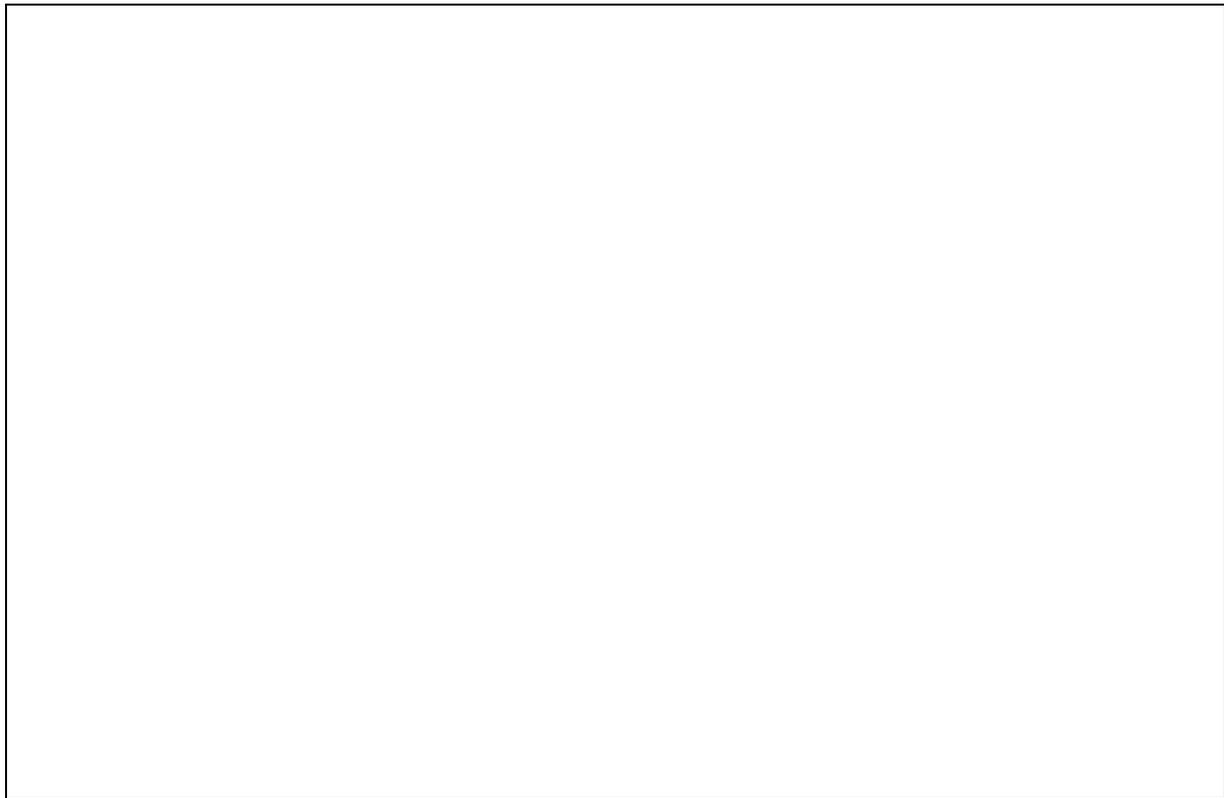


- d.) Walter Zimmer beschreibt in „Relationships between design patterns, in Pattern languages of program design“, einen Gleichartigkeitsbegriff, wie er in deutscher Übersetzung auch auf Übungsblatt 4, Aufgabe 2 angegeben war. In dem angegebenen Artikel behauptet Herr Zimmer ferner, sowohl (Abstrakte Fabrik und Erbauer) als auch (Fassade und Mediator) seien gleichartig. Stimmen Sie dieser Aussage zu? Geben Sie die **Definition** an und **begründen** Sie Ihre Antwort auf deren Grundlage. (3P)

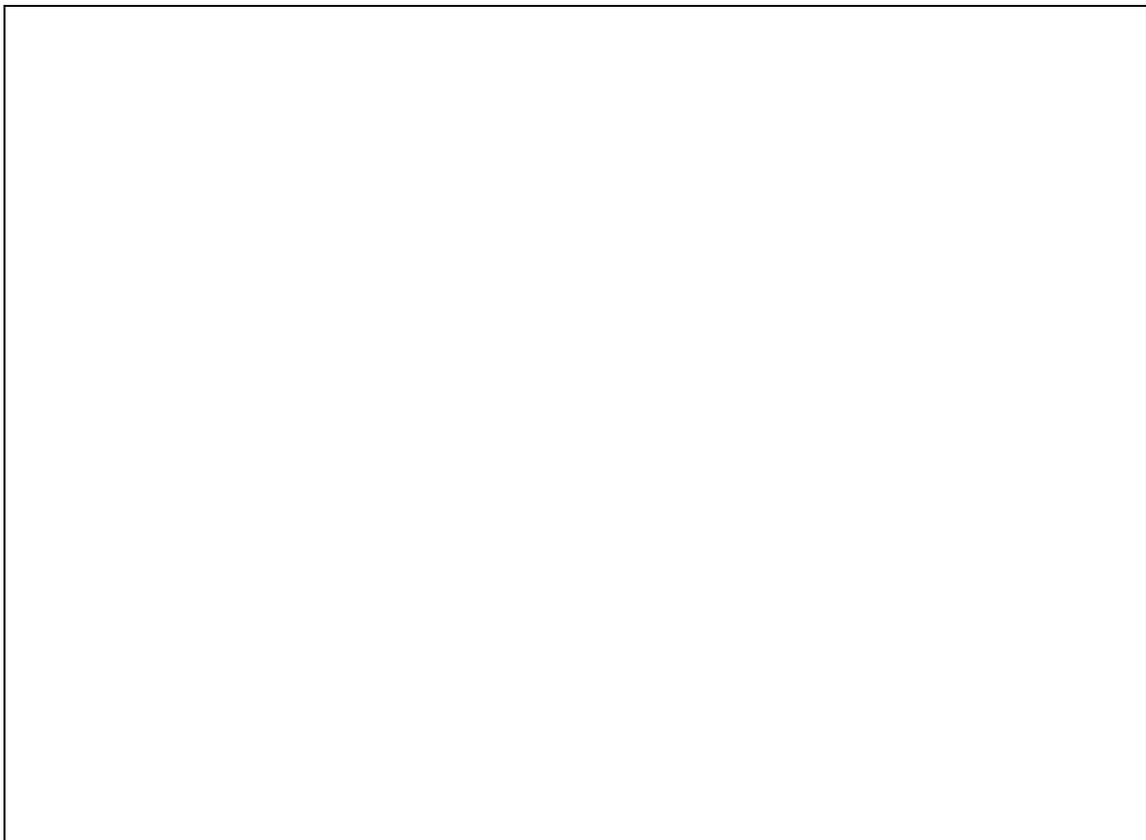
Aufgabe 3: Abbildung von UML auf Code (7+4=11P)

- a.) Gegeben sei das nebenstehende UML-Diagramm. Implementieren Sie diese Klasse **in Java**. Schreiben Sie den Code für die Verwaltung der Assoziation in die angegebenen Methoden; Sie brauchen keine weiteren Verwaltungsmethoden anzugeben. Bedenken Sie, dass ein Mitarbeiter auch einen neuen Chef bekommen könnte. Sie dürfen das „Collection Framework“ aus `java.util` verwenden. Achten Sie auf Sichtbarkeiten und korrekte Syntax! (7P)





- b.) **Welche** Schwierigkeiten ergeben sich allgemein bei der Umsetzung von den Multiplizitäten „1“ und „1..*“ und **warum**? Machen Sie einen **Vorschlag**, wie man diese Schwierigkeiten umgehen kann. Was kann man tun, wenn an **beiden Enden** der Assoziation derartige Kardinalitäten zu finden sind? (4P, jedoch nur für ganze Sätze)



Aufgabe 4: Testen & Prüfen (1+5+2+2+3=13P)

```
double c = 0;
double TOL = 0.8;
double diff = 0;
do {
    c = (a + b)/2;
    double y = f(c)*f(a);
    if (y < 0)
        b = c;
    else
        a = c;
    diff = Math.abs(a-b);
} while (diff > TOL);
System.out.println(c);
```

Gegeben sei nebenstehendes Rahmenprogramm für einen Bisektionsalgorithmus*. Die Funktion $f(x)$ mit der Signatur $f(x:\text{double}) : \text{double}$ sei verifiziert und

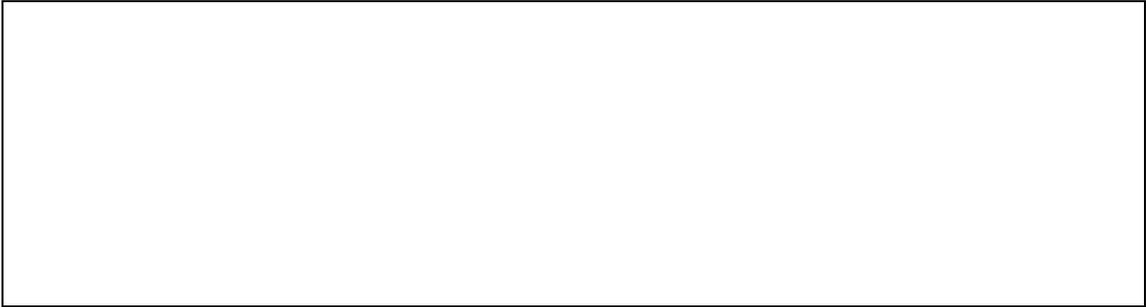
$$\text{spezifiziert als } f(x) = \begin{cases} x^2, & \text{wenn } x > 0 \\ 0, & \text{wenn } x = 0 \\ -x^2, & \text{wenn } x < 0 \end{cases}$$

**Das ist nur Kontextinformation. Für die Bearbeitung dieser Aufgabe brauchen Sie nicht zu wissen, wozu man einen Bisektionsalgorithmus braucht oder wie er funktioniert.*

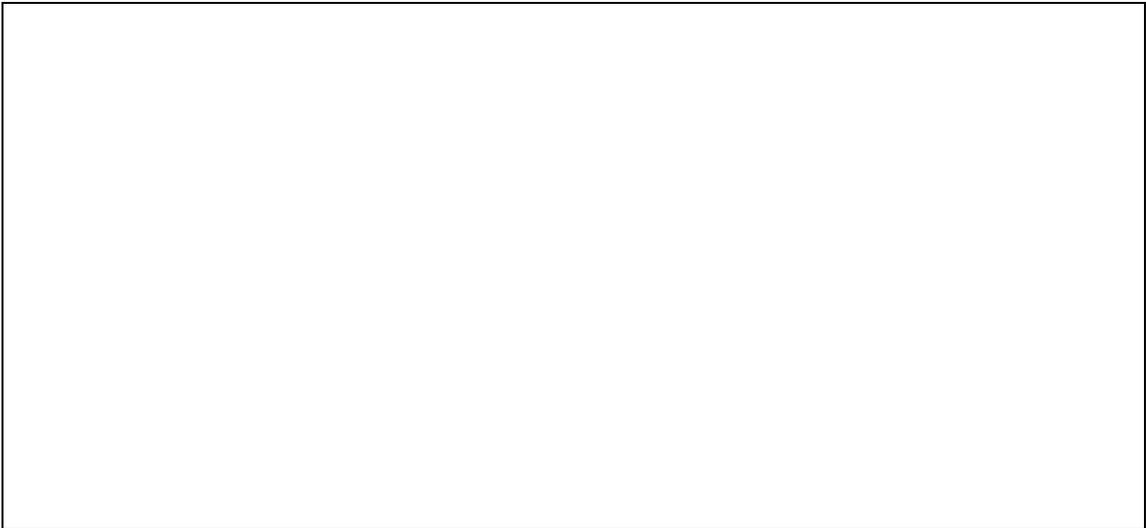
- a.) Was ist gemäß der Definition aus der Vorlesung unter „Die Funktion f ist verifiziert“ zu verstehen? (1P)

- b.) Wandeln Sie obiges Programm mit einer strukturerhaltenden Transformation in eine der Definition der Vorlesung entsprechenden Zwischensprache um. (5P)

- c.) **Wie viele** initiale Belegungen für a und b braucht man, um die Schleife einem Boundary-Interior-Test zu unterziehen? **Erläutern** Sie Ihre Antwort kurz, damit wir sehen, dass Sie nicht geraten haben! (2P)



- d.) **Welche Arten** von Überdeckung erreicht man mit $(a,b) = (-1.0, 2.0)$? **Erläutern** Sie Ihre Antwort kurz, damit wir sehen, dass Sie nicht geraten haben! (2P)
Tipp: Rechnen Sie mit Brüchen!



- e.) Für diese Teilaufgabe sei $f(x) = x^2$. **Welche Probleme** ergeben sich dann für das Testen? Wie lassen sich solche Probleme im Alltag finden? **Nennen** Sie hierfür je ein **dynamisches** und ein **statisches** Verfahren oder Werkzeug! (3P)

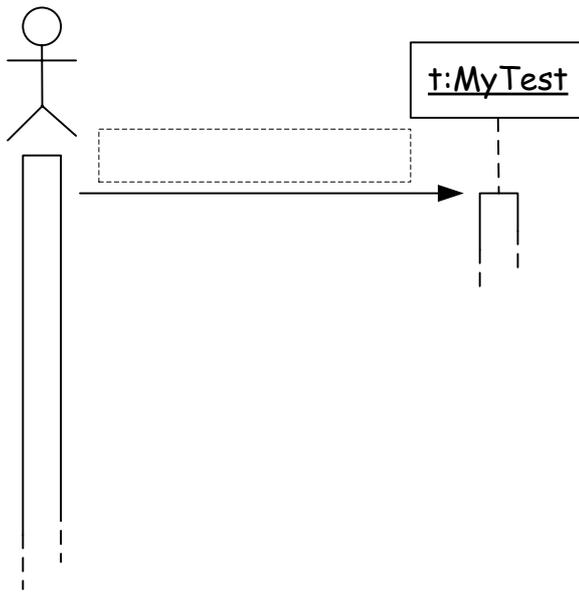


Aufgabe 5: JUnit (0,5+4+3,5=8P)

a.) Wozu verwendet man JUnit? (0,5P)

b.) Wie ist JUnit aufgebaut? Geben Sie ein Klassendiagramm an, das die Typen **Test**, **TestResult**, **TestSuite**, **TestCase** und **MyTest** in Beziehung setzt und ihre wichtigsten Methodennamen angibt. **MyTest** sei dabei Ihre eigene Implementierung eines Tests (entsprechend den JUnit-Konventionen). (4P)

- c.) Nehmen Sie an, Sie seien Softwaretechnik-Tutor und wollen Ihren Studenten erläutern, wie JUnit funktioniert. Geben Sie ein einfaches Sequenzdiagramm an, das den Kontrollfluss eines Testlaufs für einen Prüfling P mit der Methode m() verdeutlicht. (3,5P)



Aufgabe 6a: Prozessmodelle (1+6+2=9P)

(nur für Informatiker)

- a.) In der Vorlesung wurden zwei unterschiedliche Konzepte mit dem Namen „V-Modell“ vorgestellt. Worin unterscheiden sich diese bzw. wie hängen sie zusammen? (1P)

- b.) Beim Wasserfallmodell spricht man von einem „**dokumentgetriebenen Modell**“. Was versteht man hierunter? **Welche Dokumente** kommen in dem in der Vorlesung vorgestellten 6-stufigen Wasserfallmodell **an welchen Stellen** vor? (6P)

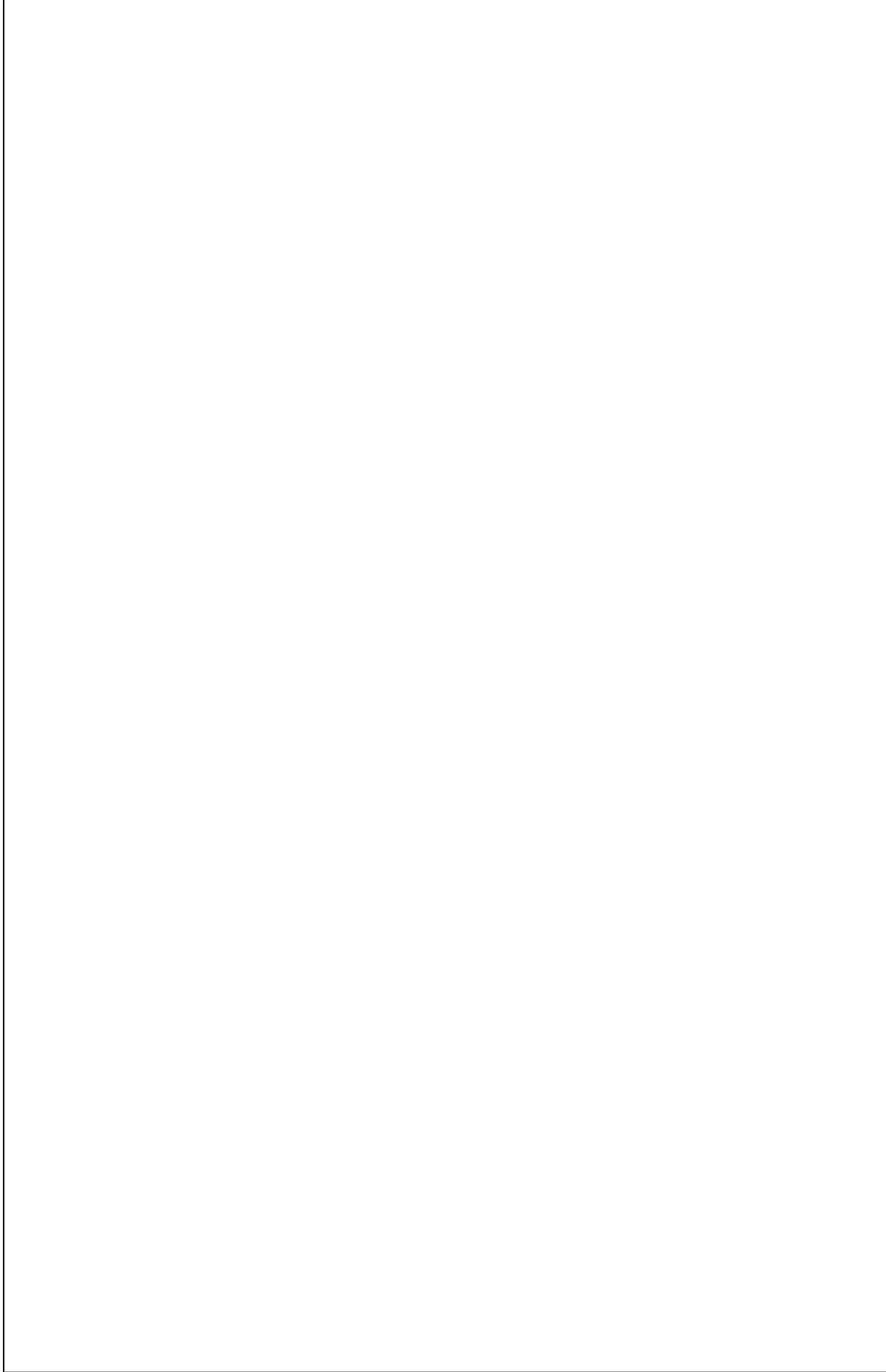
- c.) Beim Iterativen Modell unterscheiden wir den „inkrementellen“ und den „evolutionären“ Ansatz. Worin unterscheiden sich die beiden Vorgehensweisen? (2P)

Aufgabe 6b: Objektorientierte Analyse (9P) *(nur Informationswirte)*

Beschreiben Sie, wie man bei der objektorientierten Analyse die Assoziationen in einem UML-Klassendiagramm erhält. (Die Klassen seien schon vorhanden.) Beachten Sie bei Ihrer Antwort folgende Aspekte:

- a.) **Wie** sucht man nach möglichen Assoziationen in einem textuellen Anforderungsdokument? *(max. 3P)*
- b.) Was sind die **Voraussetzungen** für die Übernahme gefundener Beziehungen in das Klassendiagramm? *(max. 3P)*
- c.) Gehen Sie auf die verschiedenen Ausprägungen und Formen von Assoziationen ein: Erläutern Sie, anhand welcher **Kriterien** man welche **Auswahl** trifft! *(max. 6P)*

(Um volle Punktzahl erhalten zu können, muss auf jeden Aspekt eingegangen werden. Die Angabe der Maxima dient nur zu Ihrer Orientierung, die Aufgabe wird insgesamt mit maximal 9P bewertet!)



Notizen: