TECHNISCHE
UNIVERSITÄT
DARMSTADT

Ergänzende Informationen zur Vorlesung

Einführung in Software Engineering

Wintersemester 2011 / 2012

Fachgebiet Softwaretechnik
Fachbereich Informatik
Dr. Michael Eichberg

---

18. Oktober 2011

Hinweis

Dieses Dokument hat mittelfristig das Ziel alle wesentlichen Inhalte der Vorlesung Einführung in Software Engineering zu behandeln. Jeder, der Fehler entdeckt, oder inhaltlich etwas ergänzen möchte, wird dazu aufgerufen, zu diesem Skript beizutragen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Klausur

1.1 Allgemeines

Die Klausur wird kurz nach dem Ende der Vorlesungszeit statt finden. Alle Inhalte der Vorlesung sind klausurrelevant.

Die Bearbeitungszeit wird 90 Minuten sein. *Eine vollständige Bearbeitung der Klausur wird in der zur Verfügung stehenden Zeit nicht möglich sein.* Suchen Sie sich am Anfang der Klausur die Aufgaben heraus, die für Sie das beste Zeit-Nutzen Verhältnis haben.

Als Hilfsmittel sind beliebige schriftliche Unterlagen zugelassen (Neudeutsch: *Open-Book-Klausur*); es sind keine elektronischen Hilfsmittel zugelassen.[1]

1.2 Struktur

1.2.1 Multiplechoicefragen

Es gibt typischerweise eine Aufgabe mit Multiplechoicefragen. Bei diesen Fragen handelt es sich meist um Wissensfragen bzw. Fragen, die durch einfache Transferleistung/kurzes Nachdenken, beantwortet werden können. Beispiele für Multiplechoicefragen finden Sie in Tabelle 1.1.

Aussage	Richtig	Falsch
Durch den Einsatz von *Class-Responsibility-Cards* ist sichergestellt, dass ein gutes Design erstellt wird.		
Inversion of Discipline ist charakteristisch für Frameworks.		
...		

Tabelle 1.1: Beispiele von Multiplechoicefragen

[1] Elektronische Übersetzungshilfen sind nach expliziter Rücksprache zugelassen.

1.2.2 Allgemeine Wissensfragen

Es wird eine Aufgabe mit Wissensfragen geben. Beispiele:

- Benennen Sie drei allgemeine Entwurfsprinzipien. Begründen Sie für jedes Prinzip, warum es sich um ein allgemeines Entwurfsprinzip handelt und warum es nicht ausschließlich für die Entwicklung objektorientierter Programme gilt.
- ...

1.2.3 Software Design

Aufgaben zum Thema Software Design erfordern meist die Beurteilung eines gegebenen Designs (unabhängig von konkreten Design Patterns oder Idiomen).

Eine Aufgabe wäre:

> Geben ist folgendes Design:
>
> ```
> class Student
>
> class BachelorStudent extends Student
>
> class MasterStudent extends Student
> ```
>
> Ist dieses Design sinnvoll? Begründen Sie Ihre Antwort! Wenn Sie nicht genügend Informationen haben sollten, um eine Entscheidung treffen zu können, dann geben Sie an wovon Ihre Entscheidung ggf. abhängt.

1.2.4 Use Cases

Es gibt typischerweise eine Aufgabe, bei der ein einfaches Anwendungsfalldiagramm zu erstellen ist. Darüber hinaus gibt es meist eine Aufgabe, die das Erstellen eines "fully-dressed" Use Cases verlangt.

1.2.5 UML Klassendiagramme

Es gibt meist eine Aufgabe, im Rahmen derer ein Klassendiagramm zu erstellen oder zu übersetzen ist.

1.2.6 Testen und Testabdeckung

Eine Beispielaufgabe zum Thema Testen und Testabdeckung wäre:

> Die folgende Java-Implementierung einer Methode, zur Partitionierung einer Liste (hier ein Array) von double Werten, soll getestet werden.

```
int partition(double[] a, int left, int right) {
        int i = left - 1;
        int j = right;
        while (true) {
                while (a[++i] < a[right])
                        ;
                while (a[right] < a[--j])
                        if (j == left)
                                break;
                if (i >= j)
                        break;
                swap(a, i, j);
        }
        swap(a, i, right);
        return i;
}
```

Bewerten Sie die Testabdeckung des Testfalls $a = \{12.0, 8.3\}, left = 0, right = 1$ in Hinblick auf Anweisungsabdeckung (statement coverage). Geben Sie für jede Anweisung an, ob diese durch den Testfall abgedeckt wird.

1.2.7 Design Patterns

Fragen zum Thema Design Patterns fallen in die folgenden Kategorien:

- Ein Stück Code oder ein UML Diagramm ist gegeben und Sie müssen erkennen welche Variante eines Design Patterns umgesetzt wurde und welche Klassen welche Rollen inne haben.
- Geben ist ein bestimmtes Designproblem und Sie müssen entscheiden welches Design Pattern Sie zur Lösung einsetzten wollen. Darüber hinaus ist es oft erforderlich die Lösung – unter Verwendung des vorgeschlagenen Patterns – zu skizzieren.
- Es gibt ein partielles Design und Sie müssen dieses erweitern.

Anhang A

Abkürzungsverzeichnis

Abkürzung	
O	
OOA/D	Object-oriented Analysis and Design (*Objektorientierte Analyse und Entwurf*)
U	
UML	Unified Modeling Language

Anhang B

Kleines SE Wörterbuch

Sowohl die deutschsprachige Literatur als auch (deutschsprachige) Softwareentwickler verwenden häufig englische und deutsche Begriffe – Rund um das Thema Softwareentwicklung – wild gemischt. Dieses kleine Wörterbuch soll dabei helfen, sich im Dschungel der Begriffe besser zurecht zu finden. Fett markiert sind die Begriffe, die nach subjektiver Meinung, die gängigeren sind.

Im Rahmen der Vorlesung Einführung in Software Engineering betrachten wir die Namen der Entwurfsmuster als Eigennamen und übersetzen diese nicht. D.h. im Rahmen der Vorlesung verwenden wir z.B. immer den Namen "Template Method" und niemals "Schablonenmethode" verwenden.

Deutsch	Englisch
A	
Anforderungsmanagement	Requirements Engineering
Anwendungsfall	**Use case**
E	
Entwurf	**Design**
Entwurfsmuster	**Design Pattern**
R	
Risikomanagement	Risk Management
S	
Softwaretechnik	**Software Engineering**
Z	
Zusammenhalt	Cohesion

Anhang C

Bonusregelung

C.1 Überblick

Der maximal zu erreichende Bonus entspricht 50% der Anzahl Klausurpunkte, die notwendig wären, um in der Klausur ein Ergebnis zu erreichen, dass eine komplette Notenstufe besser wäre. D.h. wenn durch die Übungen der volle Bonus erreicht wurde, und aus der Klausur heraus 60 Punkte geholt wurden und der Abstand zwischen zwei Noten (z.B. von 3,0 → 2,0) 10 Klausurpunkte beträgt, dann wäre der Bonus 5 Klausurpunkte und die Gesamtzahl der in der Klausur erreichten Punkte 65. Der Bonus kann zum Bestehen der Klausur führen.

C.2 Erlangen des Bonuses

Durch die Bearbeitung der Übungsblätter kann immer eine bestimmte Anzahl an Übungspunkten erreicht werden. Es wird drei bewertete Übungsblätter geben. Die maximale Anzahl der Punkte, die pro Übungsblatt erreicht werden kann, ist immer identisch. Für die meisten Übungen wird es zwischen acht und zehn Übungspunkte geben. Die maximal zu erreichende Anzahl Übungspunkte ergibt sich somit aus der Summe der Übungspunkte über alle bewerteten Übungsblätter. Um ein Bonus für die Klausur zu erreichen, müssen mind. 30% der Übungspunkte erreicht werden.

C.3 Verrechnung mit der Klausur

Der Anteil der erreichten Übungspunkte wird in Klausurpunkte umgerechnet und auf die erzielten Klausurpunkte aufaddiert. Die Gesamtzahl der Klausurpunkte bestimmt dann die Endnote. D.h. wenn in der Klausur zum Beispiel zehn Punkte mehr erreicht werden müssen, um sich eine komplette Notenstufe zu verbessern (2,0 → 1,0), und im Rahmen der Übungen 90% der maximal

möglichen Übungspunkte erreicht wurden, dann ist der effektive Bonus 0,9 * $5 = 4,5$ Klausurpunkte.

Es ergibt sich somit folgende Berechnungsvorschrift: Anteil der erreichten Übungspunkte * (0,5 * Anzahl der Klausurpunkte, die notwendig sind, um eine Note, die eine volle Notenstufe besser ist, zu erreichen) = zusätzliche Klauspunkte

Sie müssen mind. 30% der Übungspunkte und mind. 25% der Klausurpunkte erreichen, um den Bonus zu erhalten.

Klausurpunkte	Note
> 80	1,0
77-79,5	1,3
73-76,5	1,7
70-72,5	2,0

Tabelle C.1: Beispielhafte Notengrenzen für die Klausur

Die Auswirkung des Bonusses auf die finale Note ist abhängig von der Anzahl der erreichten Klausurpunkte. Seien die Notengrenzen wie in Tabelle C.1, dann würde ein effektiver Bonus von drei Klausurpunkten bei 70 erreichten Klausurpunkten zu einer Verbesserung der Note von 2,0 auf 1,7 führen. Sollten in der Klausur jedoch bereits 73 Punkte erreicht worden sein, dann führen die drei Bonuspunkte zu keiner besseren Endnote.

C.4 Gültigkeit des erreichten Bonus

Der Bonus ist gültig für das aktuelle Wintersemester und das darauf folgende Sommersemester. Der Bonus verfällt insbesondere auch dann nicht, wenn die Klausur am Ende des Wintersemesters – trotz Verrechnung des Bonuses – nicht bestanden wird.

Anhang D

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die an der Erstellung dieses Dokuments in der einen oder anderen Weise mitgewirkt haben. Die Namen sind in chronologischer Reihenfolge angegeben.

Kathrin Ballweg, Kristin Rammelt, Melanie Weiland, Barbara Zöller, Heiko Guckes