

1	2	3	Sum
3	2	5	10

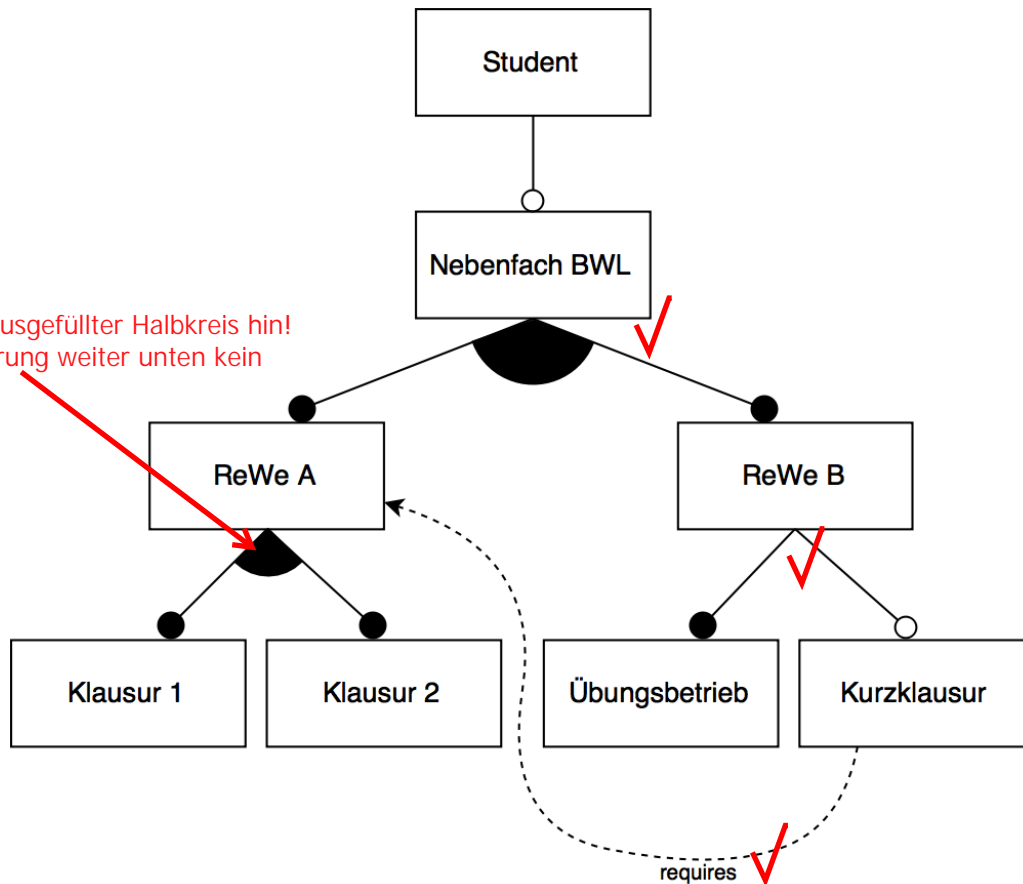
Sehr schön! Houman Biglari

Softwaretechnik, WS 2015/16 Aufgabenblatt 8

Bearbeitungsgruppe Oktoberregen

Radu Coanda (344664), Gregor Kobsik (359198), Valentin Steiner (357980), Daniel Pujiula Buhl, Jean Tekam

Aufgabe 8.1 - Feature-Diagramm

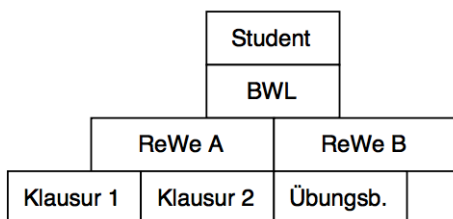
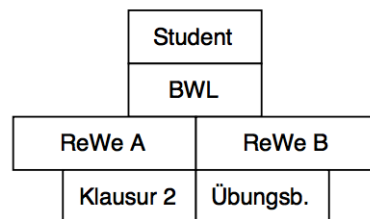
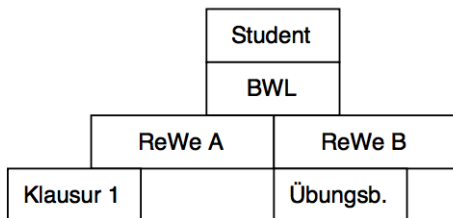
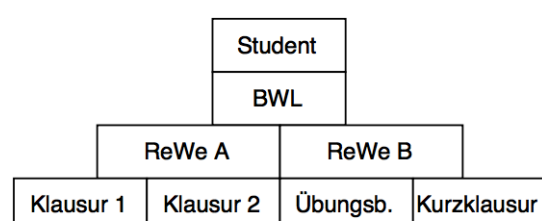
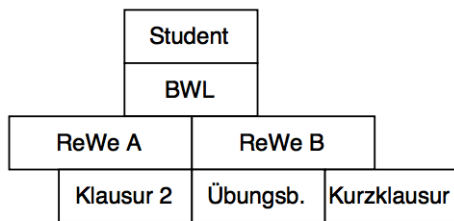
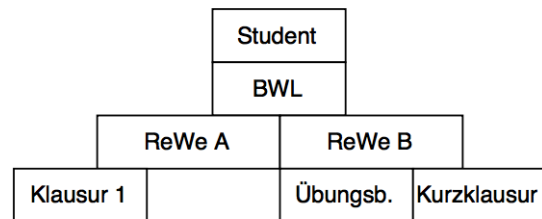
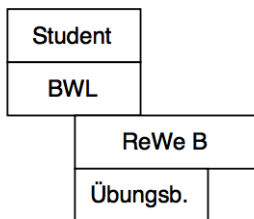
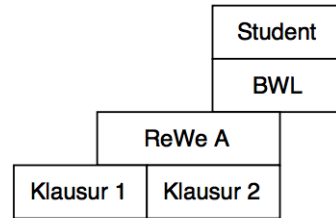
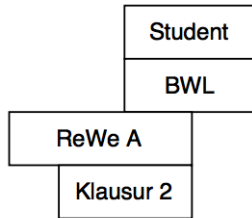
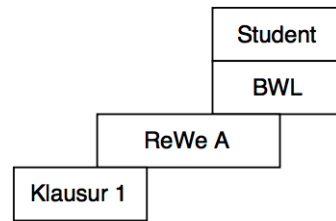
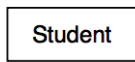


Zunächst glaubten wir, die Wahlmöglichkeiten der ReWeA-Klausuren als XOR modellieren zu müssen. Bei der Suche erlaubter Konfigurationen wählten wir aber instinktiv auch solche Fälle, in denen beide ReWeA-Klausuren bestanden wurden. Wir zogen die Aufgabenstellung zu Rate:

Zu Rechnungswesen A werden am Ende des Semesters zwei Klausuren geschrieben (man hat also zwei Versuche). Sobald man eine davon besteht, hat man Rechnungswesen A bestanden.

Nach aufmerksamem Studium der Aufgabenstellung kamen wir zum Schluss, dass es nicht verboten ist, sowohl Klausur 1 als auch Klausur 2 zu bestehen. Aus diesem Grund muss diese Fallunterscheidung als OR gewählt werden.

Es sind insgesamt 2^7 Kombinationsmöglichkeiten vorhanden, davon sind folgende 11 erlaubt: ✓



o.k. basierend auf eurer Diagramm!

3 / 3 Punkte

Aufgabe 8.2 - Komponenten

- (a) Was ist eine Komponente im Software Engineering? Welchen Zweck erfüllen Komponenten?

Eine Komponente im Software Engineering ist eine abgeschlossene Einheit eines Software-Systems, die meist eine logisch getrennte Aufgabe erfüllt und die dazu benötigten Inhalte zusammenfasst. Komponenten zeichnen sich durch eine hohe Kohäsion nach innen und lose Kopplung nach außen aus. Komponenten existieren in einem Kontext: sie werden von anderen Komponenten verwendet und verwenden ihrerseits auch andere Komponenten. Dazu ist es nötig, Schnittstellen explizit zu definieren.

Neben der Schaffung einer logischen Struktur bringen Komponenten den Vorteil der Wiederverwendung und Austauschbarkeit. Ein Softwaresystem könnte beispielsweise auf einen Kunden zugeschnitten werden, indem eine bestimmte Komponente ausgetauscht wird. ✓

- (b) Nennen Sie mindestens 3 Eigenschaften, mit denen Software-Komponenten in der Vorlesung charakterisiert wurden.

- (a) fördern Wiederverwendbarkeit und Qualität ("best in class", Produktivitätssteigerung, geringerer Schulungsaufwand etc.)
- (b) haben explizite Schnittstellen
- (c) sind von ihrer Umgebung größtenteils unabhängig ✓
- (d) können unabhängig versionsverwaltet werden

- (c) Was versteht man unter Glue Code?

Mit Glue Code wird derjenige Programmcode bezeichnet, der die Komponenten untereinander verbindet. Dies kann auf verschiedenen Arten erfolgen (z.B. durch Scripting-Sprachen) und ist recht einfach auszutauschen. ✓

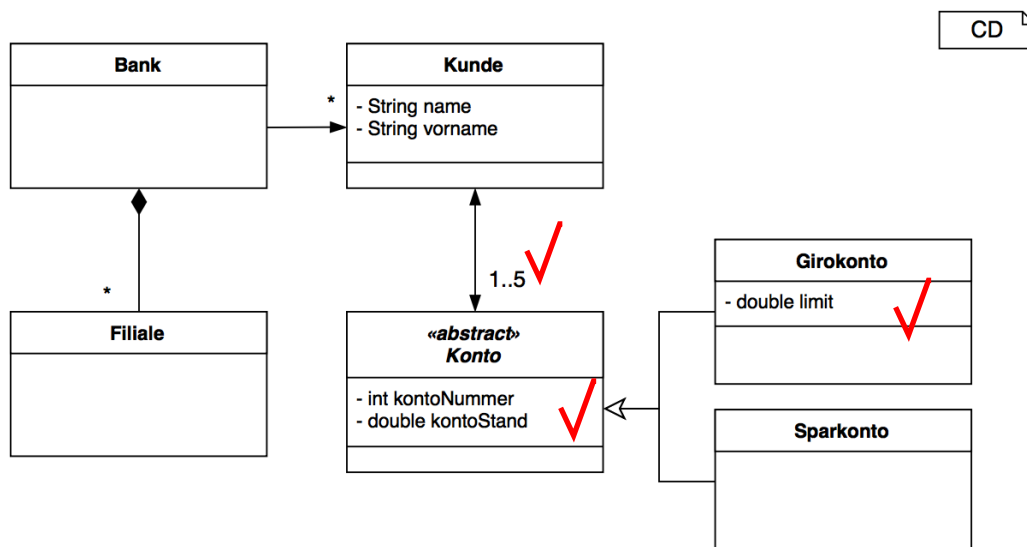
2 / 2 Punkte

Aufgabe 8.3 - Generative Softwareentwicklung

- (a) Was ist der Zweck von Code-Generierung?

Die Idee hinter der Code-Generierung ist, durch das automatisierte Generieren von Programmcode aus einer Modellsprache sowohl Zeit einzusparen als auch eine höhere Qualität zu erzielen. ✓

- (b) Erstellen Sie aus der gegebenen informellen Beschreibung ein passendes Klassendiagramm.



Schreiben Sie die Beziehung zwischen Kunde, Konto und Girokonto aus Ihrem Klassendiagramm in der textuellen Syntax für Klassendiagramme auf, welche Sie mit dem Tool DEX kennengelernt haben. Berücksichtigen Sie hier nur entsprechend relevante Klassen, Attribute und Assoziationen.

```
classdiagram Exercise83b {  
    class Kunde {  
        String name;  
        String vorname;  
    }  
    abstract class Konto {  
        int kontoNummer;  
        double kontoStand  
    }  
    class Girokonto extends Konto {  
        double limit;  
    }  
    association [1] Kunde <--> Konto [1..5];  
}
```



5 / 5 Punkte