



Probeklausur Architektur und Entwurfsmuster

2017

Ausgangssituation

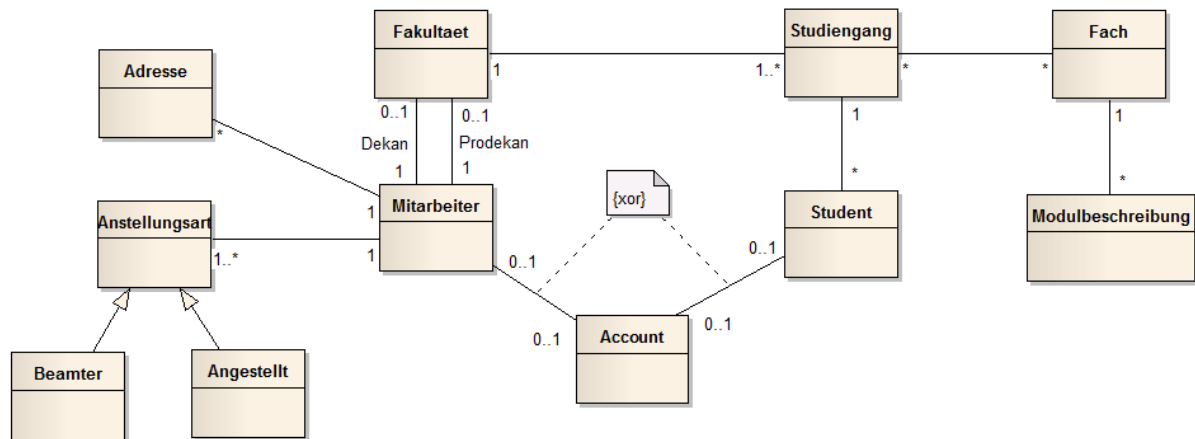
Als Mitarbeiter eines Projektteams werden Sie mit der Entwicklung eines Systems für die Administration einer Hochschule betraut.

Produktvision:

- Das System soll in der Lage sein, eine Hochschule mit beliebig vielen Fakultäten und Abteilungen zu verwalten
- Über das System soll Studenten die Möglichkeit geboten werden, möglichst viel über eine moderne Online-Plattform zu erledigen. Hierunter fällt z.B. die elektronische Prüfungsanmeldung und die Noteneinsicht
- Das System soll einen hohen Sicherheitsstandard erfüllen. Jeder Benutzer, der mit dem System arbeitet, unterliegt bestimmten Restriktionen, welche durch ein Berechtigungssystem definiert werden
- Damit ein Benutzer eine Funktion ausführen kann, muss er sich am System anmelden

1. Domain-driven Design

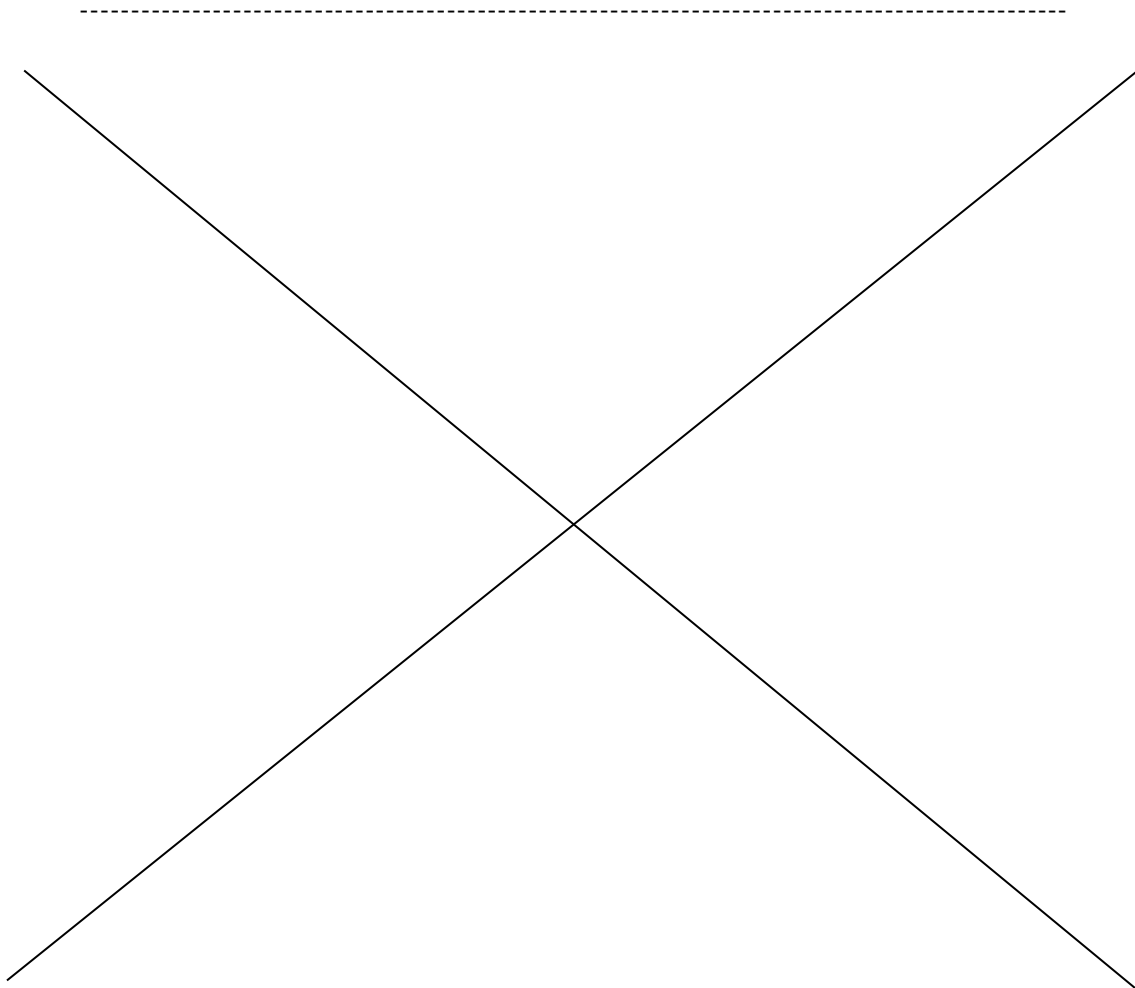
- 1a) Im Rahmen des Designs wenden Sie nun die Methode „Domain-driven Design“ Punkte ____ auf das nachstehende Produktmodell an. Analysieren Sie das nachfolgende Diagramm und kennzeichnen Sie alle Klassen in diesem Diagramm. Stellen Sie folgende Kernelemente innerhalb des Diagramms heraus: Aggregate, Entity, Root-Entity, Value-Objects. Verwenden Sie falls notwendig Stereotypen.



- 1b) Definieren Sie das Konzept einer Entity. Was ist eine Entity? Was ist eine globale bzw. lokale Identität? Geben sie jeweils ein Beispiel für die Identität an. Punkte ____

- 1c) Welche Besonderheiten sind bei der Implementierung von Assoziationen zwischen Aggregaten zu beachten? Gilt diese Einschränkung auch innerhalb eines Aggregates? Begründen Sie Ihre Aussage(n). Punkte ____

- 1d) Viele Applikationen müssen auf Altsysteme zugreifen. Definieren Sie das Problem Punkte ____
sowie die Lösung, welche durch eine „Anti-Corruption-Layer“ vorgeschlagen
wird. Begründen Sie warum diese Lösung sinnvoll ist und warum Sie zu einer
flexibleren Software führt.



2. Software Architektur & Design Prinzipien

2a) Erläutern Sie die Begriffe „Inversion of Control“ und „Dependency Injection“. Punkte ____

2b) Erläutern Sie die Arbeitsweise eines Service-Locators. Welche zwei Realisierungsvarianten existieren? Wie beurteilen Sie den Service-Locator in Bezug auf die entstehende Kopplung? Punkte ____

2c) Erläutern Sie, warum man mit Hilfe der „Kapselung“ die Einhaltung von sog. „Invarianten“ leichter sicherstellen kann. Gibt es Sonderfälle die es erlauben das Prinzip Kapselung aufzuweichen? Was ist Ihr Rat als Architekt? Punkte ____

2d) Definieren Sie das liskovsche Substitutionsprinzip. Skizzieren Sie ein Klassendiagramm und erläutern Sie daran was passiert, wenn gegen das Prinzip verstoßen wird. Begründen Sie, warum dieses Prinzip notwendig für einen sinnvollen Einsatz der Polymorphie ist. Punkte ____

2e) Definieren Sie das Interface-Segregation-Prinzip. Begründen Sie, warum der Einsatz dieses Prinzips eine leichtere Wartbarkeit, bzw. Erweiterbarkeit der Software garantiert. Punkte ____

3.) Software Architektur – Die Struktursicht

Im Rahmen einer Architekturbewertung sollen Sie nun Komponenten, sowie deren Schnittstellen bewerten. Hierzu wird Ihnen für das Repository des Hochschulsystems folgendes Quellcodefragment vorgelegt (Schnittstellenspezifikation):

```
public interface TechnicalLayer {  
    // Studiengang laden  
    Map<String,String> loadStudiengang( jdbc.Connection conn, String ident );  
    // Studiengang in der Datenbank speichern  
    void saveStudiengang(jdbc.Connection conn, Studiengang studiengang );  
    // Mail versenden  
    void sendMail( java.mail.Adress recipient, String betreff, String body );  
}
```

3a) Skizzieren Sie den Software-Kategoriegraphen unter der Annahme, dass die Anwendungskomponente „ServiceCenterStudium“ (SCS) sowie die Anwendungskomponente „Dekanat“ auf das Repository zugreifen. Darüber hinaus gelten folgende Annahmen:

Punkte ____

- Alle Datenbankspezifischen Klassen/Interfaces (jdbc) liegen in der Kategorie „JDBC“.
- Die Klassen des Produktmodells werden in der Kategorie „A“ (Anwendung) angesiedelt
- Collections liegen in Kategorie 0
- Das SCS kann automatisch Emails versenden (Email-API: Kategorie Mail)

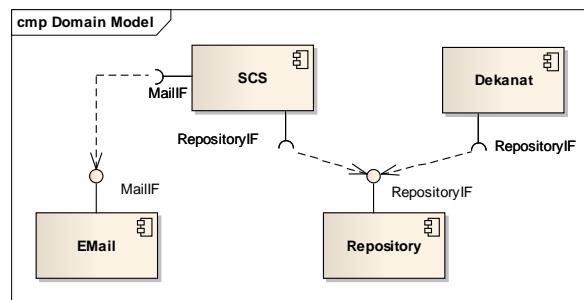
3b) Begründen Sie, warum man gemäß Quasar A und T Software immer trennen sollte. Wie löst Quasar dieses Problem, welche Art von Softwarekategorie wird hierzu verwendet und welche Einschränkung muss berücksichtigt werden?

Punkte ____

- 3c) Zeichnen Sie nun einen verbesserten Kategoriegraphen. Zerlegen Sie das Interface „TechnicalLayer“ in verschiedene Interfaces, um das Interface-Segregation-Prinzip zu gewährleisten.

Punkte ____

- 3d) Ordnen Sie die Artefakte des nebenstehenden Diagramms sinnvoll den Softwarekategorien aus Aufgabe 3d zu.



Punkte ____

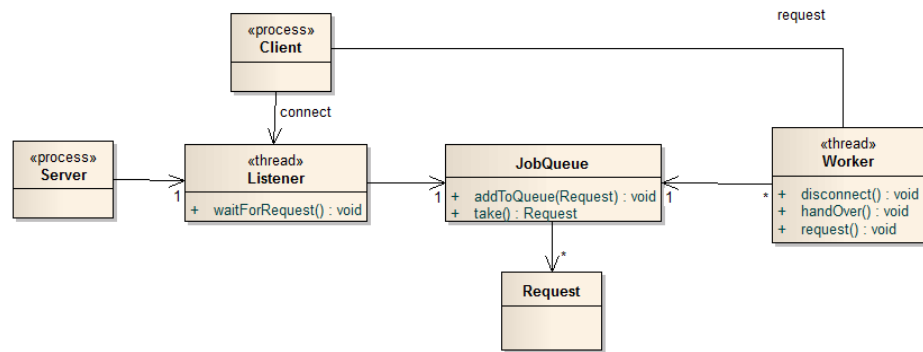
Artefakt	Typ	Kategorie
Implementierung SCS	Klasse(n)	
Implementierung Dekanat	Klasse(n)	
Implementierung Email	Klasse(n)	
MailIF	Schnittstelle	
RepositoryIF	Schnittstelle	

- 3e) Die Kopplung eines Systems kann durch den Entwurf der Schnittstellen einer Komponente stark beeinflusst werden. Nennen Sie drei Möglichkeiten wie die Kopplung beim Schnittstellenentwurf beeinflusst wird. Geben Sie für jede Variante eine kurze Beschreibung sowie den Grad der Kopplung (Hoch, Mittel, Niedrig) an.

Punkte ____

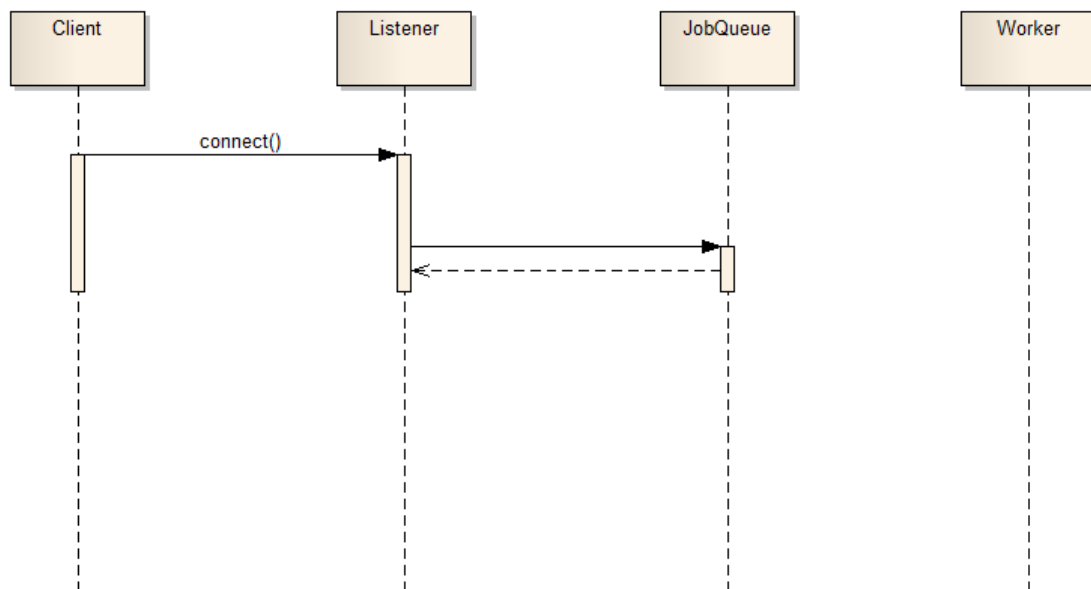
Variante	Beschreibung	Grad

4.) Die Prozesssicht



4a) Erweitern Sie das nachfolgende Sequenzdiagramm so, dass der prinzipielle Ablauf des Job-Queue-Patterns ersichtlich wird. Erläutern Sie die Arbeitsweise des Patterns anhand des Sequenzdiagramms.

Punkte ____



4b) Wie verhält sich dieses Pattern, wenn die Queue voll ist und keine freien Worker zur Verfügung stehen? Was kann man tun, um eine Überlast des Systems zu verhindern?

Punkte ____

5) Die physische Sicht

5a) Skizzieren Sie den physischen Aufbau eines active/active-Clusters (symmetrischer Cluster). Nehmen Sie in die Skizze folgende Komponenten mit auf: External Client, Loadbalancer, Application Server und Datenbank-Server. Zeichnen Sie auch redundante Komponenten ein. Punkte ____

5b) Beschreiben Sie die Funktionsweise eines active/active-Clusters. Punkte ____

5c) Nennen Sie zwei nicht-funktionale Anforderungen, welche durch eine Cluster-architektur positiv beeinflusst werden können. Geben Sie jeweils eine Begründung an. Punkte ____

5d) Beurteilen Sie folgende Aussage: „Es müssen nicht alle Komponenten eines Clusters redundant ausgelegt sein“. Welches Problem ist darin enthalten? Punkte ____

6 Design Pattern, Enterprise Patterns

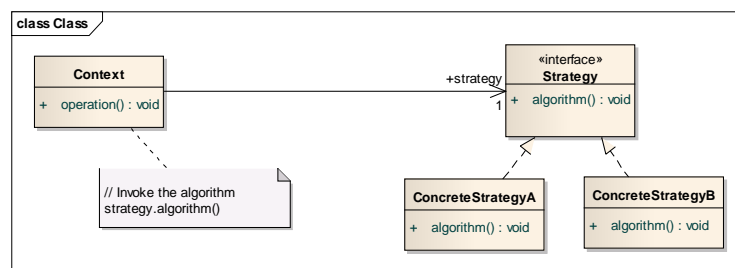
- 6a) Welches Grundproblem wird durch die Verwendung von Erzeugermustern adressiert? Geben Sie ein Quellcodefragment an, welches das Problem verdeutlicht. Begründen Sie, warum die Flexibilität einer Software durch die Anwendung dieser Kategorie von Mustern erhöht werden kann und welches Architekturprinzip dadurch positiv beeinflusst wird.

Punkte ____

Quellcodefragment:

- 6b) Erläutern Sie die Grundidee des Strategy-Patterns.

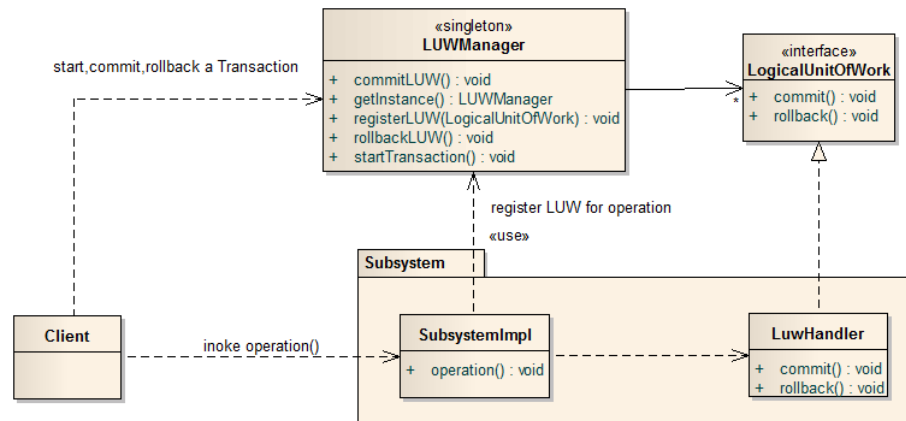
Punkte ____



- 6c) Wie kann man die Idee des Strategy-Patterns bei der Verwendung von Collections sinnvoll anwenden? Erläutern Sie die Idee und geben Sie ein Beispiel an.

Punkte ____

6 Design Pattern, Enterprise Patterns



6d) Welches Problem wird durch das Pattern „Logical Unit of Work“ gelöst? Welche Punkte __
Lösungs-idee liegt diesem Pattern zugrunde?

6e) Erweitern Sie das nachfolgende Sequenzdiagramm so, dass der prinzipielle Punkte __
Ablauf des Patterns „Logical Unit of Work“ sichtbar wird. Fügen Sie falls
notwendig neue Objektinstanzen bzw. Methodenaufrufe in das Diagramm ein.

