Probeklausur Architektur und Entwurfsmuster

2013/2014

Ausgangssituation

Als Mitarbeiter eines Projektteams werden Sie mit der Entwicklung eines Kantinensystems betraut. Über dieses System sollen Kantinen, wie z.B. eine Hochschul-Mensa einfach abgebildet werden können.

Produktvision:

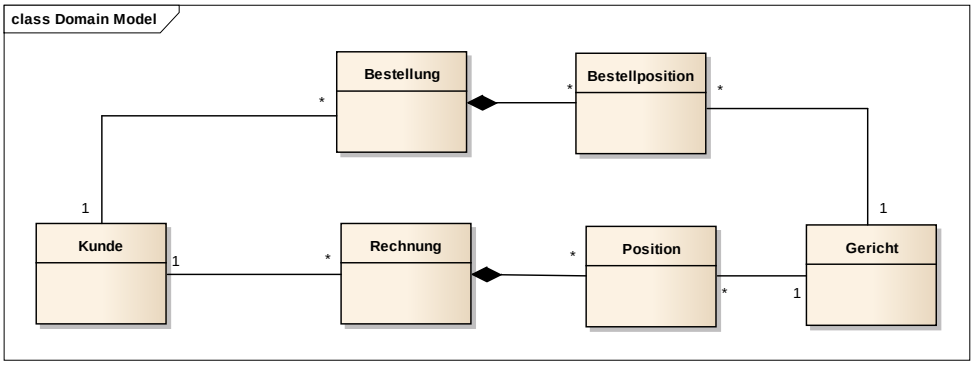
- Das System bietet die Möglichkeit den Speiseplan einer Woche online einzusehen. Dies soll sowohl per Webbrowser als auch über ein Smartphone möglich sein.
- Das System bietet dem Kantinenpersonal die Möglichkeit Speisepläne im Voraus zu erstellen und zu editieren.
- Das System verfügt über eine integrierte Lagerverwaltung. Diese kann automatisch Bestellvorschläge für die benötigten Zutaten auf Basis der Wochenpläne ermitteln.
- Zusätzlich bietet das System einen Bestell- und Lieferservice für externe Firmen und Privatkunden an. Firmen können die angebotenen Menüs vorab bestellen.
- Die Mitarbeiter der externen Firmen, bzw. die Privatkunden können auf Basis des Speiseplans ihre Menüwünsche online bestellen. Am jeweiligen Wochentag erfolgt die Auslieferung durch den Lieferservice der Kantine.
- Das System soll einen hohen Sicherheitsstandard erfüllen. Jeder Benutzer, der mit dem System arbeitet, unterliegt bestimmten Restriktionen, welche durch ein Berechtigungssystem definiert werden.
- Wichtig: Der Speiseplan kann von allen Benutzern eingesehen werden. Erst die Bestellfunktion erfordert eine Anmeldung am System.
- Die Wartung des Systems obliegt den Administratoren.
- Um eine hohe Akzeptanz zu erzielen, soll das System über eine Vielzahl von online-Schnittstellen verfügen.
- Sämtliche Zahlungsvorgänge sollen online über eine Schnittstelle zur Hausbank durchgeführt werden.

Randbedingungen:

- Das System soll modular aufgebaut sein, so dass es jederzeit möglich ist, neue Module hinzuzufügen.

1. Software Architektur – Die Logische Sicht

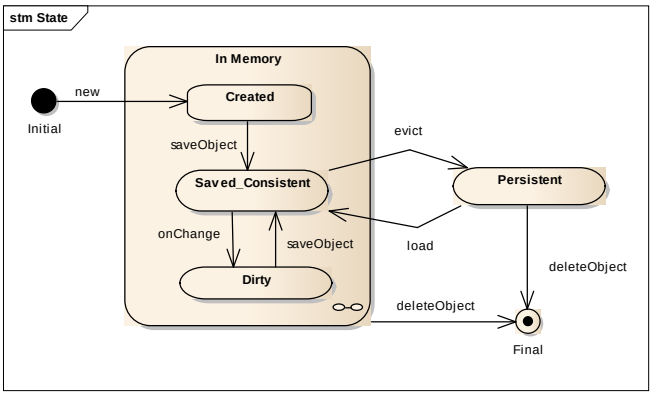
1a) Im Rahmen des Designs wenden Sie nun die Methode „Domain Driven Design" auf das nachstehende Produktmodell an. Analysieren Sie das nachfolgende Diagramm und kennzeichnen Sie alle Klassen in diesem Diagramm. Stellen Sie folgende Kernelemente innerhalb des Diagramms heraus: Aggregate, Entity, Root-Entity,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



1b) Erläutern Sie das Konzept eines „Aggregates". Beantworten Sie hierzu folgende Aspekte: Was versteht man unter einem Aggregat? Was ist die sog. Root-Entity? Welche Einschränkungen existieren in Bezug auf aggregatübergreifende Assoziationen?

1c) Was versteht man unter sog. Intension Revealing Interfaces? Warum ist dieses Konzept wichtig, wie hilft es bei der Entwicklung?

1d) Erläutern Sie das nebenstehende Zustandsdiagramm. Dieses Zustandsdiagramm stellt den Lebenszyklus eines Domänen-Objekts dar. Erläutern Sie die Zustände sowie die zugeordneten Ereignisse. Stellen Sie den Unterschied zwischen einem Objekt einer Programmiersprache und einem Domänenobjekt heraus.



1e) Erläutern Sie anhand des in Aufgabe 1d beschriebenen Lebenszyklus warum Assoziationen zwischen Aggregaten schwierig zu realisieren sind? Geben Sie eine Möglichkeit an, wie diese spezielle Form von Assoziation abgebildet werden kann.

2. Software Architektur

2a) Beschreiben Sie die 4+1 Sichten einer Architektur. Nennen Sie für jede Komponente die wichtigsten Eigenschaften/Ziele. Nennen Sie, falls möglich, jeweils eine Anforderung, welche durch eine Architektursicht gelöst wird.

2b) Was versteht man unter dem Architekturprinzip „Information Hiding“? Was passiert, wenn man gegen das Architekturprinzip verstößt?

2. Software Architektur (Fortsetzung)

2c) Welche Gründe sprechen für die Zerlegung einer Anwendung in Komponenten?

2d) Welches Merkmal einer Komponente gibt Aufschluss darüber, wie gut eine Komponente wiederverwendet werden kann? Begründen Sie Ihre Aussage. Wann ist eine Komponente sehr gut, bzw. schlecht wiederverwendbar?

3.) Software Architektur – Die Struktursicht

Im Rahmen einer Architekturbewertung sollen Sie nun Komponenten, sowie deren Schnittstellen bewerten. Hierzu wird ihnen für die Persistenzkomponente des Kantinensystems folgendes Quellcodefragment vorgelegt (Schnittstellenspezifikation):

```
public interface PersistenceKantine {

    // Gericht aus Datenbank laden
    XMLDocument loadGericht( String gerichtID  );

    // Gericht in der Datenbank speichern
    void saveGericht( XMLDocument docGericht );
}
```

3a) Skizzieren Sie den Software-Kategoriegraphen unter der Annahme, dass die Anwendungskomponente „Wochenplanung“ sowie die Anwendungskomponente „Bestellwesen“ auf die Persistenzkomponente zugreifen. Darüber hinaus gelten folgende Annahmen:

- Alle XML spezifischen Klassen/Interfaces liegen in der Kategorie „XML“
- Die Klassen des Produktmodells werden in der Kategorie „A“ (Anwendung) angesiedelt

3b) Nennen Sie zwei Probleme, die in dieser Architektur enthalten sind. Wie hilft R-Software bei der Lösung des Problems und welche Einschränkung gilt bei der Verwendung von R-Software?

3. Software Architektur (Fortsetzung)

3c) Zeichnen Sie nun einen verbesserten Kategoriegraphen. Begründen Sie warum Ihr Vorschlag die genannten Probleme löst.

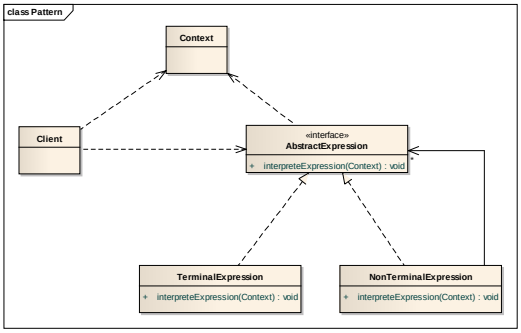
3d) Beschreiben Sie die Sichtbarkeitsregeln, welche Quasar basierend auf einem Kategoriengraphen definiert. Gehen Sie insbesondere darauf ein, welche Auswirkung auf die Schnittstellen sowie Parameter existiert. Welche besondere Stellung hat die Kategorie-0?

3e) Definieren Sie den Begriff „unreine Softwarekategorie“. Warum sollte man diese Softwarekategorie vermeiden, welches Problem existiert?

3.) Software Design – Pattern

3a) Erläutern Sie die Grundidee/-problem welches durch den Einsatz von Erzeugermuster gelöst werden soll

3b) Erläutern das nachfolgend dargestellte Interpreter-Pattern. Wann ist der Einsatz dieses Patterns sinnvoll und wo liegen die Grenzen.



3c) Skizzieren Sie ein Klassendiagramm welche das Interpretermuster auf die Berechnung von einfachen arithmetischen Ausdrücken wie z.B. 5 * (4 – 2) anwendet.