



## **Unterrichtsaufgabe**

### **LE 14 – SEP**

#### **Lernziel**

Sie haben 2 zentrale Aufgaben einer möglichen SEP selbst gelöst und kennen Ihre Stärken und Schwächen besser.

#### **Bemerkungen**

Die folgenden 2 Aufgaben entsprechen in Aufwand und Schwierigkeitsgrad ungefähr den entsprechenden Aufgaben in einer SEP.

Der Aufbau dieser Unterrichtsaufgabe weicht von den anderen Unterrichtsaufgaben ab, da er dem Aufbau von SEP-Aufgaben entspricht.

Erstellen Sie schlussendlich ein pdf Ihrer Lösungen und laden Sie es wie gewohnt zur Abgabe in MS-Teams hoch.

#### **Aufgabe 14.1 Domänenmodellierung Online-Handwerkbörse**

##### **Einleitung**

Die folgende Beschreibung dient als Basis für die Erstellung des Domänenmodells einer Online-Handwerkbörse.

Eine Online-Handwerkbörse hat 2 Arten von Mitgliedern: Auftraggeber und Handwerker. Jedes Mitglied wird mit seinem Namen identifiziert, besitzt ein Passwort und seine E-Mail-Adresse ist hinterlegt. Beim Auftraggeber muss zusätzlich noch die Adresse und beim Handwerker der offizielle Firmenname angegeben werden.

Ein Auftraggeber erfasst eine Ausschreibung mit Angabe eines Titels, einer genauen Beschreibung der Arbeit, einem gewünschten Ausführungsbeginn und den Ablauftermin der Ausschreibung.

Handwerker können auf eine Ausschreibung ein Angebot machen. Dabei erfassen sie im Angebot neben dem Preis auch weitere Details zur Auftragserledigung. Ein Angebot ist eine bestimmte Zeitdauer gültig.

Ein Auftraggeber kann nach dem Ablauftermin ein passendes Angebot auswählen, sofern das Angebot dann noch gültig ist. Falls ihm kein Angebot passt, kann er die Ausschreibung nochmals neu durchführen oder einfach stornieren.

##### **Aufgabenstellung**

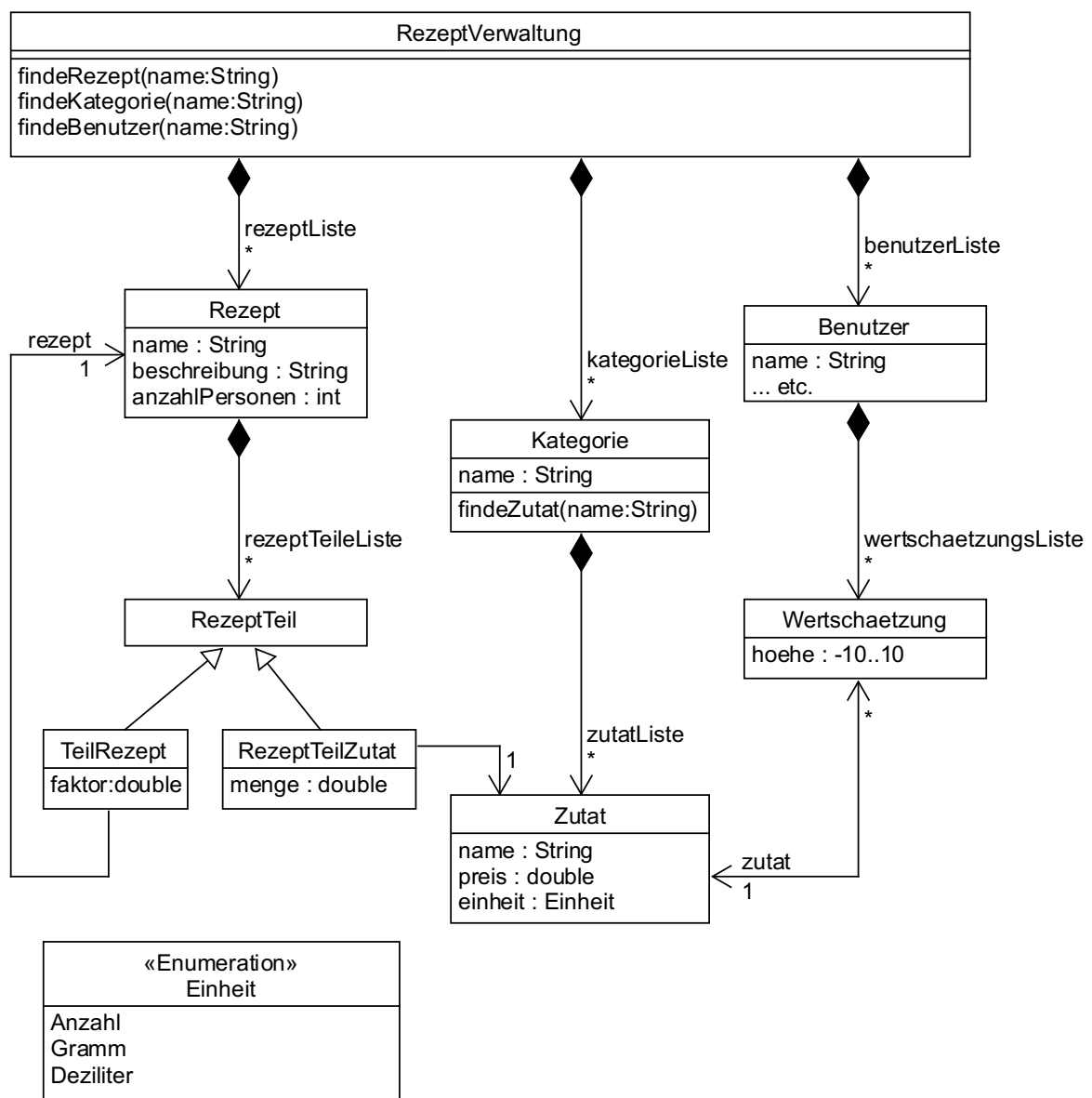
Analysieren Sie die Informationen aus der Problembeschreibung und leiten Sie davon ein Domänenmodell ab. Erfassen Sie auch wichtige Attribute pro Domänenklasse, insbesondere die, die in der Einleitung erwähnt werden.

## Aufgabe 14.2 Use-Case Realisation

### Einleitung

Die folgende Beschreibung dient als Basis für die Design Aufgaben. Es ist zwingend, dass Sie sich an diese Vorgaben halten, wenn Sie die Lösungen zu den Aufgaben erstellen.

Sie entwickeln eine Anwendung zur Verwaltung von Rezepten. Als Basis dient das folgende, unvollständige Design-Klassendiagramm.



Ein Rezept besteht aus mehreren Teilen. Ein Teil ist entweder eine Zutat und ihre Menge, oder dann ein anderes Rezept. Dies wäre z.B. bei einem Kuchen der Fall, wo die Glasur oder die Füllung ein eigenes (Teil-)Rezept darstellen, das in mehreren übergeordneten Rezepten verwendet werden kann.



Eine Zutat gehört zu einer Kategorie und hat einen Namen, Preis und die Einheit zum Preis.

Ein Benutzer dieser Anwendung kann (muss aber nicht), für Zutaten eine Wertschätzung erfassen und so festlegen, wie gerne er sie hat. Je positiver, desto lieber hat er sie, wobei 10 das Maximum darstellt. Eine negative Zahl bedeutet umgekehrt, dass er sie nicht gerne hat.

### **Aufgabenstellung «Finde Wertschätzung für Benutzer und Zutat»**

Entwerfen Sie die Systemoperation „int findeWertschaetzungFuerBenutzerUndZutat(String benutzerName, String zutatName)“.

Sie gibt für den spezifizierten Benutzer die Wertschätzung zurück, die er für die spezifizierte Zutat hat. Gibt es keine Wertschätzung, soll 0 zurückgegeben werden.

Zeichnen Sie dafür ein UML-Sequenzdiagramm.

Die Methoden, die im DCD bereits eingetragen sind, können Sie in Ihren Lösungen direkt einsetzen.

Die Zugriffe auf die Container-Objekte, die eine 1:\* Beziehung implementieren, müssen nicht explizit angegeben werden. Jedoch müssen alle Zugriffe auf die Resultatliste, inklusive ihrer Erzeugung, eingezeichnet werden.

Geben Sie im Diagramm noch an, welche GRASP Prinzipien und GoF Design Patterns Sie für die Realisierung der Systemoperation wo einsetzen.