Software Engineering I

Musterlösungen zur Klausur vom 3.8.2002

Aufgabe 1

a) In der Aufgabenstellung war ein möglichst einfaches Klassendiagramm gefordert. Daher verzichten wir auf Klassen, die zwar der Problemwelt entstammen, aber für die Lösung der geforderten Aufgabe und zur Beantwortung der vom Auftraggeber genannten Anfragen nicht erforderlich sind. In diese Kategorie fallen Klassen wie "Produktgruppe", "Warenkorb", "Produktliste", "Kasse", "Einkauf", "Zahlung", "Kreditkarte", "Preis", "Kundenkonto", "Buchung" usw.

Zur Lösung der Aufgabenstellung reicht das Klassendiagramm in Abb. 1 aus. Wichtig ist, dass bei der Bestellung von Waren deren Preis festgehalten wird und dass eine Bestellung auch mit Produkten verbunden sein kann, die nicht mehr im Sortiment sind. Daher definieren wir ein Attribut imSortiment in der Klasse Produkt und ein Attribut bestellpreis in der Klasse Einzelposten.

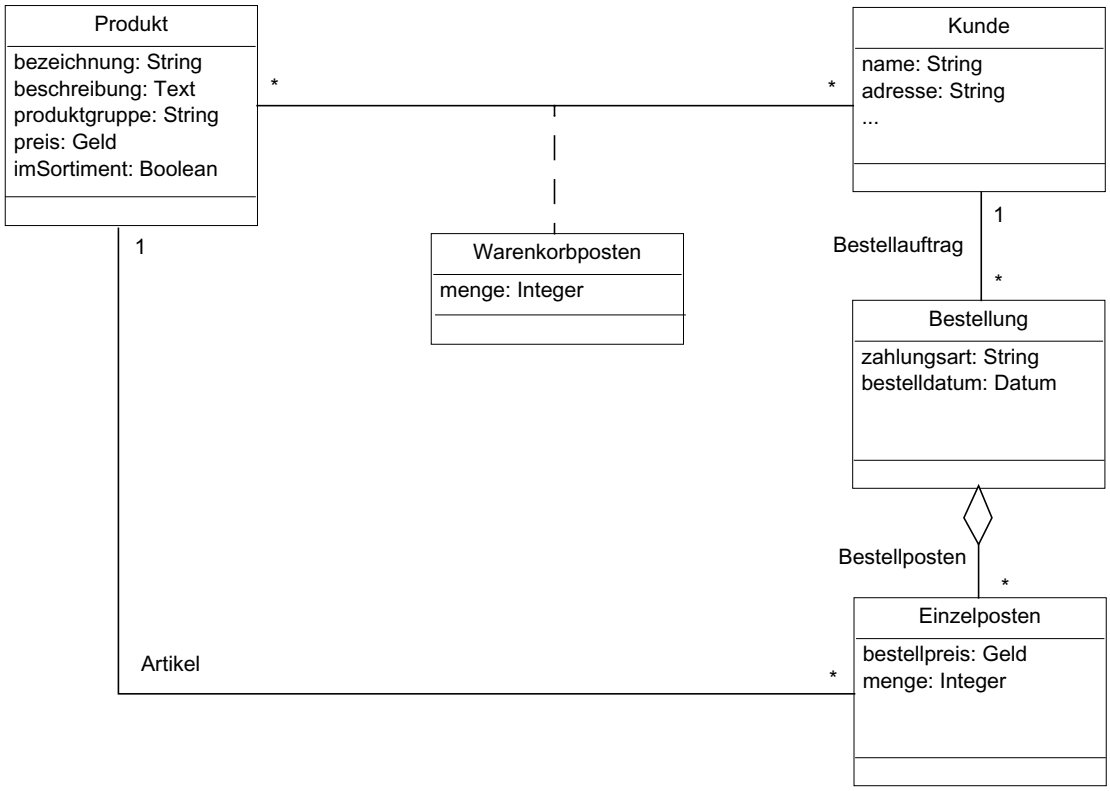

Abb. 1: Klassendiagramm des Warenbestellsystems

b) Abb. 2 zeigt das Objektdiagramm vor der Ausführung des Bezahlvorgangs, Abb. 3 zeigt das Objektdiagram danach.

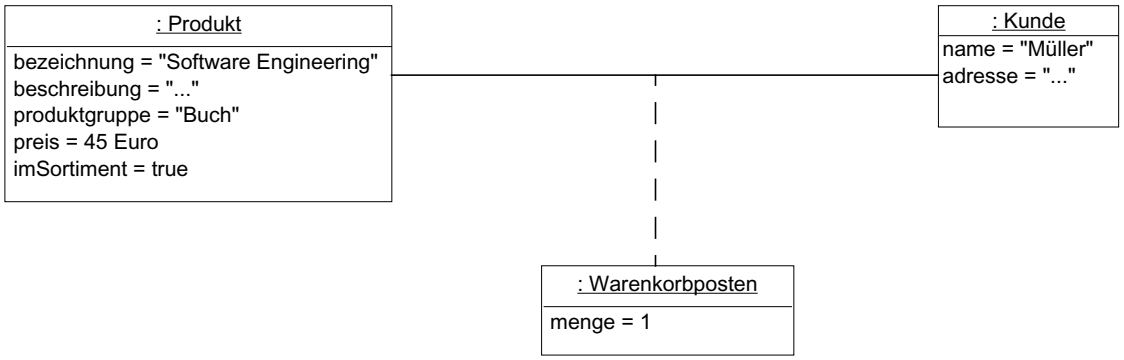

Abb. 2: Objektdiagramm vor Ausführung des Bezahlvorgangs

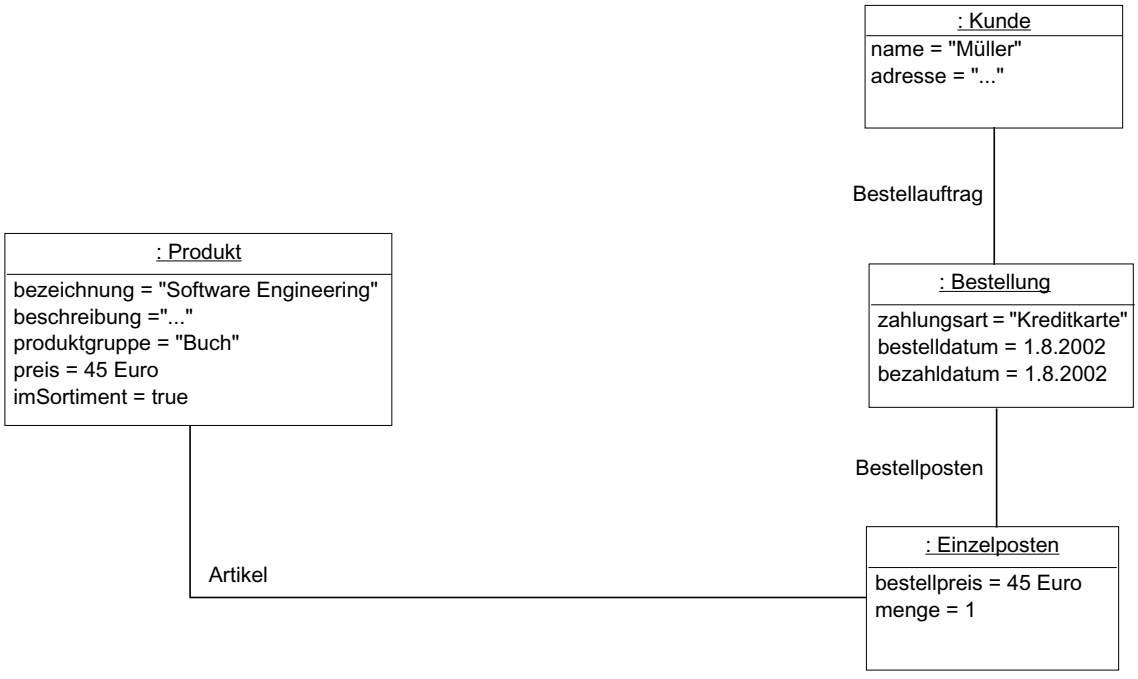

Abb. 3: Objektdiagramm nach Ausführung des Bezahlvorgangs

c) Es gibt mehrere Möglichkeiten, die beschriebenen Vorgänge als Anwendungsfälle zu modellieren. Abb. 4 zeigt eine davon.

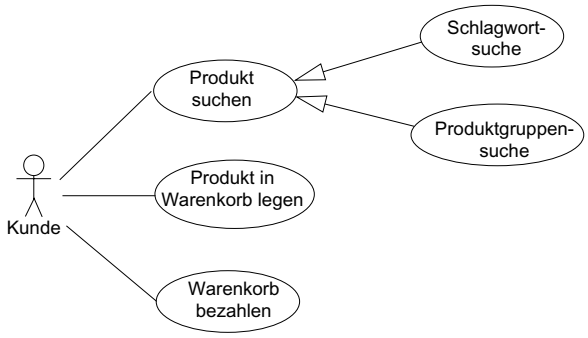

Abb. 4: Anwendungsfalldiagramm (Ausschnitt)

Aufgabe 2

Abb. 5 zeigt das Zustandsdiagramm des mobilen Roboters:

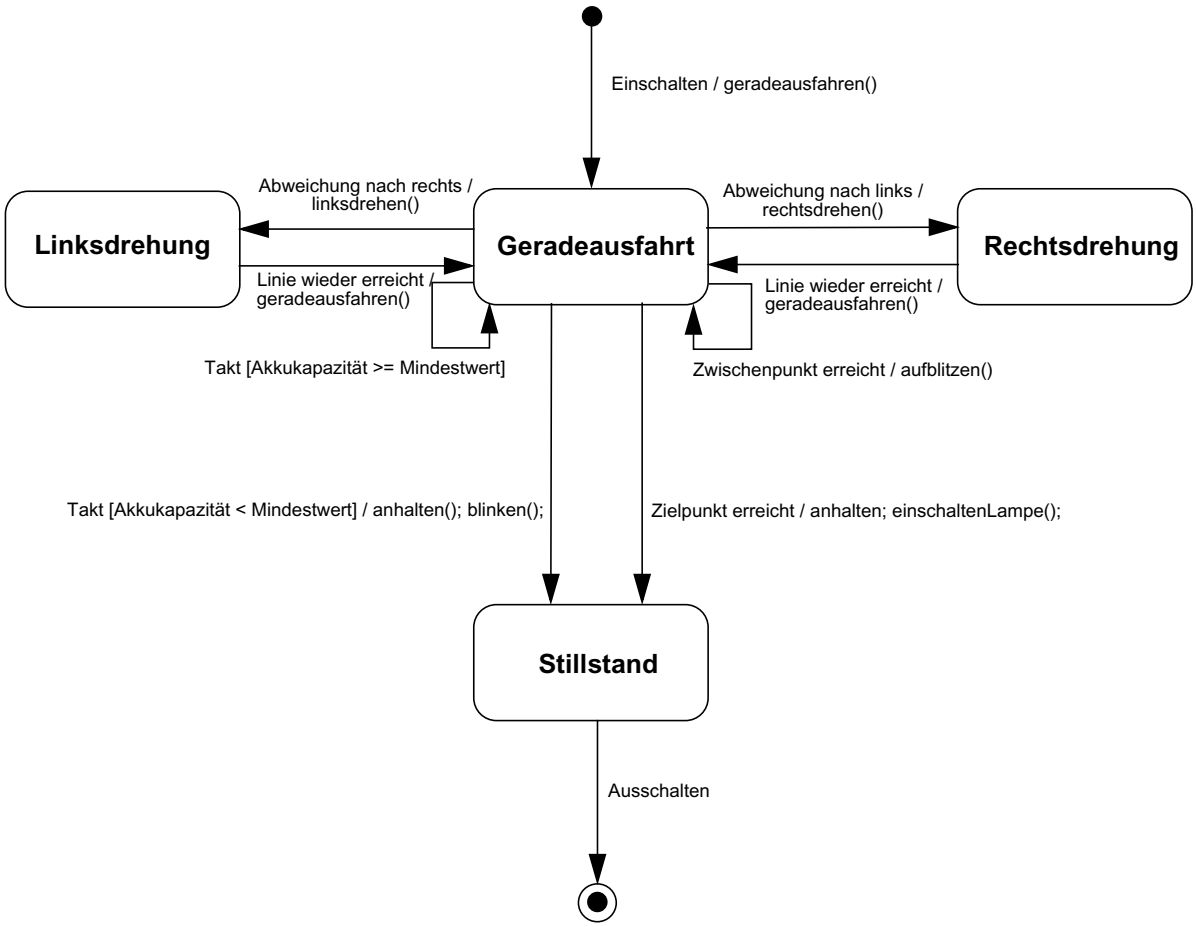

Abb. 5: Zustandsdiagramm für den mobilen Roboter

Aufgabe 3

a) Abb. 6 zeigt das Sequenzdiagramm für den Fall, dass ein Dienstnutzer die Operation dokument-Öffnen() einer Textapplikation aufruft.

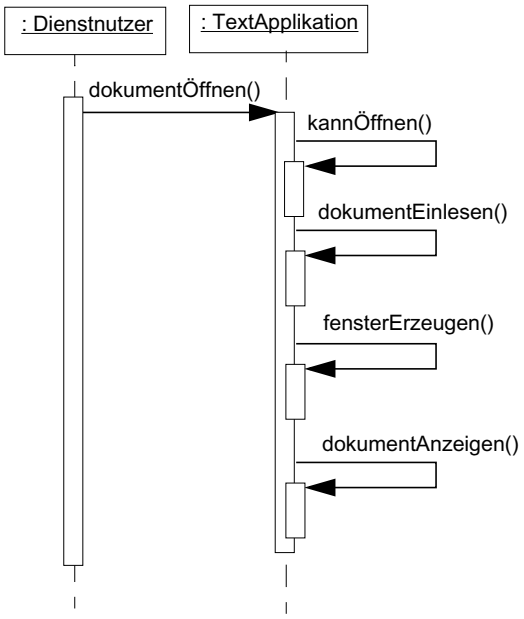

Abb. 6: Sequenzdiagramm der Operation dokumentÖffnen

b) Es handelt sich um das Muster Schablonenmethode (Template Method).

c) Lösung 1 ist ein schlechtes Beispiel für die Anwendung des Musters Schablonenmethode, da hier von allen Unterklassen jeweils die gesamte Operation kannÖffnen() überschrieben wird, obwohl sich die Unterklassenimplementierungen dieser Methode nur in einem kleinen Detail (nämlich der Erweiterung des zu überprüfenden Dateinamens) unterscheiden.

In Lösung 3 wurde dieser variable Teil von kannÖffnen() in die Operation gibErweiterung() ausgelagert, so dass Unterklassen nur die Operation gibErweiterung() überschreiben müssen (und nicht die komplexere Operation kannÖffnen()). Ein weiterer Vorteil von Lösung 3 ist, dass die Operation gibErweiterung() als abstract deklariert werden kann, so dass es nicht möglich ist, das Überschreiben in der Unterklasse zu vergessen.

Lösung 2 hat gegenüber den anderen beiden Lösungen den Nachteil, dass ein zusätzliches Attribut erforderlich ist und dass dieses auch noch im Konstruktor der Unterklasse initialisiert werden muss. Dadurch ergibt sich eine unnötige Kopplung zwischen dem Konstruktor und der Operation kannÖffnen(). Außerdem besteht die Gefahr, dass die Initialisierung des Attributs vergessen wird.

Aufgabe 4

a) Abb. 7 zeigt eine Lösung für ein Klassendiagramm, das die Anforderungen der Aufgabenstellung erfüllt.

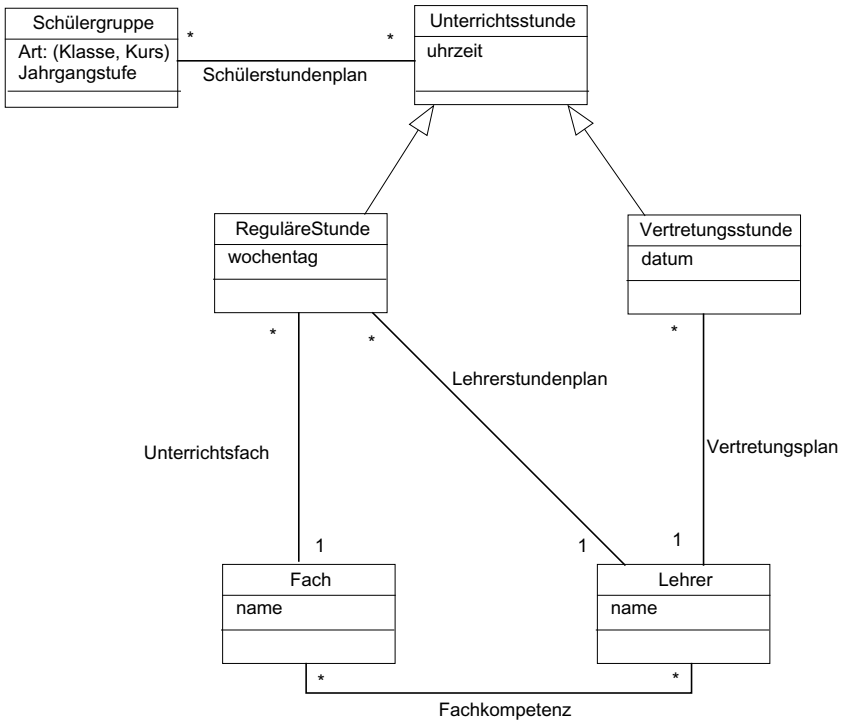

Abb. 7 Verbessertes Klassendiagramm zur Stundenplananwendung

b) Für die in a) gezeigten Vereinfachungen sprechen folgende Gründe:

- Die Information, ob es sich bei einer Schülergruppe um eine Klasse oder einen Kurs handelt, kann einfacher in einem Attribut festgehalten werden. Gleiches gilt für die Jahrgangstufe einer Klasse oder eines Kurses. Da sich das Verhalten dieser Klassen nicht (oder nur sehr geringfügig) unterscheidet, können sie nach Heuristik H 25.5 zusammengefasst werden.
- Die Information, ob eine Schülergruppe zur Ober-, Mittel- oder Unterstufe gehört, lässt sich aus der Jahrgangstufe entnehmen und braucht daher nicht extra modelliert zu werden.
- Die Angabe des Wochentags ist nicht für alle Unterrichtsstunden, sondern nur für reguläre Unterrichtsstunden erforderlich. Bei Vertretungsstunden ist der Wochentag durch das Datum gegeben.

- Für die Angabe des Wochentags ist keine eigene Klasse erforderlich, daher wurde die Klasse Wochentag durch ein Attribut Wochentag in der Klasse RegulärеStunde ersetzt.
- Die Assoziation GibtUnterricht ist zu stark generalisiert: Sie enthält sowohl Stunden aus dem Lehrerstundenplan (reguläre Stunden) als auch Vertretungsstunden. Da der Lehrerstundenplan und der Vertretungsplan in der Anwendungsdomäne separat gehandhabt werden, ist es sinnvoll, die Assoziation GibtUnterricht in der Aufgabenstellung durch die Assoziationen Lehrerstundenplan und Vertretungsplan zu ersetzen (s. Abb. 7).
- Die Klasse RegulärеStunde wurde entfernt, da sie keine Attribute und Operationen hatte. Statt dessen wird der reguläre Stundenplan jetzt durch die Oberklasse Unterrichtsstunde modelliert.