Software Engineering I

Musterlösungen zur Klausur vom 31.7.2004

Aufgabe 1

a) In der Aufgabenstellung war ein möglichst einfaches Klassendiagramm gefordert. Abb. 1 zeigt eine mögliche Lösung.

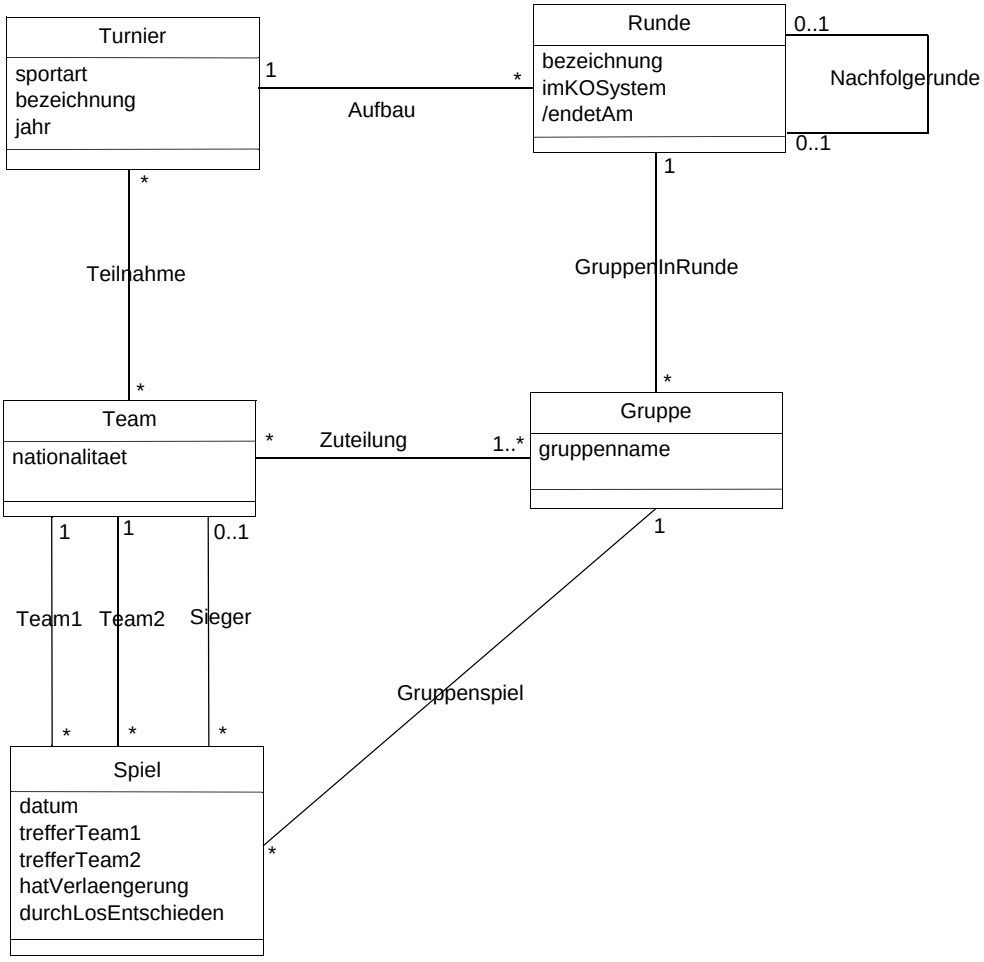


Abb. 1: Klassendiagramm für den Turnierveranstalter

Zwischen Team und Spiel muss es zwei getrennte Assoziationen "Team1" und "Team2" geben, damit die in der Klasse Spiel gespeicherten Trefferzahlen den Teams eindeutig zugeordnet werden können. Zusätzlich wird eine Asssoziation Sieger benötigt, da ein Spiel auch bei gleicher Trefferzahl einen Sieger haben kann (durch Losentscheid). Die Assoziation Nachfolgerunde beschreibt die zeitliche Reihenfolge der Runden.

Das Attribut endetAm in der Klasse Runde ist abgeleitet, da es dem Datum des letzten in dieser Runde stattfindenden Spiels entspricht.

b) Folgende Einschränkungen der Realität werden durch das Klassendiagramm in Abb. 1 nicht ausgedrückt:

- Ein Team kann nicht gleichzeitig an zwei Spielen teilnehmen.
- Eine Nachfolgerunde muss später enden als ihre Vorgängerrunde.
- Eine Runde kann nicht Nachfolgerunde von sich selbst sein.
- Team1 und Team2 müssen zur selben Gruppe gehören.
- Ein Team kann pro Runde nur einer Gruppe zugeteilt werden.
- Alle Gruppenspiele der vorhergehenden Runde müssen beendet sein, bevor die qualifizierten Teams ermittelt werden können

c) Abb. 2 zeigt ein Objektdiagramm für die in der Aufgabenstellung beschriebene Situation

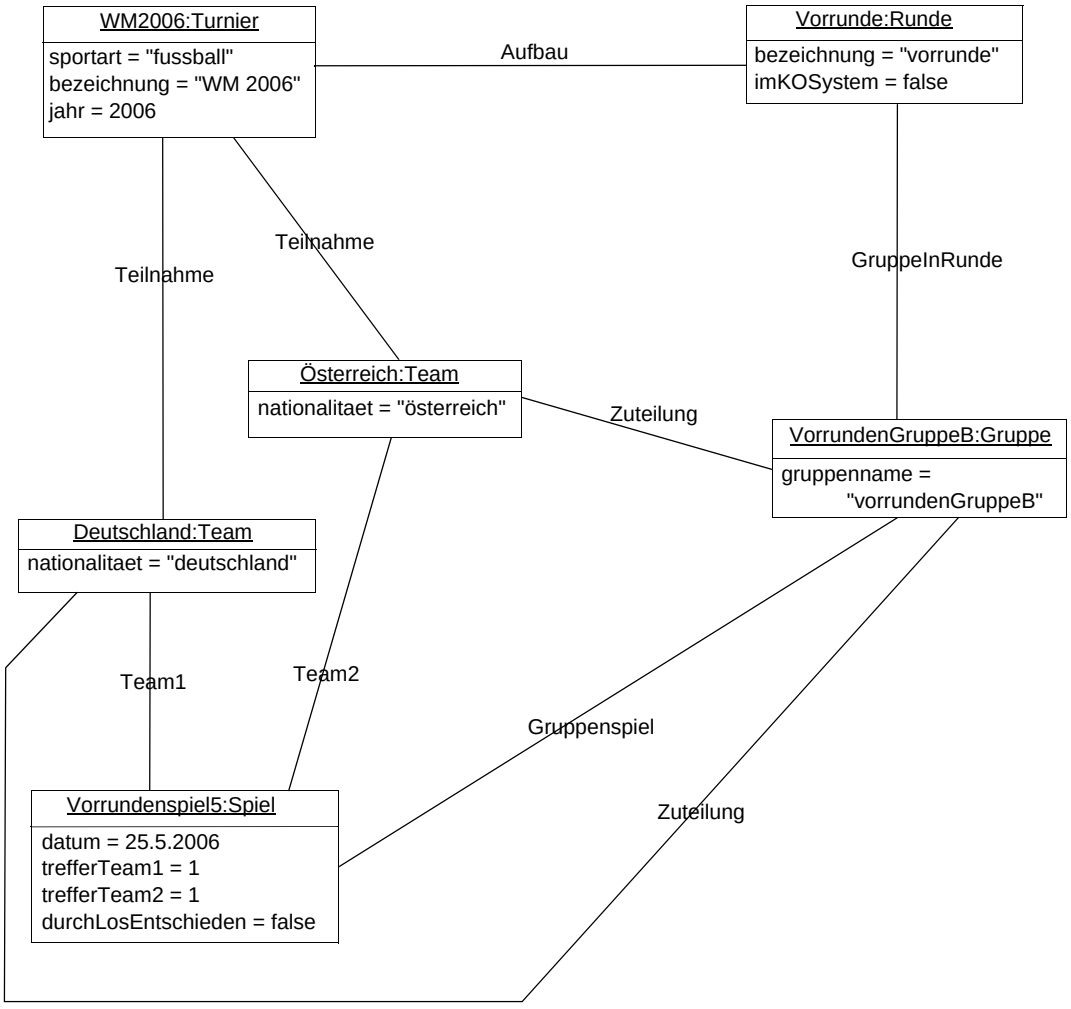


Abb. 2: Objektdiagramm zu Aufgabe 1c

d) Es gibt zwei Arten von Wetten: Spielwetten, bei denen auf ein konkretes Spielergebnis gewettet wird (Klasse Spielwette), und Turnierwetten, bei denen darauf gewettet wird, ob eine bestimmte Mannschaft eine bestimmte Runde erreicht (Klasse Turnierwette). Die gemeinsamen Eigenschaften dieser beiden Arten von Wetten fassen wir in der abstrakten Oberklasse Wette zusammen.

Das boolesche Attribut qualifiziertSich in der Klasse Turnierwette gibt an, ob auf Qualifikation oder Nichtqualifikation eines Teams gewettet wird.

Bei der Spielwette drücken wir den vom Kunden getippten Sieger durch die Assoziation GewetteterSieger aus. Soll auf Unentschieden gewettet werden, entfällt die entsprechende Verbindung.

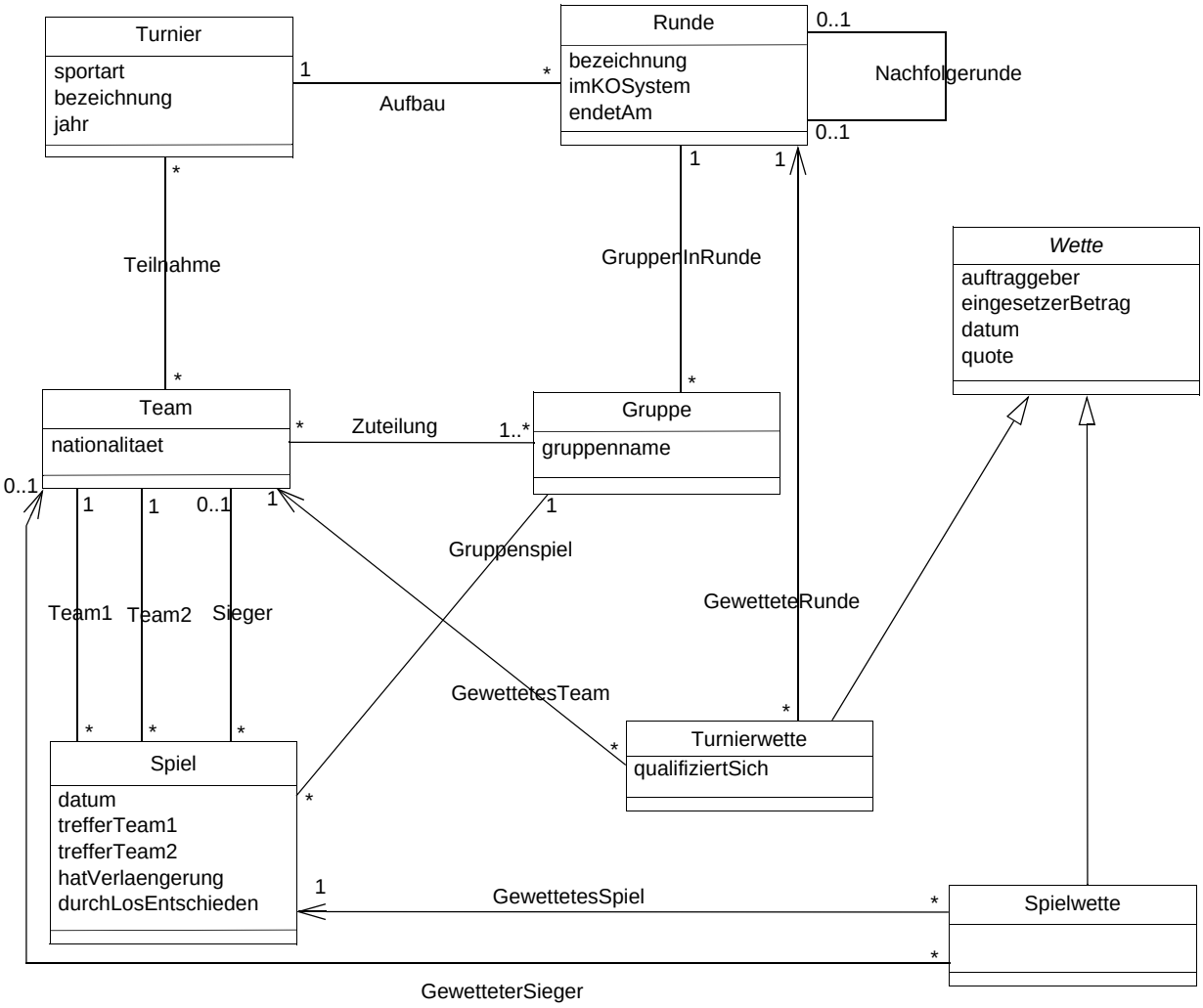


Abb. 3: Klassendiagramm für das Online-Wettbüro

Aufgabe 2

Zustandsdiagramm zur Modellierung des Lebenszyklus eines Buches in der beschriebenen Bibliothek:

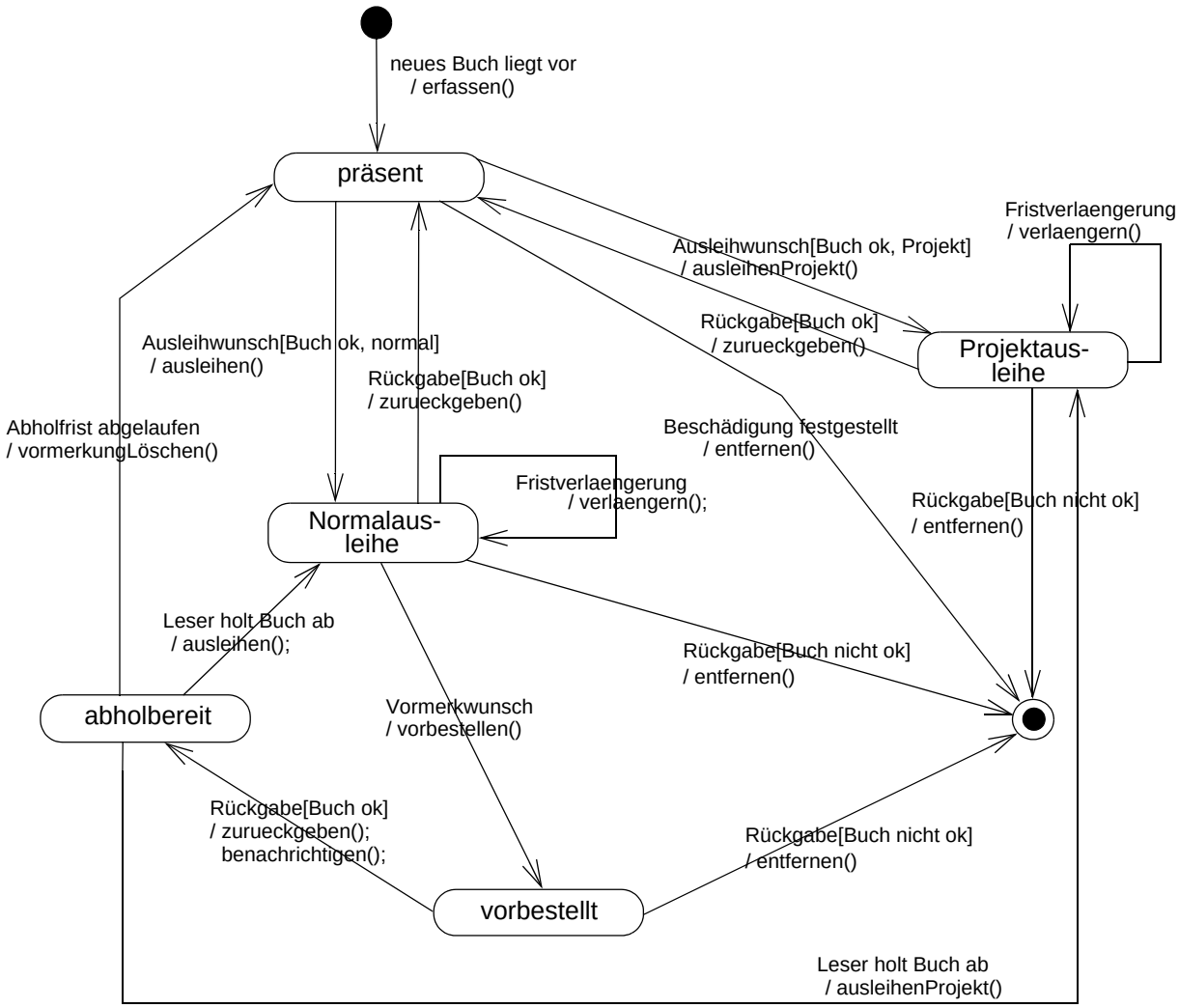


Abb. 4 Zustandsdiagramm zu Aufgabe 2

Aufgabe 3

a) Zur Transformation des Klassendiagramm-Ausschnitts zerlegen wir die Assoziation zunächst gemäß Abb. 48.7 des Kurstextes in zwei unidirektionale Assoziationen. Diese transformieren wir gemäß Abb. 48.6 und 48.7 in ein Feinentwurfs-Klassendiagramm. Als Behälterklasse definieren wir eine Klasse AngestelltenMenge. Abb. 5 zeigt das transformierte Klassendiagramm.
Statt der unidirektionalen Assoziation aktuellerVorgesetzter kann auch ein entsprechendes Attribut angegeben werden.

Abb. 5 Transformation des Klassendiagramm-Ausschnitts

b) Für jede der beiden Assoziationen definieren wir jeweils eine Operation zum Erstellen und zum Lösen einer Verbindung. Die Mengenklasse benötigt die üblichen Mengenoperationen hinzufuegen(), entfernen() und istEnthalten().

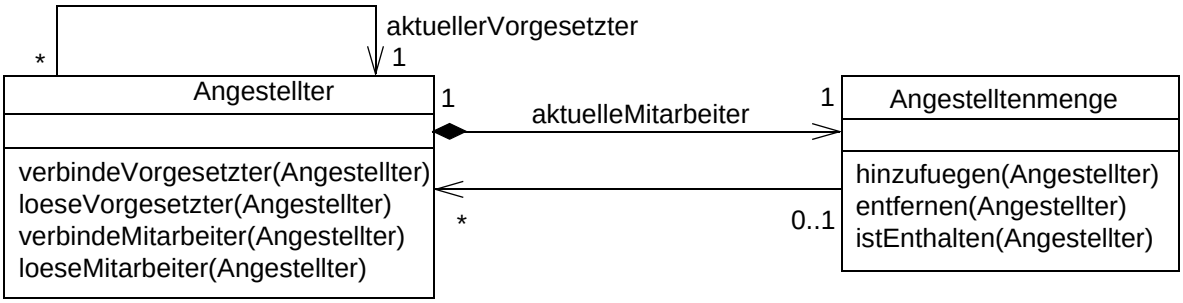
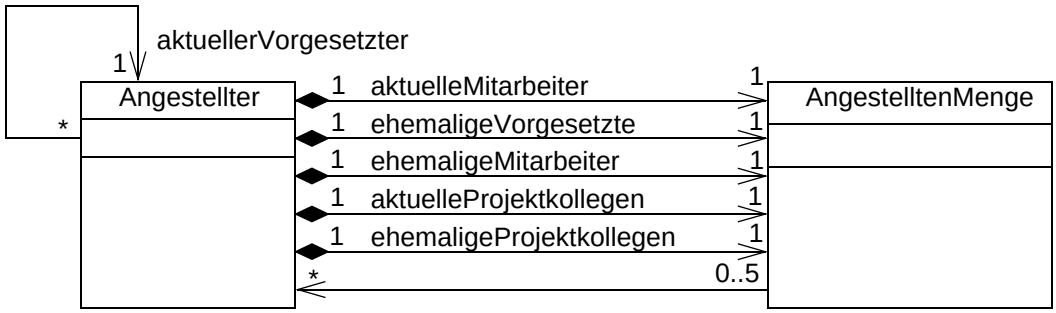


Abb. 6 Operationen zum Verbinden und Lösen

c) Die drei restlichen Assoziationen zerlegen wir wieder in jeweils zwei unidirektionale und transformieren diese zunächst schematisch nach Abb. 48.7. Von den entstehenden sechs Angestelltenmengen sind zwei redundant, da die Beziehung von einem Angestellten zu einem aktuellen oder ehemaligen Projektkollegen *kommutativ* ist: Ist (a1, a2) ein Paar von Projektkollegen, so gilt dies auch für das Paar (a2, a1). Gegenüber Abb. 6 kommen also lediglich vier Kompositionen hinzu.

Von der Klasse Angestelltenmenge zur Klasse Angestellter darf nur eine Assoziation existieren,

da eine Angestelltenmenge nur eine und nicht mehrere Mengen von Angestellten enthält.



Aufgabe 4

a) Abb. 7 zeigt ein Kollaborationsdiagramm, das dem Sequenzdiagramm in der Aufgabenstellung entspricht.

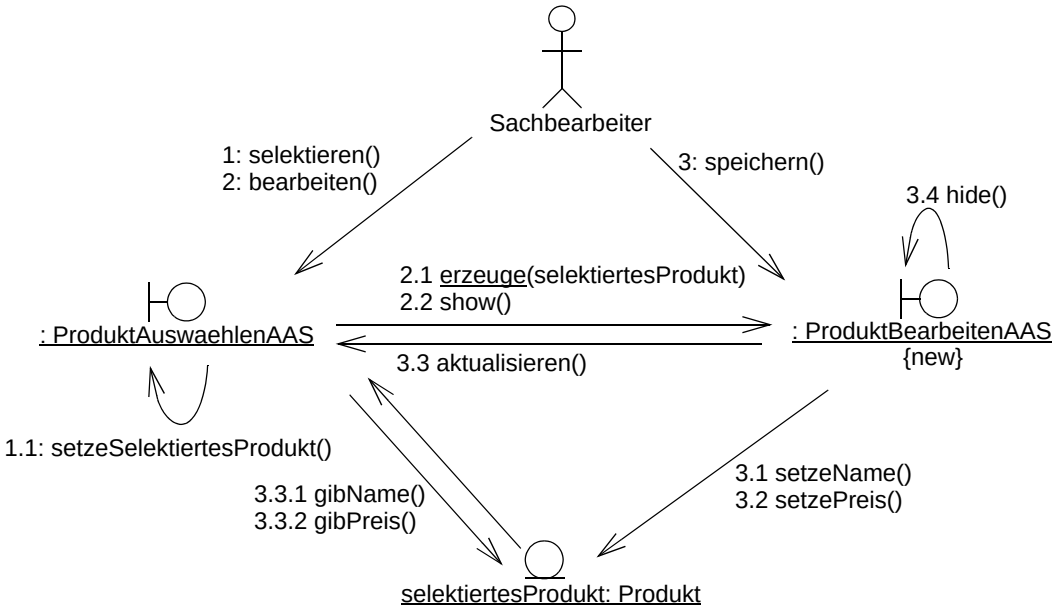


Abb. 7 Der Ablauf als Kollaborationsdiagramm

b) Wir listen einige Nachteile von zyklischen Abhängigkeiten auf:

- Sie verursachen eine hohe Kopplung zwischen den beteiligten Klassen.
- Sie widersprechen dem Prinzip hierarchischer Strukturen.
- Sie erschweren das Testen, da keine der im Zyklus enthaltenen Klassen ohne die andere Klassen des Zyklus getestet werden kann.

c) Abb. 8 zeigt das modifizierte Klassendiagramm.

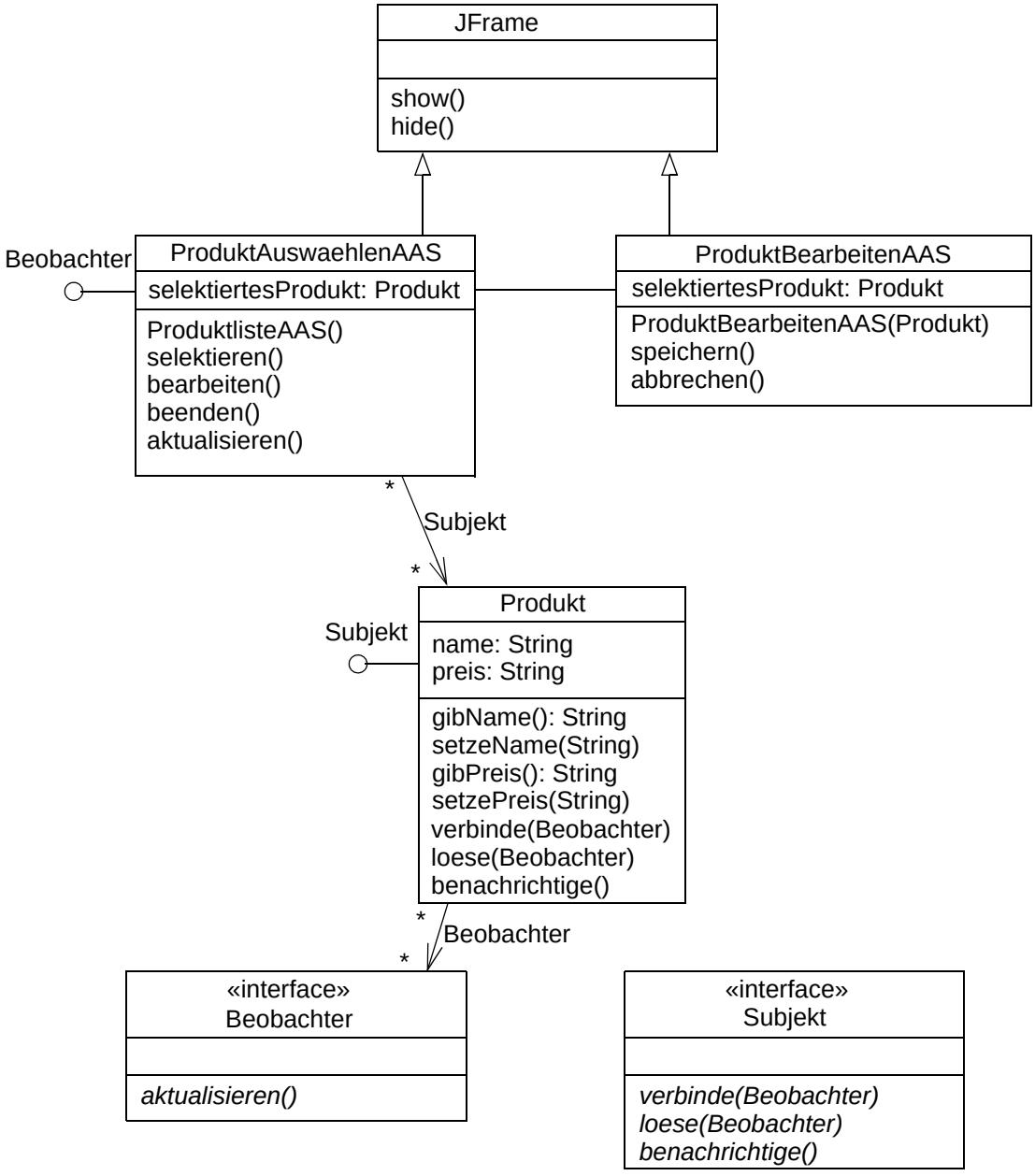


Abb. 8 Die zweite Entwurfsvariante

d) Abb. 9 zeigt den geänderten Ablauf der Operation speichern().

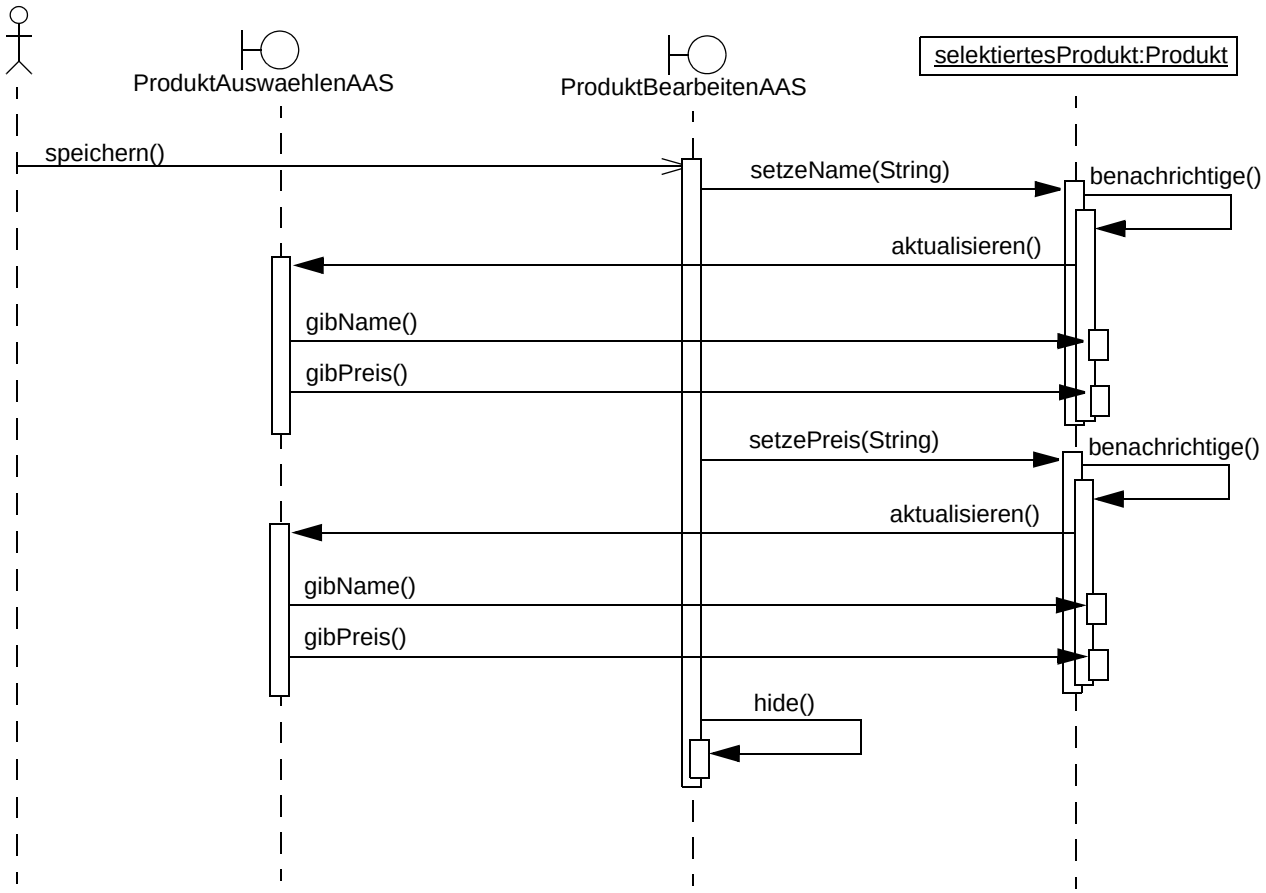


Abb. 9 Der geänderte Ablauf als Sequenzdiagramm

e) Der in Aufgabe c) erstellte Entwurf hat die Eigenschaft, dass die Liste bei jeder Änderung des selektierten Produkts aktualisiert wird. Dies ist zwar im Allgemeinen ein Vorteil, da die Liste ist dadurch jederzeit auf dem aktuellen Stand ist. Der Nachteil besteht darin, dass die Aktualisierung mehrfach erfolgt. (Die Folge ist eine geringere Performanz und ggf. ein Flackern bei der Bildschirmdarstellung.)