

Software Engineering 1



Anna Prenzel

Einführung

Ziele der heutigen Veranstaltung:

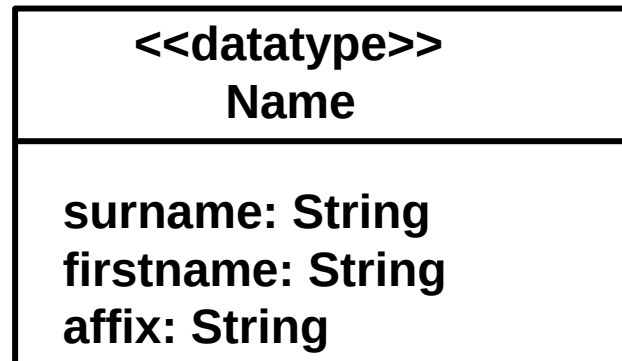
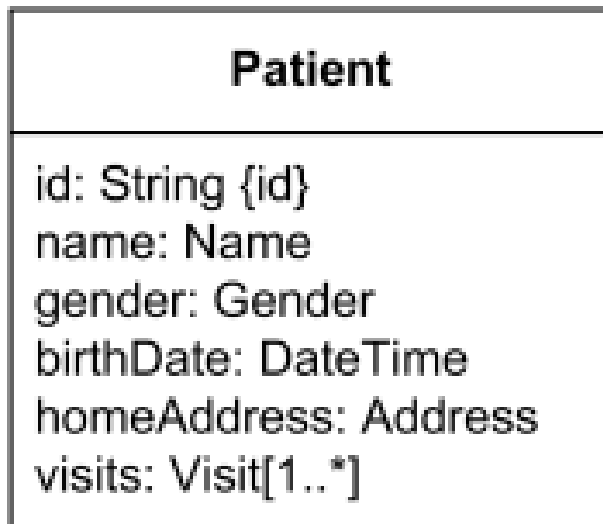
- ✎ Möglichkeiten zur Umsetzung von Klassendiagrammen in Quellcode (Java) kennen lernen
- ✎ Aufbau von Sequenzdiagrammen kennen lernen
- ✎ Fortsetzung des Vorgehensmodells zur objektorientierten Analyse

Wiederholung

- ✎ Erläutern Sie zwei Kriterien, nach denen die Paketbildung erfolgen sollte!
- ✎ Wie kann man ein Analysemodell verbessern, welches eine Klasse mit zu vielen Attributen enthält? → Vor.: die Attribute lassen sich zu einer logischen Einheit zusammenfassen

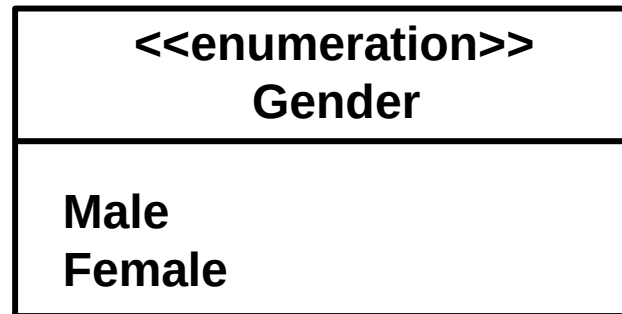
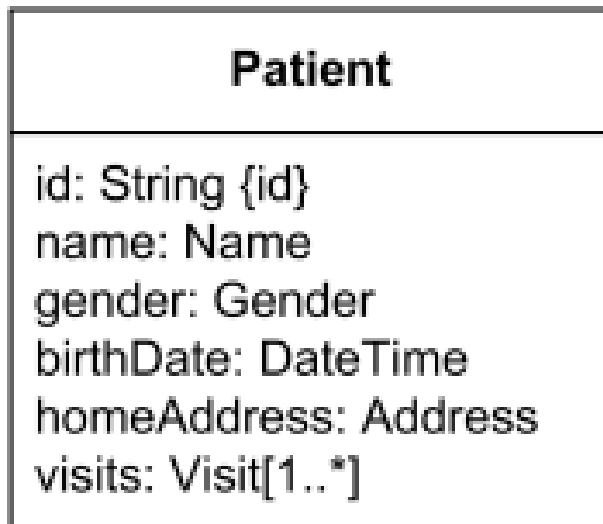
Wiederholung

Stereotyp Datatype



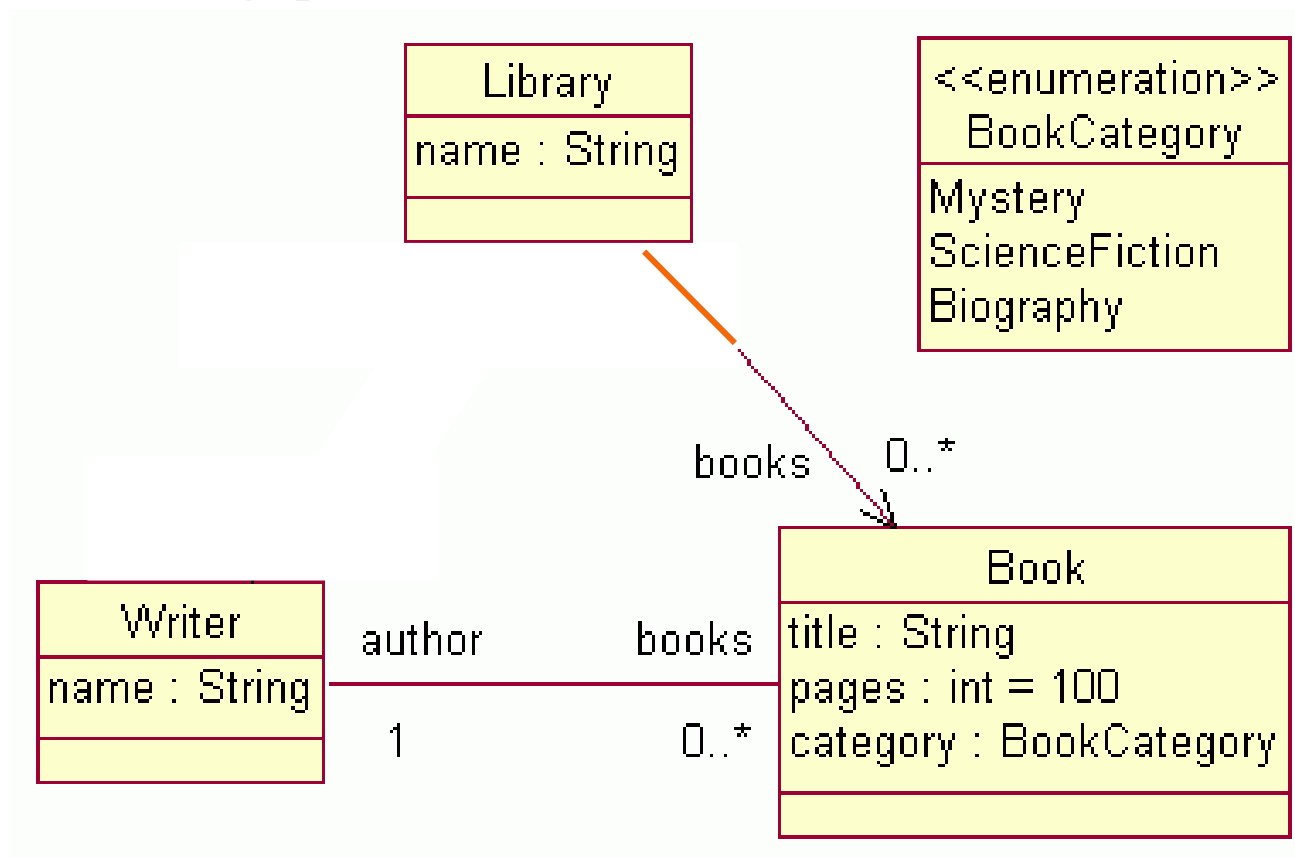
Wiederholung

Stereotyp Enumeration



Wiederholung

Stereotyp Enumeration

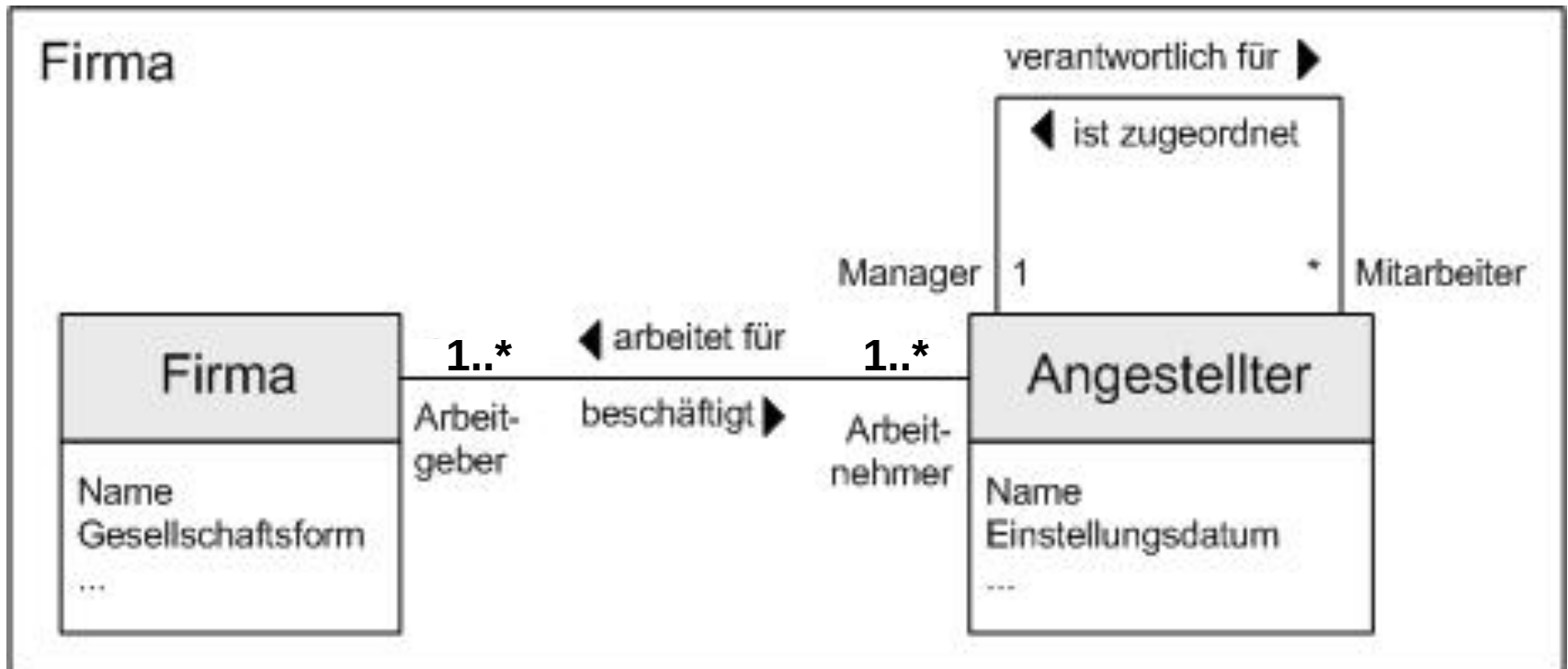


Wiederholung

- ☞ Nennen Sie zwei Eigenschaften einer Kompositionsbeziehung!
- ☞ Nennen Sie Beispiele für Kompositionsbeziehungen!

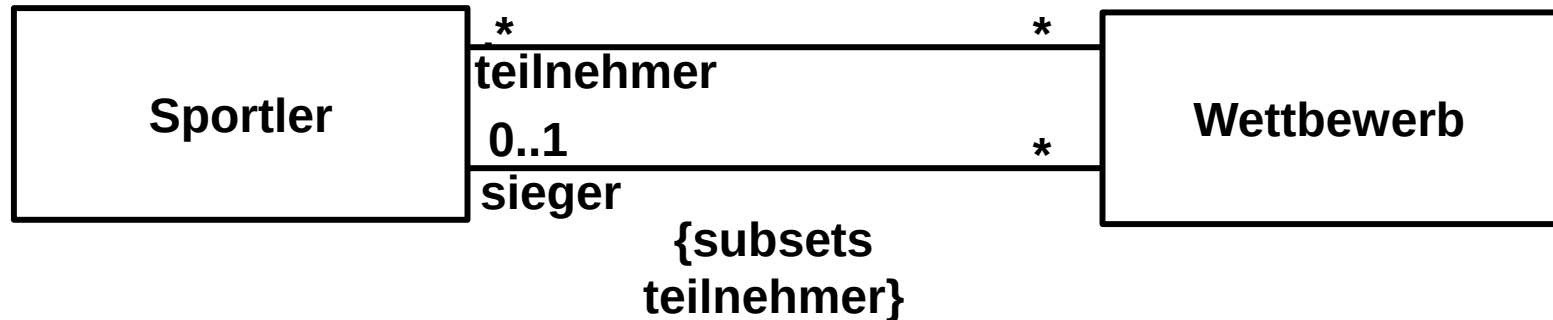
Übersetzung in Code

Setzen Sie das Klassendiagramm in Java-Code um!



Übersetzung in Code

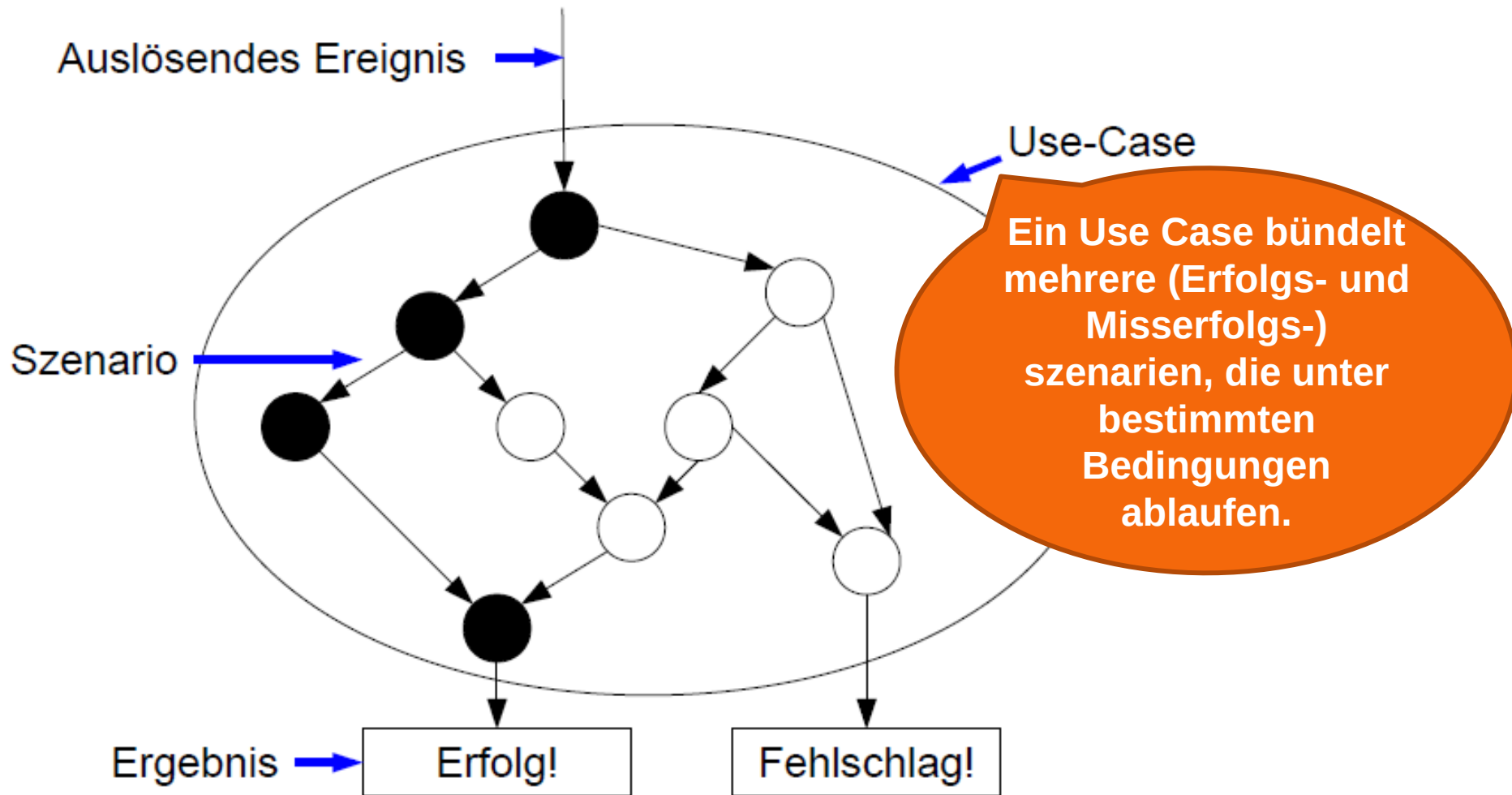
Setzen Sie das Klassendiagramm in Java-Code um!



Klassendiagramme – und jetzt?

- ✎ Klassendiagramme zeigen die Struktur des Systems
→ statische Modellierung
- ✎ Nun soll das Verhalten betrachtet werden
→ dynamische Modellierung

Use Case vs. Szenario



Szenarien

- ✎ Alternativ kann ein Szenario auch eine komplexe Operation beschreiben, die z.B. einem Interaktionsschritt im Use Case entspricht.
- ✎ Szenarien sind der Ausgangspunkt für die dynamische Modellierung.

Szenarien

Vorgehensweise zur Modellierung von Szenarien

1. Identifiziere die Klassen, die zur Realisierung eines Szenarios benötigt werden
2. Bilde den Ablauf des Szenarios auf eine Sequenz von Nachrichten zwischen den Klassen ab.

 **Interaktionsdiagramme**

Szenarien

Arten von Interaktionsdiagrammen

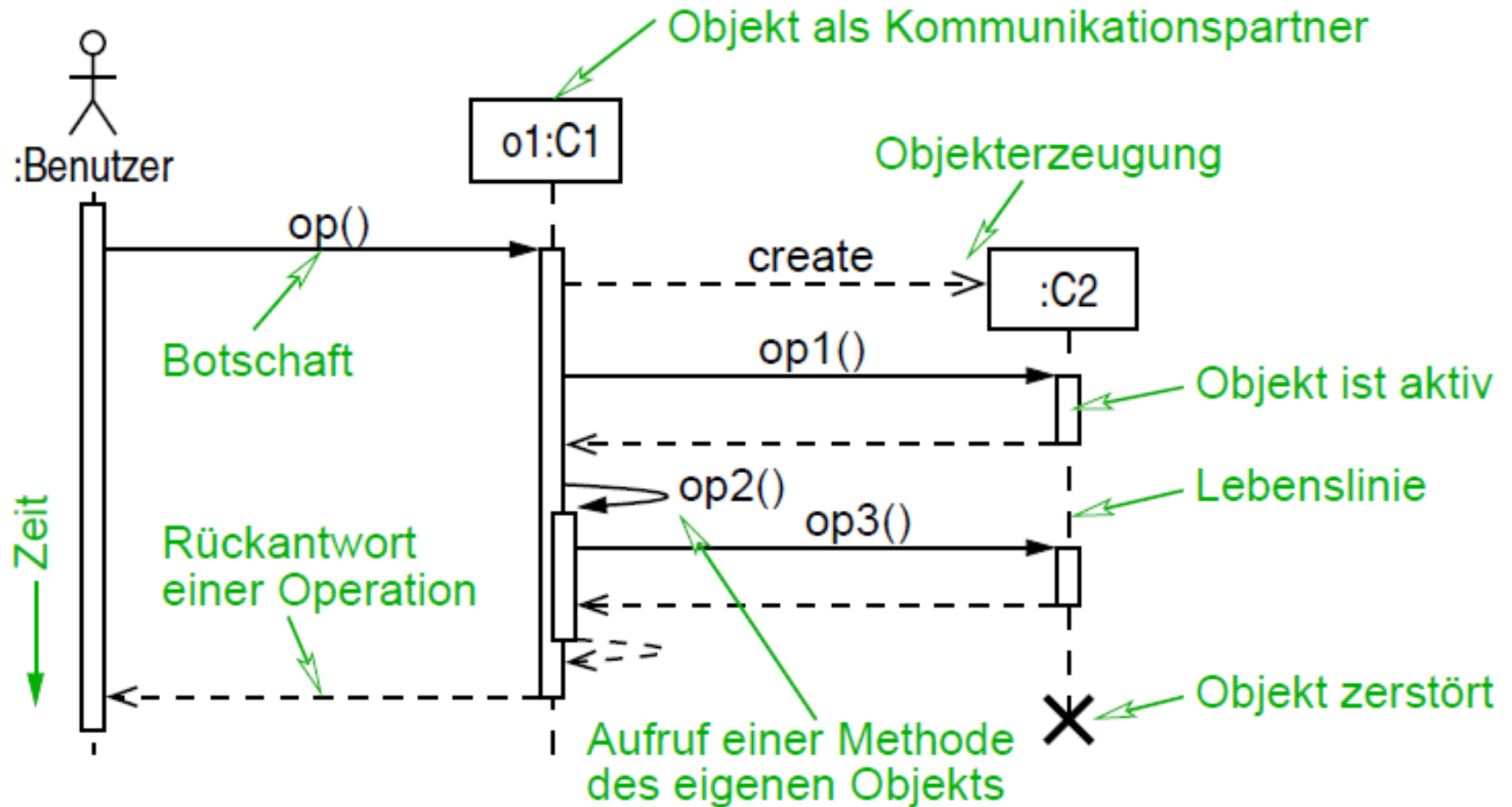
- ⌘ Sequenzdiagramme
 - ⌘ Kommunikationsdiagramme
 - ⌘ Timing-Diagramme
 - ⌘ Interaktionsübersichtsdiagramme
- unterschiedliche Sichtweisen auf die Interaktion

Sequenzdiagramme

Sichtweise

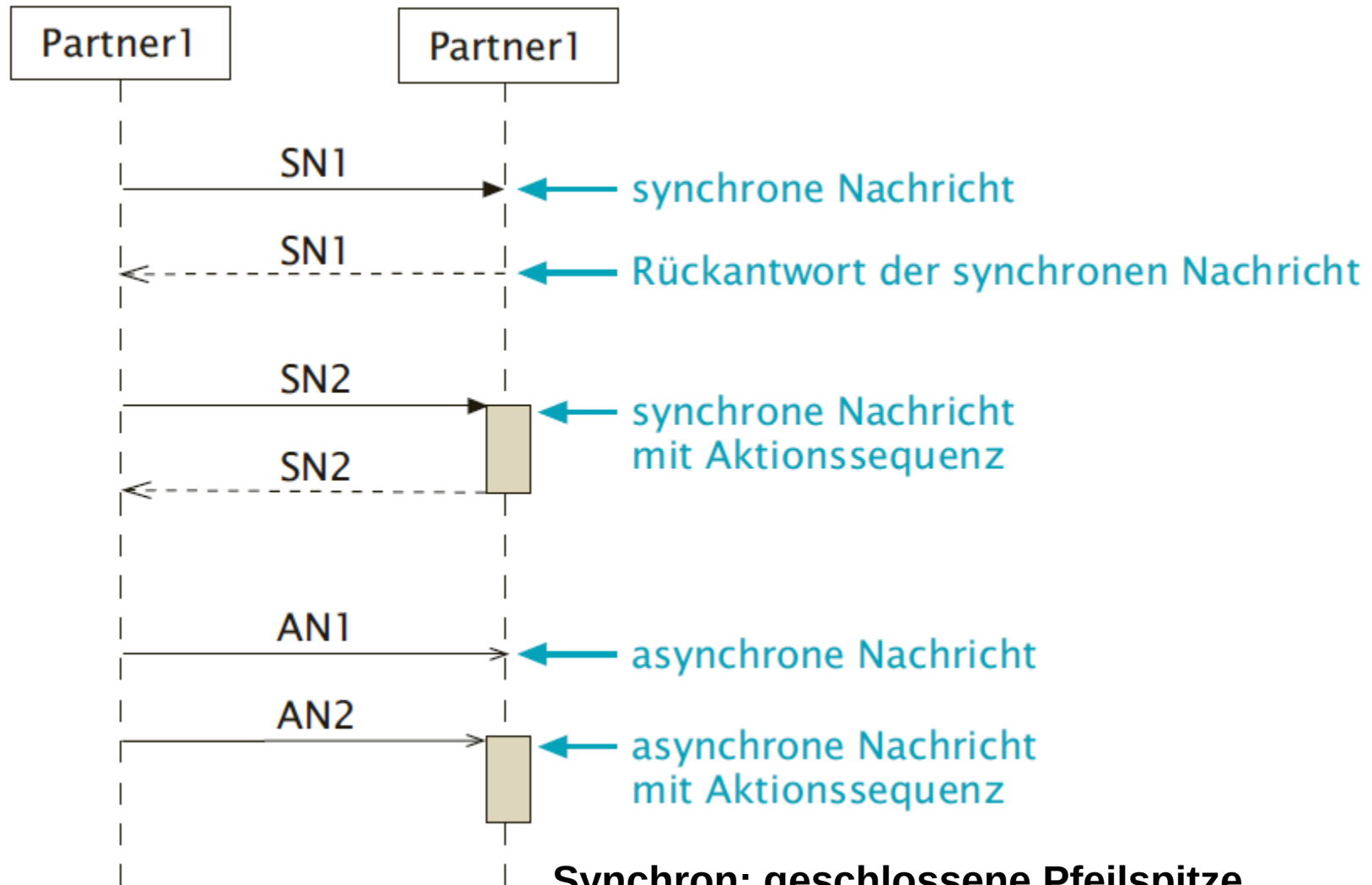
- ✎ Nachrichtenaustausch zwischen Objekten erfolgt in einem bestimmten zeitlichen Ablauf
- ✎ Erzeugung/Zerstörung von Objekten erfolgt in einem bestimmten zeitlichen Ablauf

Aufbau eines Sequenzdiagramms



Quelle: Heide Balzert, Lehrbuch der Objektmodellierung

Synchrone und asynchrone Nachrichten



Synchron: geschlossene Pfeilspitze
Asynchron: offene Pfeilspitze

Sequenzdiagramme

Einsatz in der Analyse

- ✎ Basis für Diskussion der fachlichen Korrektheit von Szenarien
- ✎ frei wählbarer Detaillierungsgrad
→ Code muss nicht unmittelbar aus Diagrammen abgeleitet werden können

Sequenzdiagramme

Einsatz in der Analyse

- ✎ Nur Klassen aus dem Domänenmodell (DM) verwenden!
- ✎ Bitte nicht versuchen, bereits den zukünftigen Programmablauf in Java zu modellieren.
- ✎ Das macht die Erstellung von SDs sehr schwierig, da wichtige technische Klassen noch nicht im DM enthalten sind!

Sequenzdiagramme

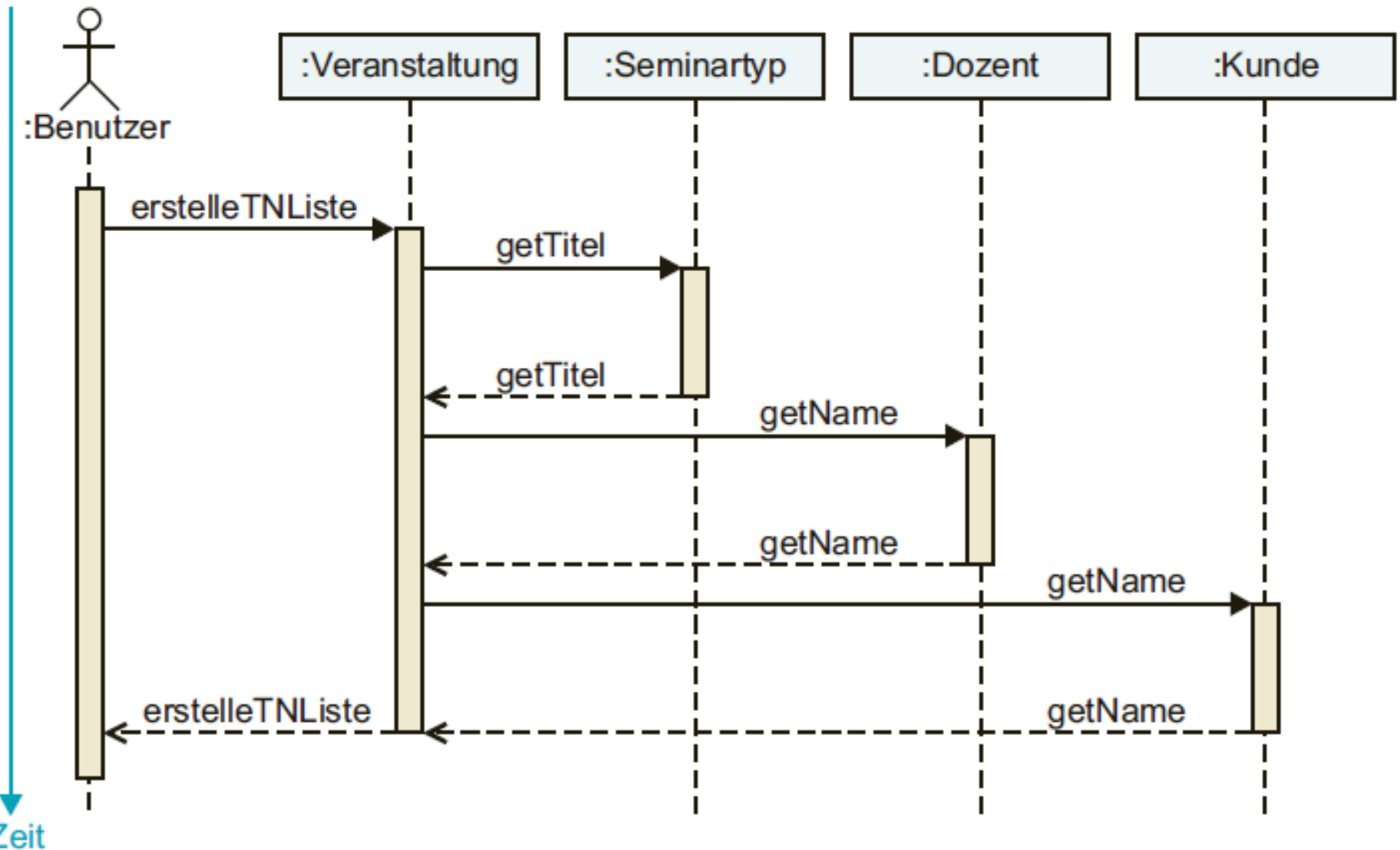
Analyse-SD spiegelt Überlegungen wieder:

- ☞ Welche Objekte aus dem DM sind an einem Szenario beteiligt?
- ☞ In welcher Abfolge und zu welchem Zweck wird auf ihre Daten oder Dienste zugegriffen?

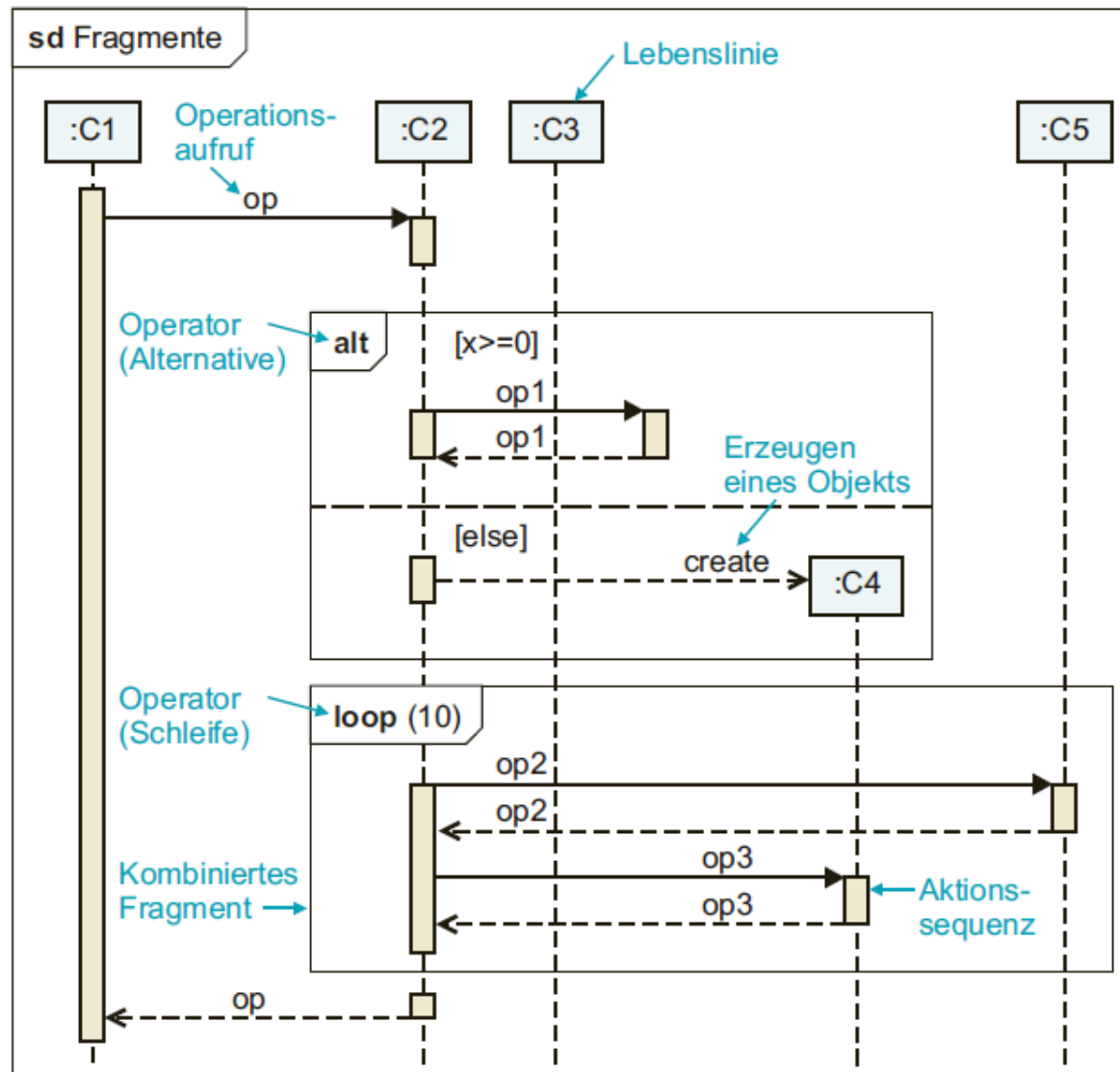
Sequenzdiagramme

- ✎ Aus SDs können erste (!) Rückschlüsse über die benötigten Methoden der einzelnen Klassen gezogen werden.
- ✎ Ggf. wird das DM aufgrund des Analyse-SDs um weitere (fachliche) Klassen ergänzt.

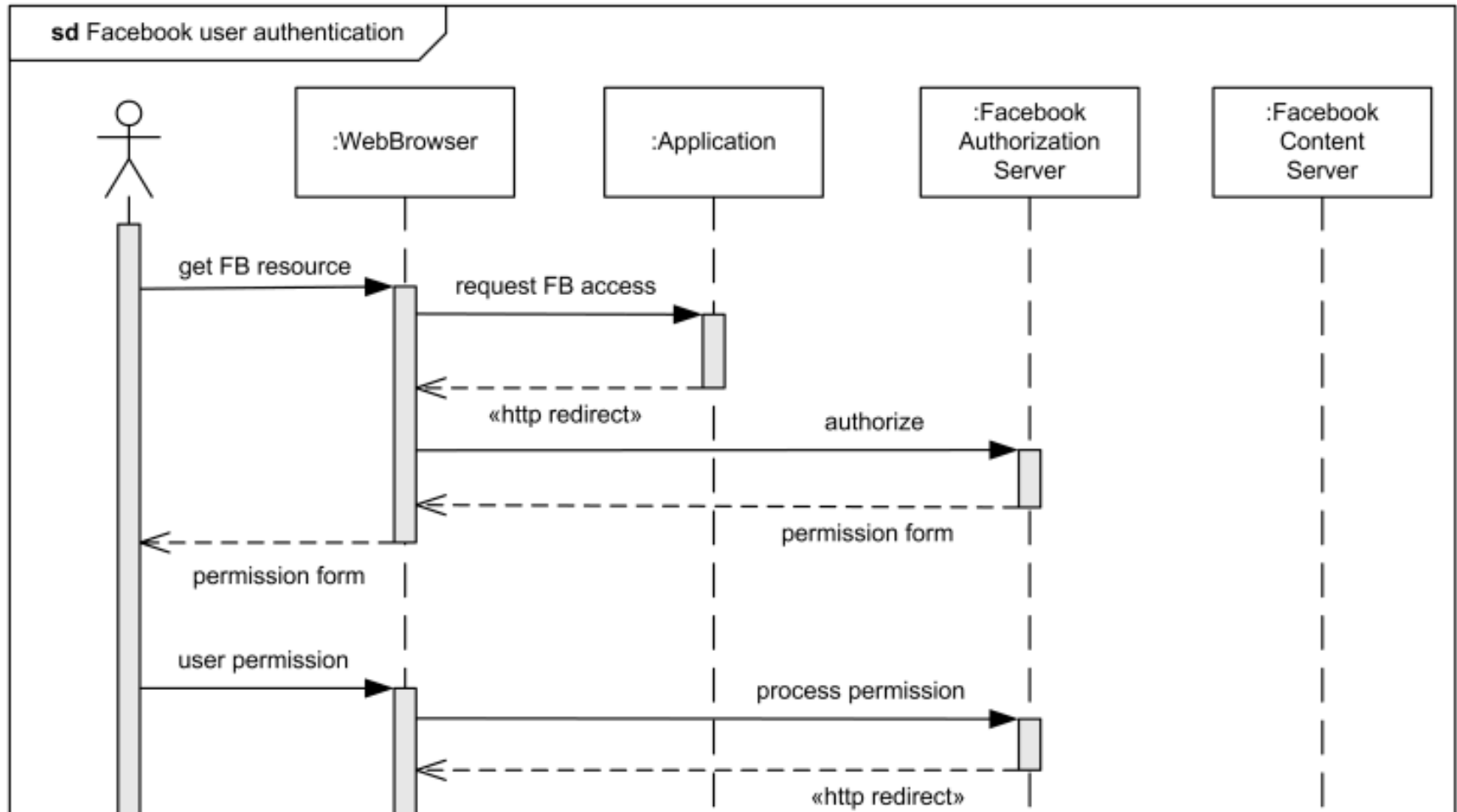
SemOrg: Szenario „Teilnehmerliste erstellen“



Bedingungen und Schleifen im SD



Szenario „Facebook-Authentifizierung“



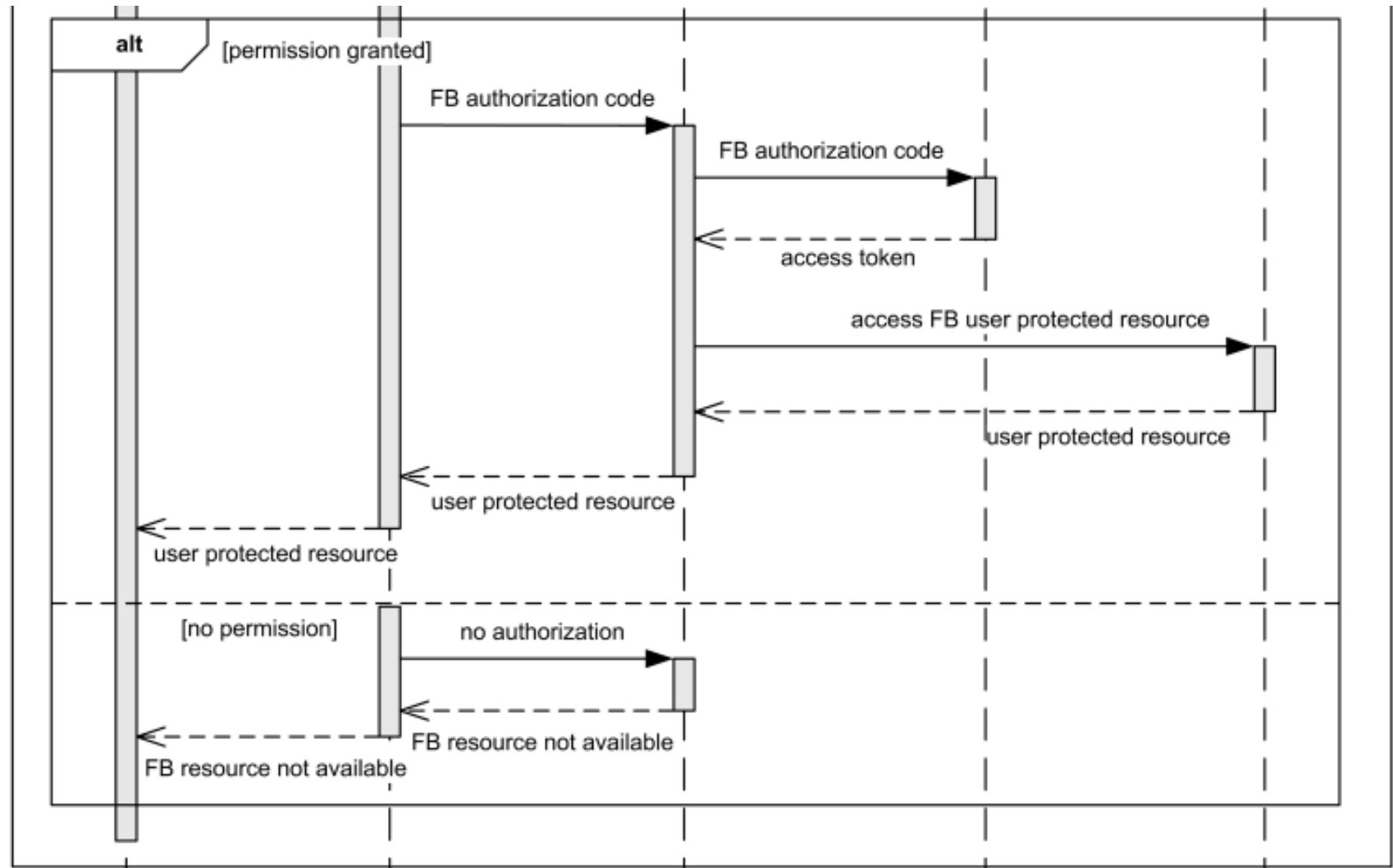


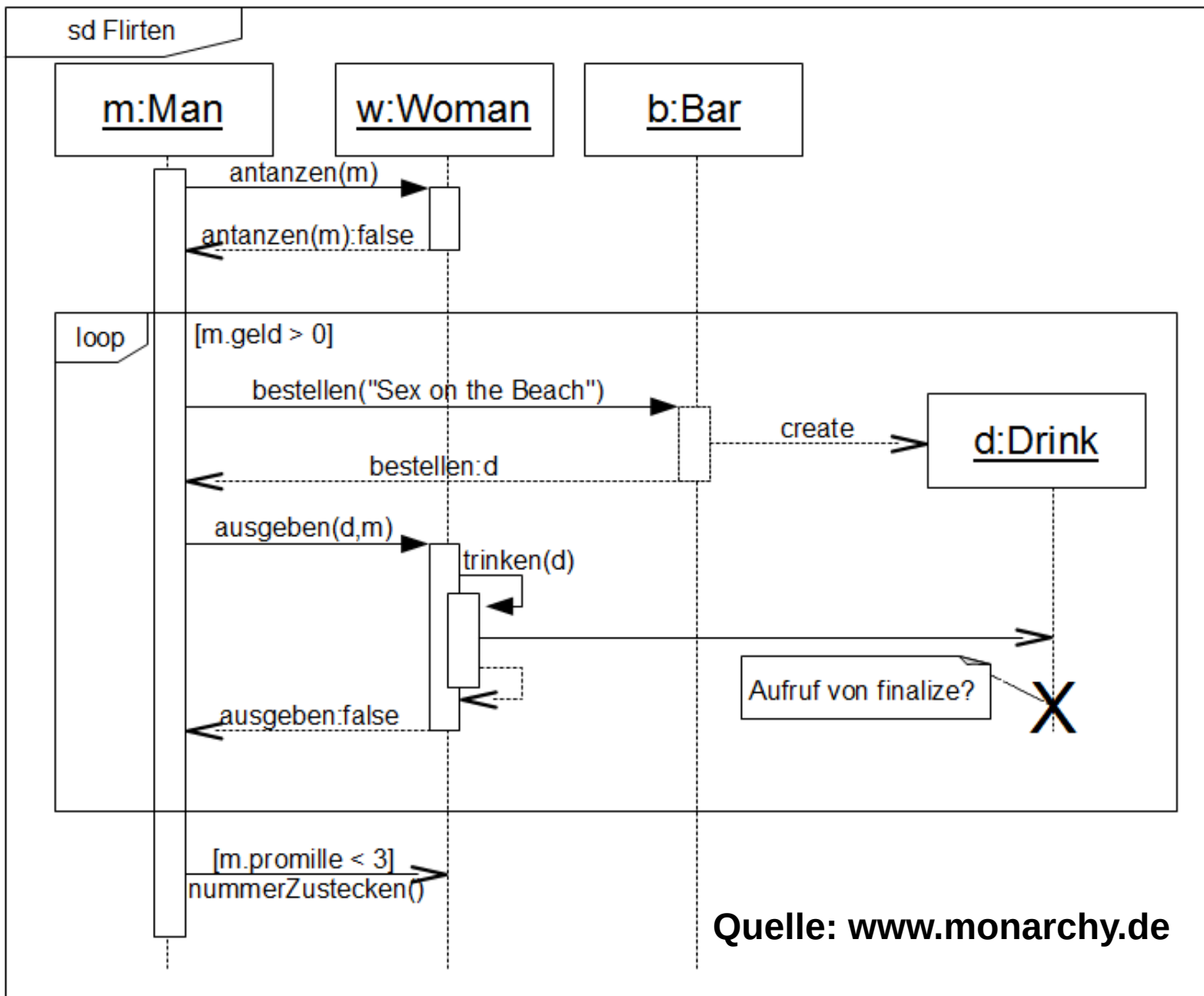
:WebBrowser

:Application

:Facebook
Authorization
Server

:Facebook
Content
Server





Sequenzdiagramme

Quellen für Szenarien

- ☞ Szenarien im Business-Use-Case → **Sequenzdiagramme**
- ☞ Szenarien im System-Use-Case → **System-Sequenzdiagramme**
- ☞ Komplexe Operation innerhalb eines Szenarios → **Sequenzdiagramme**

Sequenzdiagramme

Szenarien im Business-Use-Case

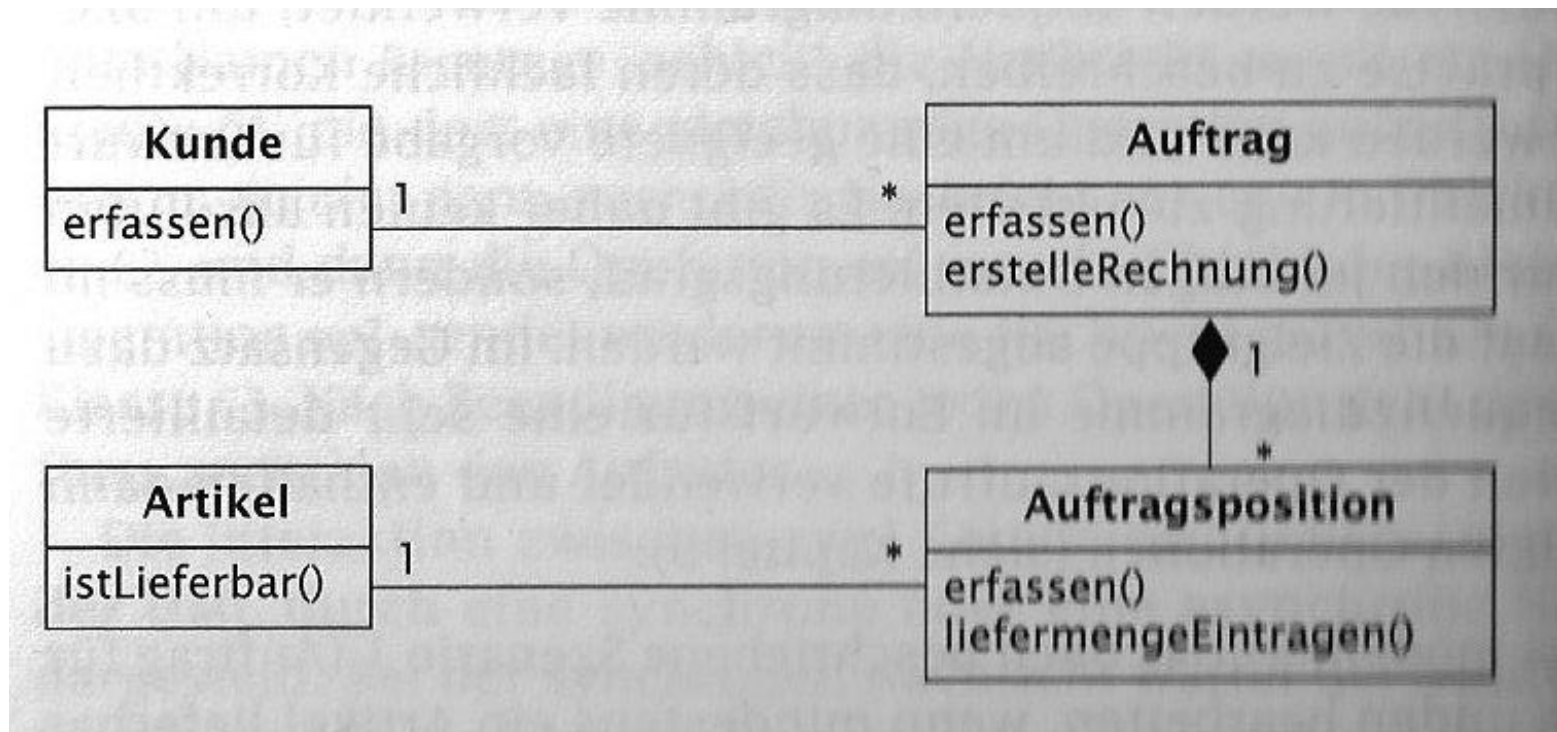
- ☞ Use Case zeigt Geschäftsprozesse, die im zukünftigen System u.U. automatisiert ablaufen.
- ☞ Beispiel: Schichtplanerstellung

Sequenzdiagramme

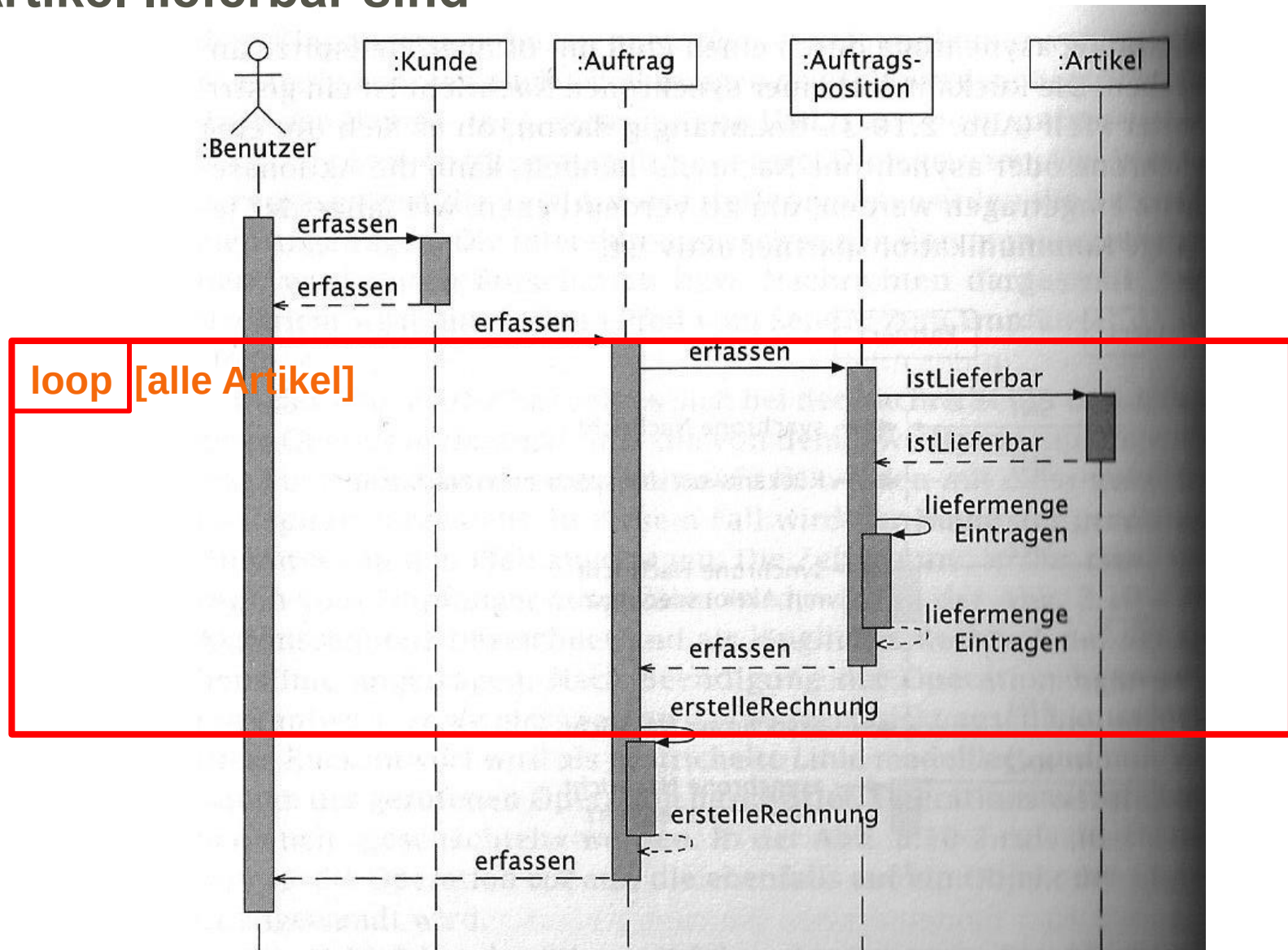
Use Case „Auftrag erfassen“

1. Kundendaten abrufen
 2. Lieferbarkeit prüfen
 3. Rechnung erstellen
 4. Auftrag vom Lager ausführen lassen
 5. Rechnungskopie an Buchhaltung senden
-
- 1.a Kunde ist nicht registriert
 - 2.a Artikel ist nicht lieferbar

Sequenzdiagramme



Szenario „Auftrag für Neukunde, wenn Artikel lieferbar sind“



Sequenzdiagramme

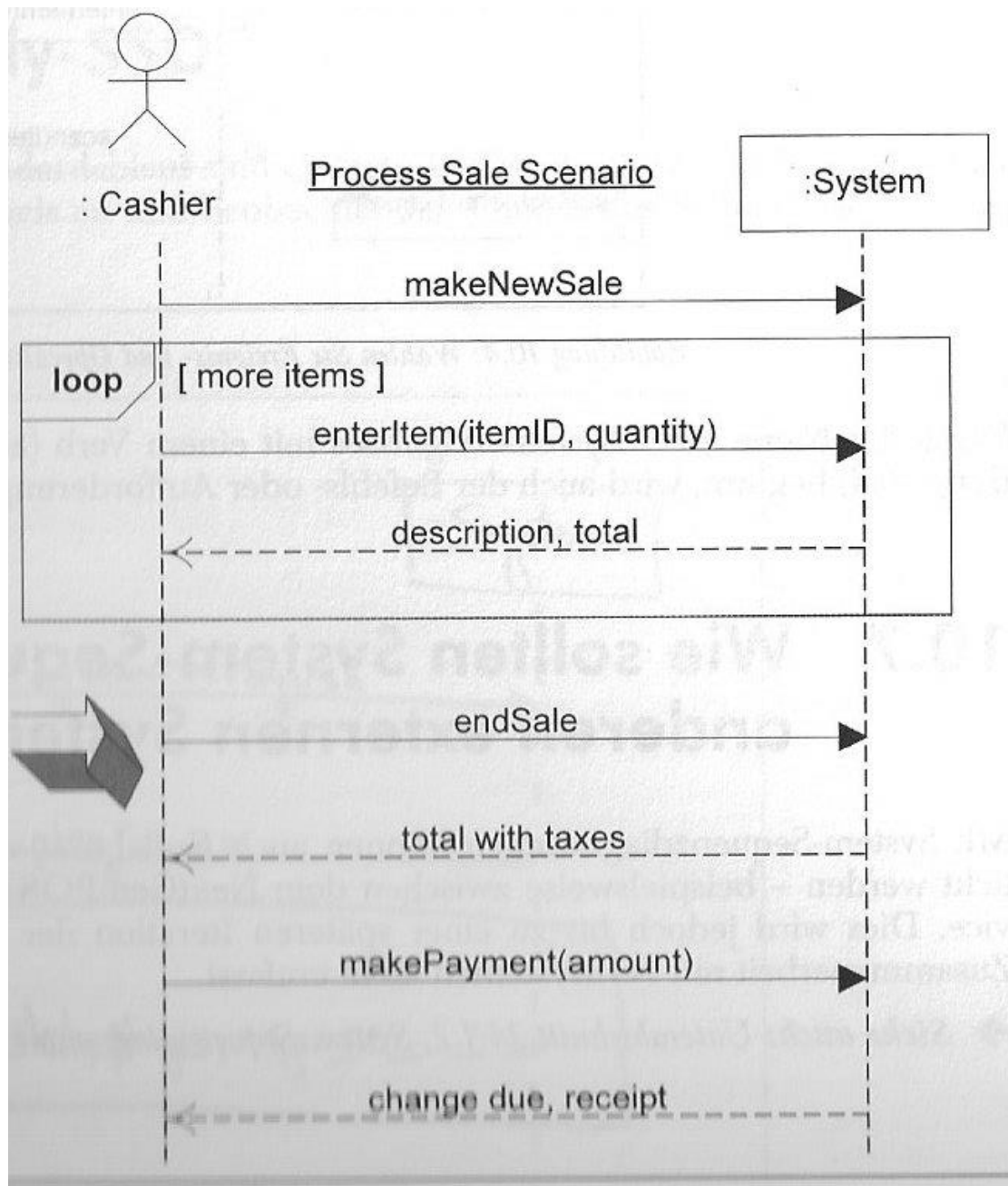
Szenarien im System-Use-Case

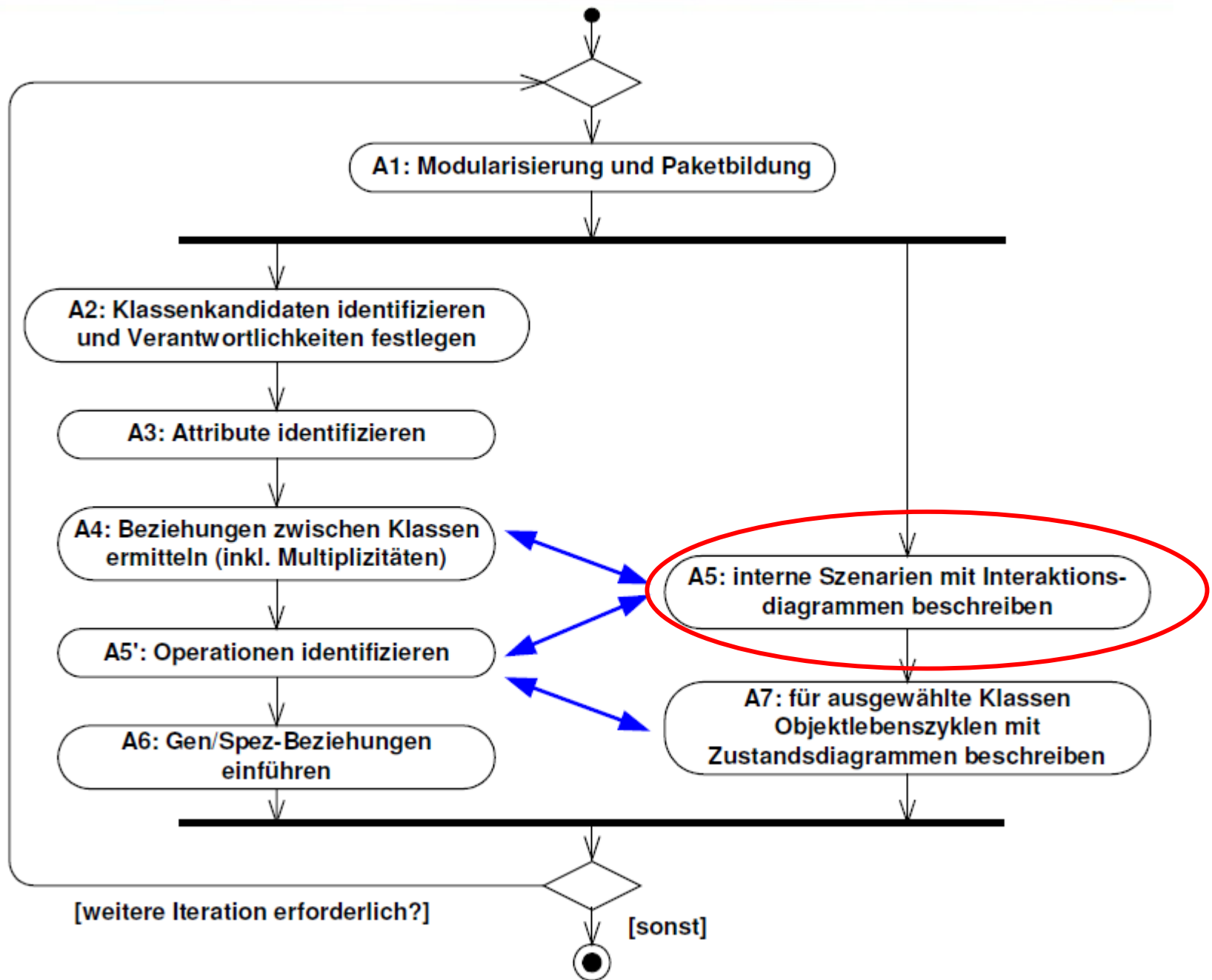
- ✎ System wird als Blackbox betrachtet
- ✎ Nur Ein- und Ausgabe an der Systemschnittstelle wird betrachtet

Sequenzdiagramme

Use Case „Verkauf“

1. Kunde kommt mit Waren an die Kasse
2. Kassierer beginnt neuen Verkauf
3. Kassierer gibt Artikelnummer an
4. System erfasst Position und zeigt Artikelbeschreibung, Preis und laufende Summe an
5. Wiederholung Schritte 3-4
6. System zeigt Summe an.
7. Kunde bezahlt
8. System verarbeitet Bezahlung





OOA-Vorgehensmodell

Aktivität 5‘

Für jede Nachricht, die ein Objekt im SD versendet, muss die Klasse des Empfängerobjektes im Klassenmodell eine entsprechende Operation zur Verfügung stellen.

 **Klassenmodell um entsprechende Operationen ergänzen.**

OOA-Vorgehensmodell

Aktivität 6 – Gen/Spez-Beziehungen

Bottom-Up-Klassifikation (Generalisierung)

Fragestellung: Gibt es im aktuellen Strukturmodell gleichartige bzw. ähnliche Klassen, aus denen sich durch Ausklammern gemeinsamer Eigenschaften / gemeinsamen Verhaltens eine gemeinsame Oberklasse bilden lässt?

→ Es wird eine Oberklasse gebildet.

OOA-Vorgehensmodell

Top-Down-Klassifikation (Spezialisierung)

Fragestellung: Gibt es im aktuellen Strukturmodell Klassen, deren Objekte sich in einigen ihrer Eigenschaften bzw. in einigen Teilen ihres Verhaltens unterscheiden?

→ Es werden Unterklassen gebildet.

Klassen- und Sequenzdiagramme

Anwendung in Ihrer Projektaufgabe

- ✎ Erstellen Sie ein Sequenzdiagramm für den Prozess der Schichtplanung!
- ✎ Ergänzen Sie die identifizierten Operationen in Ihrem Klassendiagramm!
- ✎ Ergänzen Sie ggf. das Klassendiagramm um weitere fachliche Klassen!