

Softwaretechnik 1 (ST1) im SoSe 2022 Objektorientierte Modellierung und Entwicklung

Kapitel 4a: UML – Verhaltensmodellierung: Interaktionsdiagramme

Lernziele: Nach dieser Vorlesung sollten Sie ...

- Wissen, was man unter Verhaltensmodellierung versteht, und die Verhaltensmodellierung von der Strukturmodellierung und der Funktionsmodellierung unterscheiden und abgrenzen können
- Ablauforientiertes Verhalten als Interaktionen zwischen Objekten mehrerer Klassen mit Sequenzdiagrammen und Kommunikationsdiagrammen modellieren können
- Kombinierte Fragmente und die Operatoren `opt`, `alt` und `loop` in Interaktionsdiagrammen kennen und anwenden können
- Verstehen, in wie weit Sequenzdiagramme und Kommunikationsdiagramme äquivalent sind und sie ineinander überführen können

Inhaltsüberblick

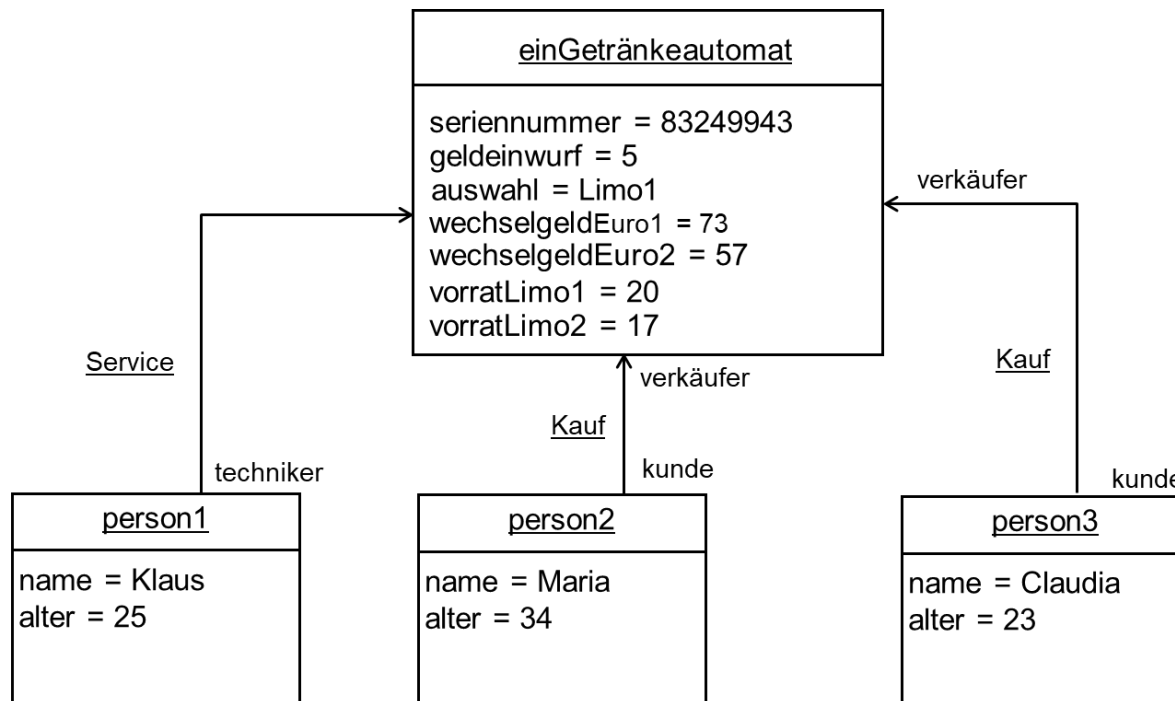
- Verhaltensmodellierung
- Sequenzdiagramm
- Kommunikationsdiagramm

Welche Modellierungssichten kennen wir?

- Strukturmodellierung
 - Fokussiert auf die Struktur des zu modellierenden Sachverhalts
 - Sachverhalte/Informationen/Daten, die sich in der Anwendung widerspiegeln müssen
 - OO: Objekte und Verbindungen bzw. Klassen und Assoziationen
 - UML: Objektdiagramm und Klassendiagramm
- Funktionsmodellierung
 - Fokussiert auf die zu unterstützenden Aufgaben des zu modellierenden Sachverhalts
 - Ziele (Goals) und Geschäftsprozesse, die sich in der Funktionalität der Anwendung widerspiegeln müssen
 - OO: Akteure und Anwendungsfälle
 - UML: Anwendungsfalldiagramm
- Was fehlt?
 - Wie werden die Funktionen von den Objekten erbracht?
 - Wie interagieren Objekte?
 - Wie und unter welchen Umständen ändern Objekte ihren Zustand?
 - Welche Daten/Informationen werden von welchen Funktionen wie verarbeitet?
 - Kurz: Wie verhalten sich das System und seine Komponenten?
 - Noch kürzer: **Verhaltensmodellierung**

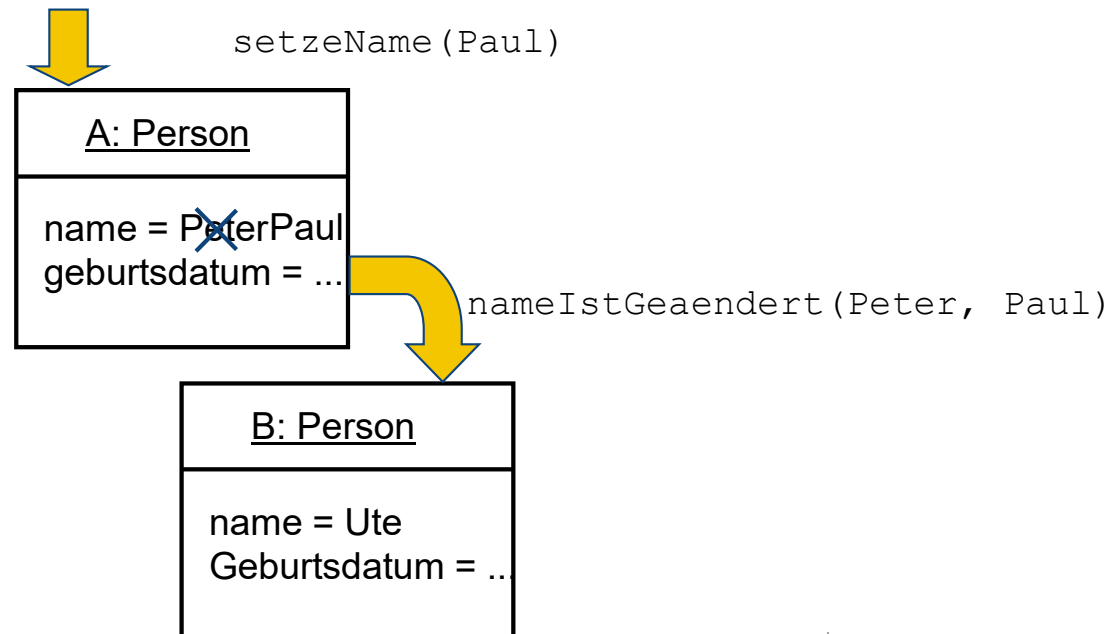
Objekte im Laufe der Zeit ...

- Zur Erinnerung: Objektdiagramm ist Schnappschuss zu einem bestimmten Zeitpunkt
 - Feste Anzahl dargestellter Objekte
 - Zustand der Objekte (Attributwerte, Verbindungen) ist statisch
- Aber: Objekte senden Nachrichten und ändern ihren Zustand im Laufe der Zeit !



Zur Erinnerung: Objektverhalten

- Objekte reagieren auf Nachrichten (z.B. Operationsaufrufe), indem sie z.B.
 - ihren Zustand wechseln, d.H. Attributwerte ändern oder Verbindungen lösen oder neu eingehen
 - selbst Nachrichten an (andere) Objekte senden
- Eine Nachricht besteht aus Name und Parametern



Arten von Verhalten in der Objektorientierung

Interaktives Verhalten (Ablauforientiertes Verhalten)

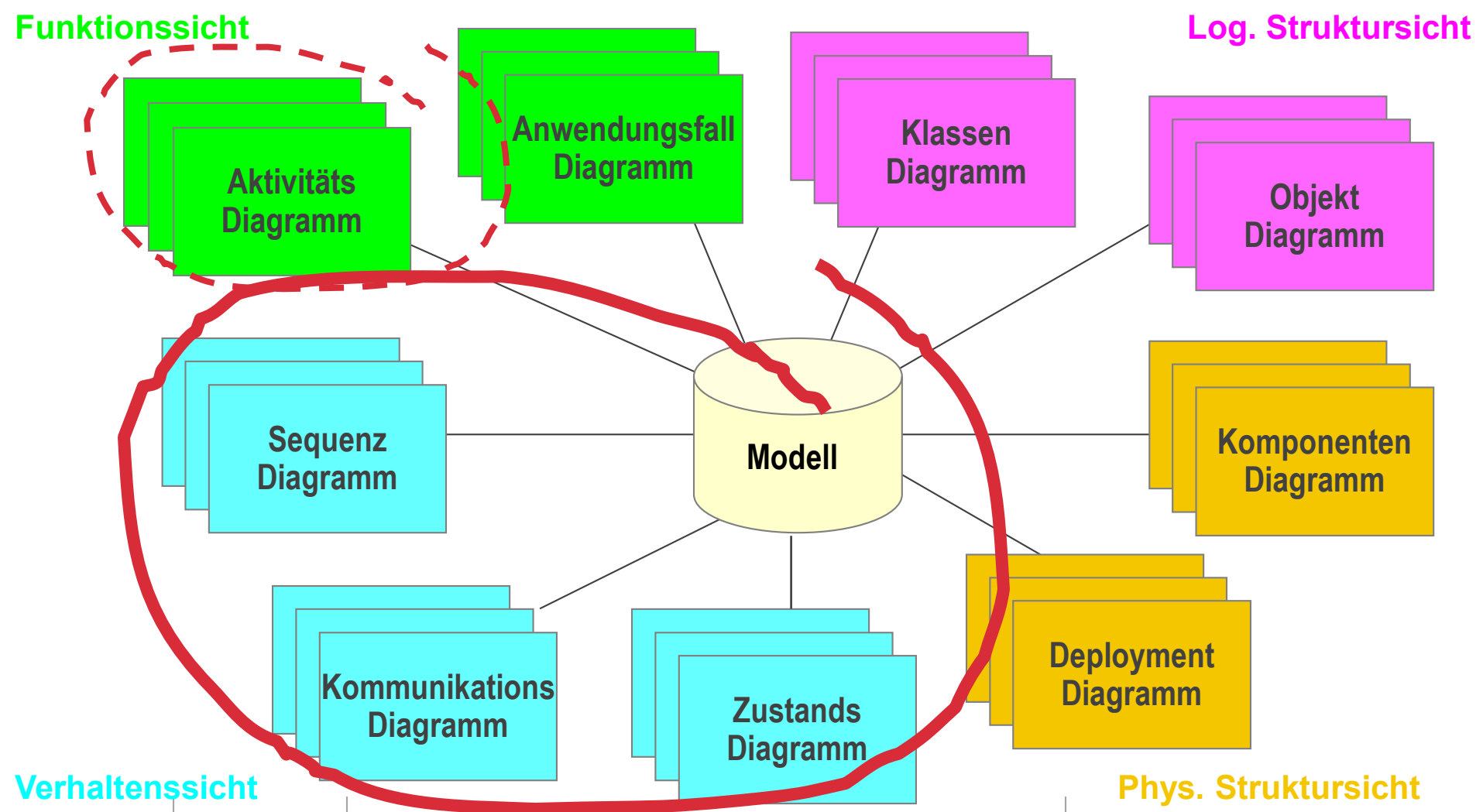
- Inter-Objektverhalten („System-Intern“ beobachtbares Verhalten)
 - Abläufe von Operationen
 - Objekte i.d.R. mehrerer Klassen arbeiten bei der Ausführung einer im Klassenmodell definierten Operation durch den Austausch von Nachrichten zusammen
 - Determiniert durch den Zustand der Objekte zu Beginn der Operationsausführung und die aktuellen Parameter des Operationsaufrufs
- Systemverhalten („System-Extern“ beobachtbares Verhalten)
 - Abläufe von Anwendungsfällen
 - Akteure interagieren in einem Szenario eines Anwendungsfalls mit dem Anwendungssystem
 - Determiniert durch den Zustand des Anwendungssystems zu Beginn des Anwendungsfalls und die aktuellen Interaktionsparameter

Vgl. auch
Aktivitätsmodellierung
(Kap. 2)

Reaktives Verhalten (Zustandsorientiertes Verhalten)

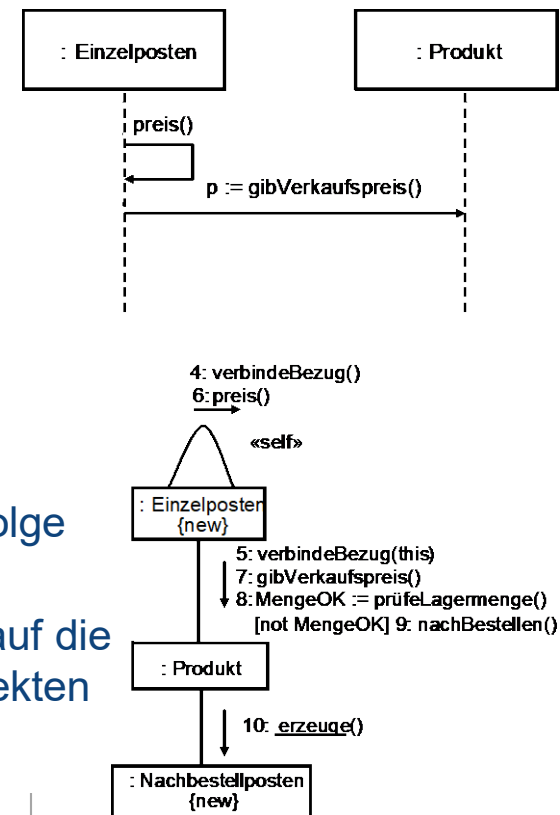
- Mögliche/erlaubte Reaktionen eines Systems (oder einer Komponente oder der Instanzen einer Klasse) über den gesamten „Lebenszyklus“ hinweg
- Ereignisse / Reaktionen / Zustandsänderungen

Verhaltensorientierte Diagramme in der UML



Interaktionsdiagramme: Interaktives Verhalten

- Interaktionsdiagramme stellen dar, wie Objekte bei bestimmten Abläufen (z.B. von Anwendungsfällen oder Operationen) durch den Austausch von Nachrichten interagieren
 - Aufrufe von Operationen (Aufruf-Nachrichten)
 - Setzen von „Variablen“ (Rückgabe-Nachrichten)
 - Erzeugen und Zerstören von Objekten
- Eine Interaktion ist die Folge (*trace*) der dabei auftretenden Vorkommen (*occurrences*) von Ereignissen (*events*)
 - Senden einer Nachricht (Sende-Ereignis)
 - Empfangen einer Nachricht (Empfangs-Ereignis)
 - Ändern eines Wertes (Änderungs-Ereignis)
 - Erzeugungs- und Zerstörungs-Ereignis
- Das Sequenzdiagramm fokussiert auf die zeitliche Reihenfolge der zwischen den Objekten gesendeten Nachrichten
- Das Kommunikationsdiagramm fokussiert zusätzlich auch auf die Verbindungen, über welche Nachrichten zwischen den Objekten gesendet werden

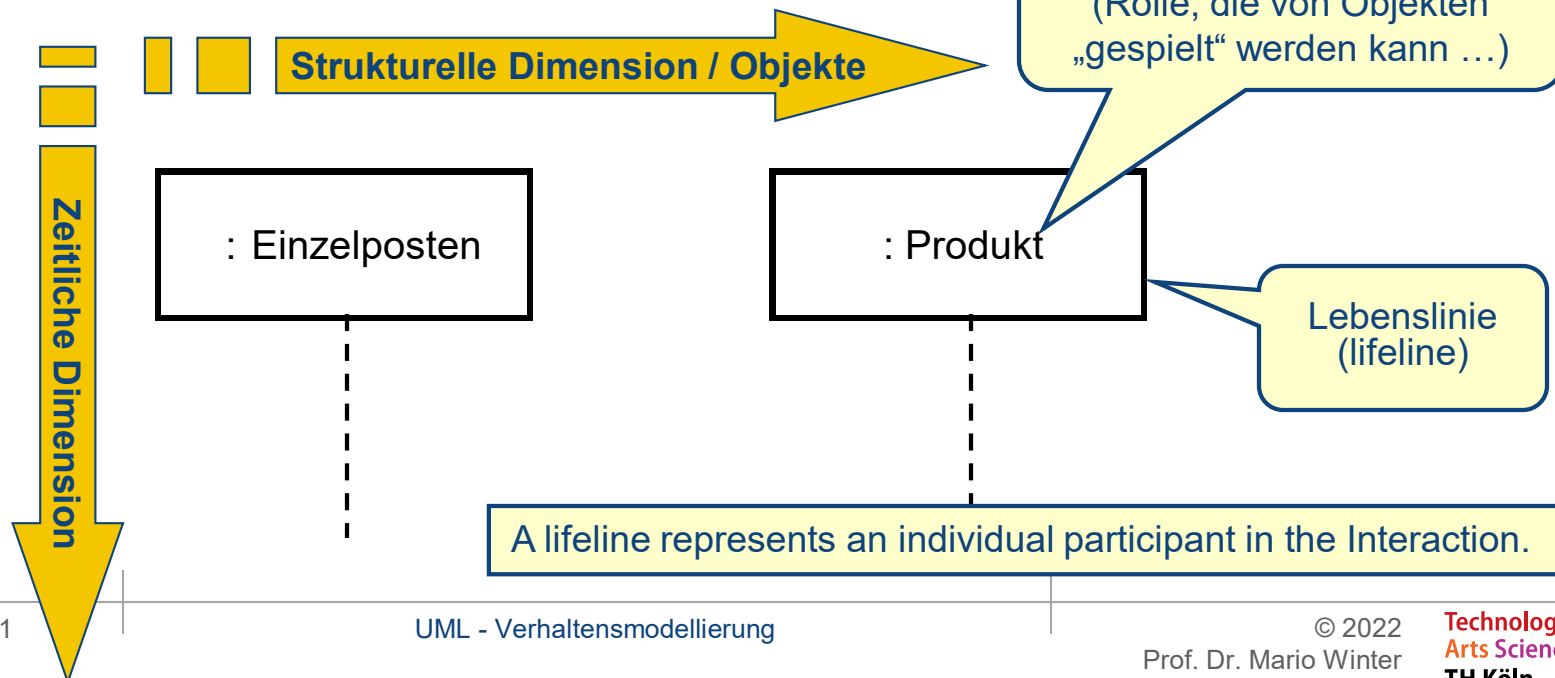


Wo sind wir?

- Verhaltensmodellierung
- **Sequenzdiagramm**
- Kommunikationsdiagramm

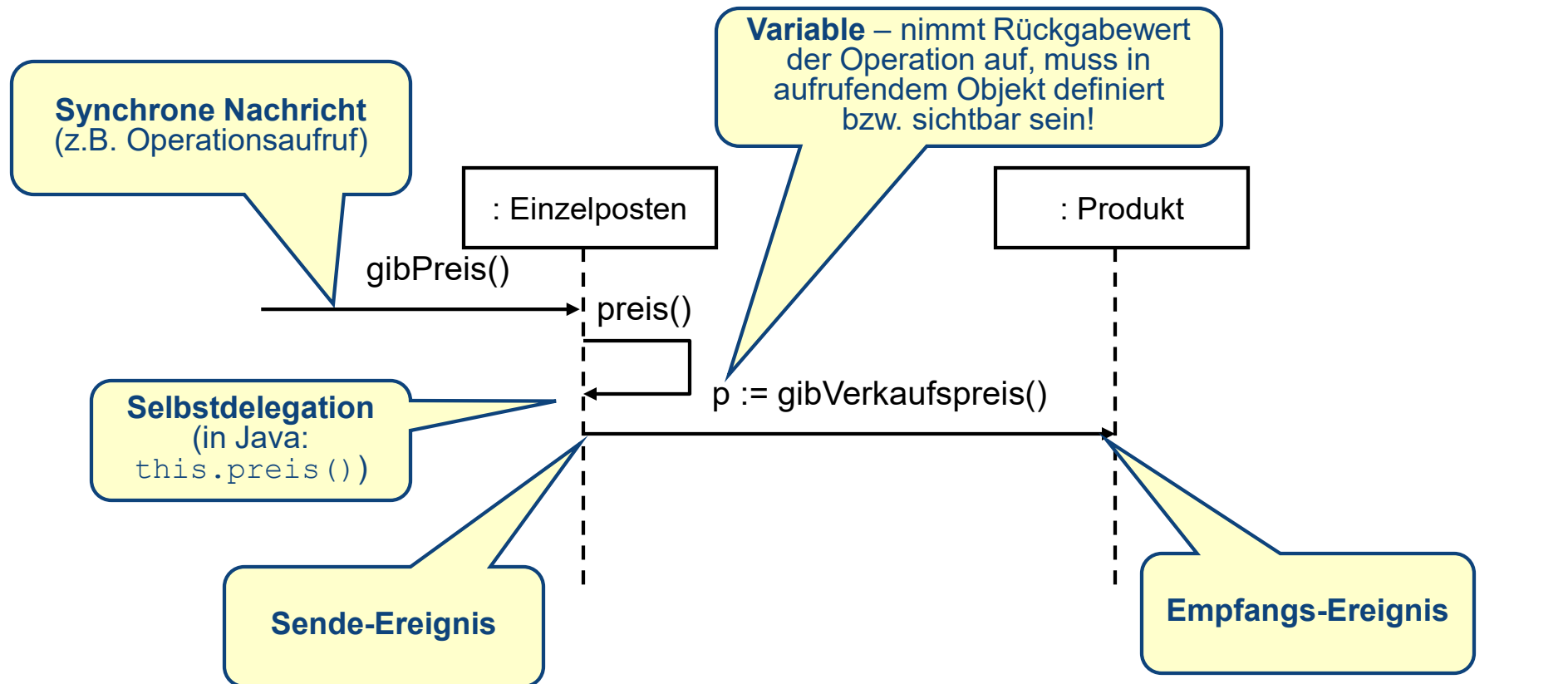
Raum und Zeit: Sequenzdiagramm

- Man braucht zur Darstellung einer Interaktion (mindestens) zwei Dimensionen
 - Die Struktur (welche Objekte in welchem Zustand interagieren)
 - Die Zeit (wann welche Nachricht gesendet/empfangen wird)
- Idee: Aufteilung der zwei Dimensionen des Zeichenblattes
 - Horizontal: Objekt-Achse
 - Vertikal: Zeitachse



Verhalten in der Raumzeit

- Verhaltensaktivierung durch Operationsaufrufe (Nachrichten)
- Synchroner Nachrichten: Pfeile mit ausgefüllter Spitze



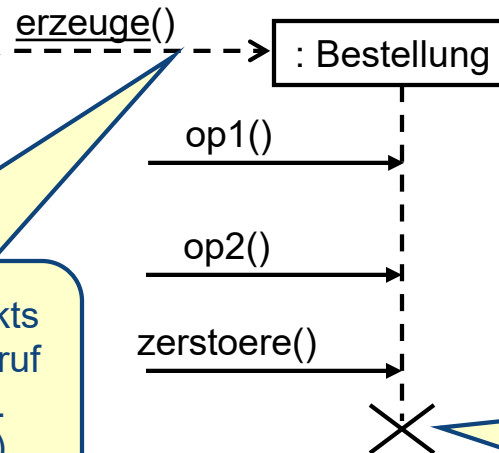
Der Objekt-Lebenszyklus

- Ein Objekt wird irgendwann erzeugt (instanziiert) ...
- ... ändert durch Aufrufe der (in seiner Klasse definierten) Operationen seinen Zustand ...
- ... und wird irgendwann zerstört

Sende-Rolle unspezifiziert

„gefundene Nachricht“
(found message)

Instanziierung des Objekts
(In Java, C++ und C# Aufruf
eines Konstruktors, z.B.
`new Bestellung()`)

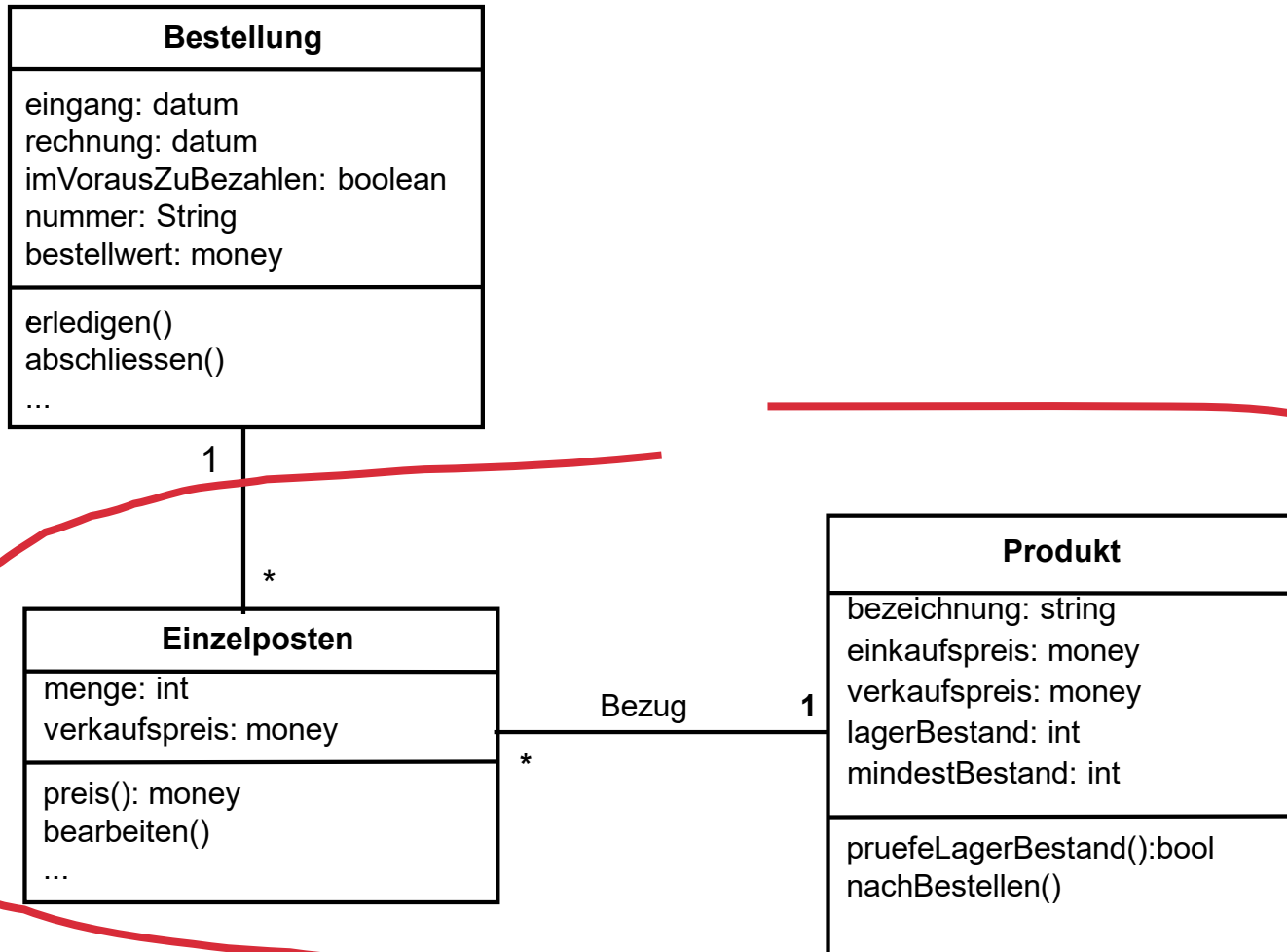


Zerstörung des Objekts
(In Java: `finalize()`;
In C++, C# Aufruf des Destruktors ~... mit
`delete`)

Kombinierte Fragmente und Interaktionsoperatoren

- Kombinierte Fragmente sind zusammenhängende Teile einer Interaktion
 - **Kombinierte Fragmente** laufen nach bestimmten Bedingungen bzw. Regeln ab
 - Werden im Sequenzdiagramm eingerahmt (ähnlich den UML-Diagrammrahmen)
- Interaktionsoperatoren steuern den Interaktionsfluss
 - **Interaktionsoperatoren** steuern, wann bzw. wie kombinierte Fragmente ablaufen sollen
 - In der oberen linken Ecke des betroffenen kombinierten Fragments stehen die Art des Operators und ggf. weitere Angaben
 - Hierarchische Schachtelung kombinierter Fragmenten / Interaktionsoperatoren erlaubt
- Bedingungen in Interaktionsoperatoren
 - Boolesche Ausdrücke
 - In eckigen Klammern angegeben, z. B. `[prüfen == true]`
 - Zur Formulierung der Bedingungen verwendet man meistens Attribute des dienstnutzenden Objekts oder "Variablen"
- Iterationsausdrücke in Interaktionsoperatoren
 - Kombinierte Fragmente werden manchmal wiederholt aufgerufen, z. B. wenn innerhalb einer Operationsausführung alle Objekte angesprochen werden sollen, die mit dem die Iteration ausführenden Objekt bez. einer mehrwertigen Assoziation verbunden sind
 - Dafür Iterationsausdruck angegeben, z. B. konkrete Werte `(1,10)`, Ausdrücke `(a, a+b)` oder allgemeine Bedingungen z.B. `[für alle verbundenen Instanzen]`
 - Iterationsausdruck ist optional - fehlt er, ist die Anzahl der Iterationen unbestimmt

Beispiel: Bestellwesen

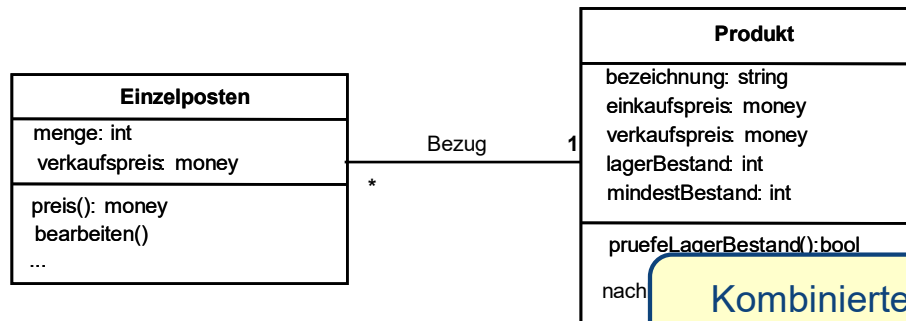


Sequenzdiagramm mit optionalem kombinierten Fragment

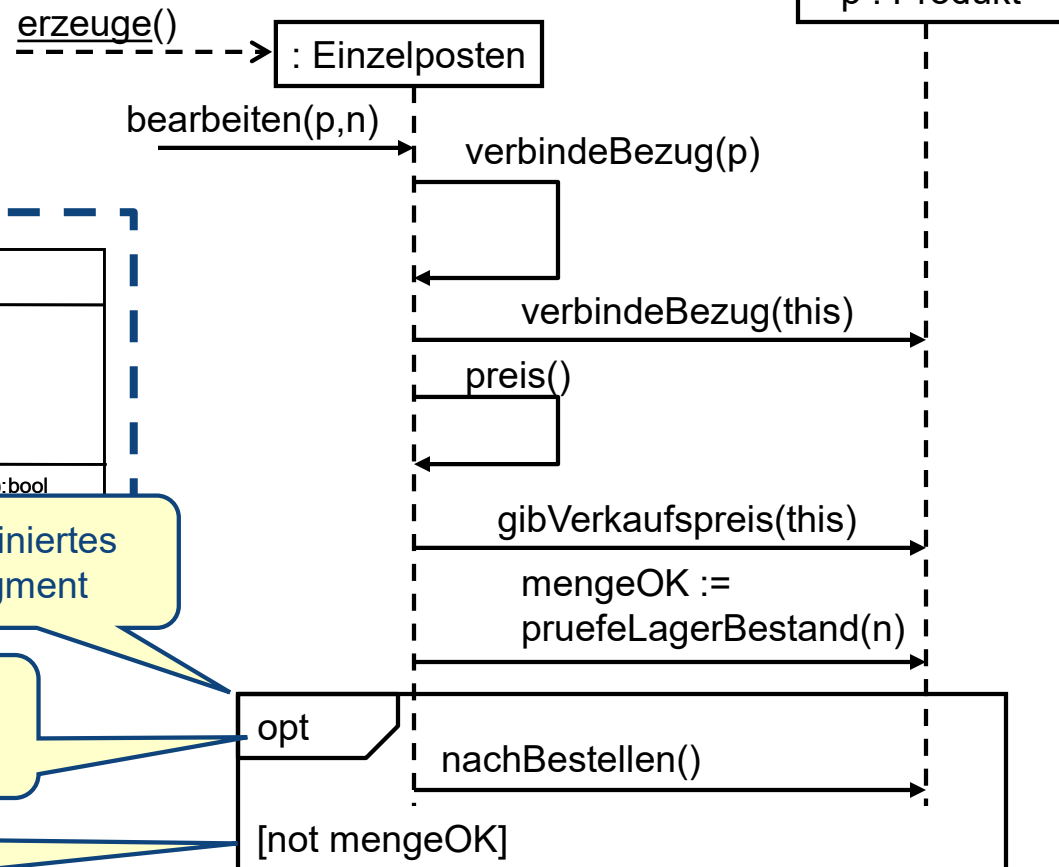


Folie 28

Klassendiagramm



Sequenzdiagramm



Kombiniertes
Fragment

opt-Operator: Optionales
kombiniertes Fragment
(Java: `if`)

Bedingung

opt

[not mengeOK]

nachBestellen()

mengeOK :=
pruefeLagerBestand(n)

gibVerkaufspreis(this)

preis()

verbindeBezug(this)

verbindeBezug(p)

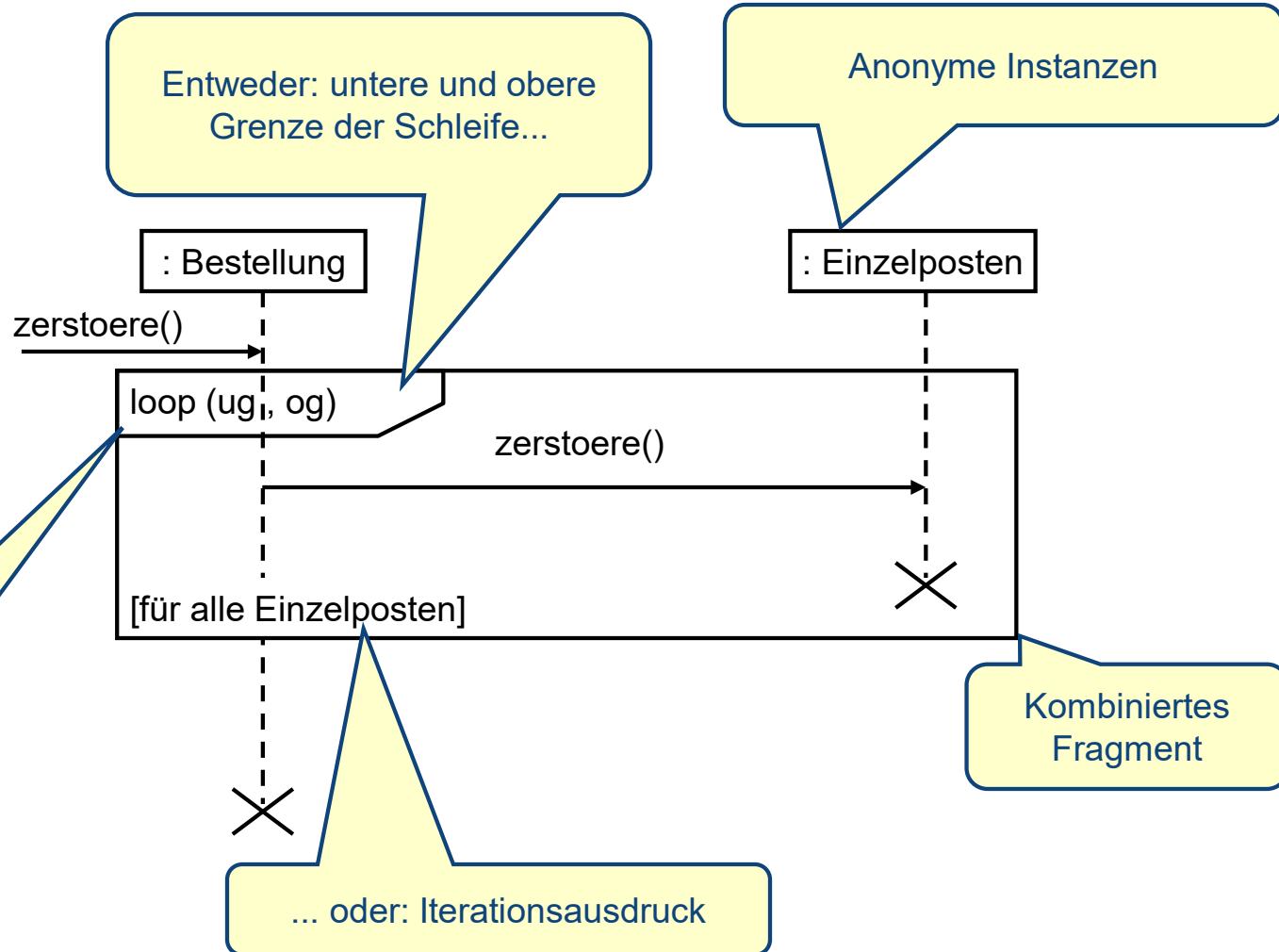
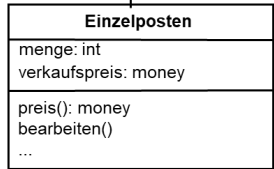
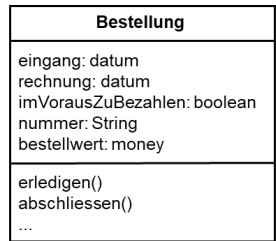
bearbeiten(p,n)

erzeuge()

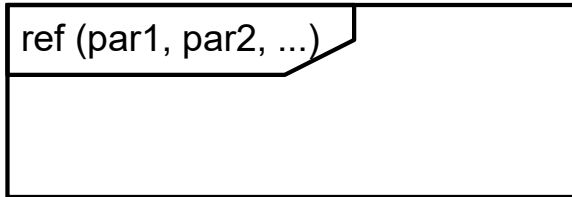
p : Produkt

: Einzelposten

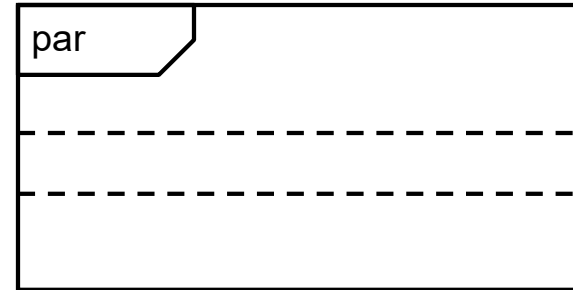
Sequenzdiagramm mit Iteration (Schleife)



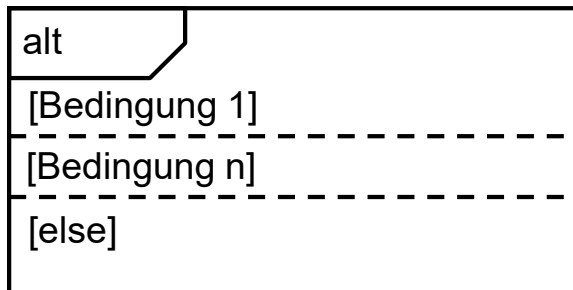
Weitere Interaktionsoperatoren (kombinierte Fragmente)



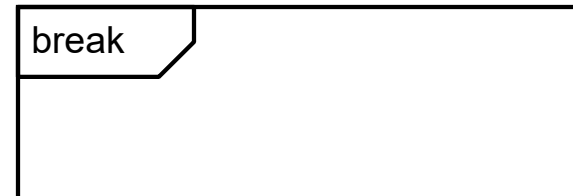
Interaktionsreferenz, „Aufruf“ eines anderen Interaktionsdiagramms



Parallele Abläufe, Nebenläufigkeit, Threads



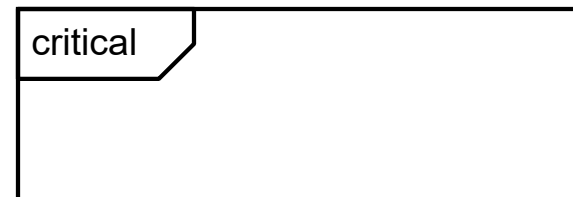
Alternative, if-then-else



Abbruchfragment



Negativ; Ungültige Interaktion



Atomare, nicht-unterbrechbare Interaktion

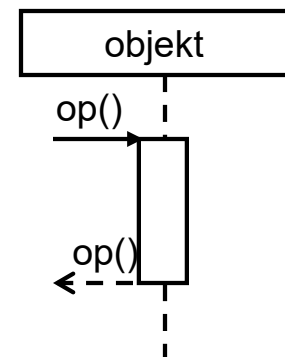
Interaktionsoperatoren: Zusammenfassung

Deutsch	Englisch	Kürzel	Bedeutung
Optionales Fragment	option	opt	Optionale Interaktionsteile (if ... then ...)
Alternative Fragmente	alternative	alt	Alternative Interaktionsteile (if ... then ... else if ...)
Schleife	loop	loop	Iterative Interaktionsteile
Interaktionsreferenz	reference	ref	“Einsetzen” einer anderen Interaktion
Abbruchfragment	break	break	Ausnahmefälle
Negation	negative	neg	Ungültige Interaktionsteile
Parallele Fragmente	parallel	par	Nebenläufige Interaktionsteile
Lose Ordnung	weak sequencing	seq	Von Lebenslinie und Operanden abhängige zeitliche Reihenfolge
Strenge Ordnung	strict sequencing	strict	Von Lebenslinie und Operanden unabhängige zeitliche Reihenfolge
Kritischer Bereich	critical region	critical	Atomare Interaktionen
Relevante Nachrichten	consider	Consider	Zu behandelnde Nachrichten
Irrelevante Nachrichten	ignore	ignore	Nicht zu behandelnde Nachrichten
Zusicherung	assertion	assert	Unabdingbare Interaktion

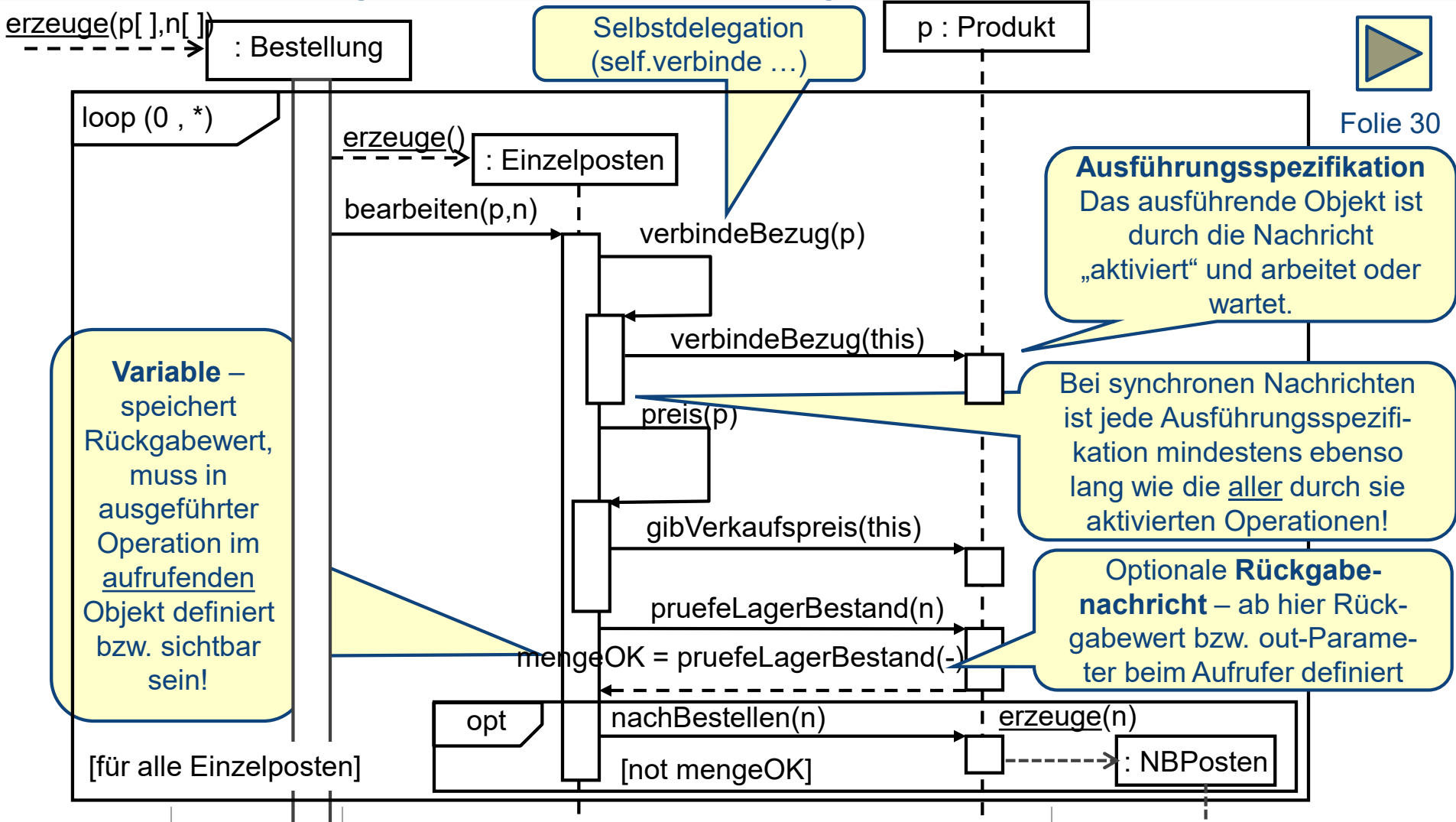
Nach: Rupp et al.: UML 2 Glasklar, 3. Aufl., Hanser Verlag, München, 2007

Synchrone Nachrichten und Ausführungsspezifikationen

- Aufruf einer Operation entspricht Senden einer Nachricht (message)
 - Jede Nachricht gibt dabei das dienstleistende und das dienstnutzende Objekt sowie den Namen und die Parameter der Operation an
 - Das Ereignis “Empfang der (Aufruf-)Nachricht“ bewirkt beim dienstleistenden Objekt die Ausführung der entsprechenden Operation
- Im Fall einer synchronen Nachricht (synchronous message) wartet der Aufrufer untätig, bis das Ergebnis der Operation zurückgeliefert wird, und fährt erst dann mit seiner Beschäftigung fort
- **Frage:** Führt ein Objekt zu einem bestimmten Zeitpunkt gerade eine Operation bzw. eine Selbstdelegation aus oder wartet auf die Rückmeldung eines Operationsaufrufs, der an ein anderes Objekt gerichtet ist?
- Zeitintervalle, in denen dies gilt, werden im Sequenzdiagramm durch Ausführungsspezifikationen (*execution specification*) als schmale Rechtecke auf der Lebenslinie dargestellt (UML1.x: Aktivierungsbalken)
- Rückgabe-Nachrichten als Antwort bzw. Rückkehr des „Kontrollflusses“ zum „Aufrufer“ (Rücksprung) können durch gestrichelte Pfeile dargestellt werden, auf denen der Name der (Aufruf-)Nachricht ohne Parameter wiederholt wird



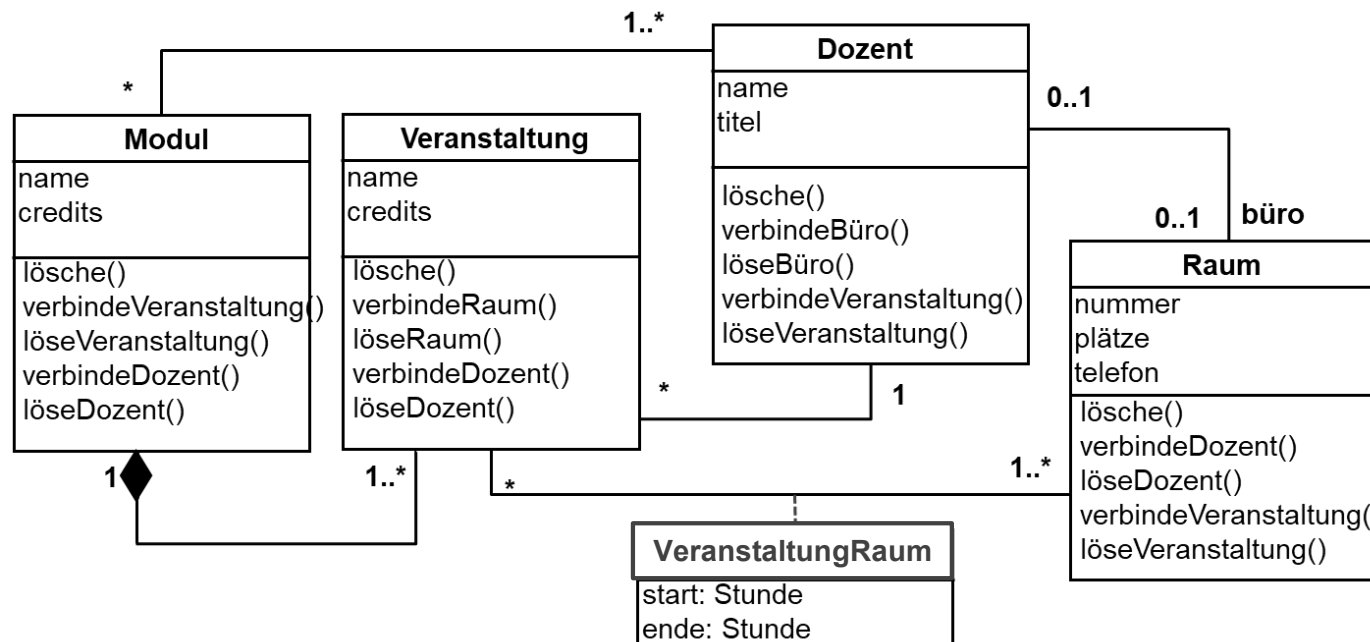
Sequenzdiagramm mit Ausführungsspezifikationen





Aufgabe 1: Interaktionsmodellierung - Sequenzdiagramm

- Modellieren Sie für das Löschen eines Moduls eine Interaktion zwischen Instanzen der dargestellten Klassen
- Verwenden Sie ein Sequenzdiagramm mit Ausführungsspezifikationen

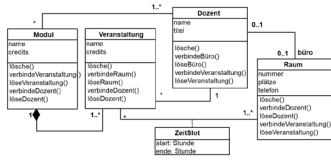
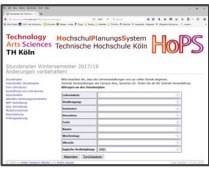


10 Minuten

Ende

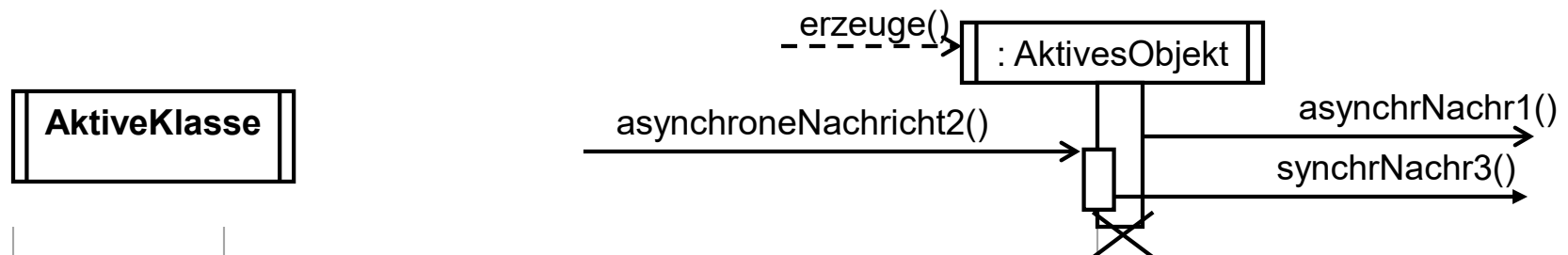


Lösungsidee Aufgabe 1: Sequenzdiagramm

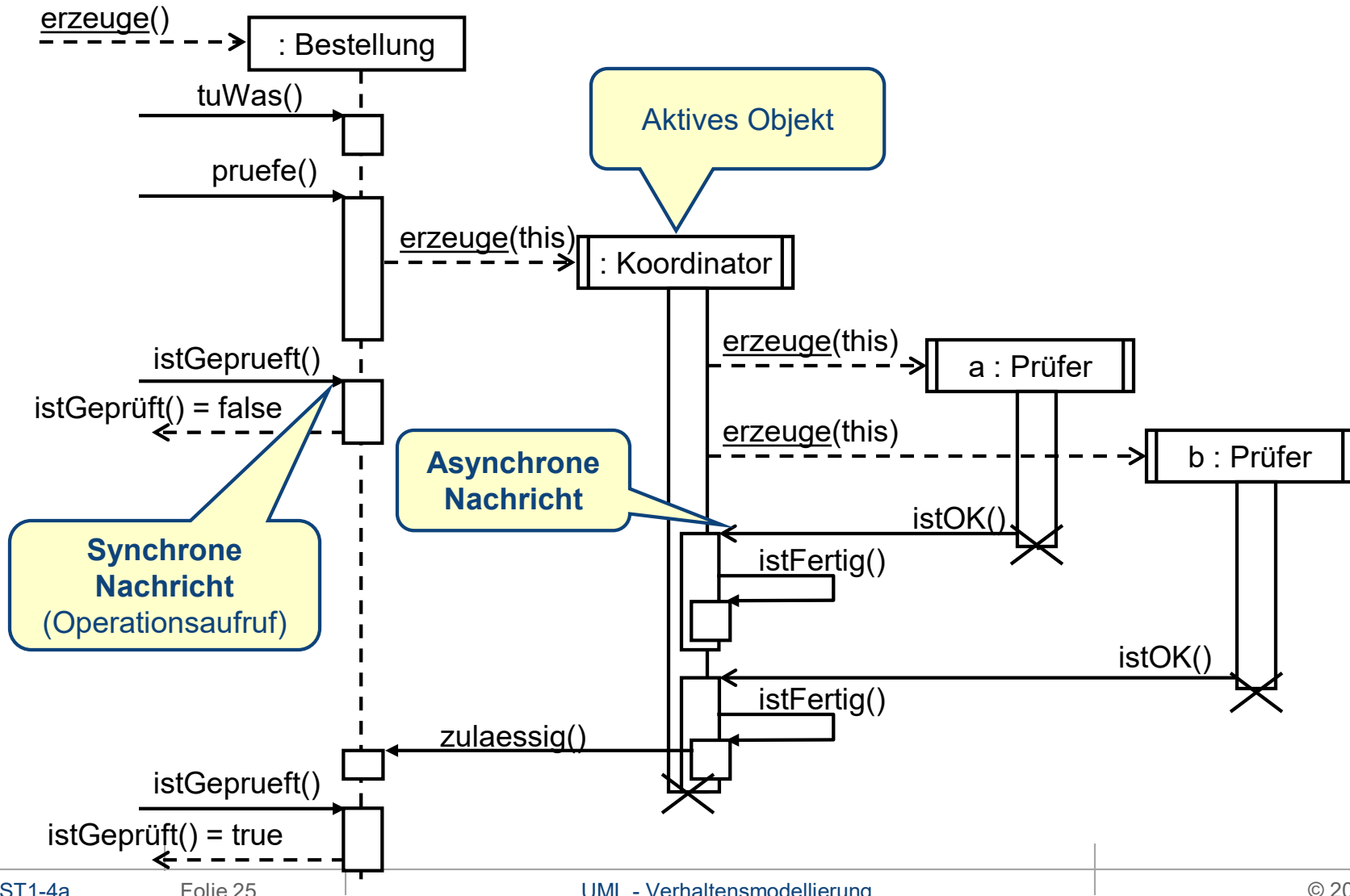


Asynchrone Nachrichten und aktive Objekte

- Bisher: Zu jedem Zeitpunkt hat höchstens ein Objekt die Ablaufkontrolle (d.h. führt eine Operation aus) → single thread
- Nun: Mehrere Objekte gleichzeitig aktiviert → multiple thread, Nebenläufigkeit
- Nachricht entspricht Brief, nach dessen Absenden der Schreiber fortfährt und erst dann warten muss, wenn er die Antwort auf seinen Brief benötigt
- Explizit modellieren, dass der “Sender” nach dem Senden einer Nachricht weiter aktiv ist, also beide Objekte die Ablaufkontrolle innehaben
- Asynchrone Nachrichten und aktive Objekte
- Aktive Objekte: Rechtecke mit doppelten Seitenlinien (ebenso die entsprechenden aktiven Klassen im Klassendiagramm)
 - Bis UML 1.5: Fett gezeichnete Rechtecke
- Asynchrone Nachrichten: Pfeile mit offener Spitze



Beispiel: Asynchrone Nachrichten und aktive Objekte

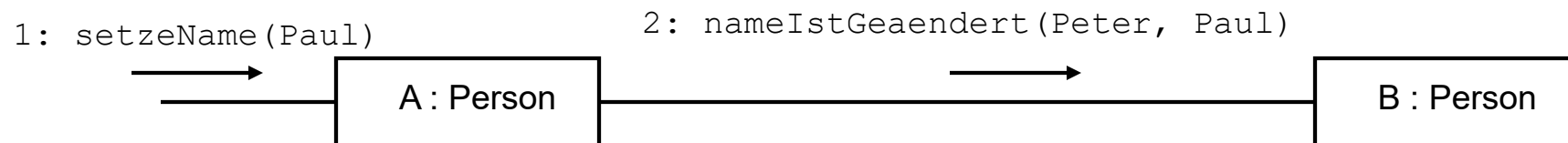


Wo sind wir?

- Verhaltensmodellierung
- Sequenzdiagramm
- **Kommunikationsdiagramm**

Raum (und Reihenfolge): Kommunikationsdiagramm

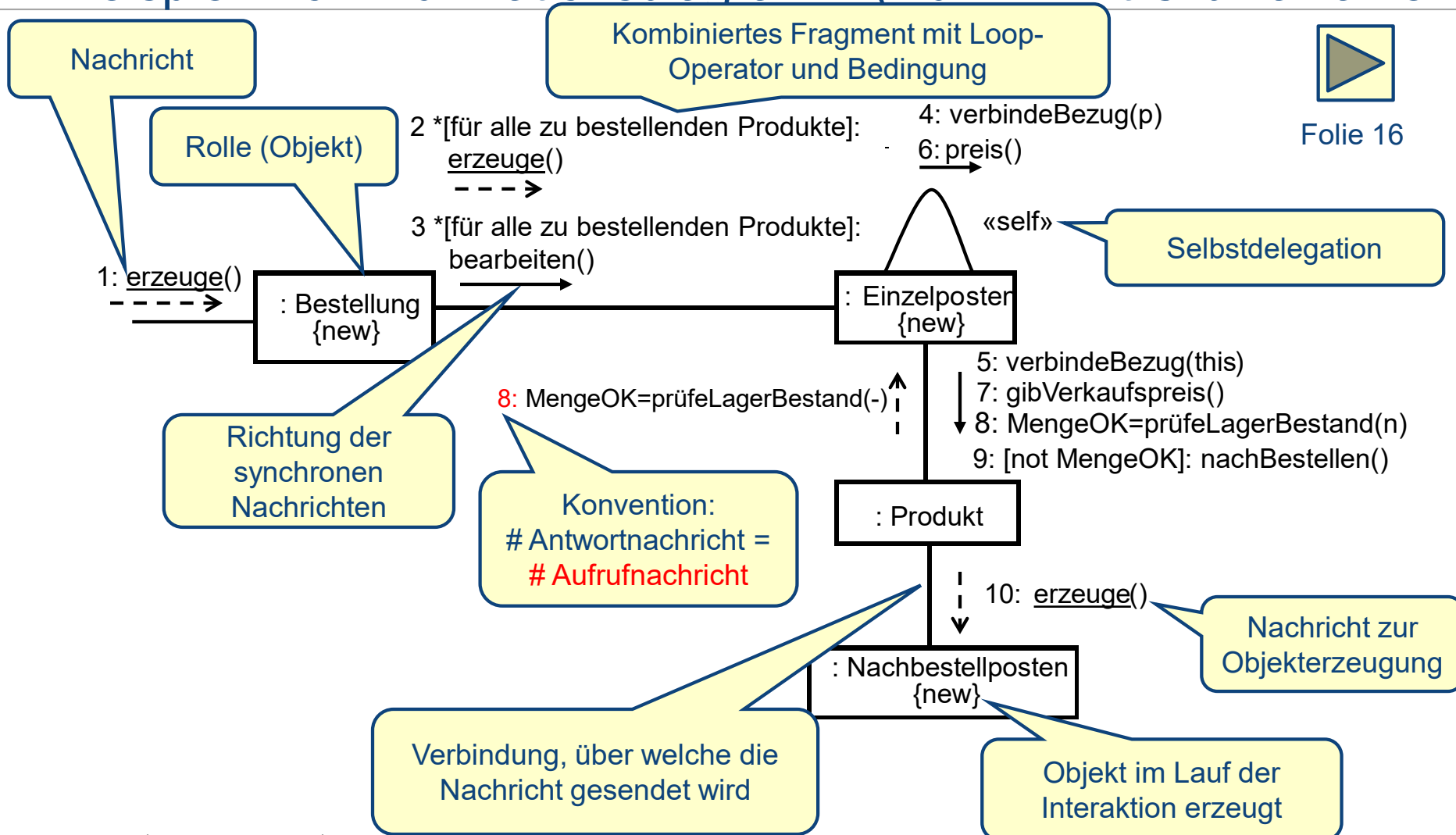
- Das Kommunikationsdiagramm kann als Objektdiagramm betrachtet werden, in dem zusätzlich ablaforientiertes Verhalten modelliert ist
- Nachrichtenaustausch an den Objektverbindungen notiert
- Nachricht-Richtung und Art (Synchron, Asynchron, Erzeugung, Rückgabewert) wie im Sequenzdiagramm durch Pfeile visualisiert
- Nachrichten-Bezeichner ggf. durch Kontrollinformation angereichert
- Da Objekte (wie im Objektdiagramm) beliebig angeordnet werden können und keine Zeitachse existiert, muss die Abfolge der Nachrichten durch eine geeignete Nummerierung verdeutlicht werden
- Objekte, die während des dargestellten Ablaufs erzeugt oder zerstört werden, erhalten die Kennung **{new}** bzw. **{destroyed}**
- Wird ein Objekt während des Ablaufs erzeugt und zerstört, erhält es die Kennung **{transient}**




Beispiel: Kommunikationsdiagramm (Numm. mit Ordinalzahlen)



Folie 16



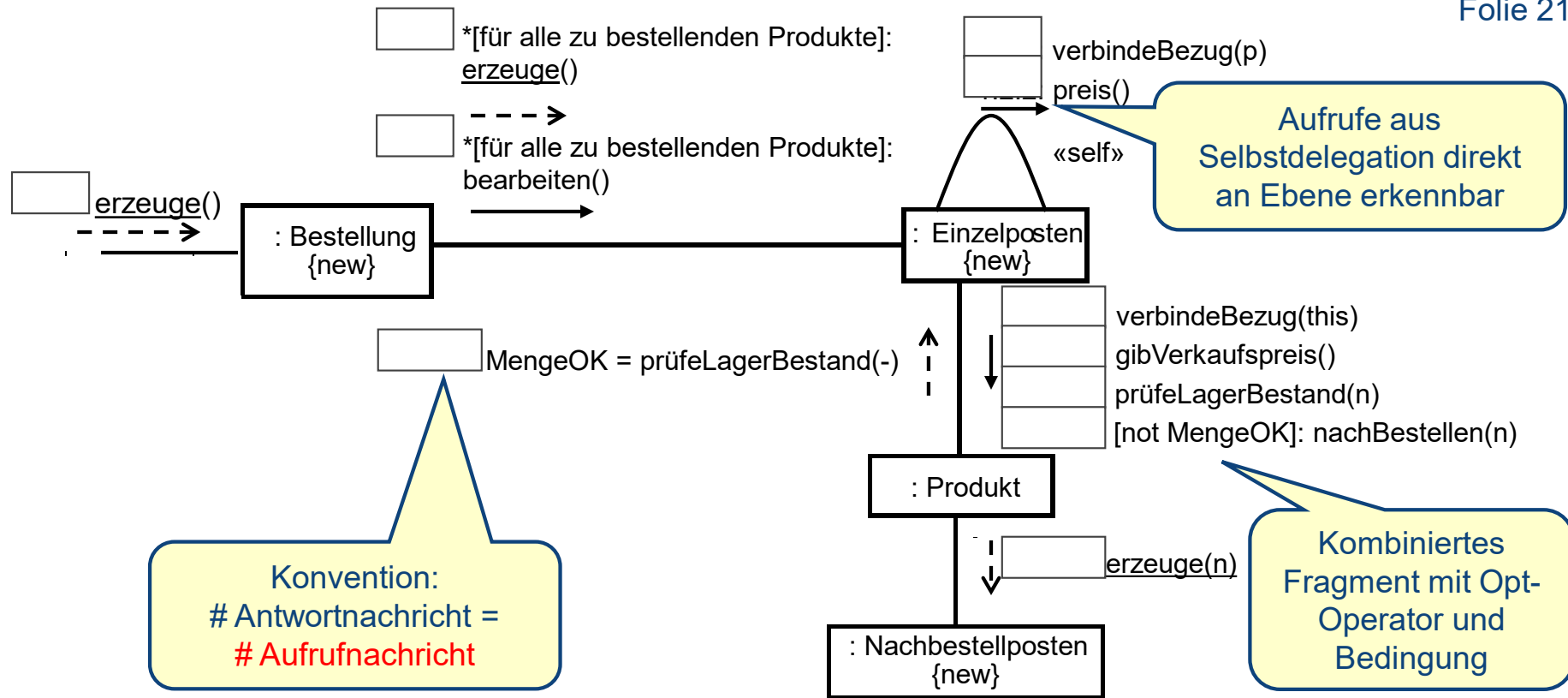
Nummerierung in Kommunikationsdiagrammen

- Nummerierung mit Ordinalzahlen
 - Nummerierung der Nachrichten von 1, ..., n
 - Beschreibt nur die Reihenfolge der Nachrichten
 - Nebenläufige Nachrichten (**par**-Operator) mit 1a, 1b, ... nummerieren
 - Problem: Ausführungsspezifikationen nicht modellierbar! Daher:
- **Hierarchische Dezimalnotation**
 - Nummerierung für jede Ausführungsspezifikation um einen Dezimalpunkt erweitern
 - Erste Ebene: 1, .., n
 - Zweite Ebene: 1.1, ..., 1.m
 - k-te Ebene: 1.1...1, 1.1...2, 1.1....p

k Stellen
 - Alle aus einer Ausführungsspezifikation (also während ein- und derselben Operationsausführung) gesendeten Nachrichten werden auf einer Hierarchie-Ebene durchnummeriert
 - Hierarchische Dezimalnotation ist ausdrucksstärker. Sie zeigt, welche Operation welche andere Operation aufruft, und ist daher vorzuziehen
- Antwortnachrichten
 - UML schweigt sich aus ... Vorschlag: Wie Aufrufnachricht nummerieren

Beispiel: Hierarchische Nummerierung



Folie 21



Max. 10 Minuten!



Aufgabe 2: Interaktionsmodellierung - Kommunikationsdiagramm

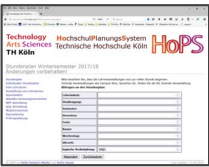
- Erweitern Sie Ihr Sequenzdiagramm für das Löschen eines Moduls um Ausführungssequenzen und wandeln Sie es in ein Kommunikationsdiagramm mit hierarchischer Dezimalnummerierung der Nachrichten um!

10 Minuten

Ende

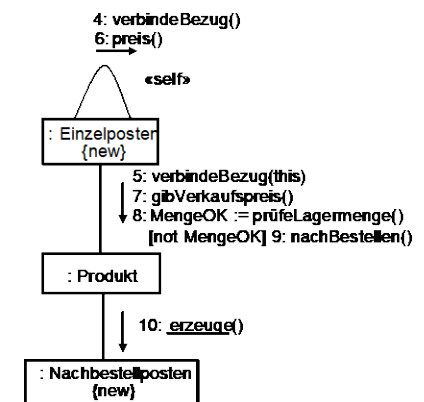
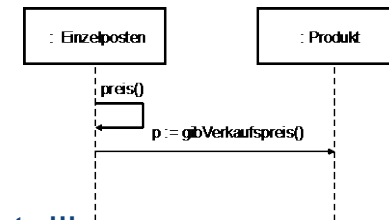


Lösungsansatz Aufgabe 2: hierarch. Dezimalnotation



Sequenzdiagramm vs. Kommunikationsdiagramm

- Sequenzdiagramm und Kommunikationsdiagramm sind bis auf wenige Eigenschaften äquivalent und können ineinander überführt werden
- Trotz der Durchnummerierung ist die Abfolge der Nachrichten im Kommunikationsdiagramm nicht so leicht zu erkennen wie im Sequenzdiagramm
- Kombinierte Fragmente im Kommunikationsdiagramm schlecht darstellbar
- Aber: Die Möglichkeit, Objekte im Kommunikationsdiagramm beliebig anzuordnen sowie Verbindungen zu zeichnen, bietet mehr Freiheiten
 - Im Kommunikationsdiagramm können Nachrichten bestimmten Verbindungen zugeordnet werden
 - Zur Verbesserung der Lesbarkeit können Objekte mit intensiven Verbindungen nahe beieinander platziert werden.
 - Andere Möglichkeit: Objekte so anordnen wie ihre zugehörigen Klassen im Klassendiagramm, um den Wiedererkennungswert auszunutzen



Diskussion: Interaktionsdiagramme

- Ablauforientiertes Verhalten von Operationen oder Anwendungsfällen
- Stärke: Einfachheit in Relation zur Ausdrucksmächtigkeit
- Sequenzdiagramm
 - Zeitliche Abläufe auf einen Blick deutlich, wenn auch zu Lasten struktureller Aussagekraft
 - Bei vielen Objekten zudem viele Kreuzungen von Lebenslinien und Pfeilen, so dass die Übersichtlichkeit leiden kann
- Kommunikationsdiagramm
 - Obiger Nachteil durch die freie Anordnung von Objekten verringert
 - Strukturkonforme Erweiterungen von Objektdiagrammen, kein neue Diagrammart
 - Hoher Wiedererkennungswert in Bezug auf Objektdiagramm und Klassendiagramm
 - Nachteil: zeitliche Abläufe und kombinierte Fragmente nicht unmittelbar erfassbar, sondern müssen mit gewissem Aufwand “heraus destilliert” werden
- Einfachheit und Klarheit von Interaktionsdiagrammen gehen rasch verloren, sobald sie viele Operatoren wie z.B. Schleifen oder Fallunterscheidungen enthalten
 - Ausweg: Verwendung eines gesonderten Diagramms für jeden konkreten Ablauf
 - Ergeben sich zu viele Diagramme bzw. Szenarien, kann man komplexe Abläufe durch Aktivitätsdiagramme beschreiben

Zusammenfassung

- Verhaltensmodellierung
 - Interaktives Verhalten = Ein Ablauf für viele Instanzen (i.d.R. mehrerer Klassen)
 - Reaktives Verhalten = Alle Abläufe für die Instanzen einer Klasse
- Interaktionsmodellierung
 - Zusammenspiel vieler Objekte (Objekt-Rollen)
 - Nachrichten und Verhaltens-Aktivierungen (Operationsausführungen)
- Sequenzdiagramm
 - Objektanordnung semantisch bedeutsam (Instanziierung!)
 - Zeitlicher Ablauf direkt erkennbar
 - Kombinierte Fragmente und Operatoren im Diagramm
 - Verschachtelte Abläufe durch Ausführungsspezifikation darstellbar
- Kommunikationsdiagramm
 - Objektanordnung semantisch nicht bedeutsam
 - Zeitlicher Ablauf nur in Nummerierung, schlecht erkennbar
 - Operatoren nur textuell darstellbar
 - Verschachtelte Abläufe nur mit hierarchischer Dezimalnotation darstellbar

