

- 01 Einführung
- 02 Prozessmodelle
- 03 Konfigurationsmanagement
- 04 Requirements Engineering
- 05 Modellierung
  - 05.1 Überblick
  - 05.2 Geschäftsprozessmodellierung
  - 05.3 Use Cases
  - 05.4 Klassen, Objekte, Assoziationen
  - 05.5 Szenarien, Zustandsautomaten
  - 05.6 Benutzungsoberflächen, Dialoge, GUI
- 06 Qualitätsmanagement

## ● Möglicher Ablauf

- Ermittlung der relevanten Geschäftsprozesse und Use Cases
  - Use Case-Diagramme, Aktivitätsdiagramme
- Erstellen des statischen Modells
  - Ableiten von Klassen aus den Use Cases
  - Klassen- und Objektdiagramme
- Erstellung des **dynamischen** Modells
  - **Szenarien** erstellen  
(jeden Use Case durch eine Menge von Szenarien präzisieren)
  - **Sequenz-** und **Kommunikationsdiagramme**, **Zustandsdiagramme**, **Aktivitätsdiagramme**
- Berücksichtigung der Wechselwirkung beider Modelle

## ● Definition

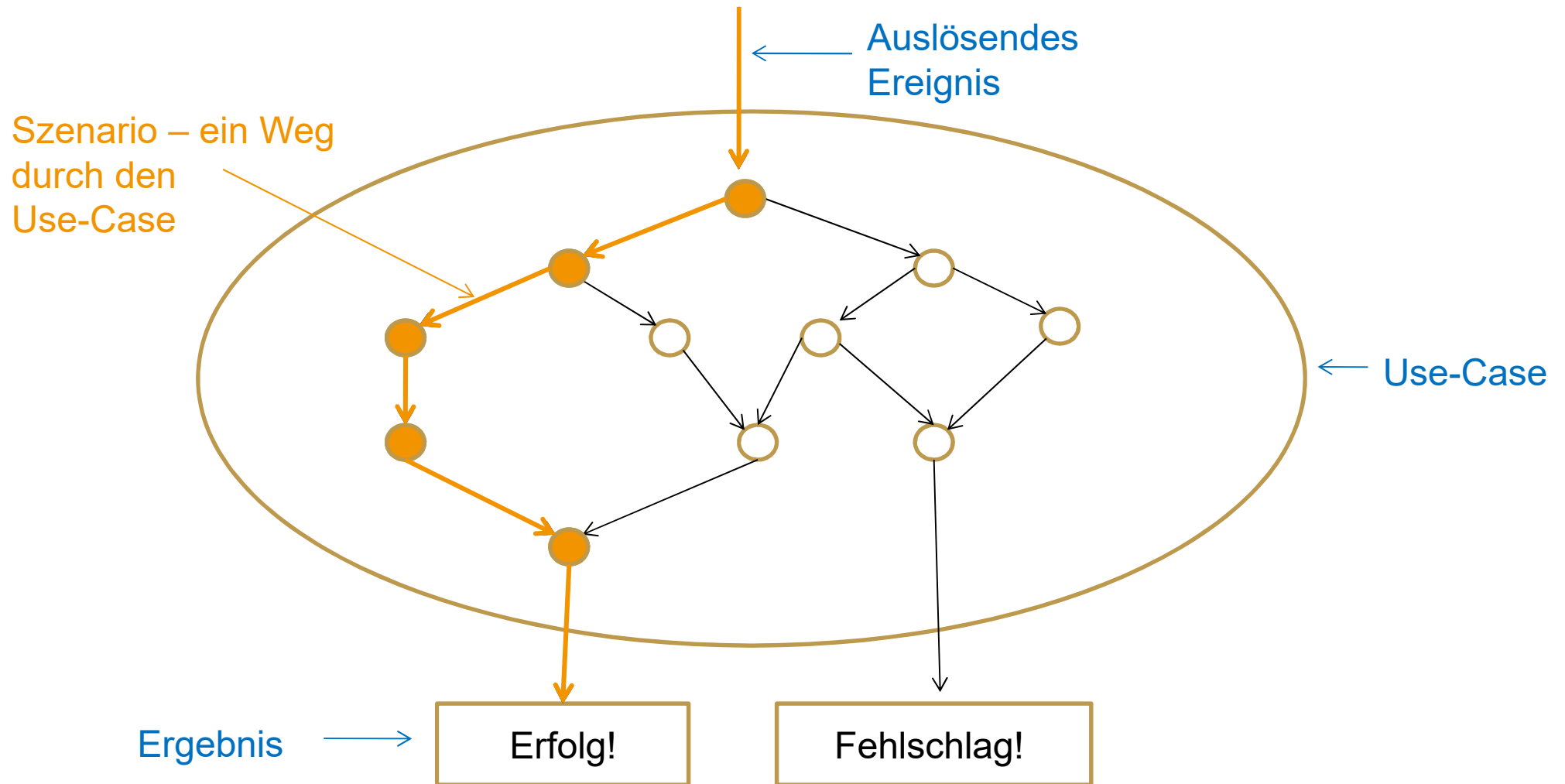
- Sequenz von Verarbeitungsschritten, die unter bestimmten Bedingungen auszuführen ist
- Schritte sollen das Hauptziel eines Akteurs realisieren und ein entsprechendes Ergebnis liefern
- Beginnt mit dem auslösenden Ereignis und wird solange fortgesetzt, bis das Ziel erreicht ist oder aufgegeben wird

## ● Einordnung

- Use Case dokumentiert durch eine Kollektion von Szenarien
- 2 Ausprägungen
  - Szenarien, die erfolgreiche Bearbeitung des Anwendungsfalls beschreiben
  - Szenarien, die zu einem Fehlschlag führen

# Szenario (2)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten



- Szenarien werden durch **Interaktionsdiagramme** modelliert
- UML bietet verschiedene Diagrammartent
  - **Sequenzdiagramm**
  - **Kommunikationsdiagramm**
  - Timing-Diagramm
  - Interaktionsübersichtsdiagramm

# Sequenzdiagramm (1)

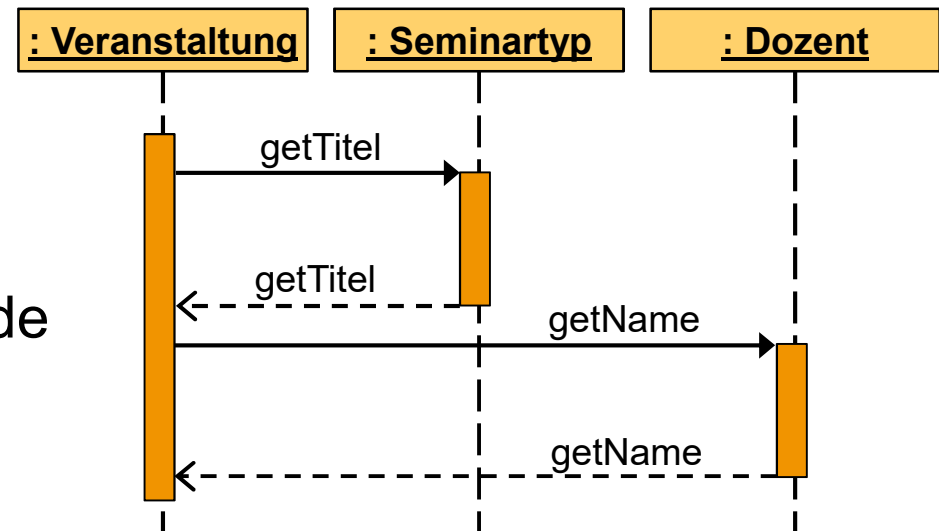
05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Eigenschaften

- Zeigt die **Interaktion** zwischen **mehreren Kommunikationspartnern**
- Oft miteinander kommunizierende Objekte (von Klassen)

## ● Notation

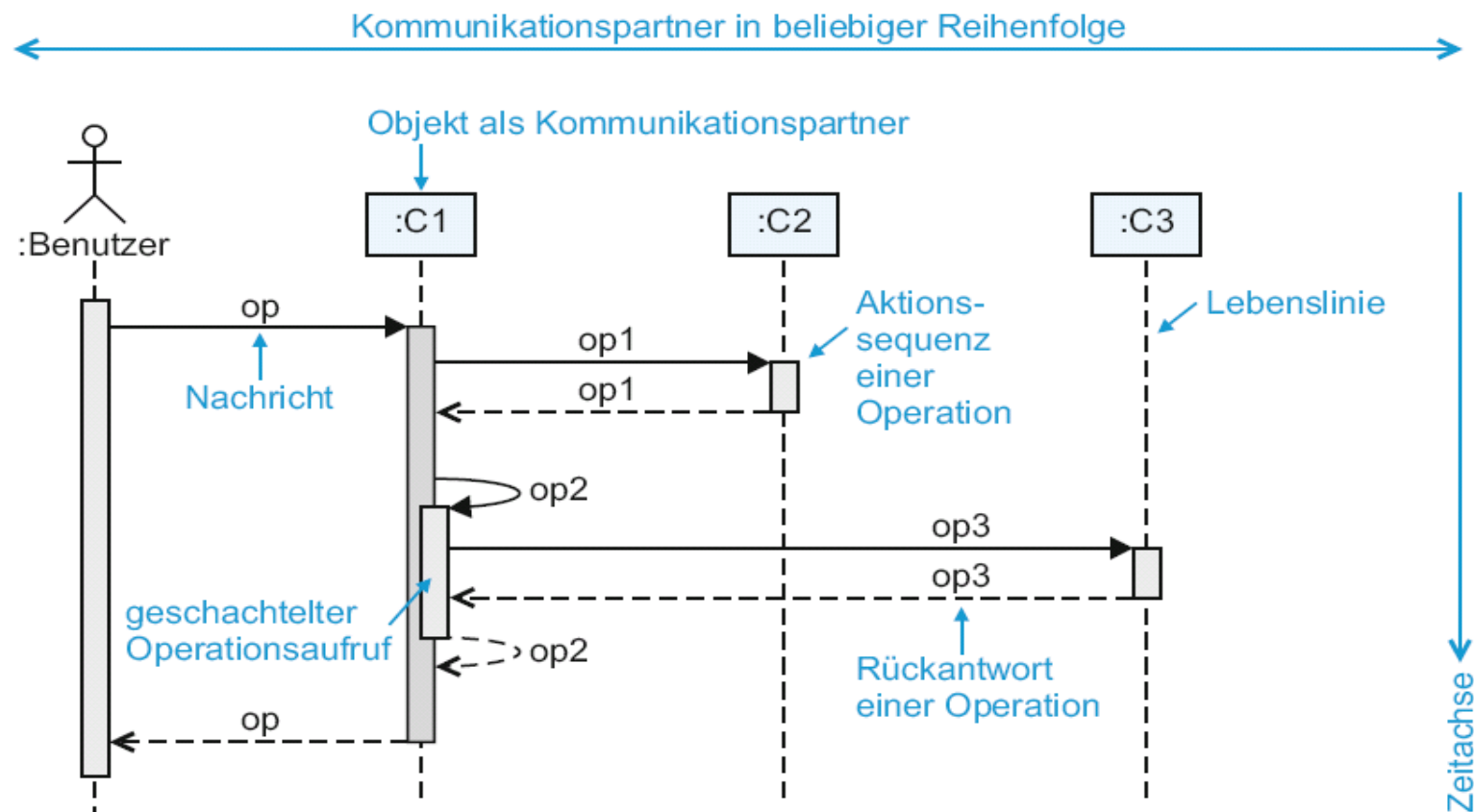
- Kommunikationspartner als Rechtecksymbol zusammen mit einer gestrichelten Linie (**Lebenslinie**)
- Zwei Dimensionen
  - **Vertikale**: repräsentiert **Zeit**
  - **Horizontale**: Interaktionen werden in Form von **Nachrichten** (messages) bzw. Botschaften eingetragen



# Sequenzdiagramm (2)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Notation

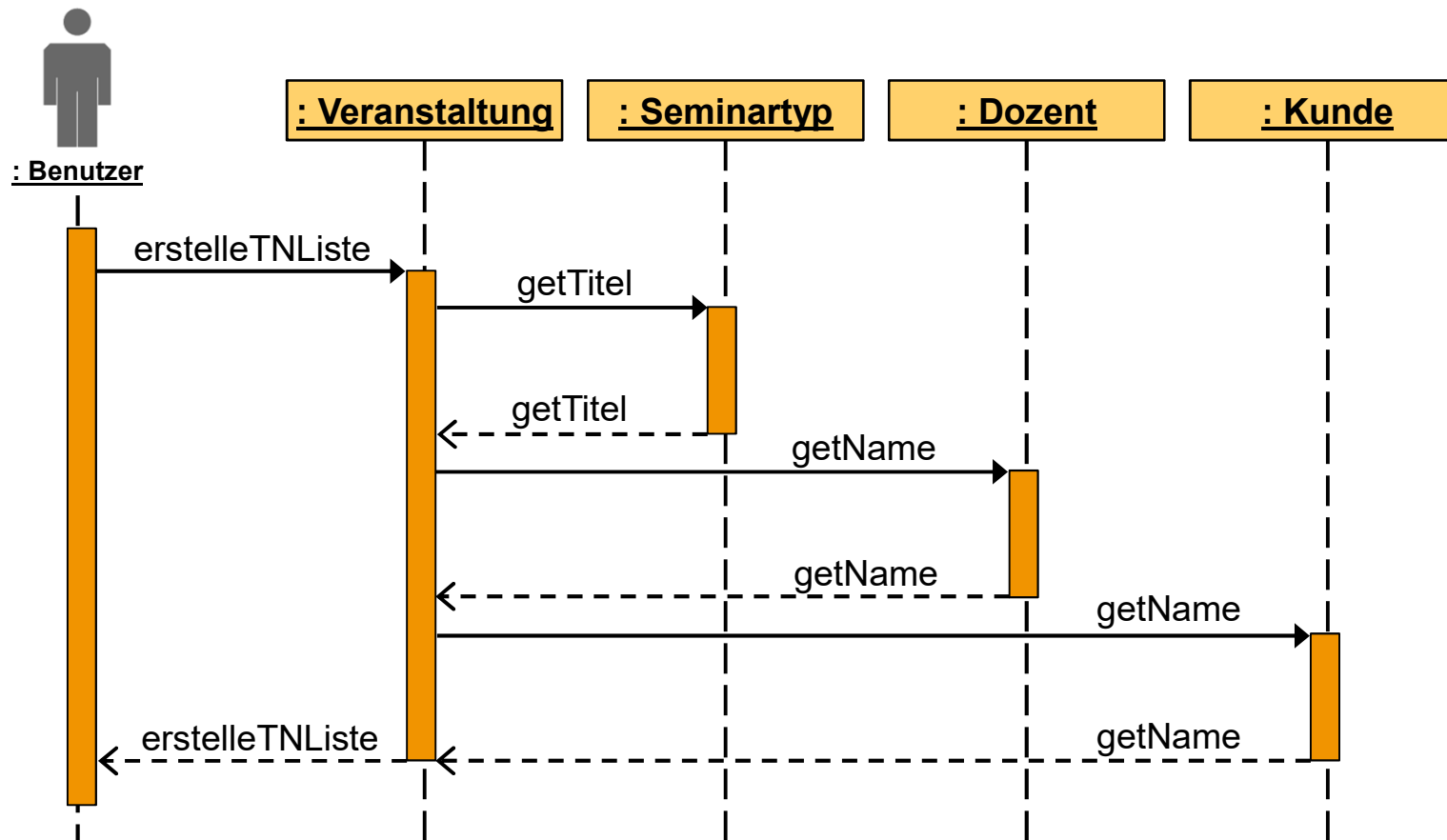


Quelle: Balzert, H.(2009): Lehrbuch der Softwaretechnik, S. 333

# Sequenzdiagramm (3)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Beispiel



*In Anlehnung an: Balzert, H. (2009): Lehrbuch der Softwaretechnik, S. 334*

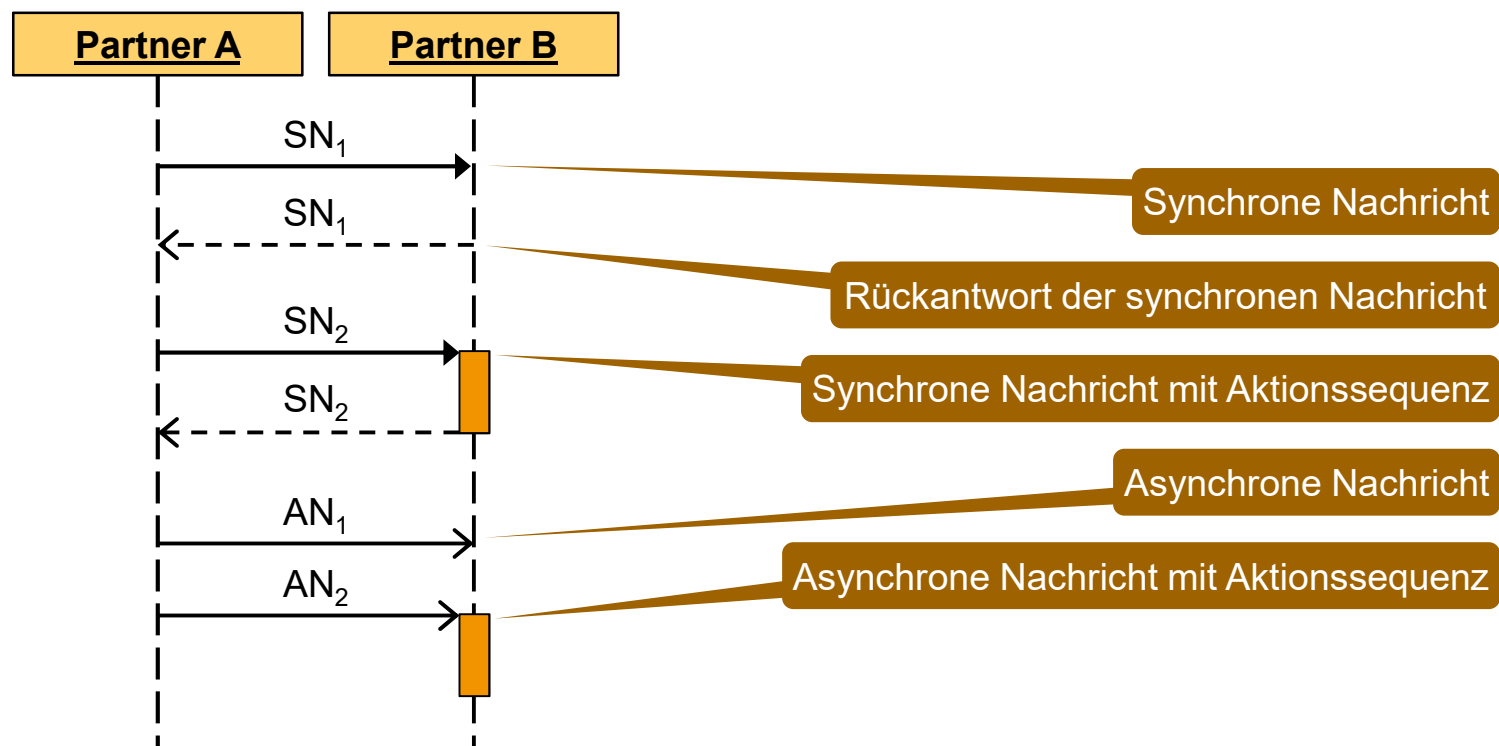




# Sequenzdiagramm (4)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

- Interaktion zwischen zwei Kommunikationspartnern wird dargestellt durch
  - **synchrone** Nachrichten
  - oder **asynchrone** Nachrichten



In Anlehnung an: Balzert, H. (2009): Lehrbuch der Softwaretechnik, S. 335



## ● Aufgabe

Beschreiben Sie folgende Problemstellung als Sequenzdiagramm und skizzieren Sie das Klassendiagramm!

*Ein neuer Kunde eröffnet bei einer Bank ein Sparkonto. Zuerst werden die Daten dieses Kunden erfasst. Bei der Kontoeröffnung muss der Kunde gleich eine Einzahlung vornehmen, d.h. es findet die erste Kontobewegung für dieses Konto statt.*

# Kommunikationsdiagramm (1)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Eigenschaften



## ● Alternative zu Sequenzdiagramm

- Sachverhalt ist äquivalent

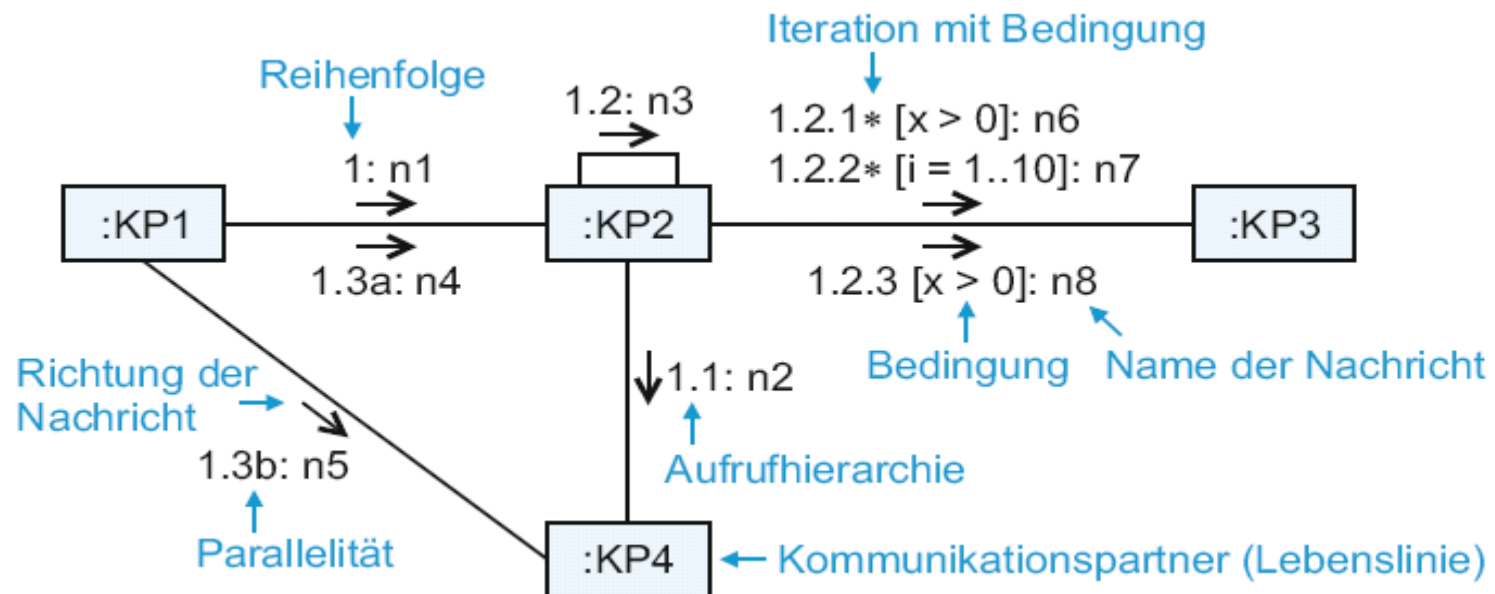
## ● Strukturelle Sicht

- Modelliert das **grundsätzliche Zusammenspiel** mehrerer Kommunikationspartner
- Zeigt wie Objekte für die Ausführung bestimmter Operationen zusammenarbeiten

# Kommunikationsdiagramm (2)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Notation

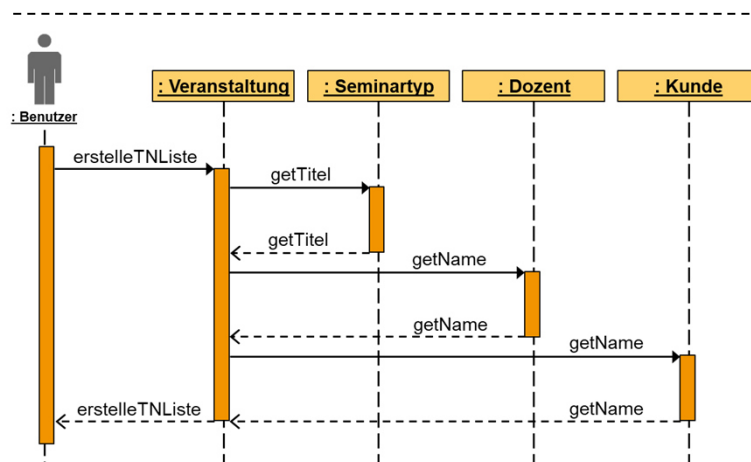
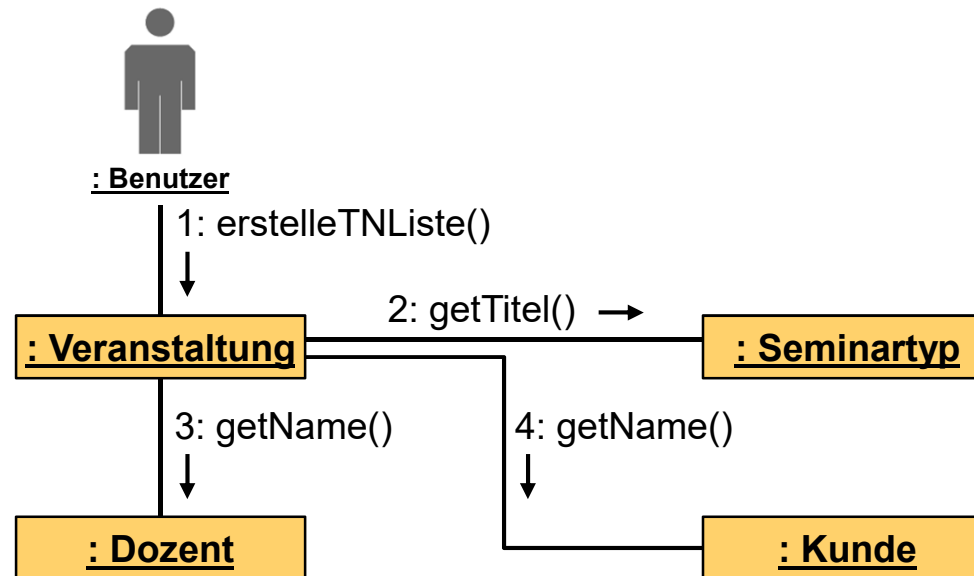


Quelle: Balzert, H.(2009): Lehrbuch der Softwaretechnik, S. 345

# Kommunikationsdiagramm (3)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Ein Beispiel

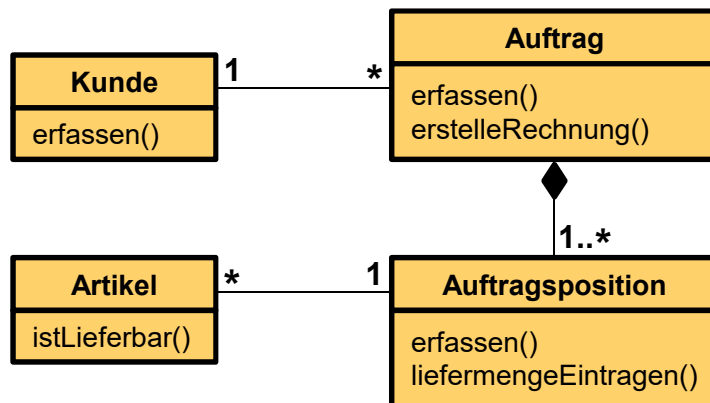
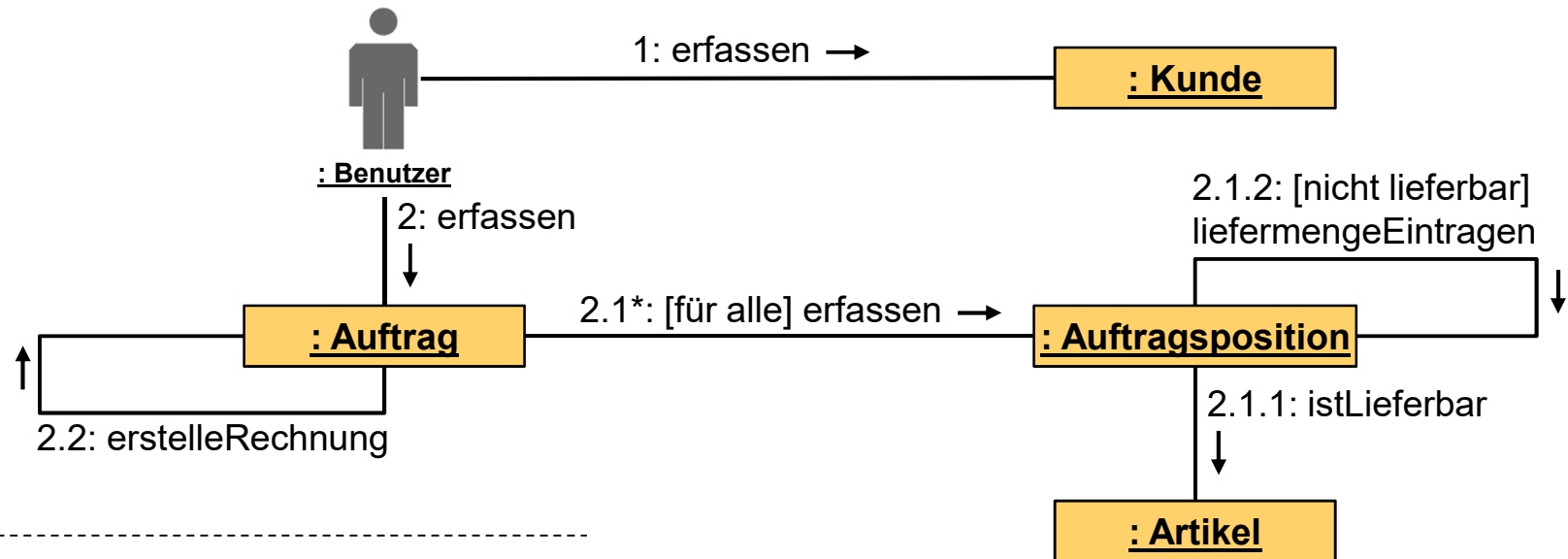


In Anlehnung an: Balzert, H. (2009): Lehrbuch der Softwaretechnik, S. 345

# Kommunikationsdiagramm (4)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● .. und noch ein Beispiel



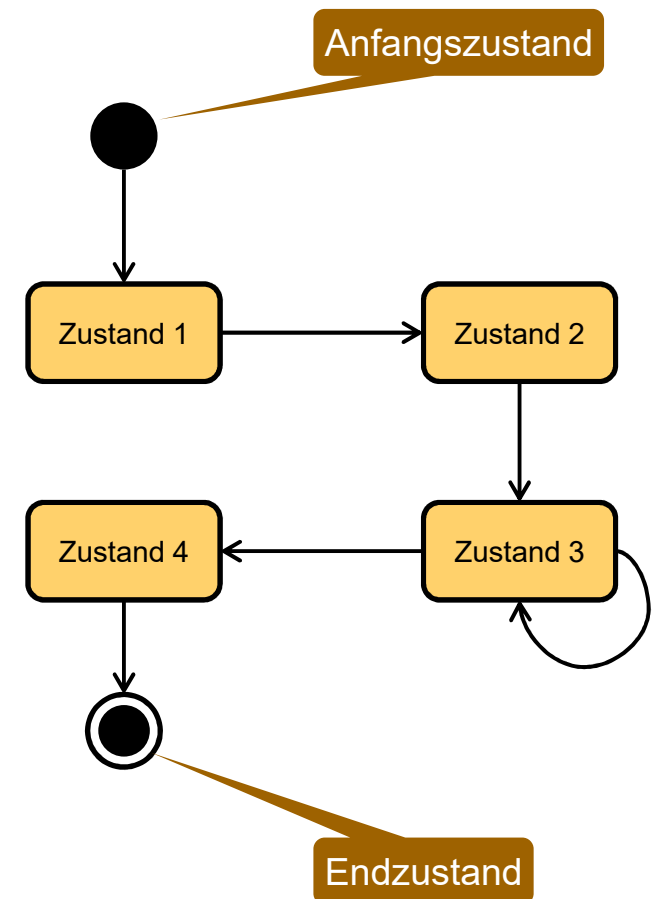
In Anlehnung an: Balzert, H.(2009): Lehrbuch der Softwaretechnik, S. 346



- Wieso Zustände modellieren?
  - Bei vielen Systemen hängt das Ergebnis oder die Ausgabe nicht nur von einer Eingabe oder einem Ereignis ab...
  - ...sondern auch von der **Historie**, welche das System bis dahin durchlaufen hat
  - Zum **Beschreiben solcher Zusammenhänge** eignen sich Zustandsautomaten

## ● Definition Zustandsautomat

- Besteht aus einer endlichen Anzahl
  - von **Zuständen**
  - und **Zustandsübergängen** (Transitionen)
- Zustand beinhaltet (implizit) die Informationen,
  - die sich aus den bisherigen Eingaben ergeben haben
  - und die benötigt werden, um die Reaktion des Systems auf noch folgende Eingaben zu bestimmen



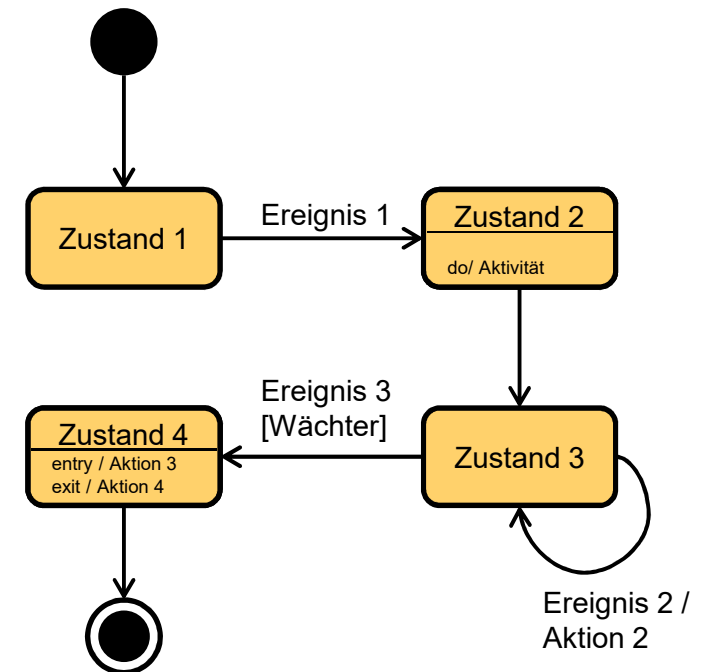


# Zustandsautomat (3)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Eigenschaften

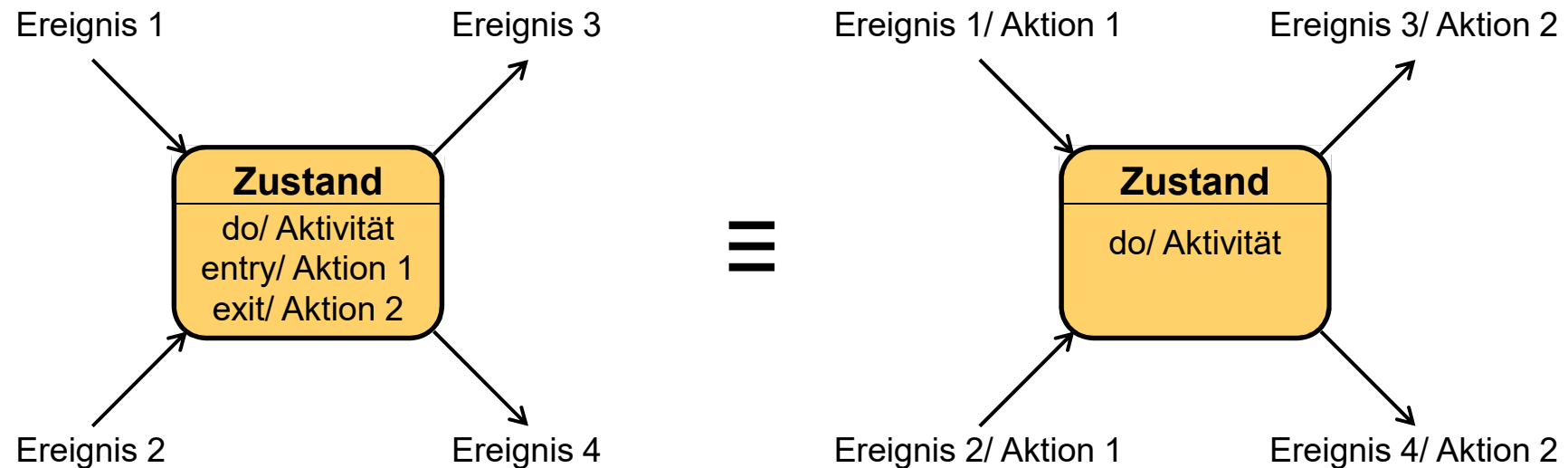
- Zustandsübergang wird durch **Ereignis** ausgelöst
  - Auslösendes Ereignis an Transition
  - Optional: Kombination mit einem **Wächter**
    - Übergang findet nur statt, wenn zu dem Zeitpunkt, zu dem Ereignis eintritt, auch die Bedingung gültig ist
- Mit Transition kann eine **Aktion** verbunden sein
  - Unmittelbar („Bruchteil einer Sekunde“)
- **Aktivitäten** benötigen eine bestimmte Zeitdauer
  - Ausführung einer längeren Berechnung, Anzeigen von Informationen oder Aussenden eines Pieptons, ...
- Mit **Zustand** können auch **Aktionen** verbunden sein
  - bei Eintritt oder Verlassen eines Zustands



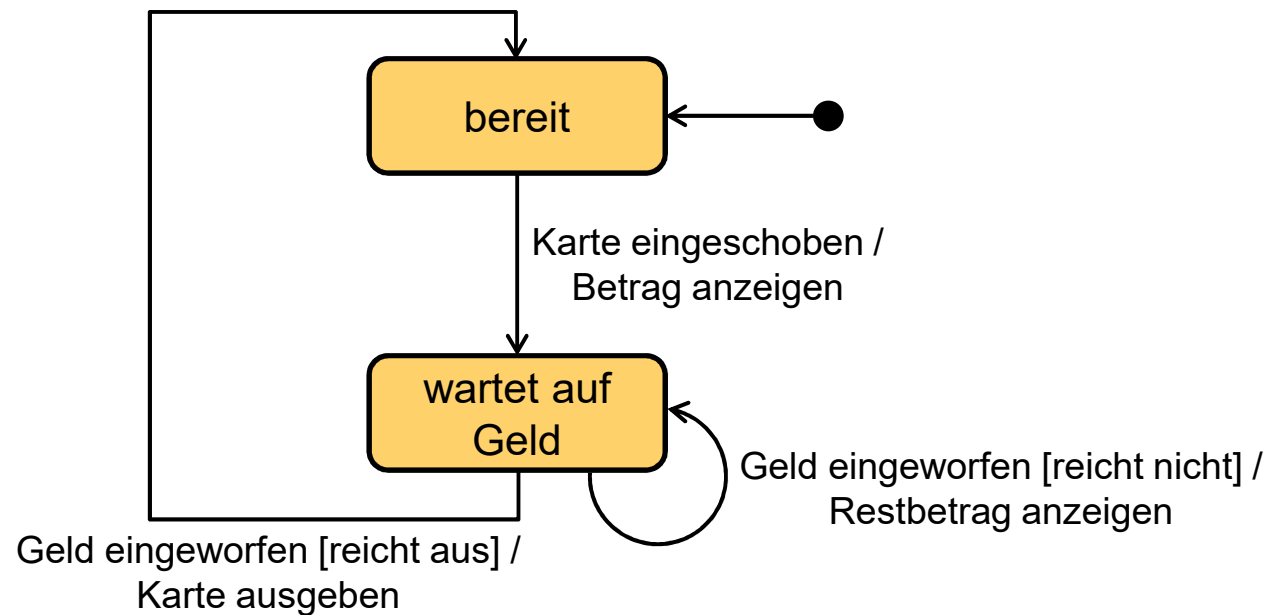
# Zustandsautomat (4)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Aktionen in Zuständen – mögliche Notationen



## ● Beispiel: Kartenautomat im Parkhaus



*In Anlehnung an: Balzert, H.(2009): Lehrbuch der Softwaretechnik, S. 279*

## ● Vorgehensweise bei Erstellung

### 1. Identifizierung von

- Zuständen
- Eingaben bzw. Ereignisse
- und Ausgaben bzw. Ergebnisse

### 2. Festlegen der Übergänge

- UML: zwei Arten von Zustandsautomaten

- Verhaltenszustandsautomat

- Ermöglicht das dynamische Verhalten von **verschiedenen Modellelementen** (z.B. Objekten) zu modellieren

- Protokollzustandsautomat

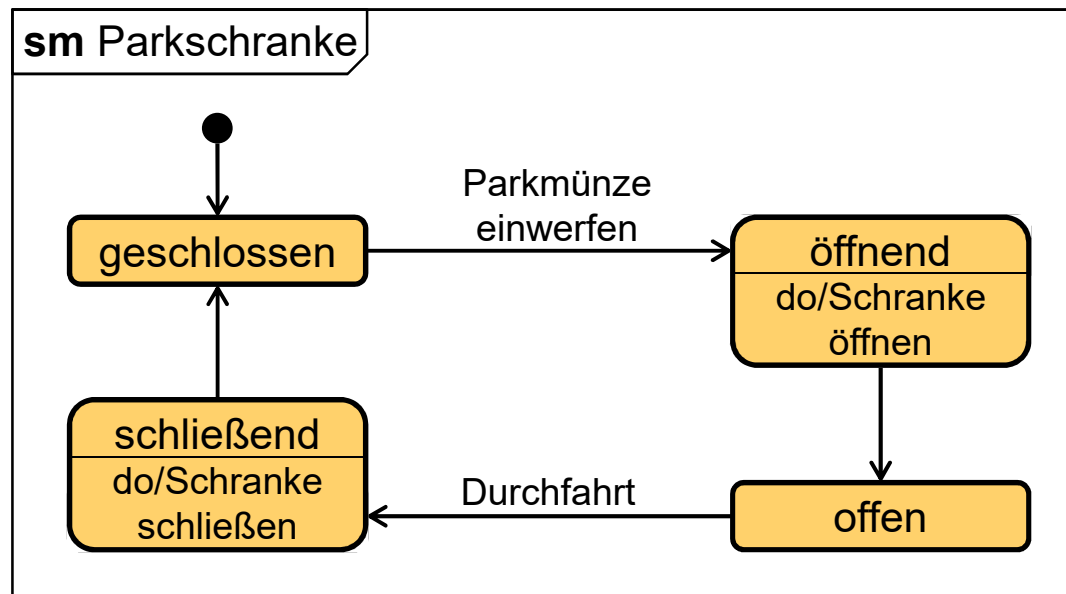
- Beschreibt für ein Objekt einer Klasse, in welchem Zustand und bei welchen Bedingungen die Operationen aufgerufen werden dürfen (**Lebenszyklus eines Objekts**)
- Beschreiben die zulässige Reihenfolge der Operationen
- Nur für Klassen mit Operationen, die nur ausgeführt werden dürfen, wenn bestimmte Bedingungen erfüllt sind

# Zustandsautomat (8)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Verhaltenszustandsautomat

- Ermöglicht das dynamische Verhalten von **verschiedenen Modellelementen** (z.B. Objekten) zu modellieren



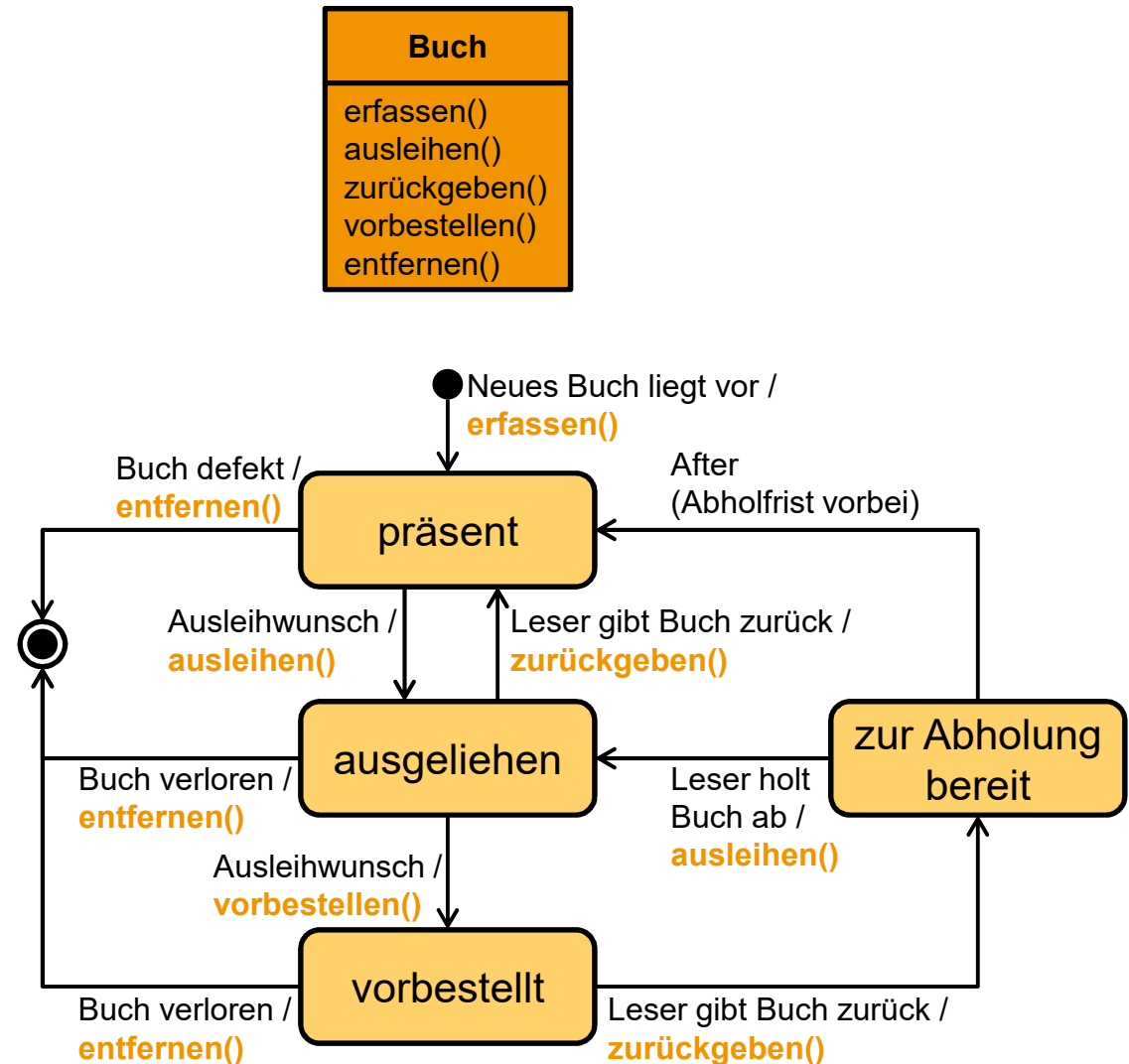
Quelle: Balzert, H.(2009): Lehrbuch der Softwaretechnik, S. 289

# Zustandsautomat (9)

05 Modellierung / 05.5 Szenarien, Zustandsautomaten

## ● Protokollzustandsautomat

- Für Klassen mit Operationen, die nur **unter bestimmten Bedingungen** ausgeführt werden dürfen
- Beschreibt die **zulässige Reihenfolge der Operationen**
- Beschreibt für ein Objekt einer Klasse, in welchem Zustand und bei welchen Bedingungen die Operationen aufgerufen werden dürfen (**Lebenszyklus eines Objekts**)



Quelle: Balzert, H. (2009): Lehrbuch der Softwaretechnik, S. 290



## ● Aufgabe

Spezifizieren Sie basierend auf folgender Problembeschreibung einen Verhaltenszustandsautomaten!

*Der Kauf von Konzertkarten soll modelliert werden. Zu Beginn sind die Konzertkarten verfügbar und der Kunde prüft das Angebot. Sobald der Kunde bestimmte Plätze ausgewählt hat, werden die Karten für maximal eine Woche reserviert. In dieser Zeit hat der Kunde die Möglichkeit, den Preis für die reservierten Karten zu überweisen. Trifft die Überweisung ein, werden die Karten als verkauft angesehen. Verkaufte Karten kann der Kunde bis zum Konzerttermin abholen, andernfalls verfallen die Karten. Der Kunde kann auch direkt zum Konzerthaus gehen, die reservierten Karten bezahlen und mitnehmen.*