

Unified Modeling Language

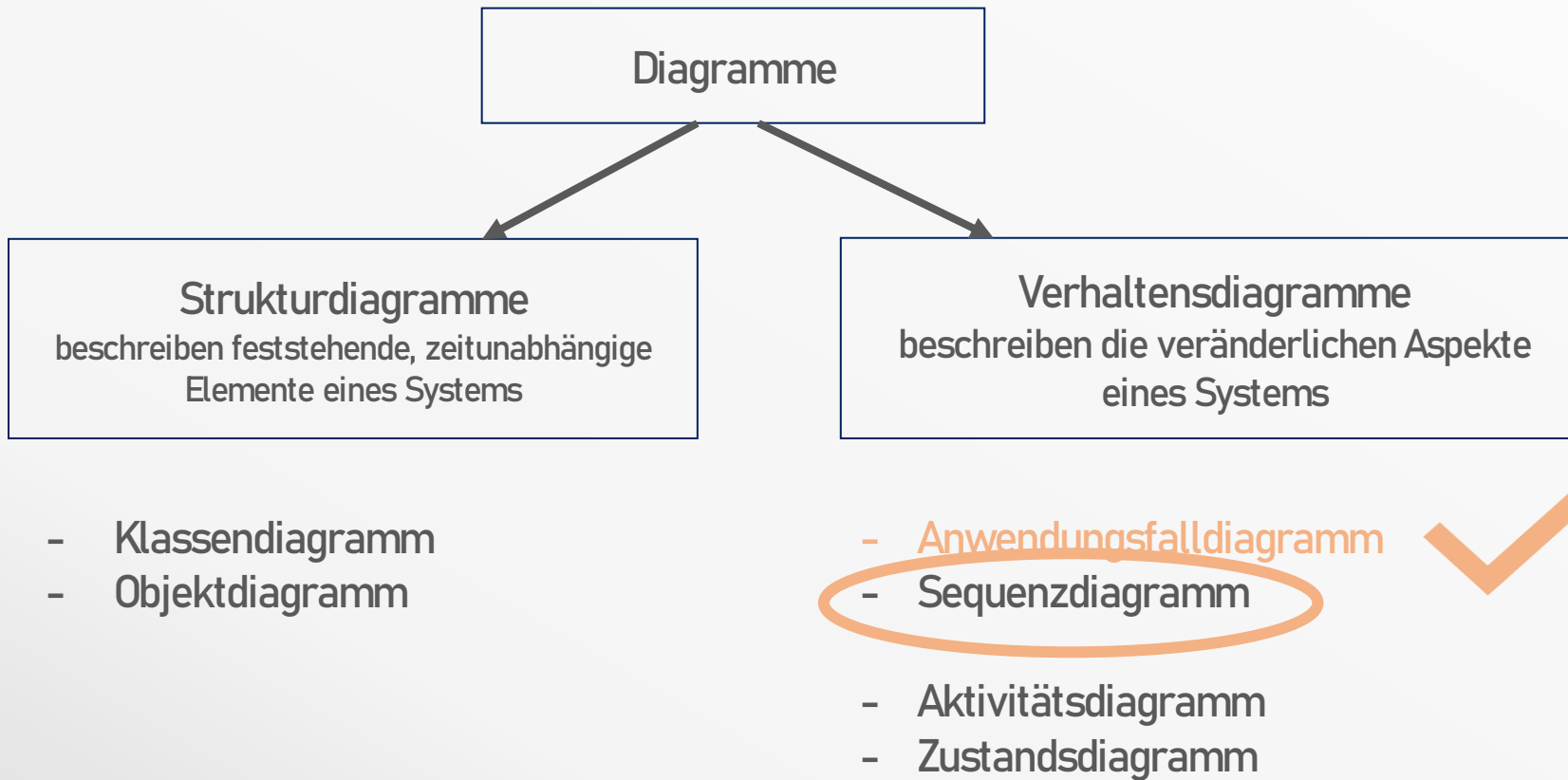
UML

Inhalt

- Projektmanagement
- Modellierungssprache UML – Use Case, Klassendiagramm, etc.
- Programmiersprachen – Einordnen
- Modellierungssprachen – Pseudocode, Struktogramm, PAP
- Programmierung – Algorithmen, Operatoren, Kontrollstrukturen, etc.
- Datenbanken – Grundlagen, ERM, Normalisierung, SQL-Befehle
- Webprogrammierung – HTML, CSS, PHP, XML, JSON
- Testen – Testverfahren, Testdaten, Unit-Test

Inhalt

- Datensicherheit, Datenschutz
- IT-Sicherheit
- IT-Systeme
- Netzwerke
- Arbeits- und Geschäftsprozesse



UML

Sequenzdiagramm

Sequenzdiagramm

Sequenzdiagramm:

- Beschreibt die zeitliche Abfolge von Interaktionen zwischen einer Menge von Objekten in einem einzigen Anwendungsfall

Sequenz, sequenziell oder sequentiell
(von lateinisch *sequentia* „Aufeinanderfolge“)

Quelle: <https://www.duden.de/rechtschreibung/Sequenz>

Ziel:

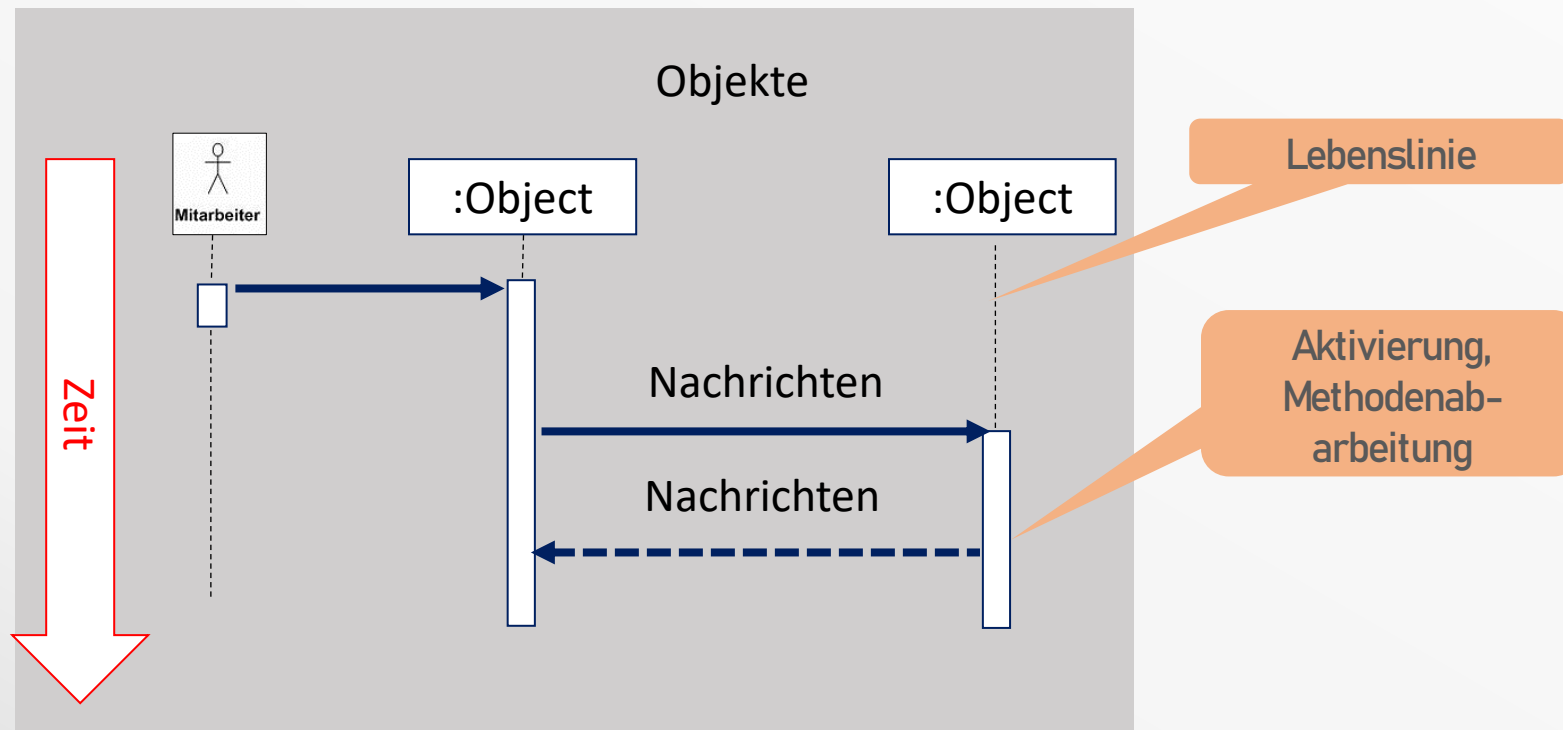
- Die Zusammenarbeit der Objekte darzustellen
- Erkennbar zu machen, wann (bei welcher Nachrichten oder bei welchem Ereignis) ein Objekt aktiviert wird und mit welcher Nachricht es reagiert

Sequenzdiagramm

- Gehört zu den Interaktionsdiagrammen
- Fokus liegt auf dem genauen Nachrichtenfluss innerhalb eines Systems im Verlauf der Zeit
- Gut ablesbar,
 - in welcher Reihenfolge Nachrichten ausgetauscht werden
 - In welchem Zeitraum ein Objekt beschäftigt ist
 - Wann ein Objekt "Leerlauf" hat
- Die Reihenfolge und die Verschachtelung der Operationen sind gut zu erkennen
- Hilfreich in den verschiedensten Phasen (bspw. Designphase) der Softwareentwicklung
- Dient zur Kommunikation mit dem Auftraggeber bzw. Anwender

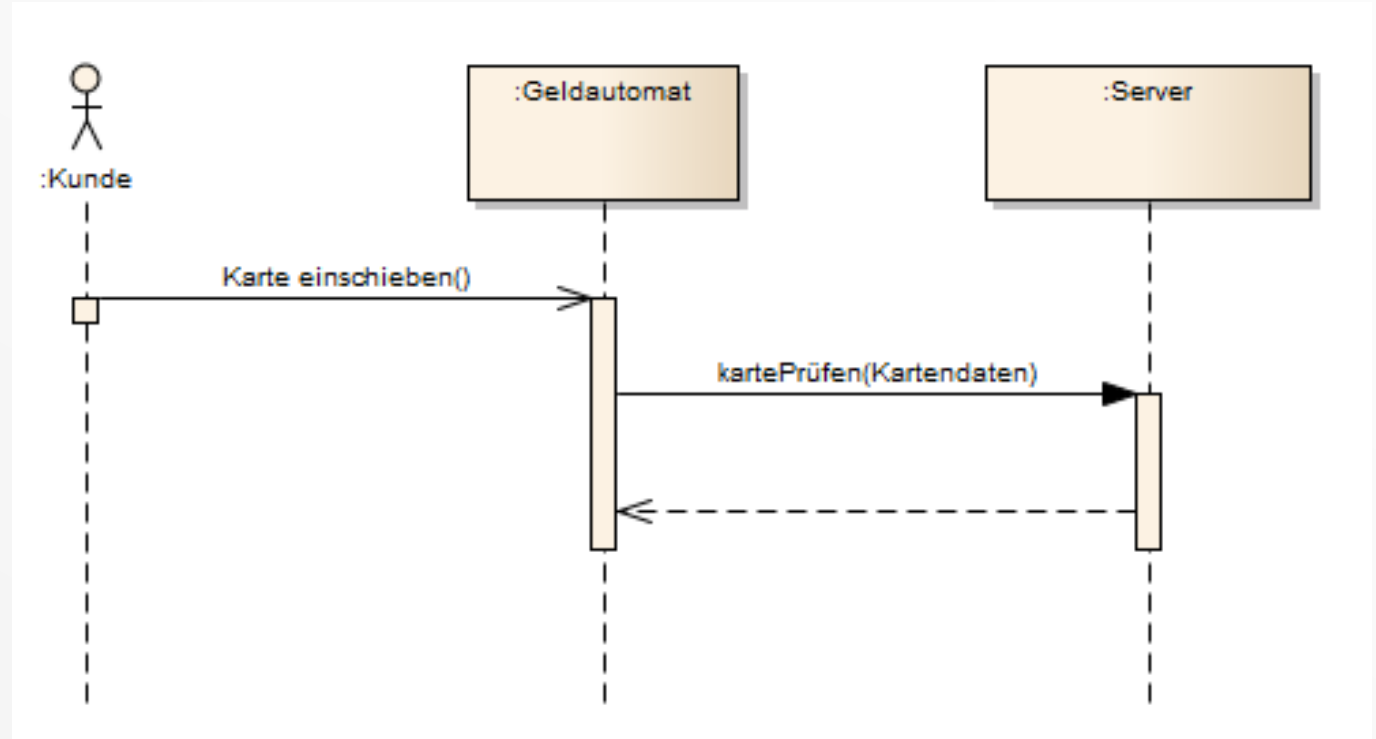
Sequenzdiagramm

Eine Lebenslinie beginnt immer mit einem Strichmännchen für Menschen oder einem beschrifteten Rechteck für eine Sache



Sequenzdiagramm

- Welche Methoden sind hier für die Kommunikation zwischen ausgewählten Objekten zuständig?
- Wie findet der zeitliche Ablauf von Methodenaufrufen zwischen ausgewählten Objekten statt?
- Welche Objekte werden in einer Sequenz neu erstellt und / oder zerstört?



Quelle: <https://www.sparxsystems.de>

Sequenzdiagramm

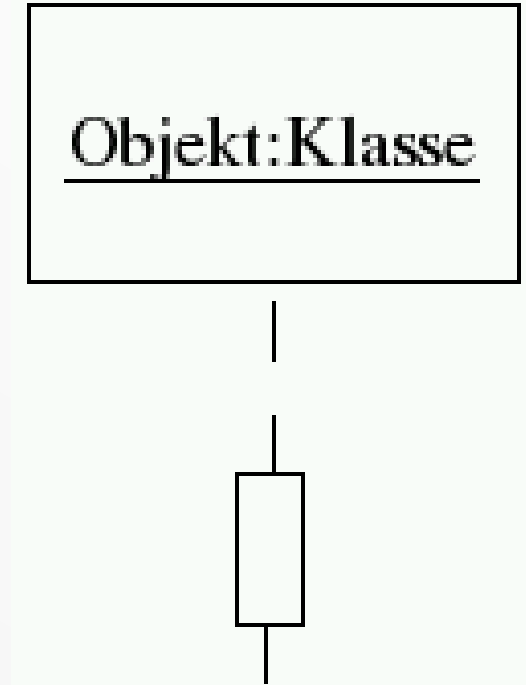
Klasse, Objekt

In dem Rechteck oberhalb der gestrichelten Linie wird der Objektname und der Klassenname angegeben. Der Name wird unterstrichen.

Die senkrechte, gestrichelte Linie stellt die Lebenslinie (lifeline) eines Objekts dar. In diesem zeitlichen Bereich existiert das Objekt.

Das schmale Rechteck auf der gestrichelten Linie stellt eine Aktivierung dar.

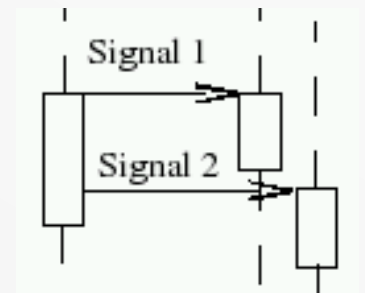
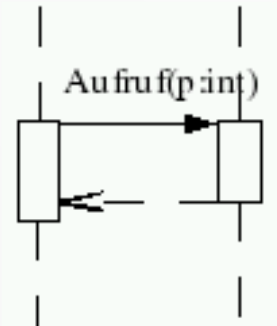
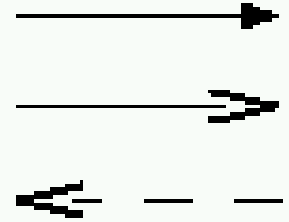
Eine Aktivierung ist der Bereich, in dem eine Methode des Objektes aktiv ist (ausgeführt wird). Auf einer Lebenslinie können mehrere Aktivierungen enthalten sein.



Sequenzdiagramm

Nachricht, Botschaft

- Objekte kommunizieren über Nachrichten.
- Nachrichten werden als Pfeile zwischen den Aktivierungen eingezeichnet.
- Der Name der Nachricht steht an dem Pfeil.
- Eine Nachricht liegt immer zwischen einem sendenden und einem empfangenden Objekt.



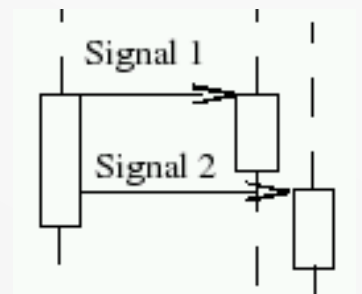
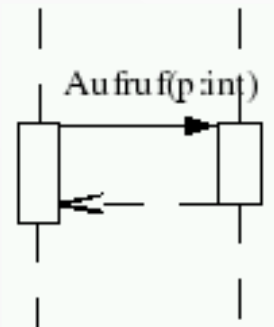
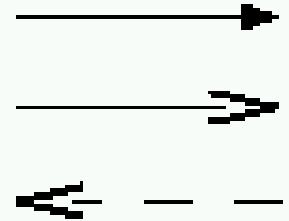
Sequenzdiagramm

Synchrone Nachricht (Au)

Mit einer gefüllten Pfeilspitze werden synchrone Nachrichten gekennzeichnet.
Die ausgehende Lebenslinie ist für weitere Nachrichten "blockiert", bis diese eine Antwort erhält.

Asynchrone Nachricht

Mit einer offenen Pfeilspitze werden asynchrone Nachrichten gekennzeichnet.
Der Aufruf erfolgt von der Quelle zum Ziel. Die Quelle wartet mit der Verarbeitung nicht auf die Zielklasse, sondern setzt ihre Arbeit nach dem Senden der Nachricht fort.
Asynchrone Aufrufe werden verwendet, um parallele Threads zu modellieren.



Sequenzdiagramm

Beispiel:

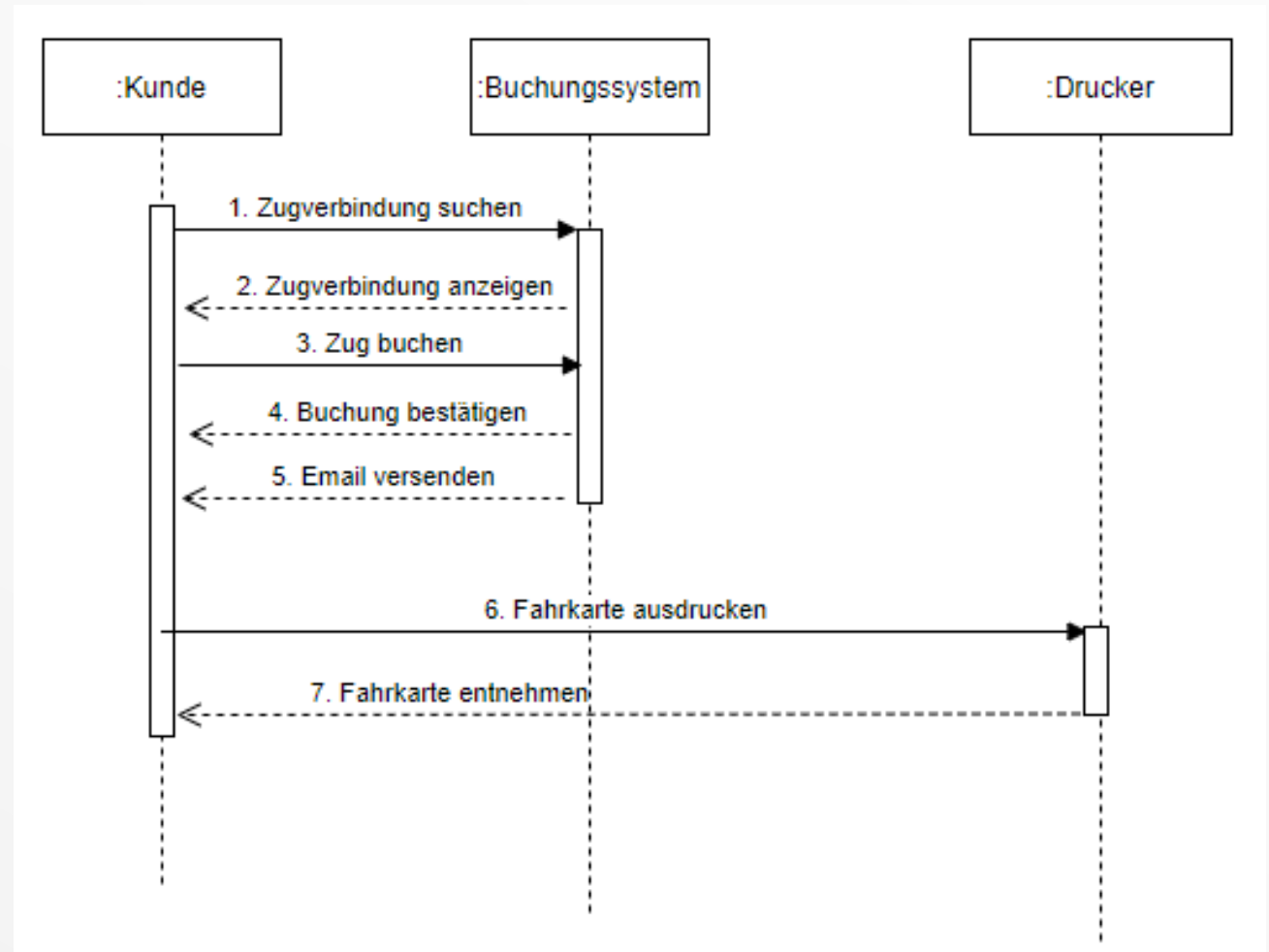
Sie möchten mit der Deutschen Bahn nach Hamburg reisen und bestellen für diese Fahrt übers Internet ein Online-Ticket zum Selbstausdrucken.

Wer ist beteiligt?

- Kunde
- Buchungssystem der Deutschen Bahn
- Drucker

Aktivitäten?

- Zugverbindung suchen
- Zugverbindung anzeigen
- Zug buchen
- Buchung bestätigen
- Email versenden
- Fahrkarte ausdrucken
- Fahrkarte entnehmen

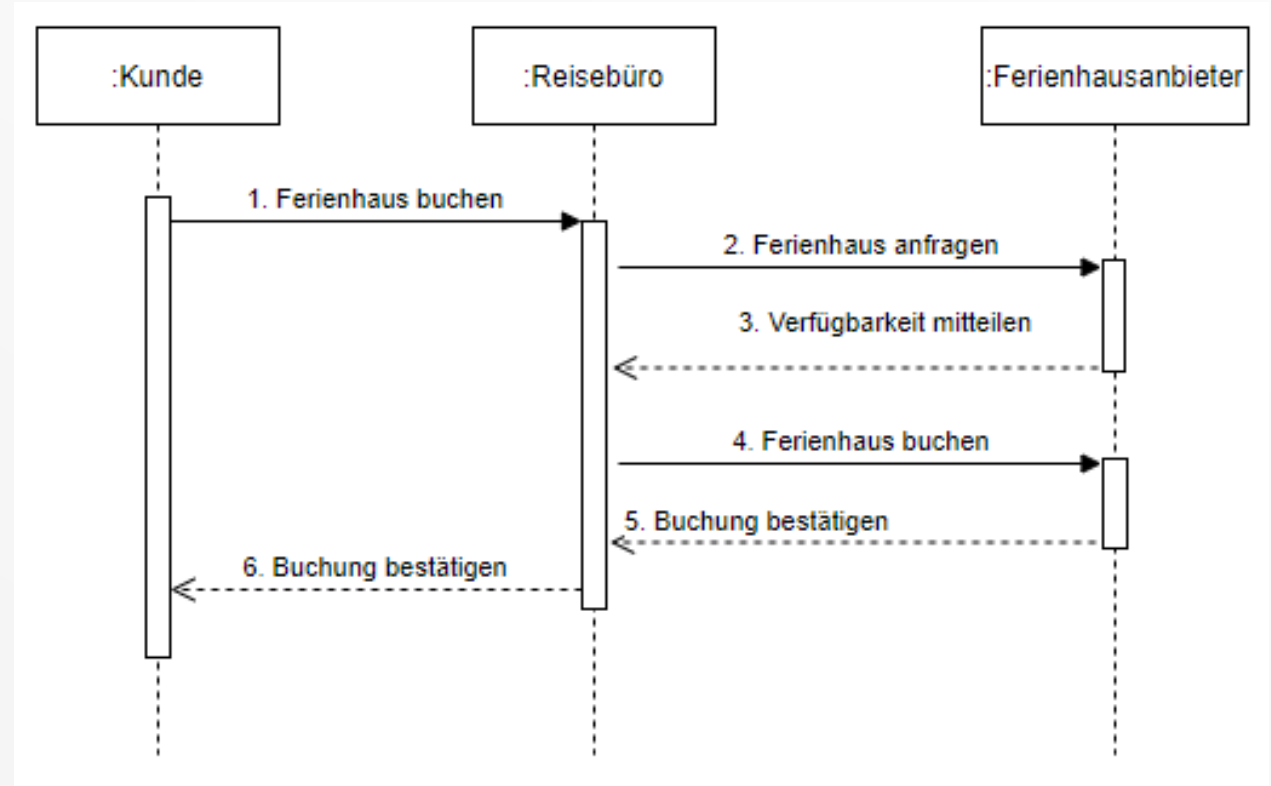


Sequenzdiagramm

Beispiel:

Reisebüro

Ein Sequenzdiagramm für die
Buchung einer Ferienreise in
einem Reisebüro

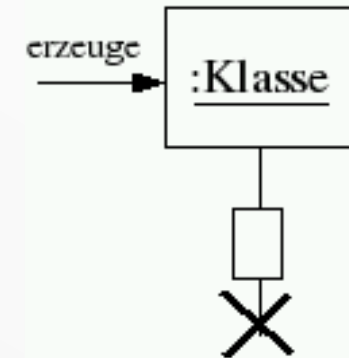


Sequenzdiagramm

Objekt erzeugen, zerstören

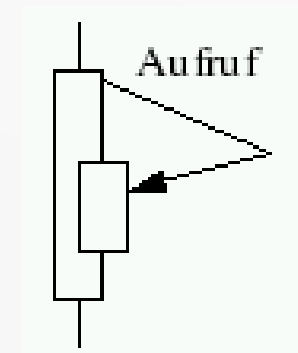
Das Erzeugen eines Objektes wird durch eine Nachricht, die im Kopf des Objekts endet dargestellt.

Das Zerstören (Löschen) eines Objektes wird durch ein x auf der Lebenslinie markiert.



Rekursion

Sendet ein Objekt eine Nachricht an sich selbst, so ruft es eine Methode auf, die es selbst implementiert.



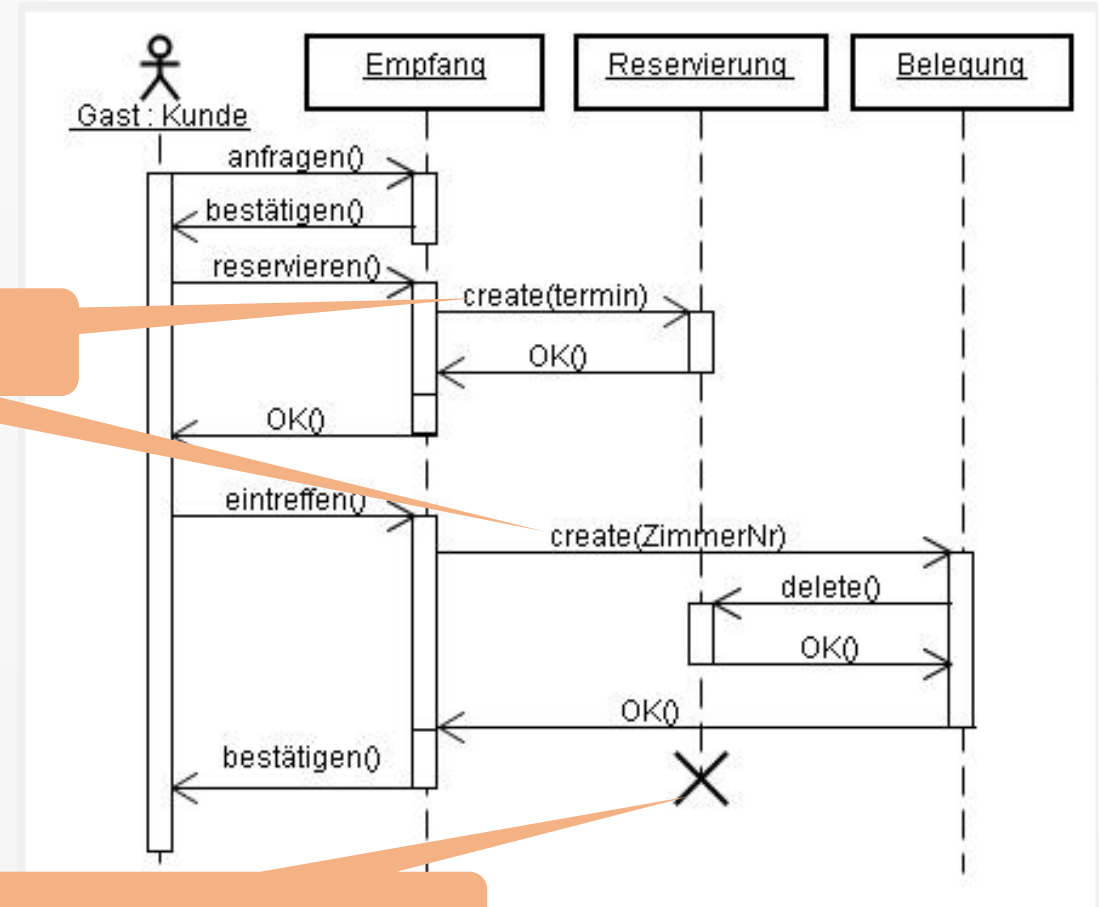
Sequenzdiagramm

Beispiel:

Hotelbetrieb

- Zimmerbelegung nach Reservierung im Hotel

Objekt wird erzeugt



Objekt wird gelöscht

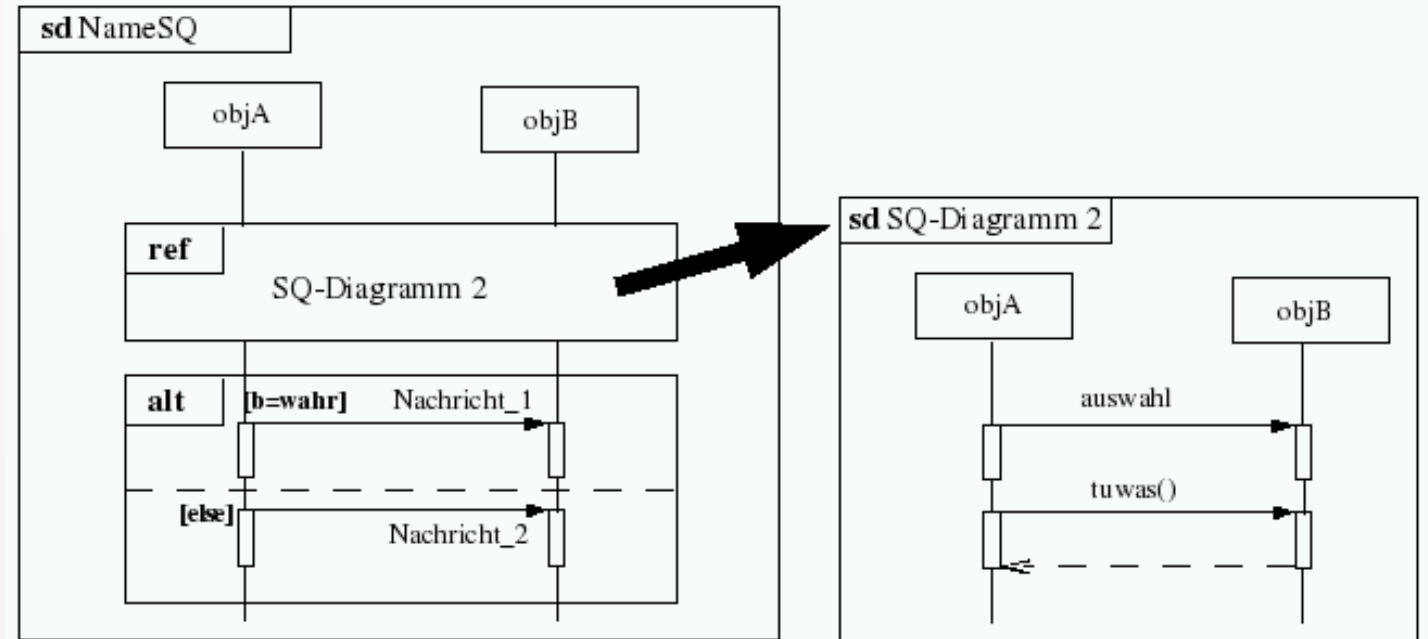
Sequenzdiagramm

Interaktionsrahmen

Ein Sequenzdiagramm enthält Objekte, Lebenslinien, Aktivierungen, Nachrichten und kombinierte Fragmente (combined fragment).

Mit kombinierten Fragmenten können in Sequenzdiagrammen die Kontrollstrukturen höherer Programmiersprachen ausgedrückt werden.

Jedes kombinierte Fragment wird als rechteckiger Block dargestellt. In der linken, oberen Ecke trägt er eine Bezeichnung, die den Typ der Kontrollstruktur angibt (interaction operator).



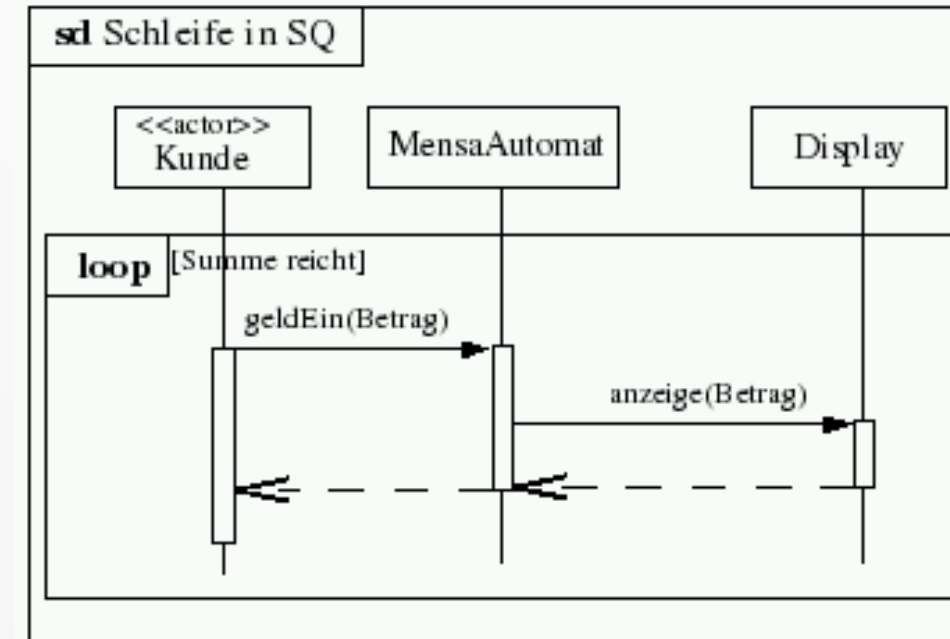
Sequenzdiagramm

Schleife

Diese Abbildung zeigt eine Schleife in einem Sequenzdiagramm.
Die modellierte Szene ist eine Variation des Beispiels Mensaautomat.

Es sind drei Objekte an der Szene beteiligt.

- Das Objekt Kunde ist vom Stereotyp <<actor>>. Er ist ein Akteur (aus dem Use Case Diagramm) und greift auf den MensaAutomat zu, indem er ihm die Botschaft "geldEin" sendet; diese Botschaft übergibt den Parameter "Betrag". Danach sendet der "MensaAutomat" die Botschaft "anzeige(Betrag)" an das "Display".
- Das "Display" führt die Methode aus und gibt die Kontrolle über den Return wieder an den "MensaAutomat" zurück.
- Der "MensaAutomat" gibt die Kontrolle wieder an den "Kunden" zurück.
Diese Sequenz von Botschaften wird solange ausgeführt bis die Abbruchbedingung [Summe reicht] erfüllt ist.
Das Prüfen des Geldes auf Echtheit ist in diesem Beispiel nicht modelliert worden.



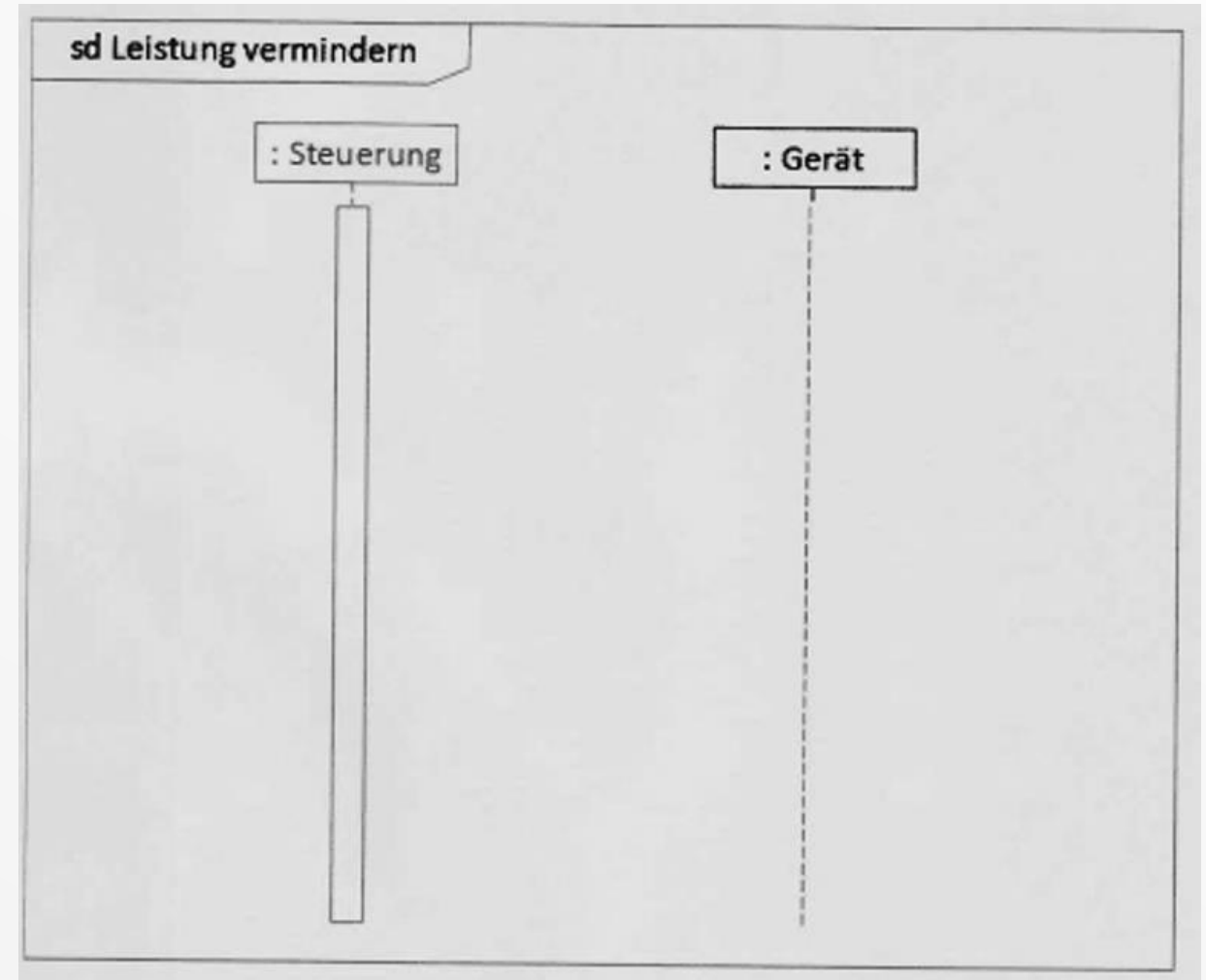
Sequenzdiagramm

Beispiel:

Die Steuerung fragt alle 5 Sekunden die Pulsfrequenz vom Gerät ab.

- Wenn die gemessene Pulsfrequenz größer als die festgelegte maximale Pulsfrequenz ist, löst die Steuerung eine Verminderung der Leistungsstufe im Gerät aus.
- Sonst bleibt die momentane Leistungsstufe erhalten.

Modellieren Sie dieses Szenario in einem UML-Sequenzdiagramm, dessen Grundstruktur wie folgt gegeben ist.



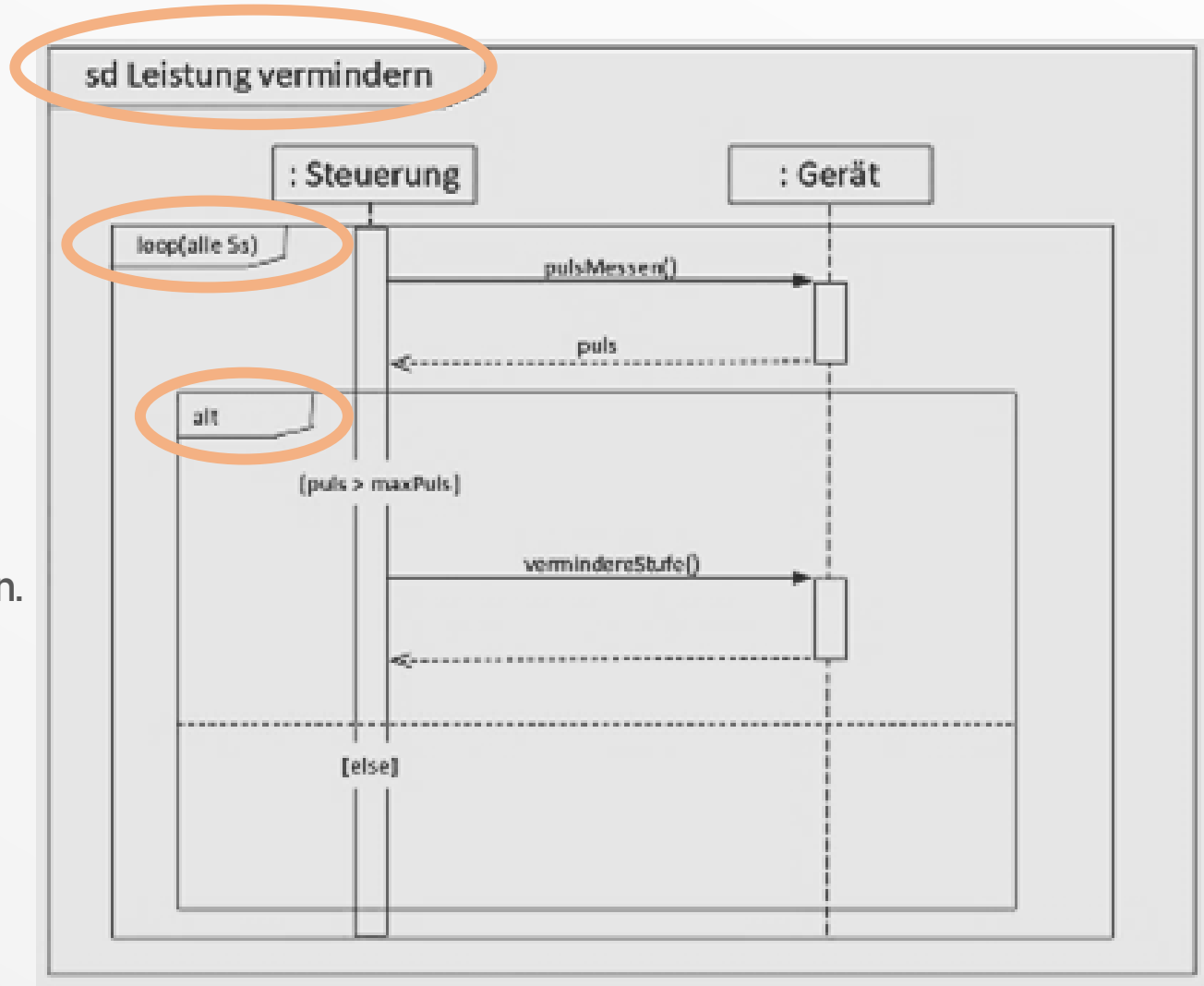
Sequenzdiagramm

Beispiel:

Die Steuerung fragt alle 5 Sekunden die Pulsfrequenz vom Gerät ab.

- Wenn die gemessene Pulsfrequenz größer als die festgelegte maximale Pulsfrequenz ist, löst die Steuerung eine Verminderung der Leistungsstufe im Gerät aus.
- Sonst bleibt die momentane Leistungsstufe erhalten.

Modellieren Sie diese Szenario in einem UML-Sequenzdiagramm, dessen Grundstruktur wie folgt gegeben ist.



Sequenzdiagramm

Fragmente	Bedeutung
critical	„Fehler“ möglich
alt	Alternative, Bedingung (IF ... THEN ... ELSE)
loop	Wiederholung

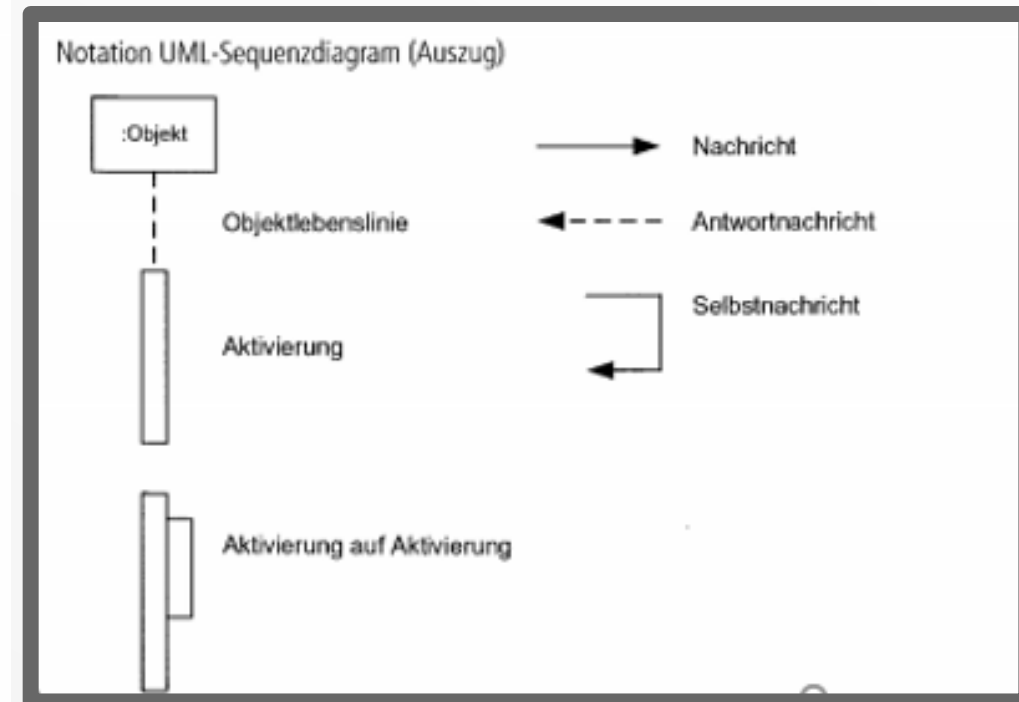
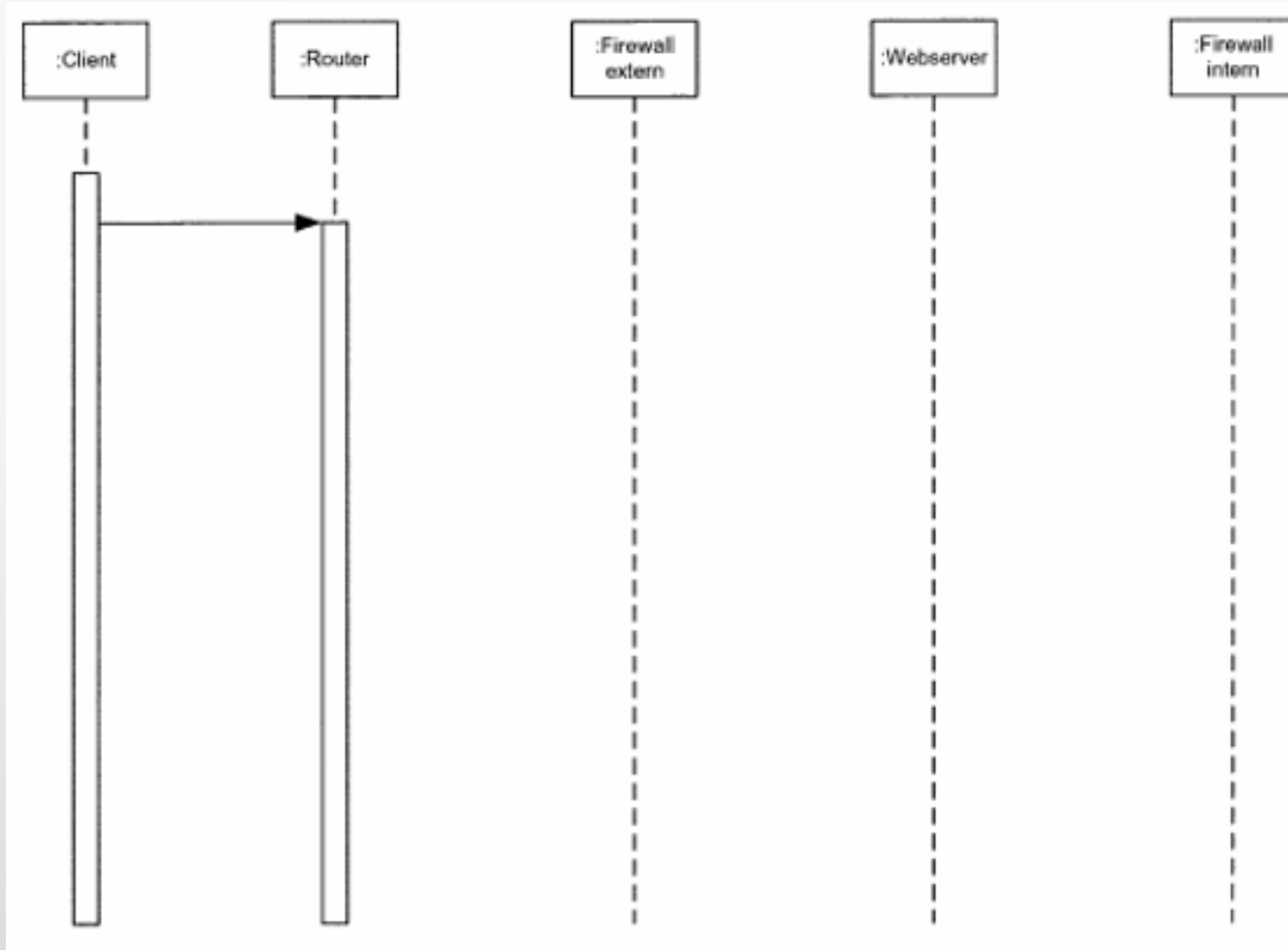
Sequenzdiagramm

In einem Unternehmen wird der Webserver durch eine Firewall in einer Demilitarisierten Zone (DMZ) geschützt.

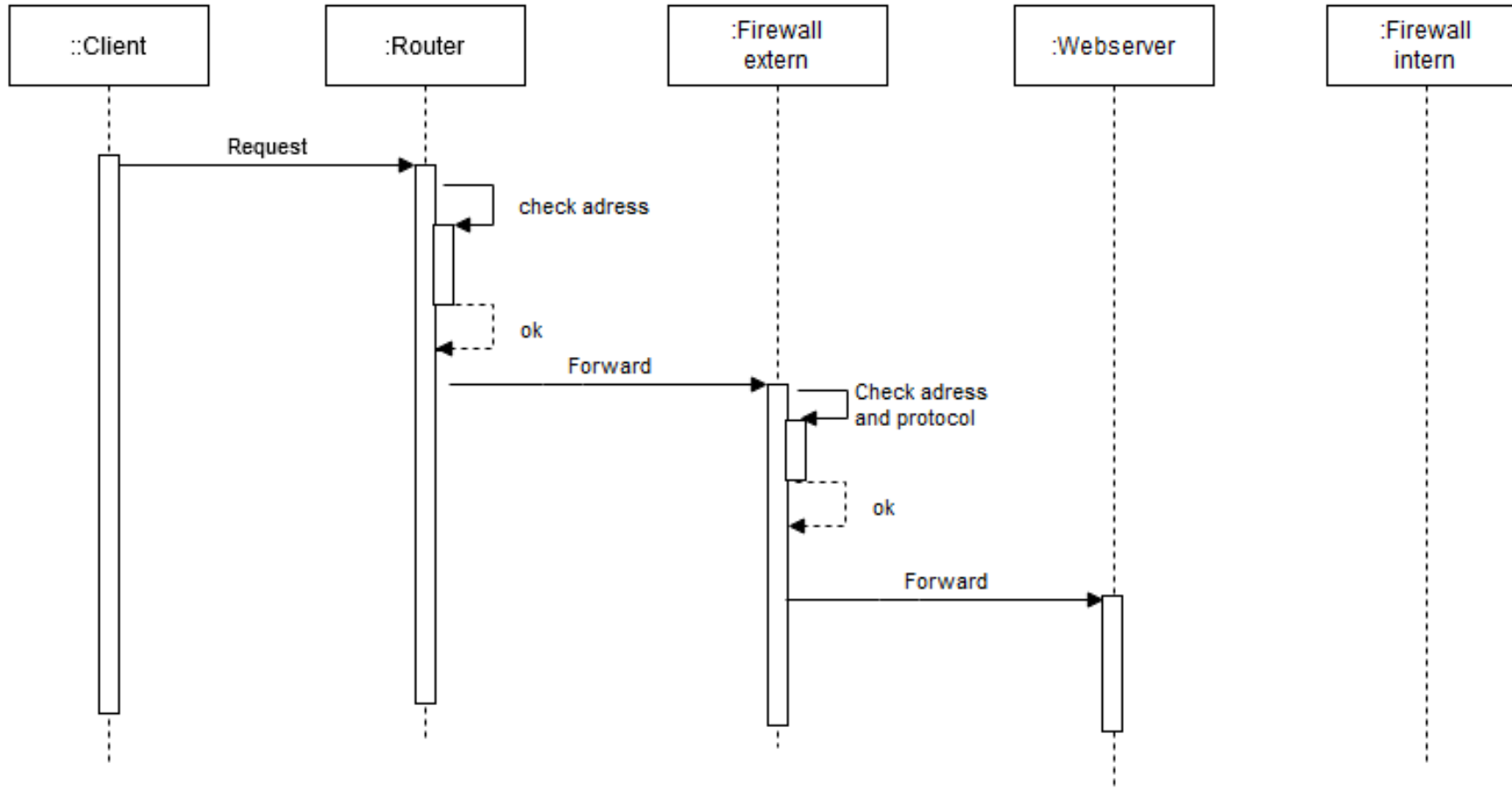
Ergänzen Sie das Sequenzdiagramm für eine positiv gefilterte Anfrage an den Webserver von einem externen Client.

Client	Stellt Anfragen an Webserver
Router des Providers	Leitet die Anfragen an die Firewall weiter, wenn er einen Eintrag für die Zieladresse in seiner Routingtabelle findet
Firewall	Untersucht den Datenverkehr und verhindert nicht erwünschten Datenverkehr
Webserver	Nimmt Anfragen an

Sequenzdiagramm



Sequenzdiagramm





Beispiel Aufgabe

UML Sequenzdiagramm entwerfen
(Einzelnen bearbeiten 15 Minuten)

Prüfungsaufgabe

Die CARMED GmbH verwendet zur Konstruktion und Dokumentation von Software- und IT-Systemen die Unified Modeling Language (UML)

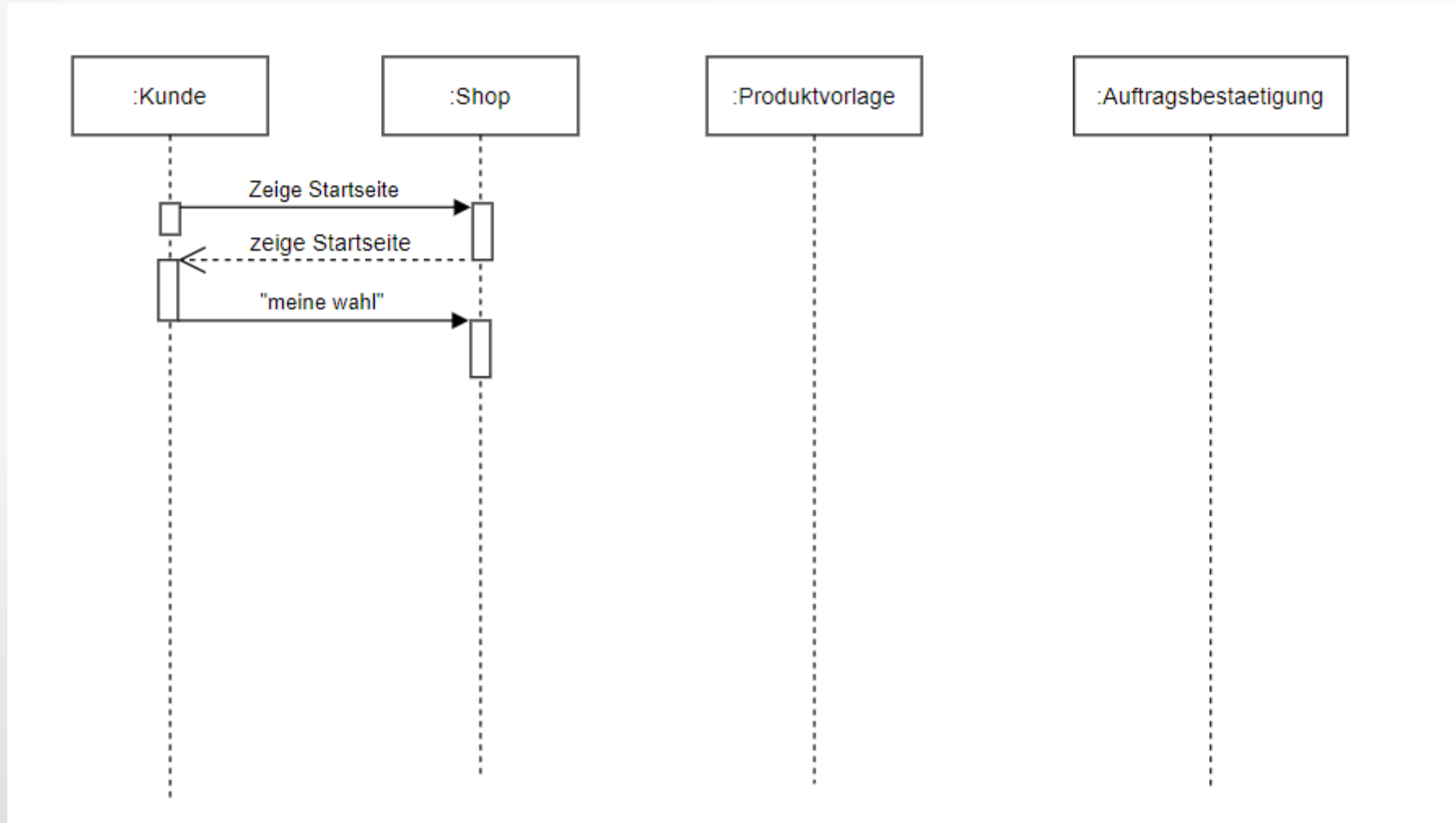
- a. Der Ablauf einer Bestellung im Onlineshop der FoPo AG soll in einem UML-Sequenzdiagramm dargestellt werden. Der Ablauf wird wie folgt beschrieben:
- Kunde ruft die Startseite mit dem Angebot des Shops auf.
 - Shop zeigt Kunden die Startseite mit den zur Wahl stehenden Produktarten (Foto, Poster).
 - Kunde teilt Shop Produktwahl mit.
 - Shop erstellt Produktvorlagen-Objekt; Aufruf der Funktion *create()* der entsprechenden Produktvorlagen-Klasse.
 - Shop zeigt dem Kunden auf HTML-Seite die leere Produktvorlage und die Mitteilung, dass das Bild hochgeladen werden kann.
 - Kunde lädt Bild hoch.
 - Shop übergibt Bild an Produktvorlage; Aufruf *bild()* des Produktvorlagen-Objekts.
 - Produktvorlage meldet „Bild ist eingefügt“.
 - Shop zeigt Kunden HTML-Seite mit fertiger Produktvorlage und fordert Kunden zur Eingabe der Adress- und Zahlungsdaten auf.
 - Kunde übermittelt Daten.
 - Shop ruft eigene Funktion *pruefung()* zur Prüfung der Daten auf.
 - Funktion *pruefung()* gibt „Daten o.k.“ zurück. Der Fall „Daten nicht o.k.“ wird nicht betrachtet.
 - Shop teilt Kunden mit, dass er zur Annahme des Auftrags bereit ist.
 - Kunde erteilt Auftrag.
 - Shop erstellt ein Auftragsbestätigungs-Objekt mit dem Aufruf der Funktion *create()* der Klasse *Auftragsbestaetigung*
 - Shop verschickt Auftragsbestätigung per E-Mail an den Kunden.

Prüfungsaufgabe

Die CARMED GmbH verwendet zur Konstruktion und Dokumentation von Software- und IT-Systemen die Unified Modeling Language (UML)

- a. Der Ablauf einer Bestellung im Onlineshop der FoPo AG soll in einem UML-Sequenzdiagramm dargestellt werden. Der Ablauf wird wie folgt beschrieben:
- Kunde ruft die Startseite mit dem Angebot des Shops auf.
 - Shop zeigt Kunden die Startseite mit den zur Wahl stehenden Produktarten (Foto, Poster).
 - Kunde teilt Shop Produktwahl mit.
 - Shop erstellt Produktvorlagen-Objekt; Aufruf der Funktion *create()* der entsprechenden Produktvorlagen-Klasse.
 - Shop zeigt dem Kunden auf HTML-Seite die leere Produktvorlage und die Mitteilung, dass das Bild hochgeladen werden kann.
 - Kunde lädt Bild hoch.
 - Shop übergibt Bild an Produktvorlage; Aufruf *bild()* des Produktvorlagen-Objekts.
 - Produktvorlage meldet „Bild ist eingefügt“.
 - Shop zeigt Kunden HTML-Seite mit fertiger Produktvorlage und fordert Kunden zur Eingabe der Adress- und Zahlungsdaten auf.
 - Kunde übermittelt Daten.
 - Shop ruft eigene Funktion *pruefung()* zur Prüfung der Daten auf.
 - Funktion *pruefung()* gibt „Daten o.k.“ zurück. Der Fall „Daten nicht o.k.“ wird nicht betrachtet.
 - Shop teilt Kunden mit, dass er zur Annahme des Auftrags bereit ist.
 - Kunde erteilt Auftrag.
 - Shop erstellt ein Auftragsbestätigungs-Objekt mit dem Aufruf der Funktion *create()* der Klasse Auftragsbestaetigung
 - Shop verschickt Auftragsbestätigung per E-Mail an den Kunden.

Sequenzdiagramm

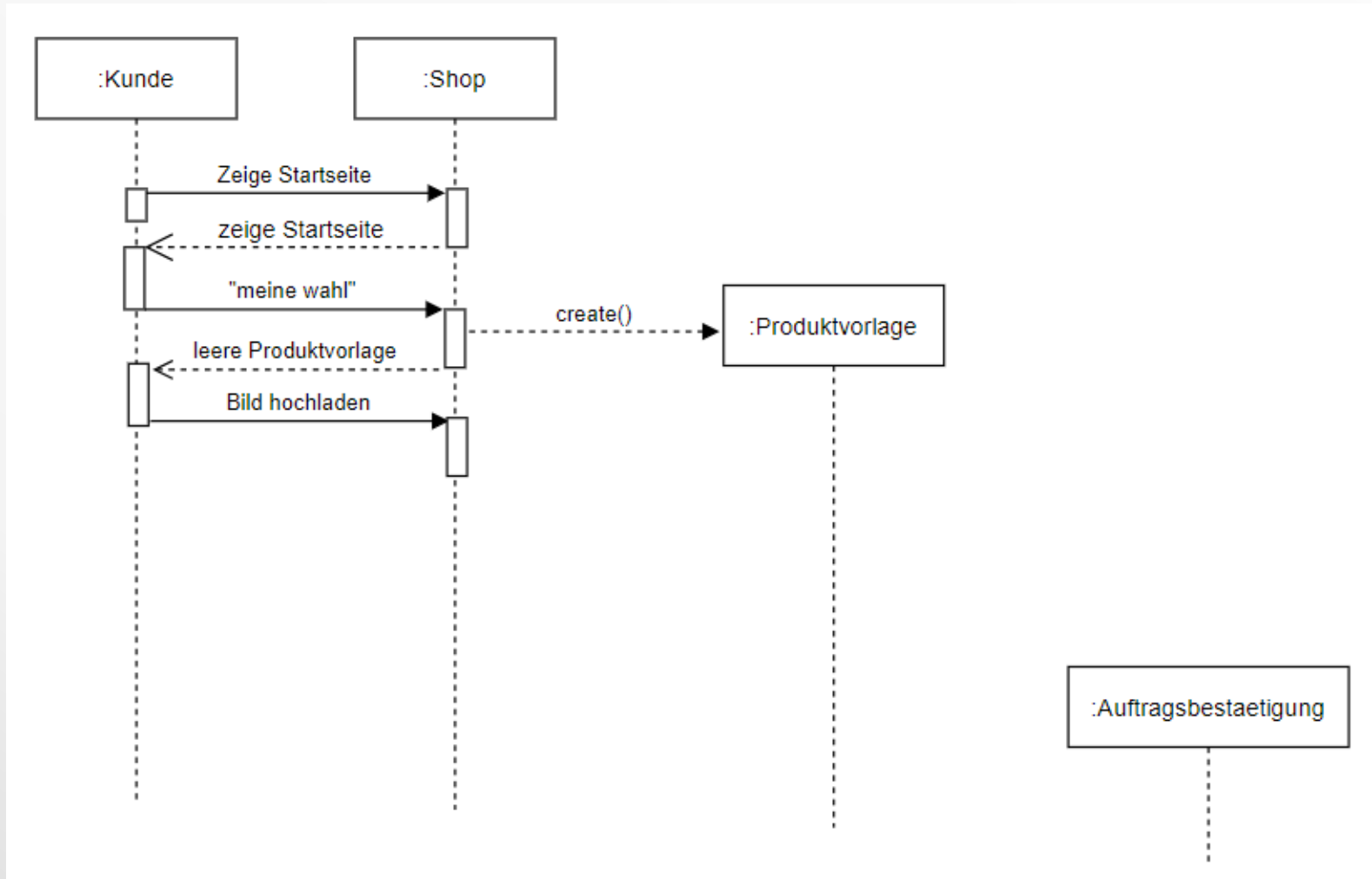


Prüfungsaufgabe

Die CARMED GmbH verwendet zur Konstruktion und Dokumentation von Software- und IT-Systemen die Unified Modeling Language (UML)

- a. Der Ablauf einer Bestellung im Onlineshop der FoPo AG soll in einem UML-Sequenzdiagramm dargestellt werden. Der Ablauf wird wie folgt beschrieben:
- Kunde ruft die Startseite mit dem Angebot des Shops auf.
 - Shop zeigt Kunden die Startseite mit den zur Wahl stehenden Produktarten (Foto, Poster).
 - Kunde teilt Shop Produktwahl mit.
 - Shop erstellt Produktvorlagen-Objekt; Aufruf der Funktion *create()* der entsprechenden Produktvorlagen-Klasse.
 - Shop zeigt dem Kunden auf HTML-Seite die leere Produktvorlage und die Mitteilung, dass das Bild hochgeladen werden kann.
 - Kunde lädt Bild hoch.
 - Shop übergibt Bild an Produktvorlage; Aufruf *bild()* des Produktvorlagen-Objekts.
 - Produktvorlage meldet „Bild ist eingefügt“.
 - Shop zeigt Kunden HTML-Seite mit fertiger Produktvorlage und fordert Kunden zur Eingabe der Adress- und Zahlungsdaten auf.
 - Kunde übermittelt Daten.
 - Shop ruft eigene Funktion *pruefung()* zur Prüfung der Daten auf.
 - Funktion *pruefung()* gibt „Daten o.k.“ zurück. Der Fall „Daten nicht o.k.“ wird nicht betrachtet.
 - Shop teilt Kunden mit, dass er zur Annahme des Auftrags bereit ist.
 - Kunde erteilt Auftrag.
 - Shop erstellt ein Auftragsbestätigungs-Objekt mit dem Aufruf der Funktion *create()* der Klasse Auftragsbestaetigung
 - Shop verschickt Auftragsbestätigung per E-Mail an den Kunden.

Sequenzdiagramm

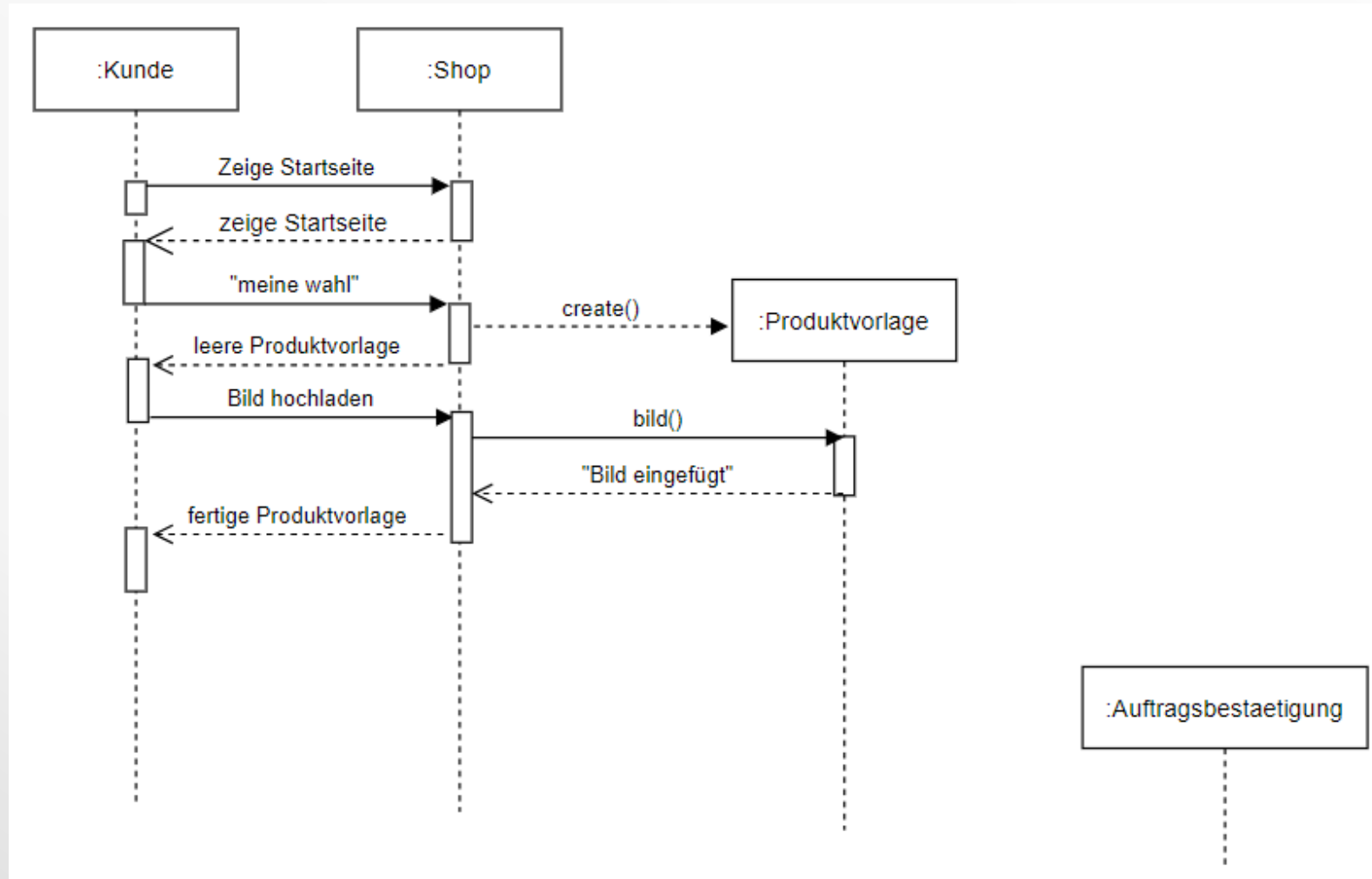


Prüfungsaufgabe

Die CARMED GmbH verwendet zur Konstruktion und Dokumentation von Software- und IT-Systemen die Unified Modeling Language (UML)

- a. Der Ablauf einer Bestellung im Onlineshop der FoPo AG soll in einem UML-Sequenzdiagramm dargestellt werden. Der Ablauf wird wie folgt beschrieben:
- Kunde ruft die Startseite mit dem Angebot des Shops auf.
 - Shop zeigt Kunden die Startseite mit den zur Wahl stehenden Produktarten (Foto, Poster).
 - Kunde teilt Shop Produktwahl mit.
 - Shop erstellt Produktvorlagen-Objekt; Aufruf der Funktion *create()* der entsprechenden Produktvorlagen-Klasse.
 - Shop zeigt dem Kunden auf HTML-Seite die leere Produktvorlage und die Mitteilung, dass das Bild hochgeladen werden kann.
 - Kunde lädt Bild hoch.
 - Shop übergibt Bild an Produktvorlage; Aufruf *bild()* des Produktvorlagen-Objekts.
 - Produktvorlage meldet „Bild ist eingefügt“.
 - Shop zeigt Kunden HTML-Seite mit fertiger Produktvorlage und fordert Kunden zur Eingabe der Adress- und Zahlungsdaten auf.
 - Kunde übermittelt Daten.
 - Shop ruft eigene Funktion *pruefung()* zur Prüfung der Daten auf.
 - Funktion *pruefung()* gibt „Daten o.k.“ zurück. Der Fall „Daten nicht o.k.“ wird nicht betrachtet.
 - Shop teilt Kunden mit, dass er zur Annahme des Auftrags bereit ist.
 - Kunde erteilt Auftrag.
 - Shop erstellt ein Auftragsbestätigungs-Objekt mit dem Aufruf der Funktion *create()* der Klasse Auftragsbestaetigung
 - Shop verschickt Auftragsbestätigung per E-Mail an den Kunden.

Sequenzdiagramm

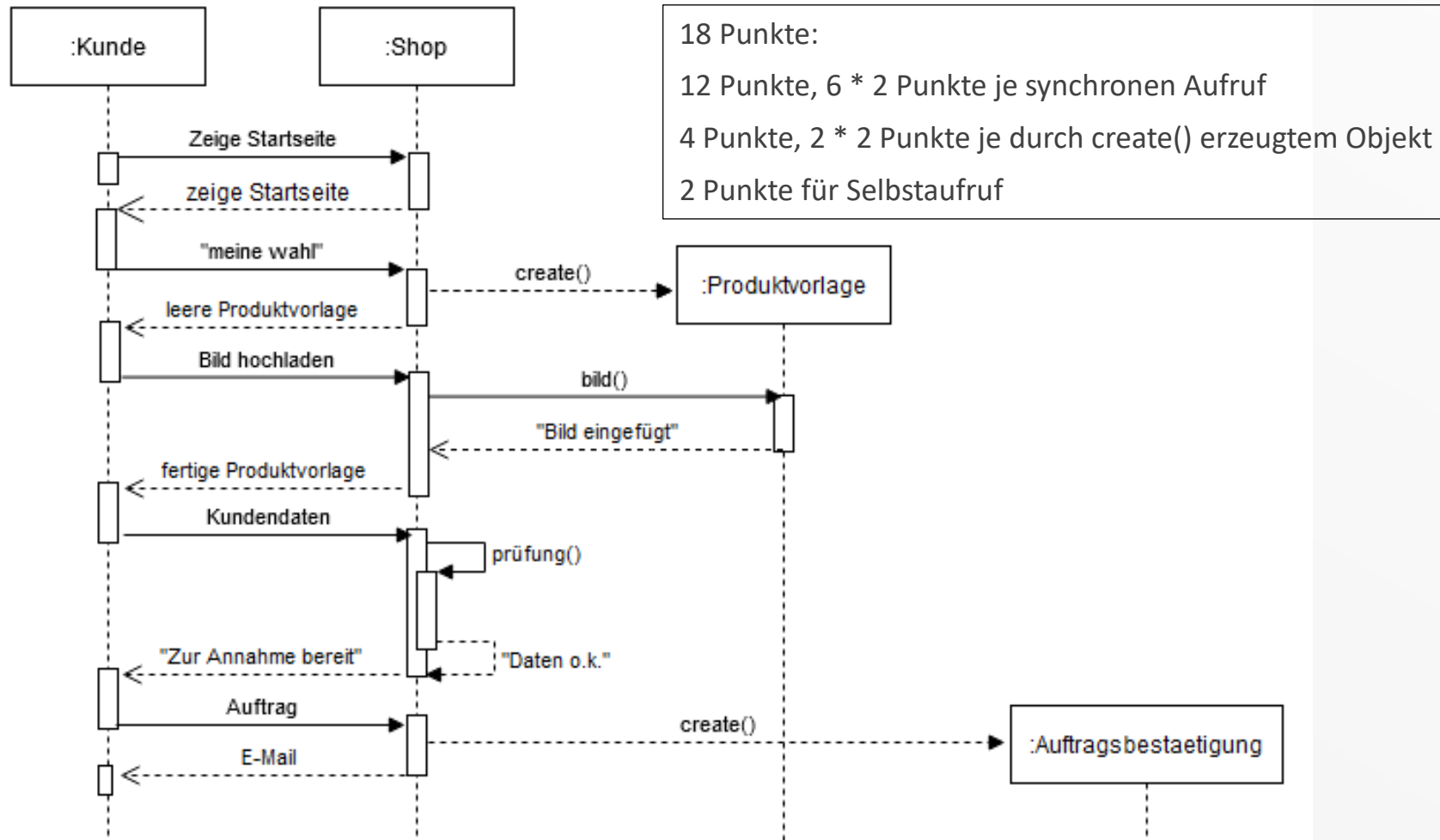


Prüfungsaufgabe

Die CARMED GmbH verwendet zur Konstruktion und Dokumentation von Software- und IT-Systemen die Unified Modeling Language (UML)

- a. Der Ablauf einer Bestellung im Onlineshop der FoPo AG soll in einem UML-Sequenzdiagramm dargestellt werden. Der Ablauf wird wie folgt beschrieben:
- Kunde ruft die Startseite mit dem Angebot des Shops auf.
 - Shop zeigt Kunden die Startseite mit den zur Wahl stehenden Produktarten (Foto, Poster).
 - Kunde teilt Shop Produktwahl mit.
 - Shop erstellt Produktvorlagen-Objekt; Aufruf der Funktion *create()* der entsprechenden Produktvorlagen-Klasse.
 - Shop zeigt dem Kunden auf HTML-Seite die leere Produktvorlage und die Mitteilung, dass das Bild hochgeladen werden kann.
 - Kunde lädt Bild hoch.
 - Shop übergibt Bild an Produktvorlage; Aufruf *bild()* des Produktvorlagen-Objekts.
 - Produktvorlage meldet „Bild ist eingefügt“.
 - Shop zeigt Kunden HTML-Seite mit fertiger Produktvorlage und fordert Kunden zur Eingabe der Adress- und Zahlungsdaten auf.
 - Kunde übermittelt Daten.
 - Shop ruft eigene Funktion *pruefung()* zur Prüfung der Daten auf.
 - Funktion *pruefung()* gibt „Daten o.k.“ zurück. Der Fall „Daten nicht o.k.“ wird nicht betrachtet.
 - Shop teilt Kunden mit, dass er zur Annahme des Auftrags bereit ist.
 - Kunde erteilt Auftrag.
 - Shop erstellt ein Auftragsbestätigungs-Objekt mit dem Aufruf der Funktion *create()* der Klasse Auftragsbestaetigung
 - Shop verschickt Auftragsbestätigung per E-Mail an den Kunden.

Sequenzdiagramm

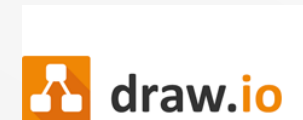


Fragen?

UML Tools

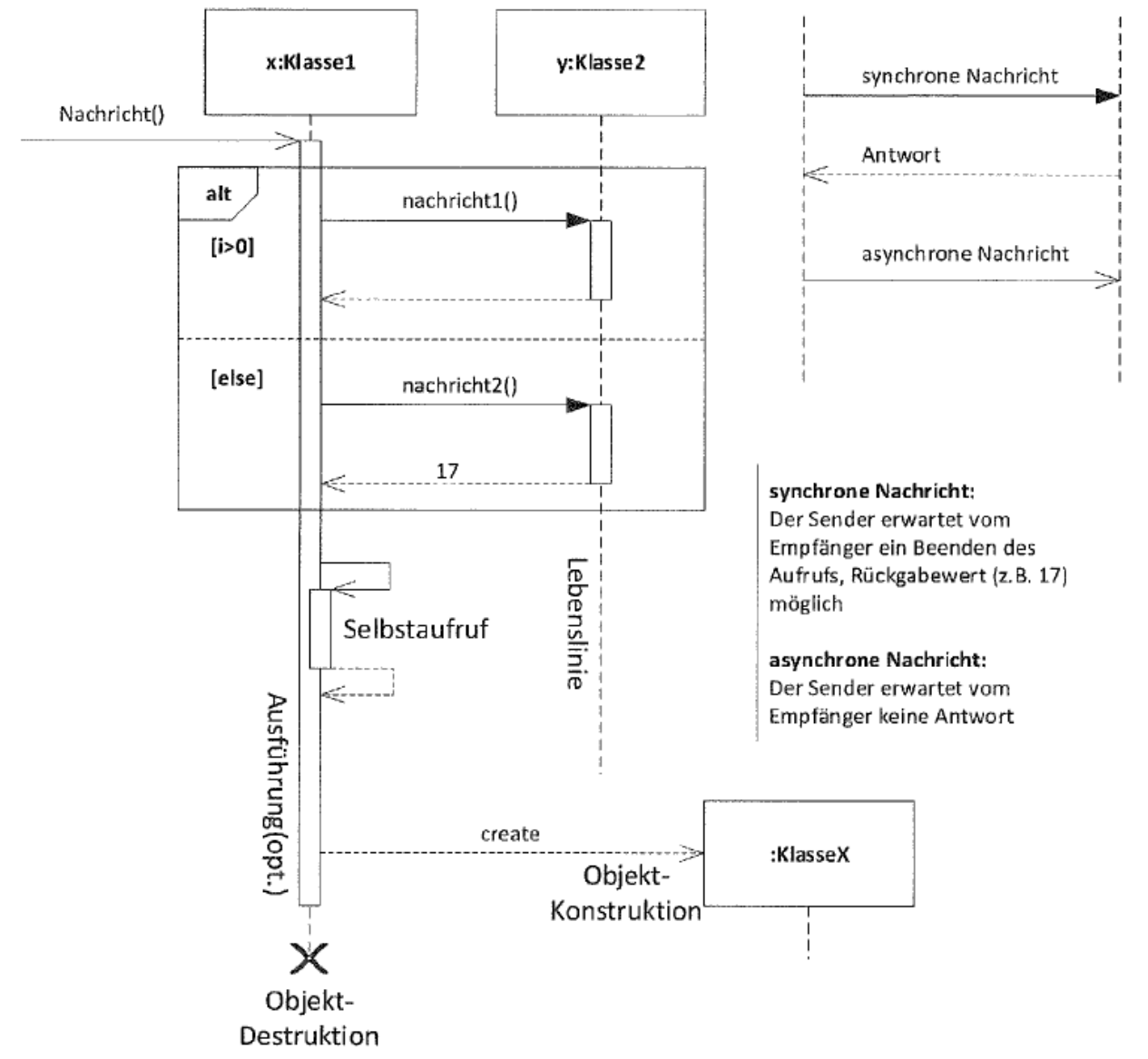


ArgoUML

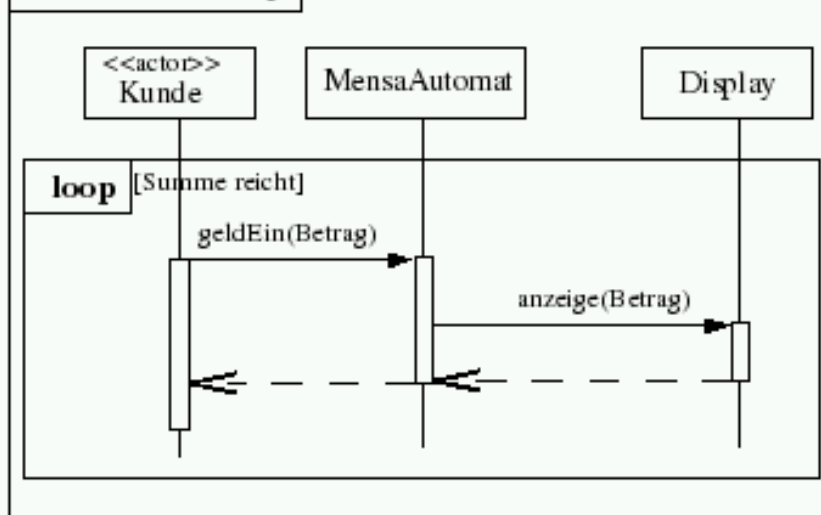


Sequenzdiagramm Zusammenfassung

Sequenzdiagramm



sd Schleife in SQ



Sequenzdiagramm

Style Guide

- Objekte sollten so angeordnet werden, dass in einem Diagramm möglichst wenig Kreuzungen entstehen.
- Nachrichten sollten von links nach rechts zeigen, Returns von rechts nach links.
- Nachrichtennamen sollten an der Spitze der Pfeile stehen.
- Beteiligte Akteure sollten am linken Rand der Diagramme stehen.
- Namen der Akteure sollten konsistent zu den Use Case Diagrammen sein.
- Klassennamen sollten konsistent zu dem Klassendiagramm sein.

