



Vienna University of Technology

Objektorientierte Modellierung

Sequenzdiagramm



Business Informatics Group

*Institute of Software Technology and Interactive Systems
Vienna University of Technology*

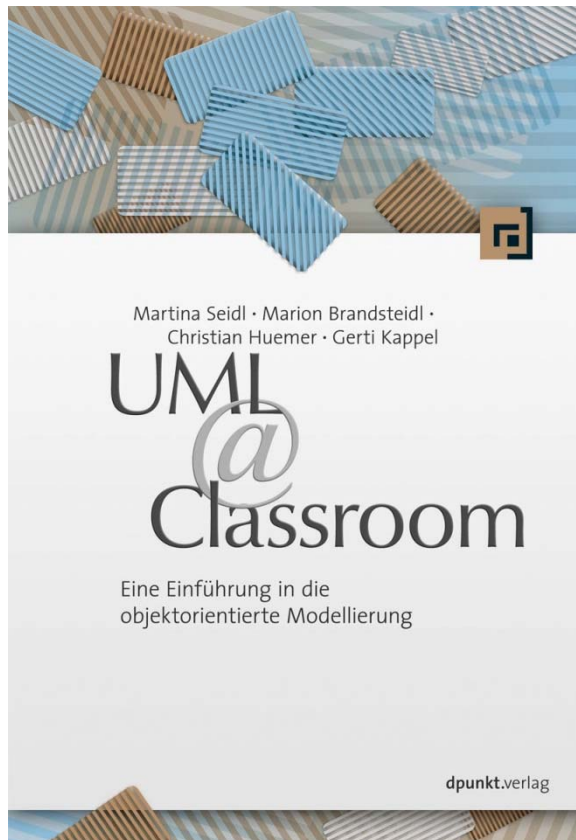
Favoritenstraße 9-11/188-3, 1040 Vienna, Austria

phone: +43 (1) 58801-18804 (secretary), fax: +43 (1) 58801-18896

office @big.tuwien.ac.at, www.big.tuwien.ac.at

Literatur

- Die Vorlesung basiert auf folgendem Buch:



UML @ Classroom:

Eine Einführung in die objekt- orientierte Modellierung

*Martina Seidl, Marion Brandsteidl,
Christian Huemer und Gerti Kappel*

dpunkt.verlag

Juli 2012

ISBN 3898647765

- *Anwendungsfalldiagramm*
- *Strukturmodellierung*
- *Zustandsdiagramm*
- ***Sequenzdiagramm***
- *Aktivitätsdiagramm*

Inhalt

- Interaktionen und Nachrichten
- Überblick Interaktionsdiagramme
- Basiselemente des Sequenzdiagramms
 - Diagrammrahmen
 - Lebenslinie
 - Nachrichten
 - Parameter, Lokale Attribute
- Zeiteinschränkungen und Zustandsinvarianten
- Kombinierte Fragmente
 - Verzweigungen und Schleifen
 - Nebenläufigkeit und Ordnung
 - Filterungen und Zusicherungen

Interaktionen und Nachrichten

- Interaktion
 - Zusammenspiel mehrerer Kommunikationspartner
 - Nachrichten- und Datenaustausch
- Interaktionen durch
 - Signale
 - Operationsaufrufe
 - Aufruf einer Operation einer Klasse
 - Antwort: Ergebnis der aufgerufenen Operation
- Steuerung der Interaktionen durch
 - Bedingungen
 - Zeitereignisse

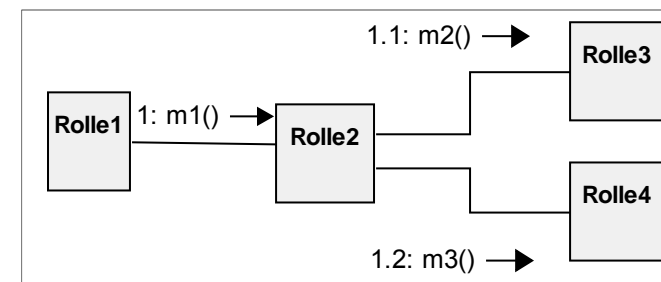
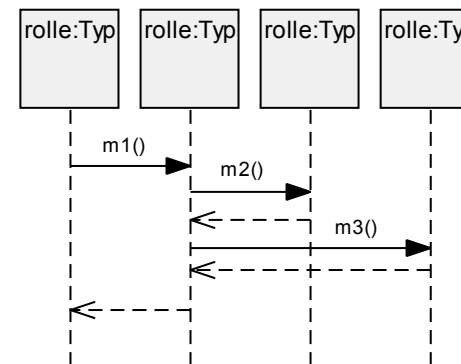
Interaktionsdiagramme

- Zeigen wie Nachrichten zwischen verschiedenen Interaktionspartnern in einem bestimmten Kontext ausgetauscht werden
- Beschreibung von Kommunikationssituationen durch:
 - Kommunikationspartner und deren Lebenslinien
 - Interaktionen
 - Nachrichten
 - Mittel zur Flusskontrolle
- Unterschiedliche Anforderungen und Betonung unterschiedlicher Aspekte
 - ⇒ 4 verschiedene Typen von Interaktionsdiagrammen

Interaktionsdiagramme – Arten (1/2)

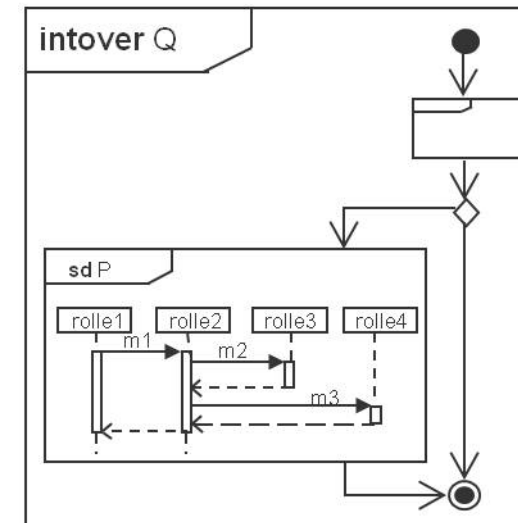
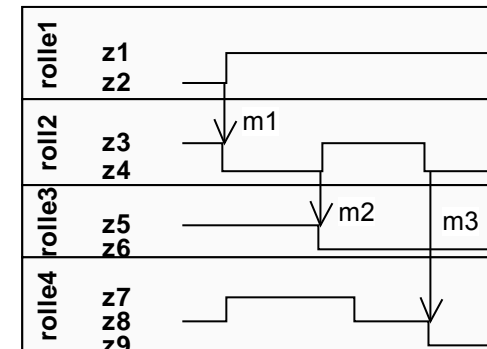
- Die 4 Arten von Interaktionsdiagrammen sind für einfache Interaktionen semantisch äquivalent
- Betonung unterschiedlicher Aspekte

- Sequenzdiagramm** zeigt den zeitlichen und logischen Nachrichtenfluss
 - Zeit ist eigene Dimension
- Kommunikationsdiagramm** ist »strukturell« orientiert
 - Zeigt die Beziehungen zwischen Interaktionspartnern – Kontextaspekt
 - Reihenfolge von Nachrichten nur über Dezimalklassifikation ausgedrückt
 - Zeit ist keine eigene Dimension



Interaktionsdiagramme – Arten (2/2)

- **Zeitdiagramm** zeigt Zustandsänderungen der Interaktionspartner aufgrund von Zeitereignissen
 - Vertikale Dimension repräsentiert Interaktionspartner und ihre möglichen Zustände
 - Horizontale Dimension repräsentiert die Zeitachse
- **Interaktionsübersichtsdiagramm** zeigt das Zusammenspiel von verschiedenen Interaktionen
 - Visualisiert in welcher Reihenfolge und unter welchen Bedingungen Interaktionsabläufe stattfinden

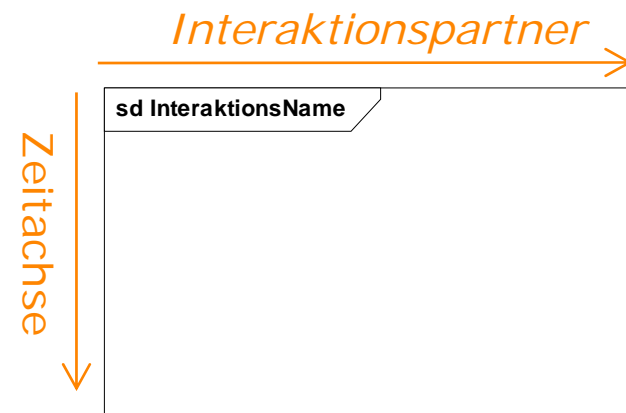


Einsatzbereiche

- Modellierung der **Interaktionen eines Systems mit seiner Umwelt** (Systemgrenzen festlegen, System als Black-Box)
- Modellierung der **Realisierung eines Anwendungsfalls**
- Modellierung des **Zusammenspiels der internen Struktur** einer Klasse, Komponente oder Kollaboration
- Modellierung der Spezifikation von **Schnittstellen zwischen Systemteilen** (Zusammenspiel angebotene/benutzte Schnittstelle)
- Modellierung der **Operationen einer Klasse**

Sequenzdiagramm

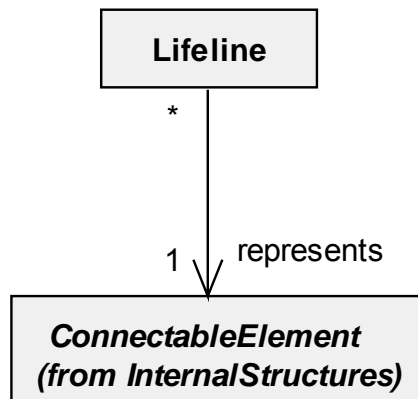
- Darstellung von Interaktionen in **2 Dimensionen**:
 - "**horizontal**": Interaktionspartner in Form von Rollen
Reihenfolge der Partner wird für eine möglichst übersichtliche Darstellung gewählt
 - "**vertikal**": Zeitachse
Darstellung des zeitlichen Ablaufs der Kommunikation
- wichtigste Notationselemente:
 - **Lebenslinien**: Kommunikationspartner
 - **Nachrichten**: Pfeile



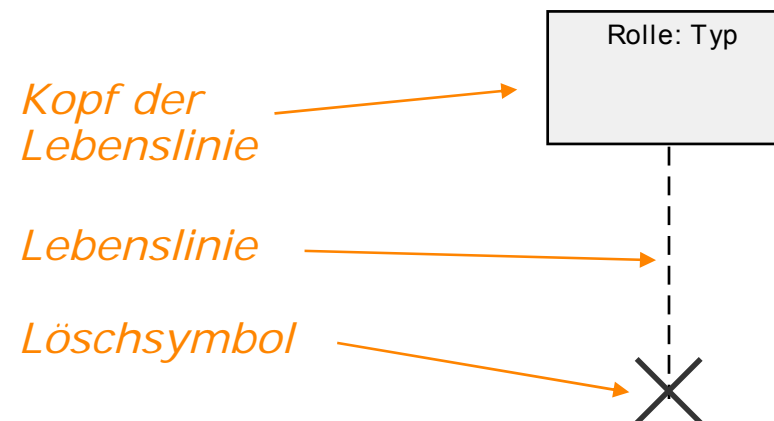
Lebenslinie

- Eine **Lebenslinie** beschreibt **genau einen Interaktionspartner**
- Als Interaktionspartner können alle Rollen des Kontext-Classifiers auftreten
 - Rollen sind vom Typ `ConnectableElement` (z.B. Klassen, Attribute oder Ports)

- **Metamodell**



- **Notation im Sequenzdiagramm**

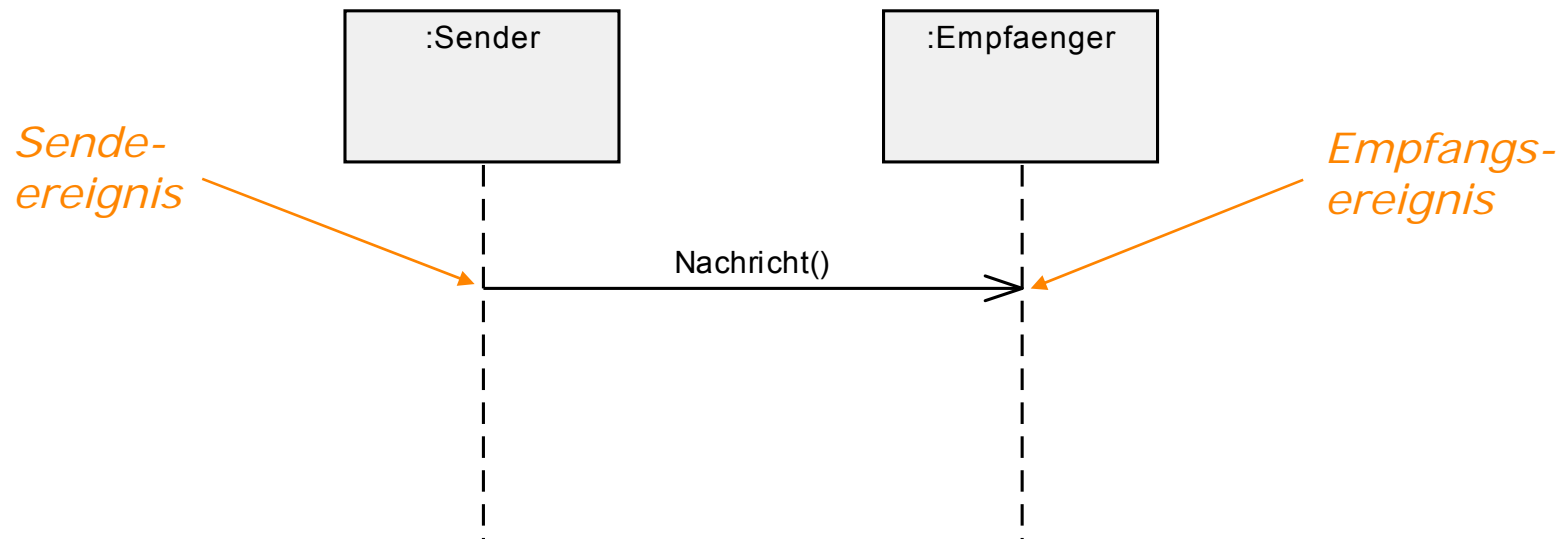


Typ- vs. Instanzebene

- Modellierung des Nachrichtenaustauschs zwischen Rollen und damit prinzipiell auf **Rollenebene**
 - Kontext der Interaktion durch strukturierte Classifier festgelegt = Kontext-Classifier
 - Die Rollen der Classifier stellen die Interaktionspartner dar
 - Tatsächliche Interaktion findet selbstverständlich auf Instanzebene zwischen Objekten statt
- Modellierung auf **Instanzebene** möglich, um eine Abfolge von Nachrichten zwischen konkreten Objekten darzustellen = Trace

Lebenslinie: Ereignisspezifikation (1/2)

- Interaktionen werden als **Folge von Ereignisspezifikationen** auf Lebenslinien betrachtet
- Beispiel für Ereignisspezifikationen
 - **Senden** und **Empfangen** von Nachrichten auf verschiedenen Lebenslinien oder der gleichen Lebenslinie

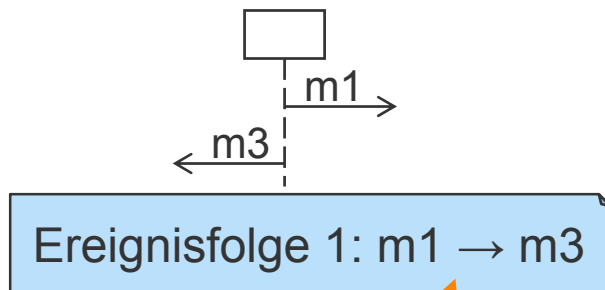


Lebenslinie: Ereignisspezifikation (2/2)

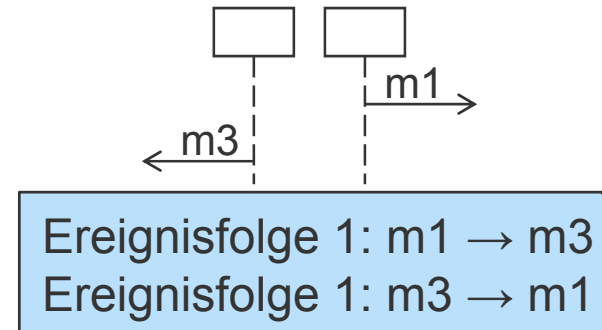
- Reihenfolge von Ereignisspezifikationen
 - Vertikale Zeitachse bestimmt nur die Ordnung der Ereigniseintritte pro Lebenslinie
 - Jedoch nicht die Reihenfolge von Ereigniseintritten auf verschiedenen Lebenslinien
 - Erst durch **Nachrichten zwischen Lebenslinien** wird eine Ordnung über Lebenslinien hinweg erzwungen

Lebenslinie: Reihenfolge von Ereignisseintritten

... auf einer Lebenslinie

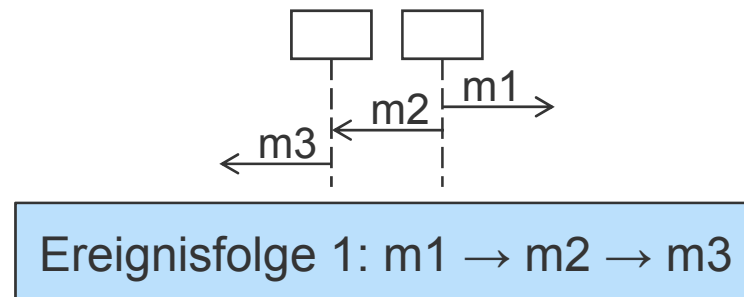


... auf verschiedenen Lebenslinien



»zeitlich vor«

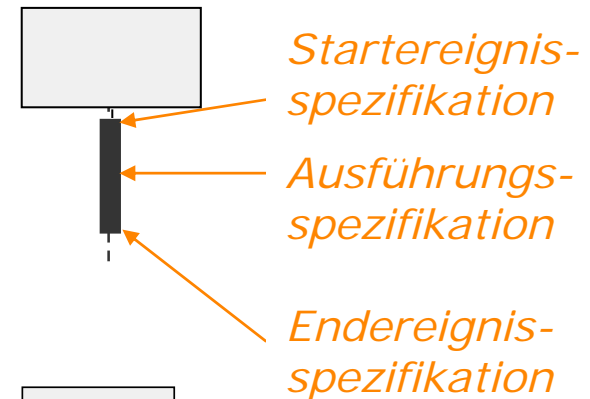
... auf verschiedenen Lebenslinien,
verbunden durch Nachrichtenaustausch



Lebenslinie: Ausführungsspezifikation

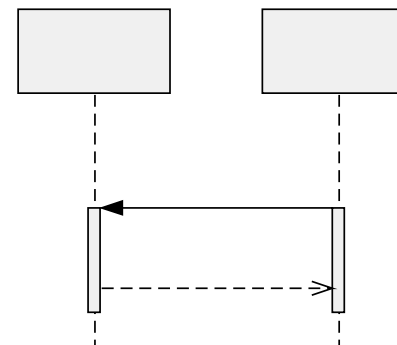
- Die **Ausführung** einer Aktivität/Operation wird durch zwei Ereignisspezifikationen (Start und Ende) auf der gleichen Lebenslinie definiert

- Diese sogenannte Ausführungsspezifikation kann durch einen Balken dargestellt werden



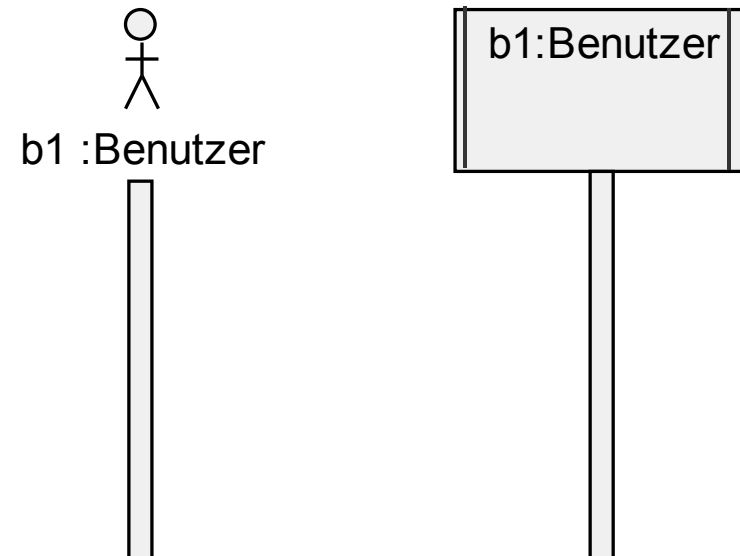
- Ausführungsarten**

- Direkt**
 - Interaktionspartner führt Verhalten selbst aus
- Indirekt**
 - Ausführung wird an andere Interaktionspartner delegiert



Lebenslinie: Aktives Objekt

- **Aktive Objekte** verfügen über **eigenen Kontrollfluss** (Prozess oder Thread)
- Können **unabhängig** von anderen Objekten operieren
- **Notation**
 - Kopf der Lebenslinie wird links und rechts mit doppeltem Rand versehen
 - durchgehender Balken über gesamte Lebenslinie



Nachricht

- Arten der Kommunikation

- Synchrone Kommunikation**

- Der Sender wartet bis zur Beendigung der Interaktion, die durch die Nachricht ausgelöst wurde

m1(p1,p2)
→

- Asynchrone Kommunikation**

- Die Nachricht wird als Signal betrachtet
 - Der Sender wartet nicht auf das Ende der Interaktion

m2
→

- Antwortnachricht (optional)**

- att**: Name eines Attributs, dem der Rückgabewert zugewiesen werden soll

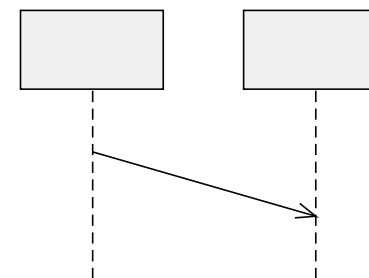
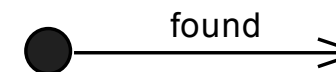
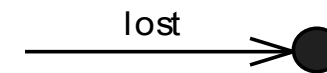
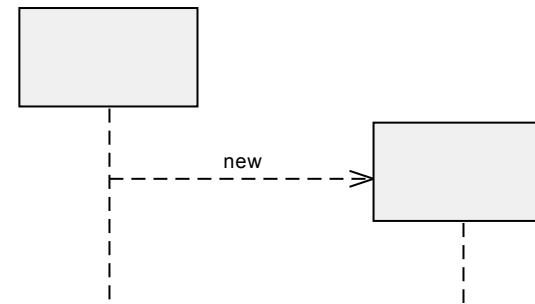
- m1**: Name der Nachricht, auf die geantwortet wird

- wert**: Rückgabewert

att=m1:wert
←-----

Nachricht: Spezielle Nachrichtenarten

- **Objekterzeugung**
 - Ermöglicht, einen Interaktionspartner erst im Laufe der Interaktion zu erzeugen
- **Verlorene Nachricht**
 - Senden einer Nachricht an unbekanntes oder nicht relevanten Interaktionspartner
- **Gefundene Nachricht**
 - Empfang einer Nachricht von einem unbekanntes oder nicht relevanten Interaktionspartner
- **Zeitkonsumierende Übertragung**



Basiskonzepte – Parameter, lokale Attribute

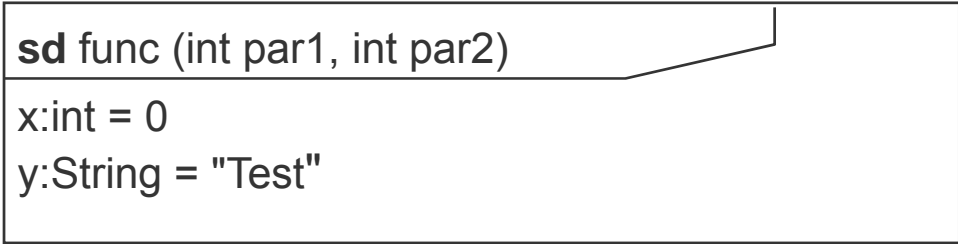
- Darstellung von Parametern und lokalen Attributen
- Beispiel: Modellierung der Operation func:

```
void func (int par1, int par2)
{
  int x = 0;
  String y = "Test";
  ...
}
```

Parameter

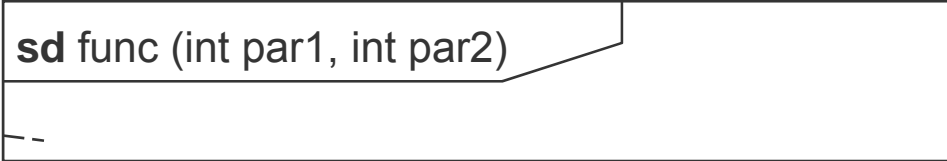
Variante 1:

lokale Attribute



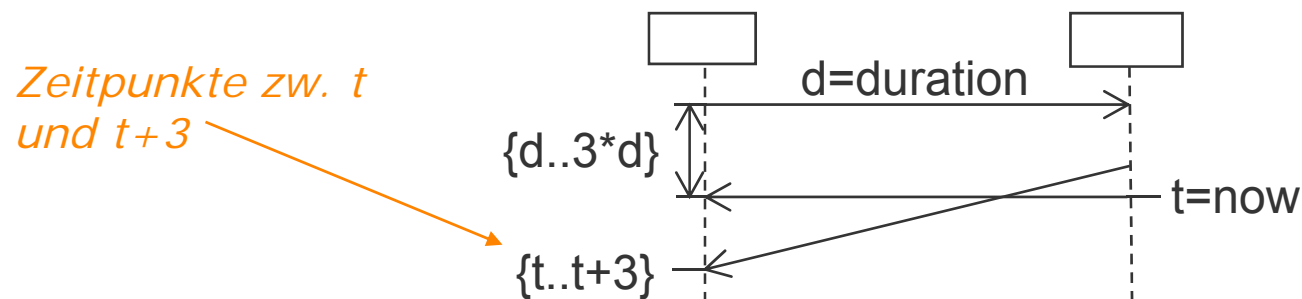
Variante 2:

x:int = 0
y:String = "Test"



Zeiteinschränkungen

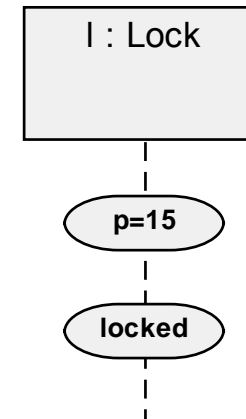
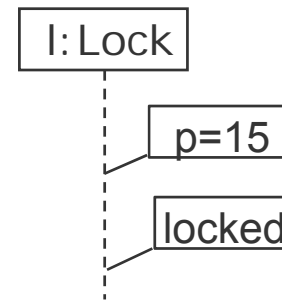
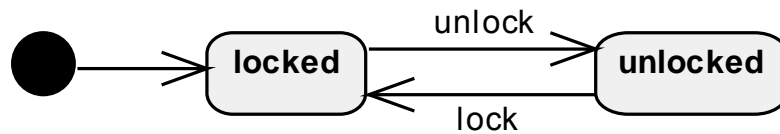
- Arten
 - **Zeitpunkt** (time constraint)
 - Bezieht sich auf einzelne Ereignisspezifikationen
relativ: z.B. after(5sec); absolut: z.B. at(12.00)
 - **Zeitdauer** (duration constraint)
 - Bezieht sich auf Zeitintervall zwischen zwei Ereigniseintritten
z.B. {12.00 .. 13.00}
- Vordefinierte Aktionen zur Zeitberechnung
 - **now**: Berechnung der aktuellen Zeit
 - **duration**: Berechnung einer Zeitdauer
 - Erhaltene Werte müssen Variablen zugewiesen werden
 - Variablen können in Zeitausdrücken verwendet werden



Zustandsinvariante

- **Zusicherung**, dass eine bestimmte **Bedingung** zu einem bestimmten Zeitpunkt **erfüllt** ist
- Bezieht sich immer auf eine bestimmte Lebenslinie
- Wird vor Eintritt des darauf folgenden Ereignisses ausgewertet
- Falls Zustandsinvariante nicht erfüllt ist - Fehler
- Notationsvarianten – Beispiel: Schloss (Lock)

Zustände, die ein Schloss annehmen kann:

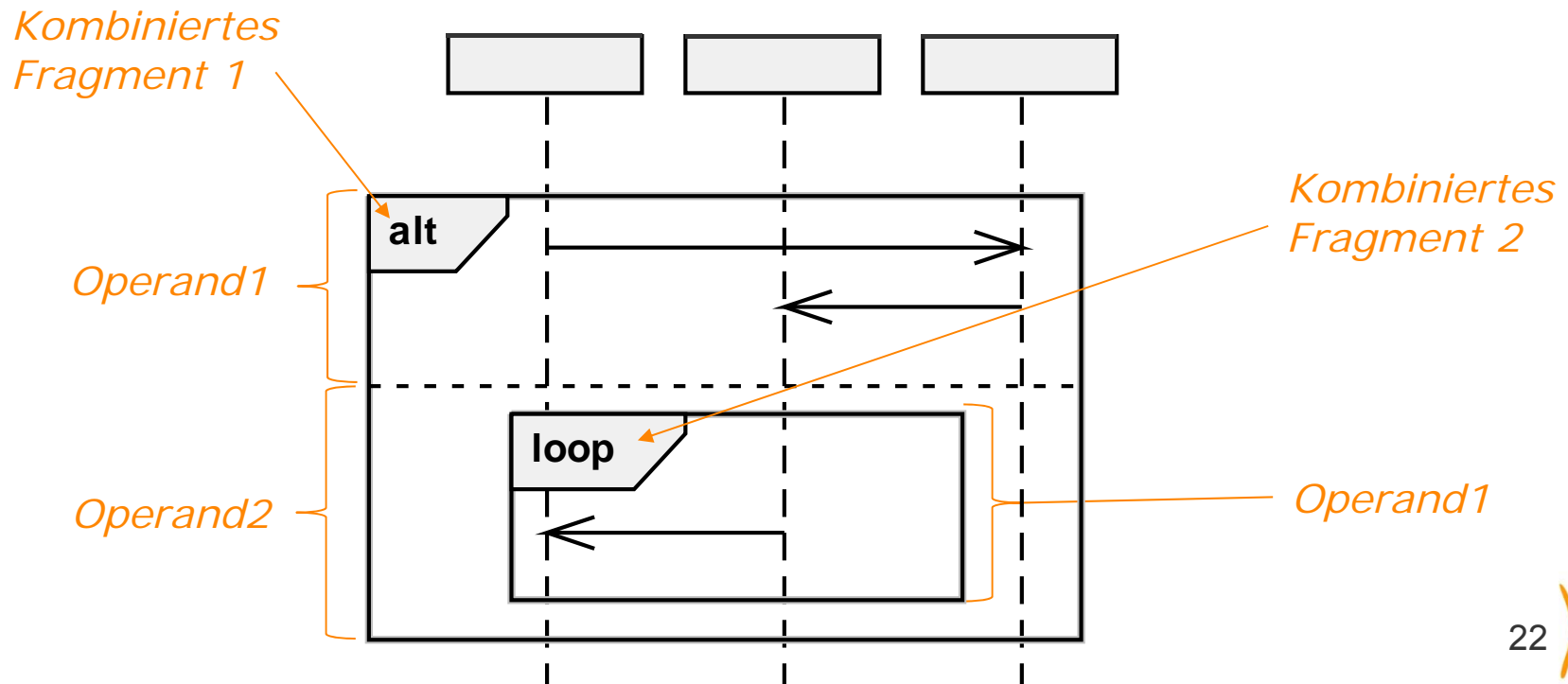


Kombinierte Fragmente

- Modellierung von **Kontrollstrukturen**
- Bestandteile: Operator und Operanden
- **Operator**
 - Definiert Art des kombinierten Fragments
 - 12 vordefinierte Operatoren
- **Operand**
 - Ein Operator enthält 1 oder mehrere Operanden, je nach Operatorart
 - Kann Interaktionen, kombinierte Fragmente (Schachtelung!) und Referenzen auf Sequenzdiagramme umfassen

Kombinierte Fragmente – Notation

- Kombiniertes Fragment wird wie Sequenzdiagramm mit Rahmen dargestellt
- Art des Fragments wird durch Operator im Pentagon festgelegt
 - default: seq
- Operanden werden durch gestrichelte Linien voneinander getrennt

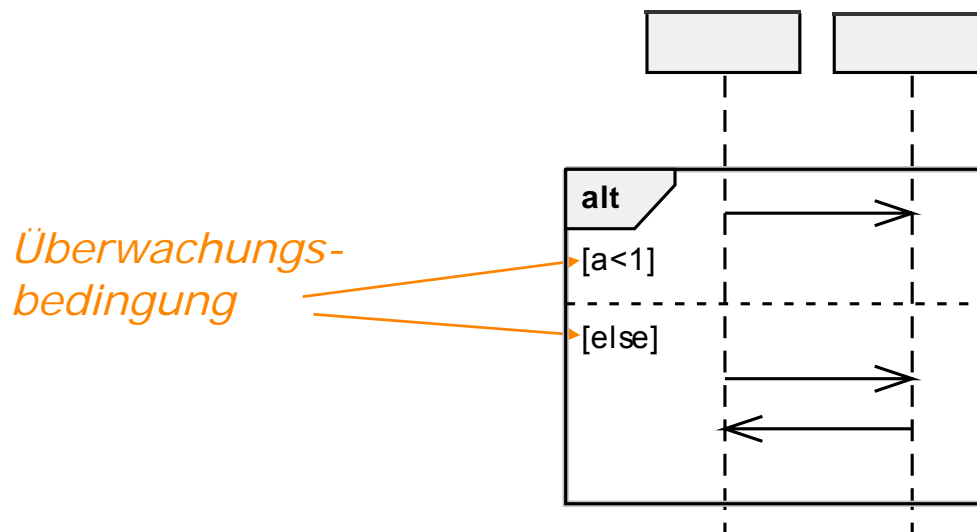


Kombinierte Fragmente – Operatorarten

	Operator	Zweck
Verzweigungen und Schleifen	alt	Alternative Interaktionen
	opt	Optionale Interaktionen
	break	Ausnahme Interaktionen
	loop	Iterative Interaktionen
Nebenläufigkeit und Ordnung	seq	Sequentielle Interaktionen mit schwacher Ordnung (Default-Operator)
	strict	Sequentielle Interaktionen mit strenger Ordnung
	par	Nebenläufige Interaktionen
	critical	Atomare Interaktionen
Filterungen und Zusicherungen	ignore	Irrelevante Interaktionen
	consider	Relevante Interaktionen
	assert	Zugesicherte Interaktionen
	neg	Ungültige Interaktionen

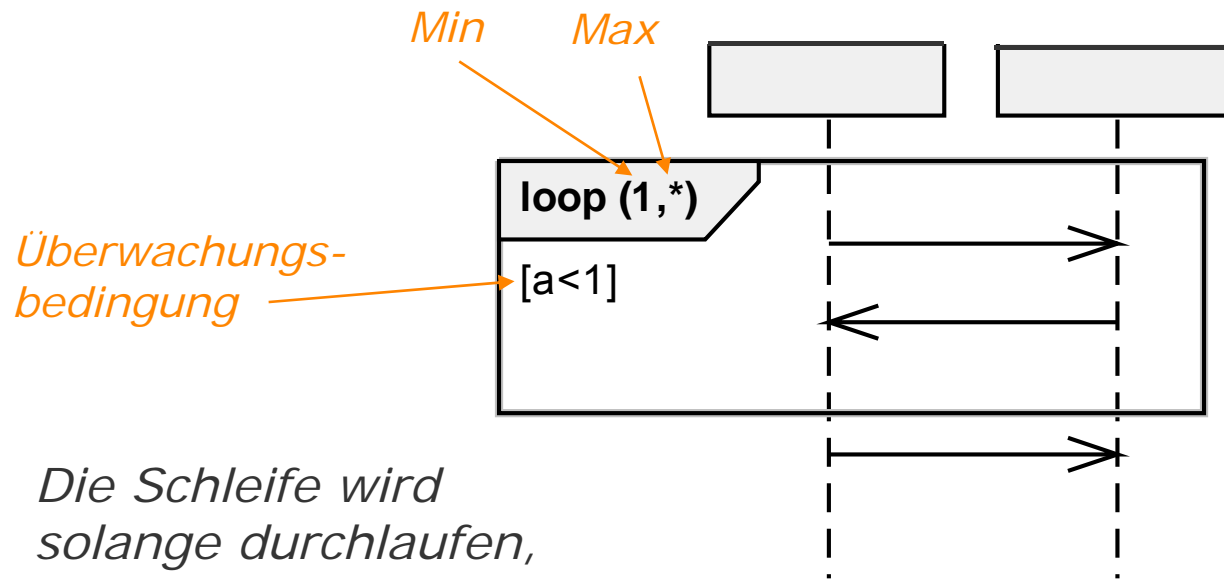
Verzweigungen und Schleifen: alt-Operator

- Darstellung von zwei oder mehreren **alternativen Interaktionsabläufen** (mind. 2)
- Zur Laufzeit wird maximal ein Operand ausgeführt
- Auswahl eines Operanden anhand von Überwachungsbedingungen
 - Boolescher Ausdruck in eckigen Klammern
 - Vordefinierte else-Bedingung: Operand wird ausgeführt, falls die Bedingungen aller anderen Operanden nicht erfüllt sind
 - default: true



Verzweigungen u. Schleifen: loop-Operator

- Darstellung einer **Schleife** über einen bestimmten Interaktionsablauf
 - Fragment enthält nur einen Operanden
 - Ausführungshäufigkeit wird durch Zähler mit Unter- und Obergrenze dargestellt
 - Optional: Überwachungsbedingung; wird bei jedem Durchlauf überprüft, sobald die minimale Anzahl an Durchläufen stattgefunden hat



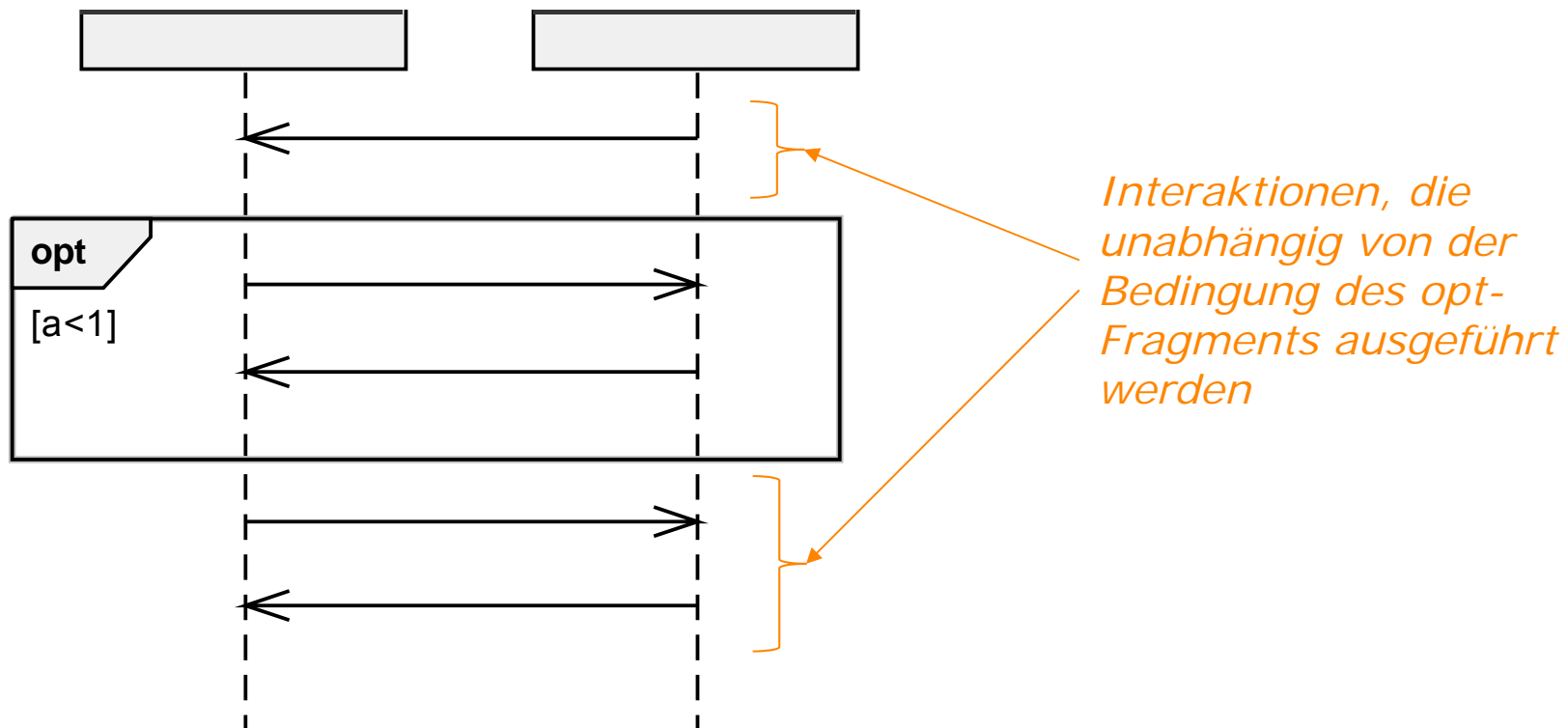
Notationsvarianten:

$loop(3,8) = loop(3..8)$
 $loop(8,8) = loop(8)$
 $loop = loop(0,*)$

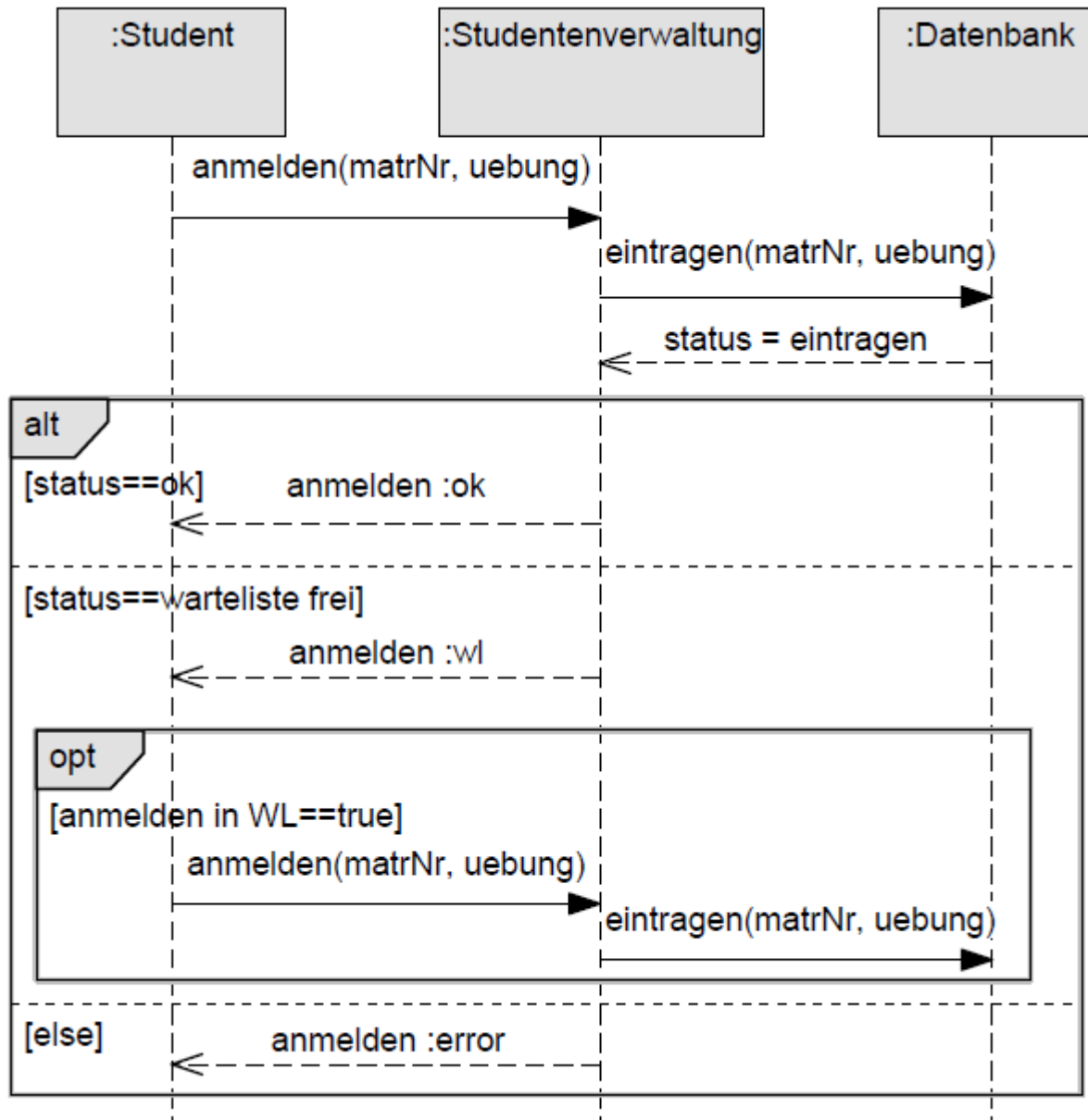
Die Schleife wird solange durchlaufen, bis $a \geq 1$

Verzweigungen u. Schleifen: opt-Operator

- **Optionale** Interaktionen
- Überwachungsbedingung steuert Durchlauf der Interaktionen
- Fragment wird nur aktiv, wenn Bedingung erfüllt ist
 - Modellierung von "wenn ..., dann ..."

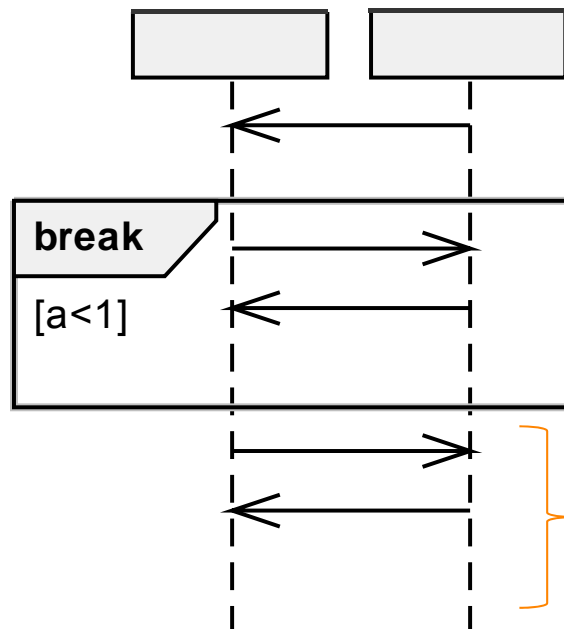


Opt-Operator - Beispiel



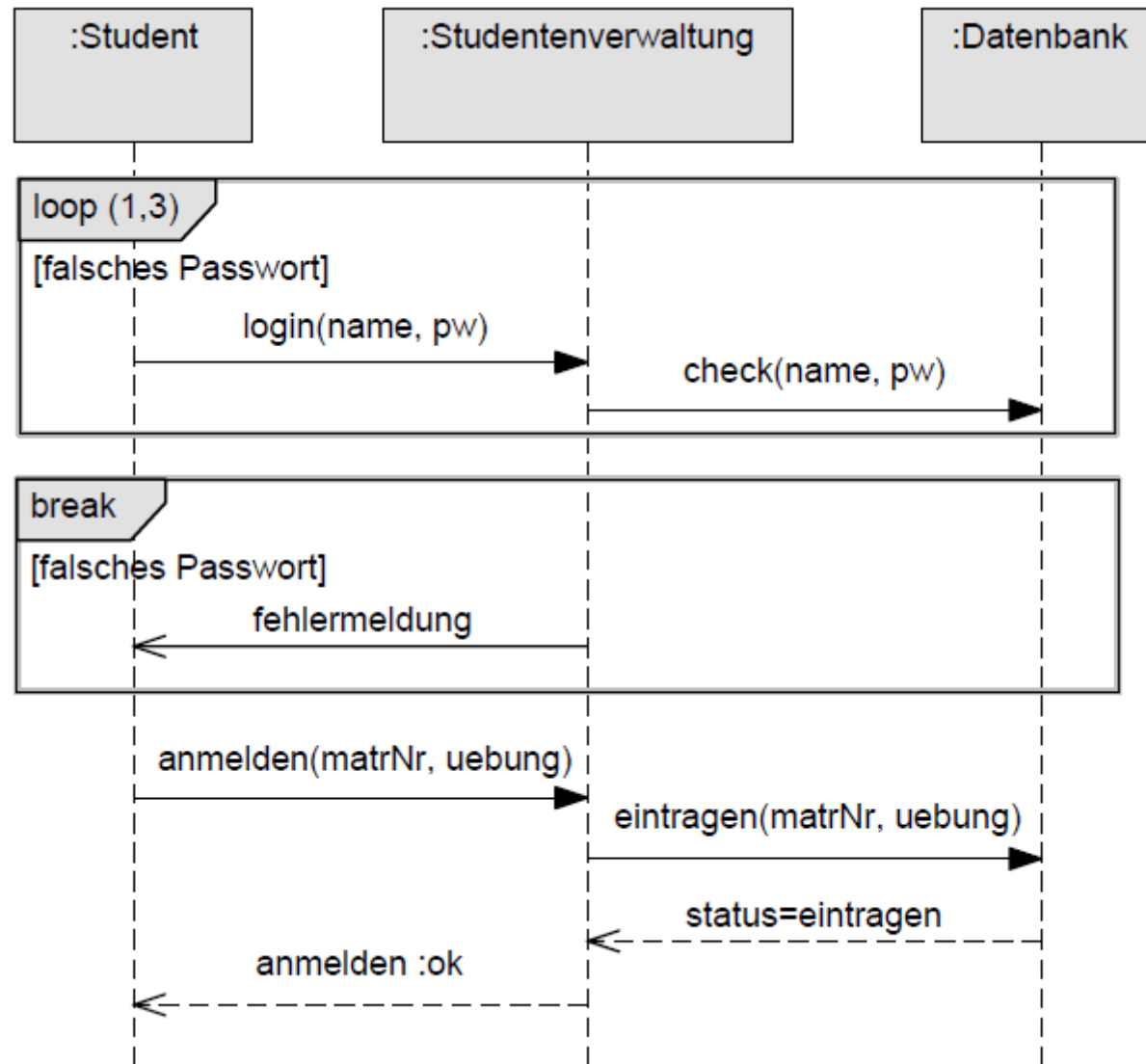
Verzweigungen u. Schleifen: break-Operator

- Ausnahme-Interaktionen
- Überwachungsbedingung steuert Durchlauf der Interaktionen
- Behandlung von Sonderfällen und Ausnahmen



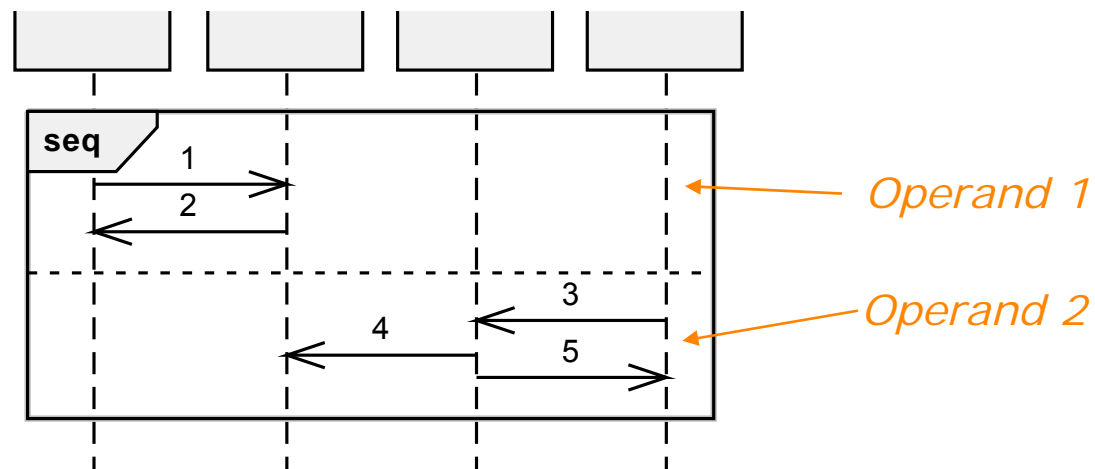
Interaktionen, die im Falle des break ($[a < 1]$) nicht ausgeführt werden

Break-Operator - Beispiel

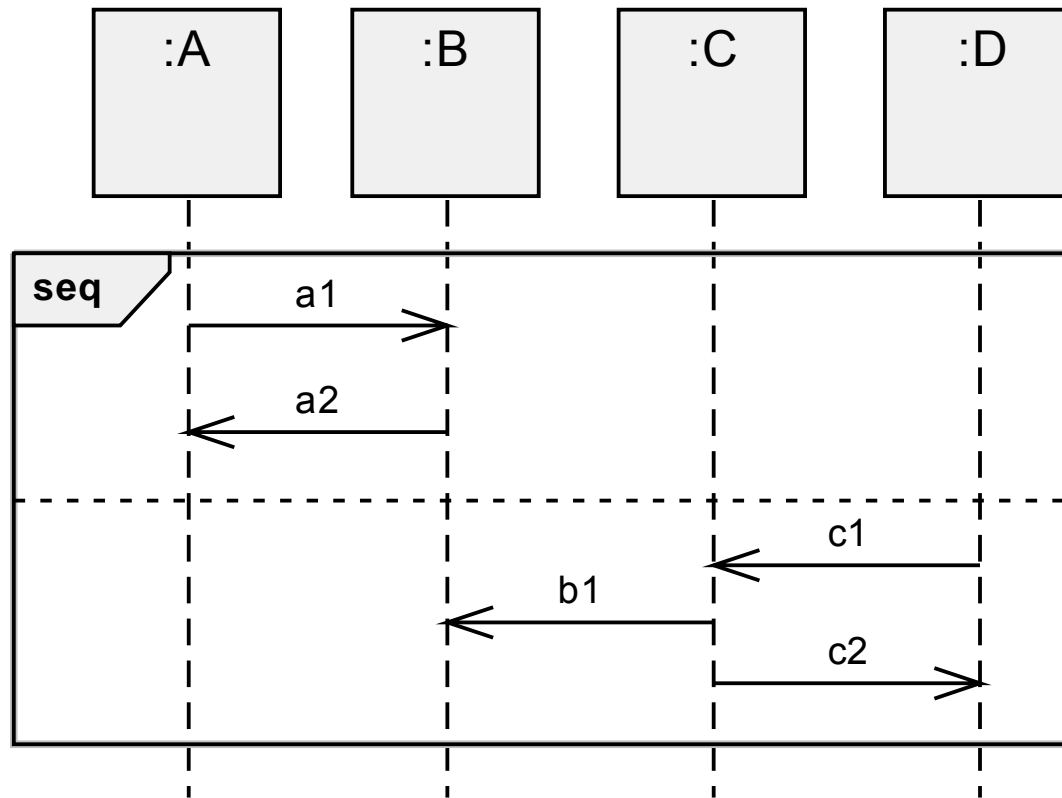


Nebenläufigkeit u. Ordnung: seq-Operator

- Sequentielle Interaktion mit schwacher Ordnung (default!)
- mind. 1 Operand
- Reihenfolge der Ereignisseintritte:
 - Reihenfolge der Ereignisse pro Lebenslinie gilt über Operandengrenze hinaus (Reihenfolge der Operanden im Diagramm ist relevant)
 - Reihenfolge auf unterschiedlichen Lebenslinien von unterschiedlichen Operanden ist nicht signifikant
 - Reihenfolge auf unterschiedlichen Lebenslinien in einem Operanden ist nur signifikant, wenn hier ein Nachrichtenaustausch stattfindet

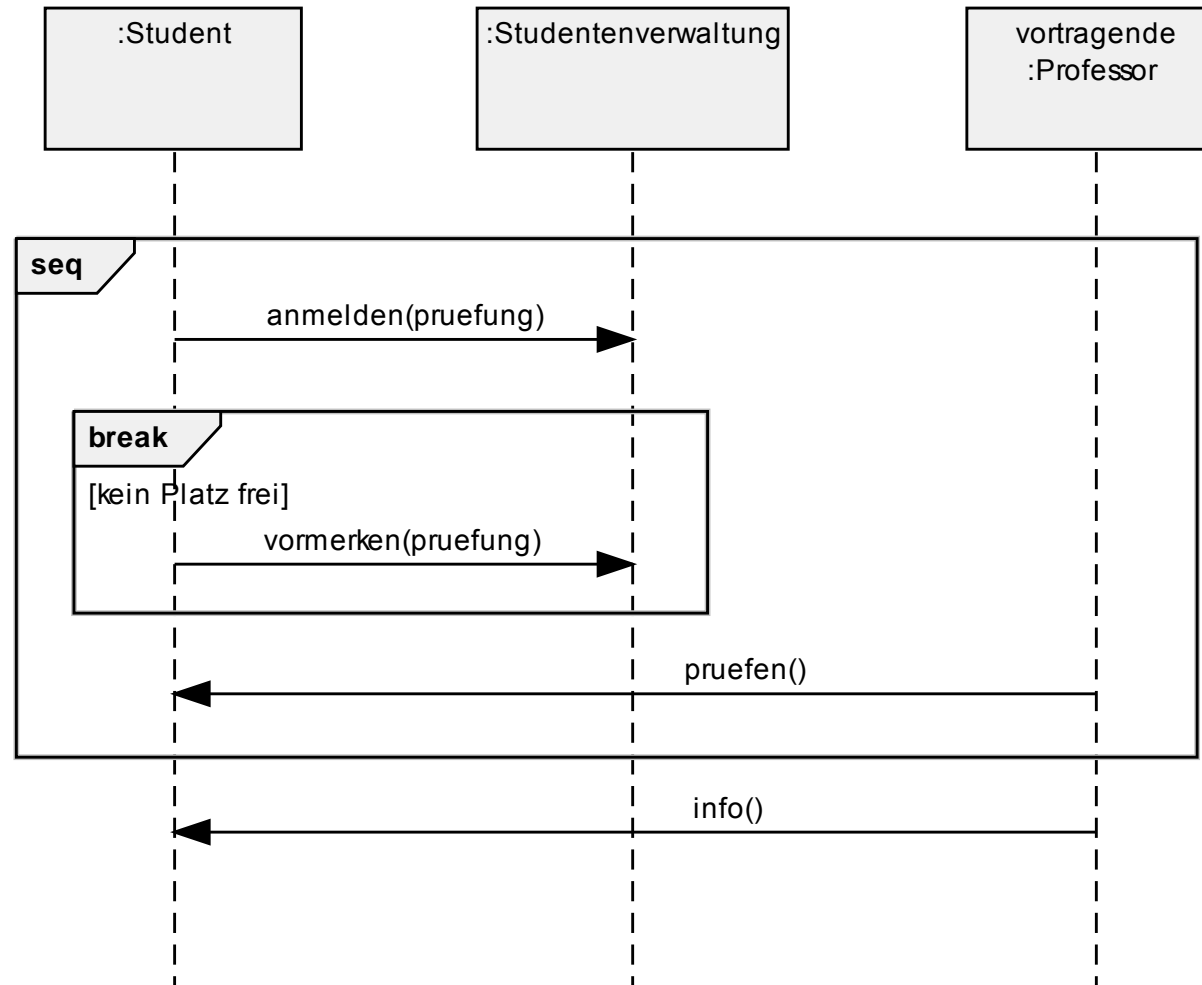


Nebenläufigkeit u. Ordnung: seq-Operator - Beispiel



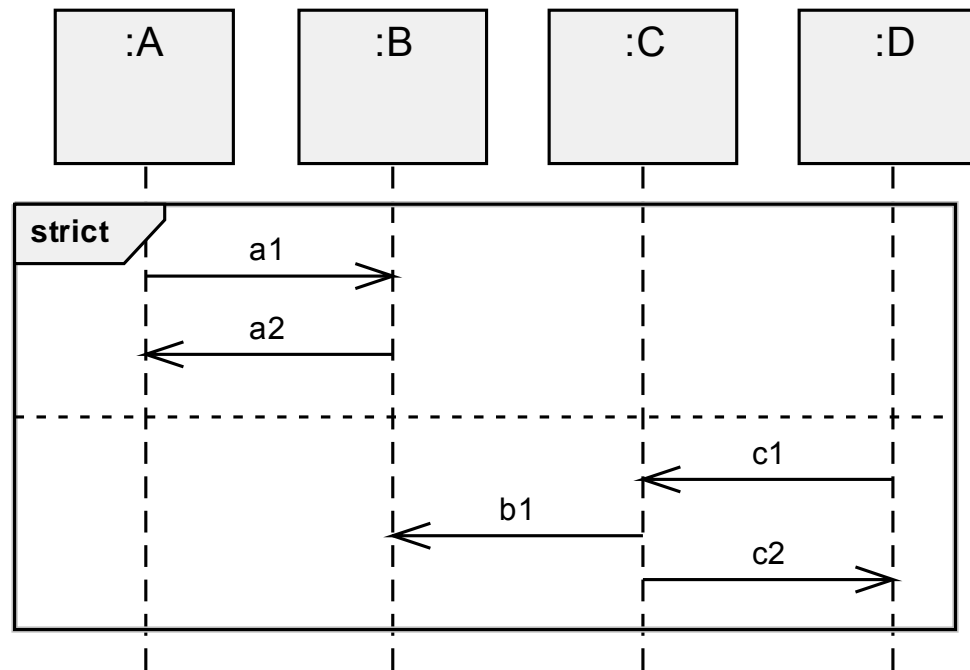
- Ereignisfolge 1: $a1 \rightarrow a2 \rightarrow c1 \rightarrow b1 \rightarrow c2$
- Ereignisfolge 2: $a1 \rightarrow c1 \rightarrow a2 \rightarrow b1 \rightarrow c2$
- Ereignisfolge 3: $c1 \rightarrow a1 \rightarrow a2 \rightarrow b1 \rightarrow c2$

Nebenläufigkeit u. Ordnung: seq-Operator – Beispiel (2/2)



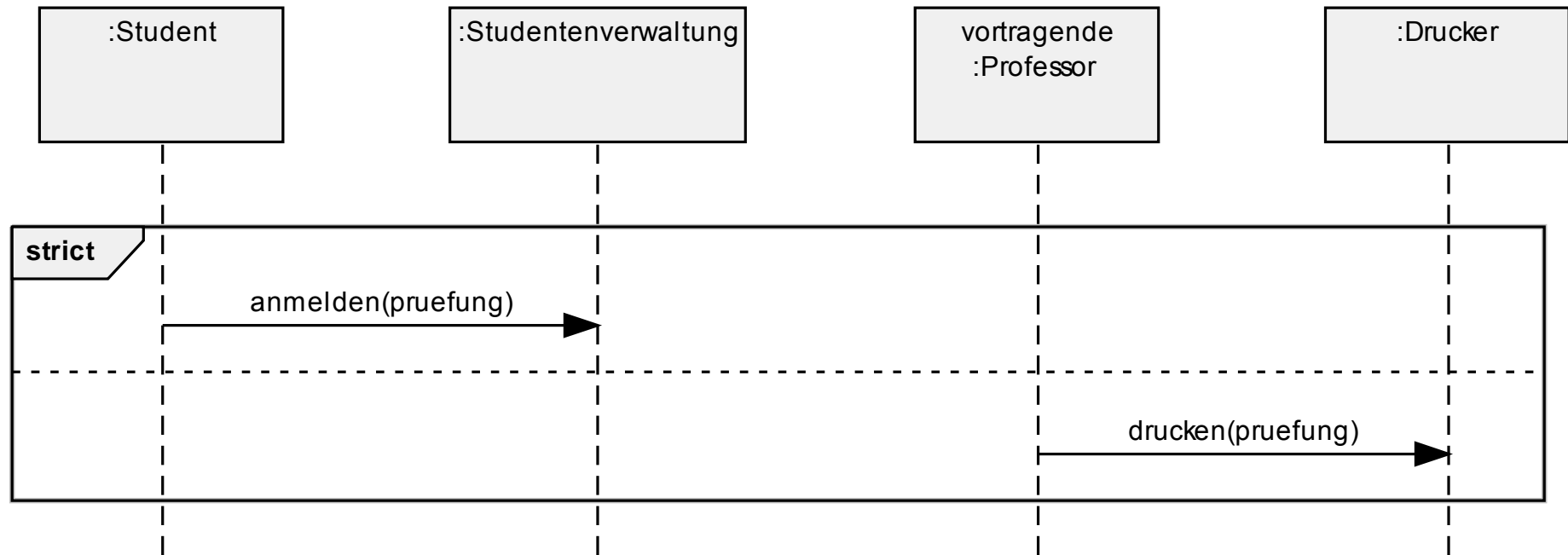
Nebenläufigkeit und Ordnung: strict-Operator

- Sequentielle Interaktion mit **strenger** Ordnung
- Reihenfolge auf unterschiedlichen Lebenslinien von unterschiedlichen Operanden ist signifikant



Ereignisfolge 1: a1 → a2 → c1 → b1 → c2

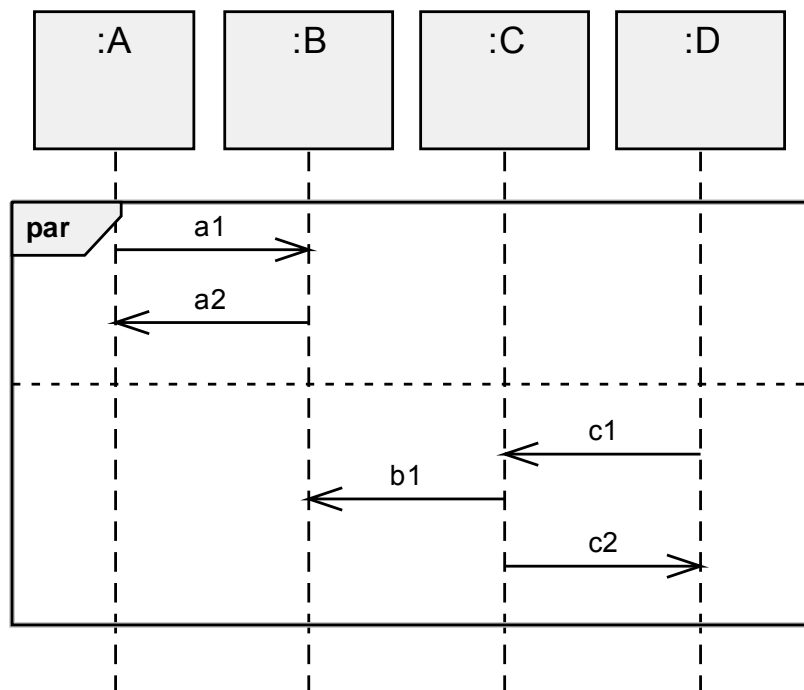
Strict-Operator - Beispiel



Nebenläufigkeit und Ordnung: par-Operator

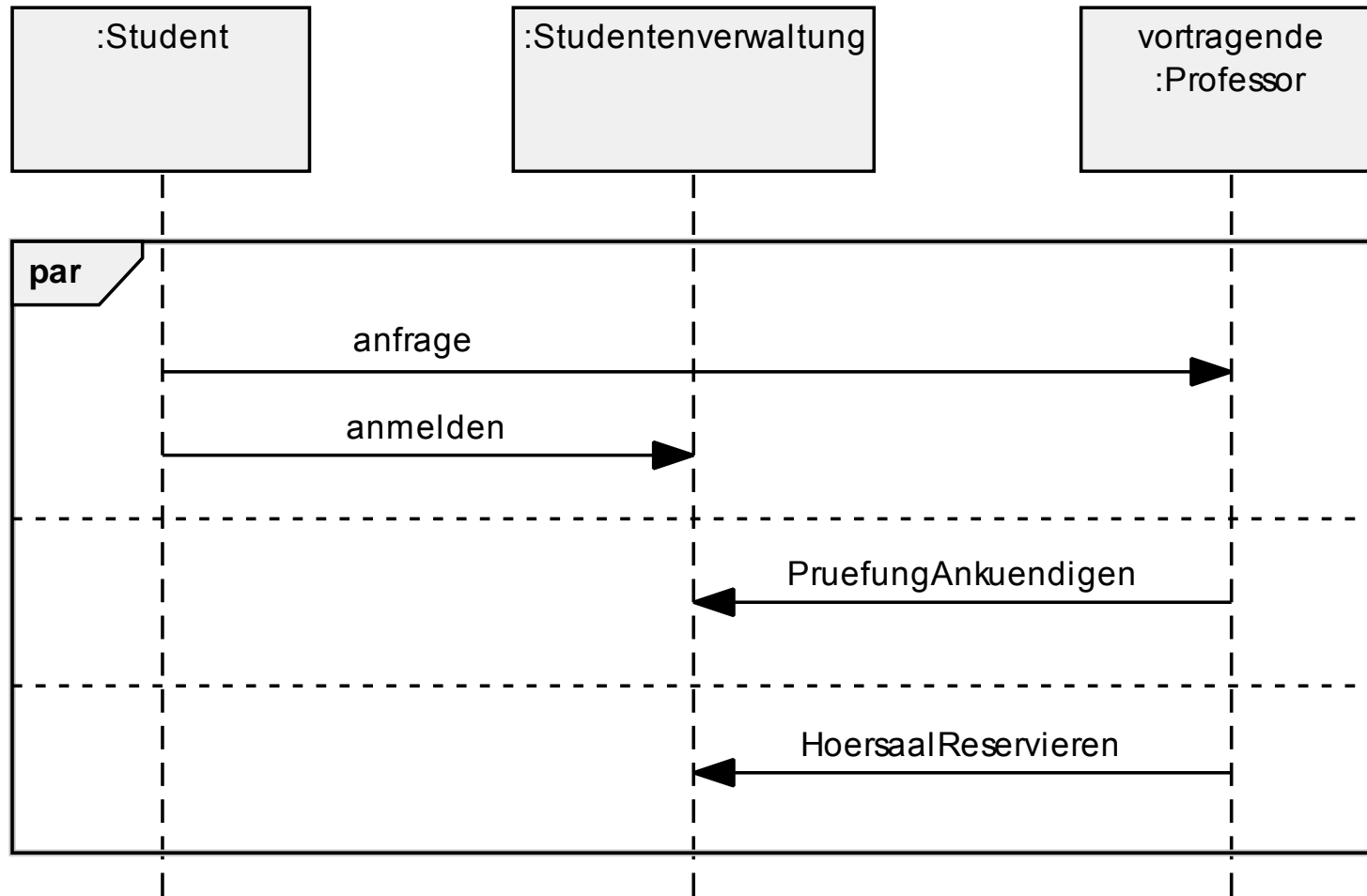
- Nebenläufige Interaktionen

- Lokale Reihenfolge pro Operand muss erhalten bleiben
- Reihenfolge der Operanden im Diagramm ist irrelevant!
- mind. 2 Operanden



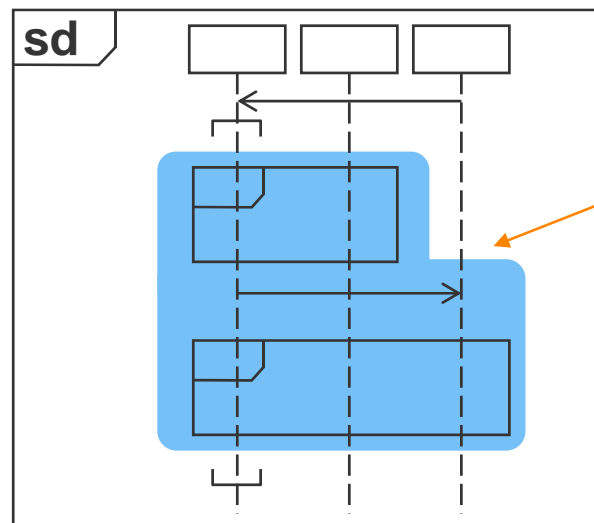
Ereignisf. 1: a1 → a2 → c1 → b1 → c2
Ereignisf. 2: a1 → c1 → a2 → b1 → c2
Ereignisf. 3: a1 → c1 → b1 → a2 → c2
Ereignisf. 4: a1 → c1 → b1 → c2 → a2
Ereignisf. 5: c1 → a1 → a2 → b1 → c2
Ereignisf. 6: c1 → a1 → b1 → a2 → c2
Ereignisf. 7: c1 → a1 → b1 → c2 → a2
Ereignisf. 8: c1 → b1 → a1 → a2 → c2
Ereignisf. 9: c1 → b1 → a1 → c2 → a2
Ereignisf. 10: c1 → b1 → c2 → a1 → a2

Par-Operator - Beispiel

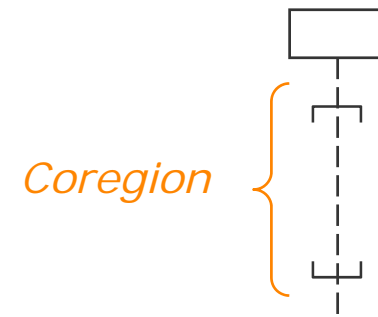


Nebenläufigkeit und Ordnung: Coregion

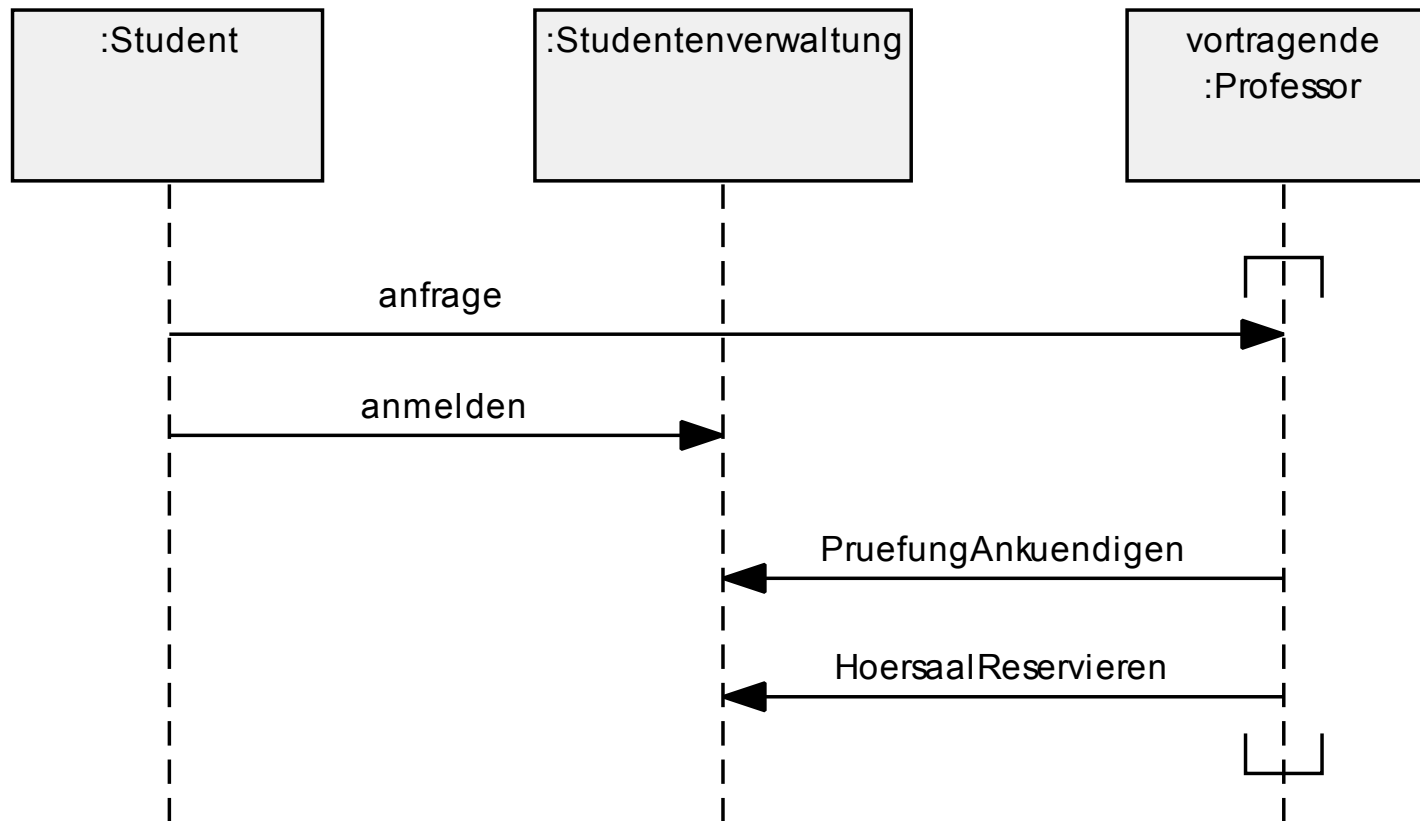
- **Coregion:** Darstellung von nebenläufigen Abläufen auf EINER Lebenslinie
- Reihenfolge der Ereignisseintritte innerhalb von Coregions ist auf keine Weise beschränkt ("Aufhebung der Zeitdimension")
- Coregion kann weitere kombinierte Fragmente beinhalten – kombinierte Fragmente können als Ganzes in bel. Reihenfolge ausgeführt werden



Auswirkung einer Coregion

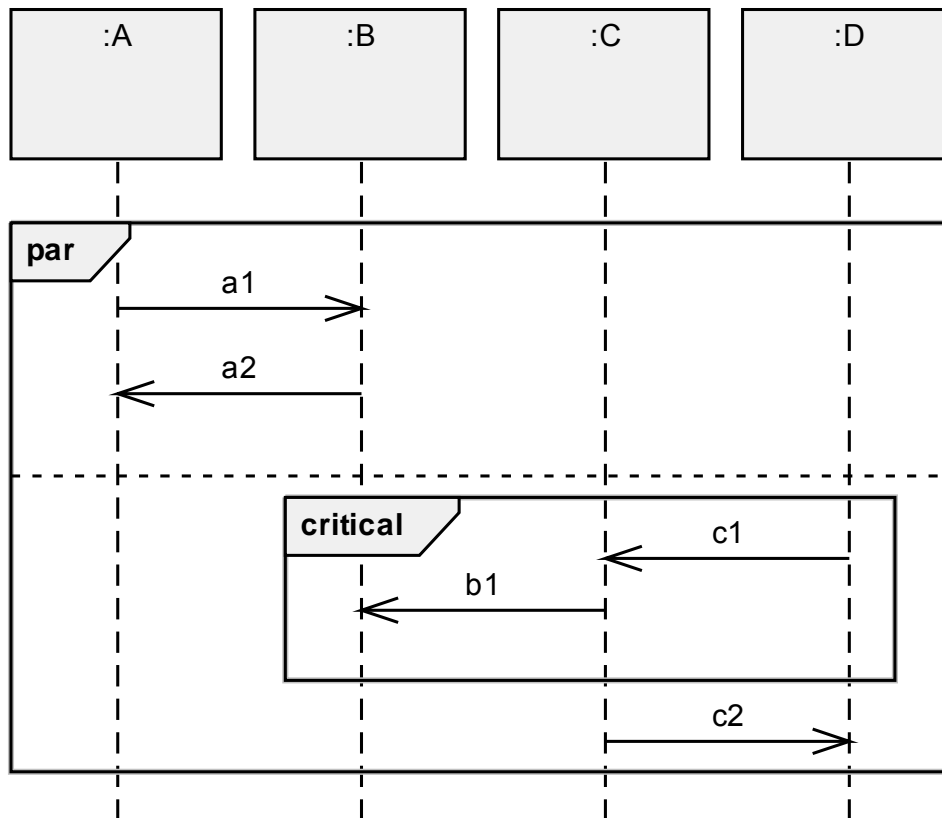


Coregion – Beispiel



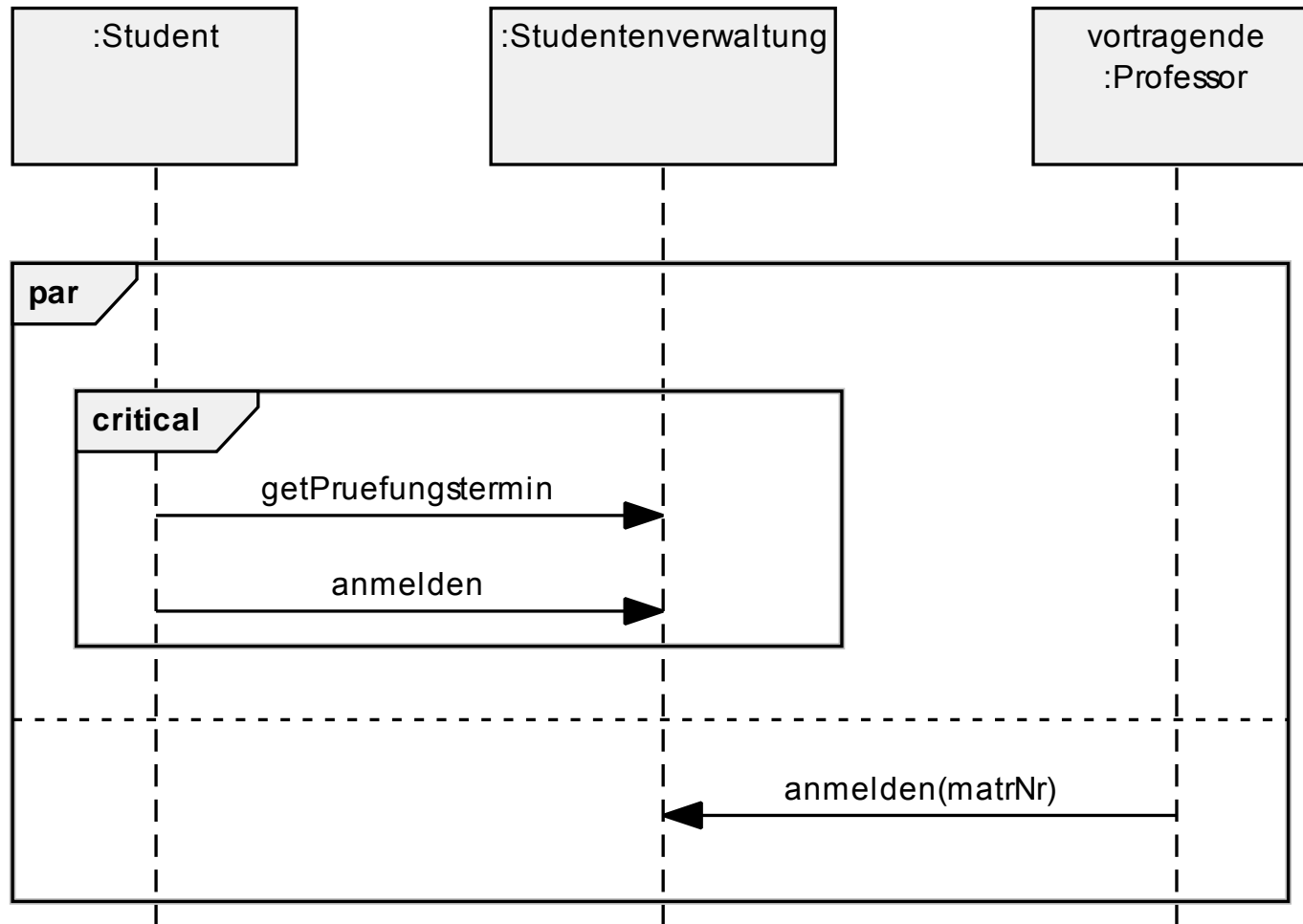
Nebenläufigkeit und Ordnung: critical-Operator

- Kritischer Bereich: atomarer (nicht unterbrechbarer) Interaktionsablauf
- Keine Beschränkung auf Interaktionen außerhalb des kritischen Bereichs



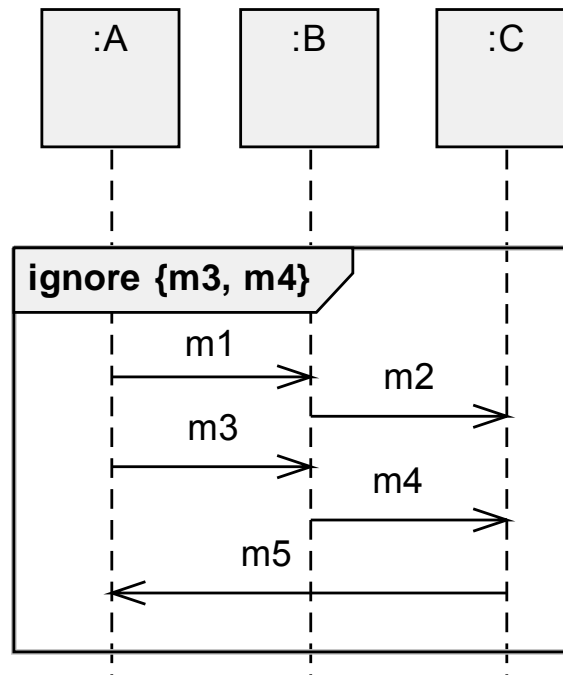
- Ereignisf. 1: a1 → a2 → c1 → b1 → c2
- Ereignisf. 2: a1 → c1 → b1 → a2 → c2
- Ereignisf. 3: a1 → c1 → b1 → c2 → a2
- Ereignisf. 4: c1 → b1 → a1 → a2 → c2
- Ereignisf. 5: c1 → b1 → a1 → c2 → a2
- Ereignisf. 6: c1 → b1 → c2 → a1 → a2

Critical-Operator - Beispiel



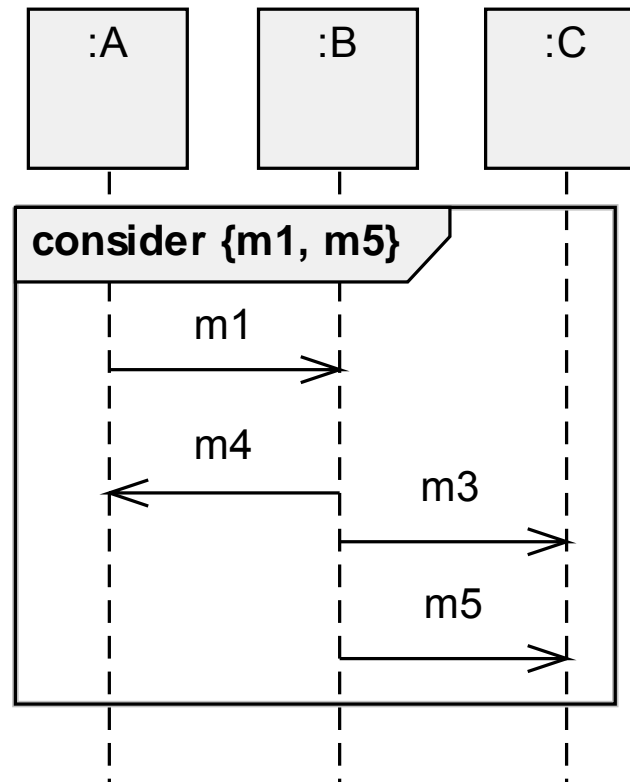
Filterungen u. Zusicherungen: ignore-Operator

- Darstellung von irrelevanten Nachrichten
 - Modellierung von Nachrichten aus technischen Gründen oder wegen syntaktischer Vollständigkeit
 - Nachrichten, die zur Laufzeit auftreten können (z.B. keep-alive Signale)

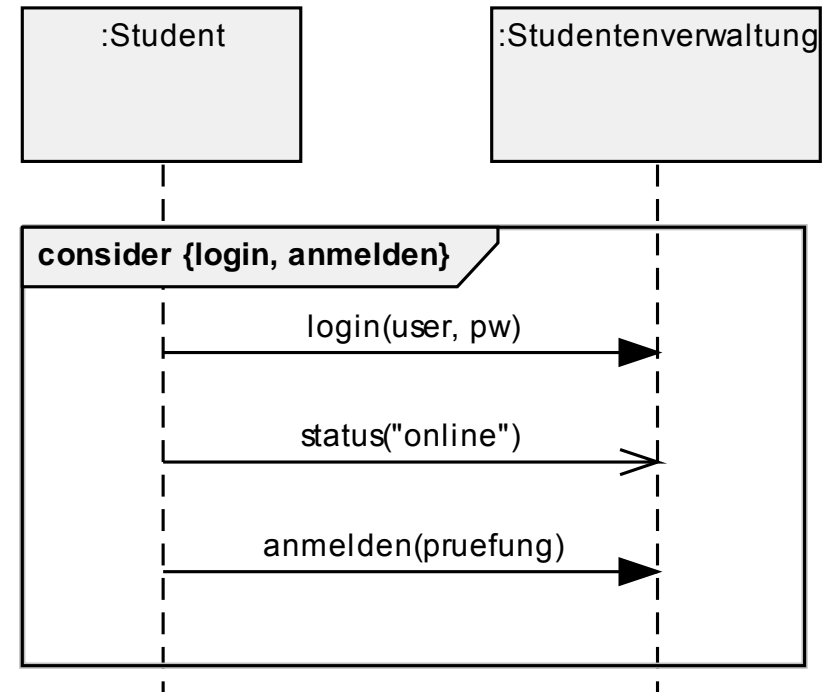
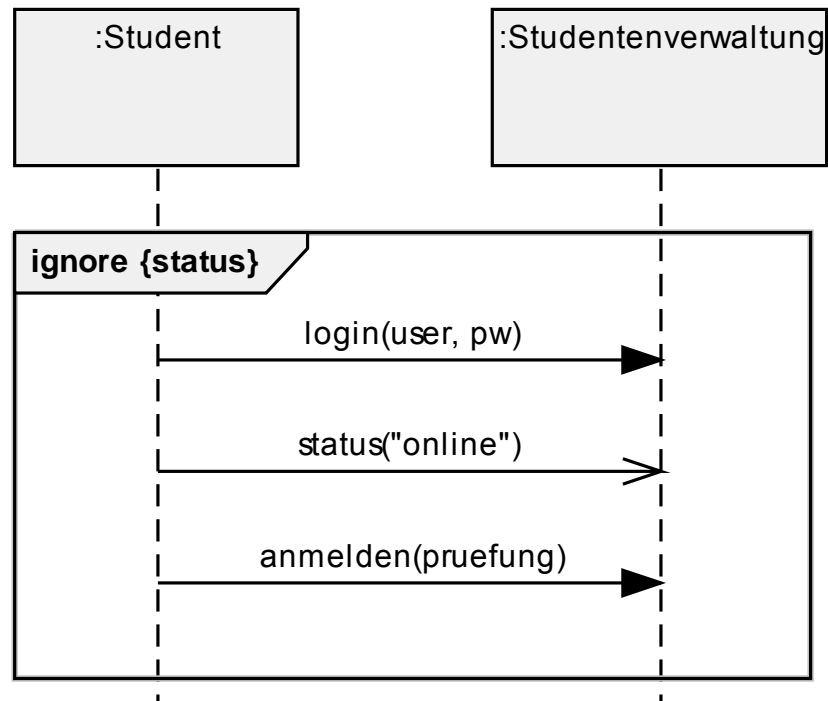


Filterungen u. Zusicherungen: consider-Operator

- Gegenstück von Ignore
- Spezifikation von besonders relevanten Nachrichten
- andere Nachrichten im Operanden werden automatisch als nicht relevant eingestuft

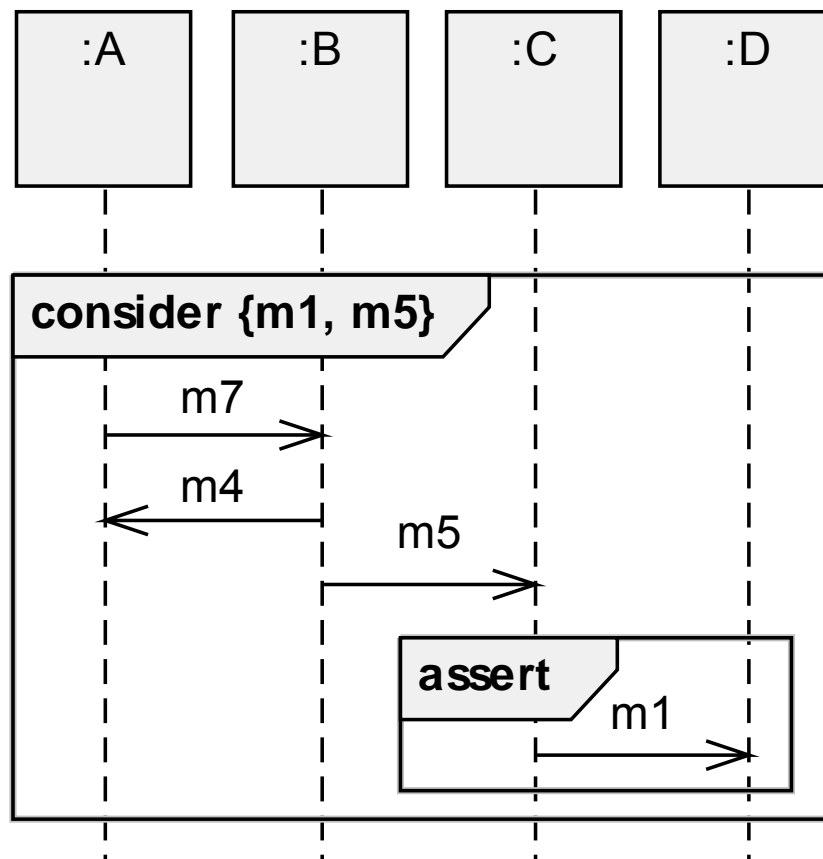


Consider-Operator - Beispiel

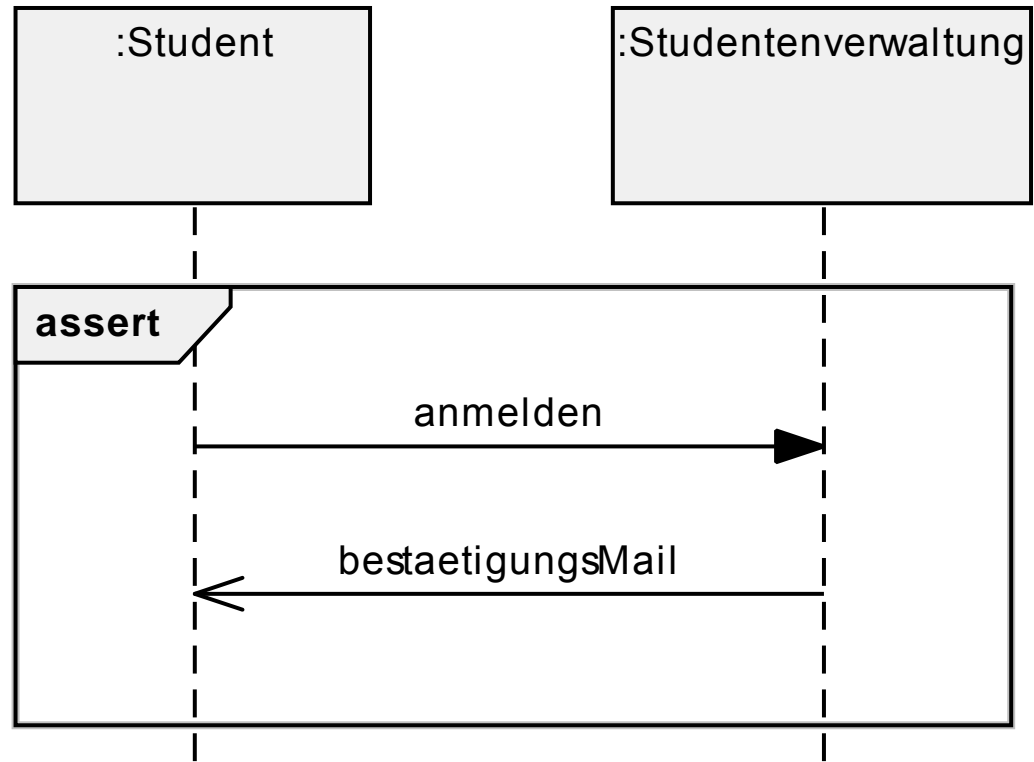


Filterungen u. Zusicherungen: assert-Operator

- Zugesicherte Interaktionen: Kennzeichnung der Interaktion als verpflichtend – Abweichungen, die im Diagramm nicht berücksichtigt sind, aber in der Realität auftreten, sind nicht zulässig
 - ⇒ Forderung von getreuer Abbildung in der Implementierung

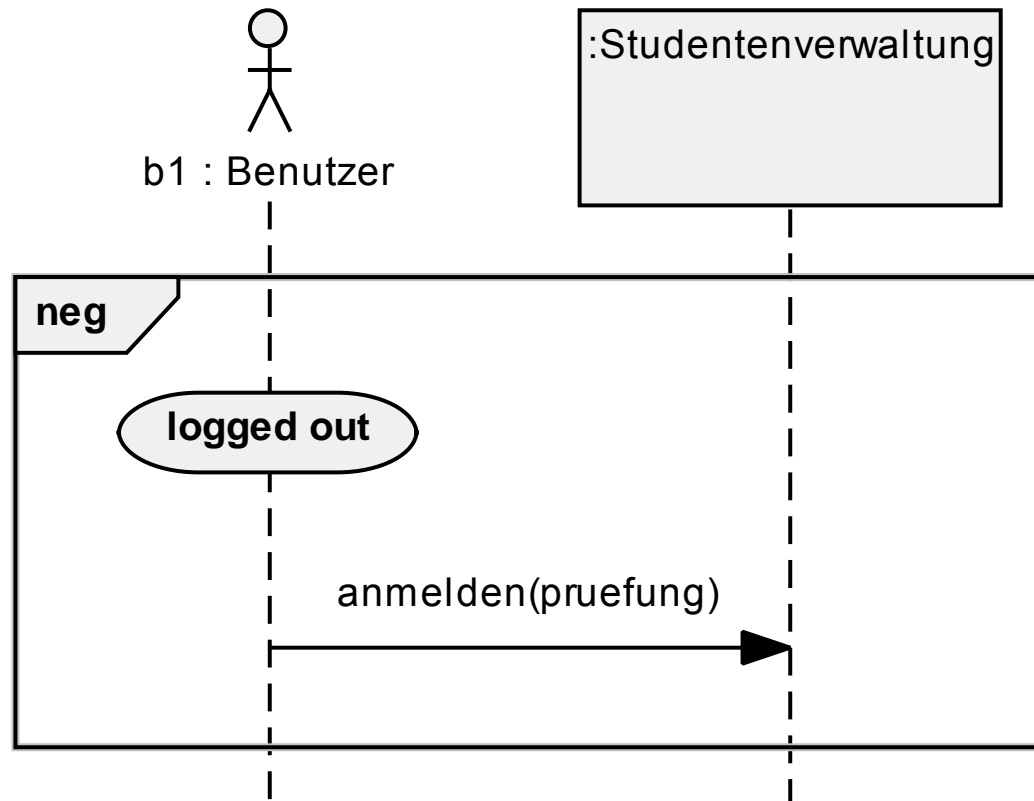


Assert-Operator – Beispiel Prüfungsanmeldung

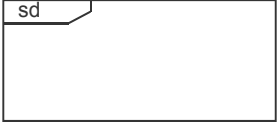
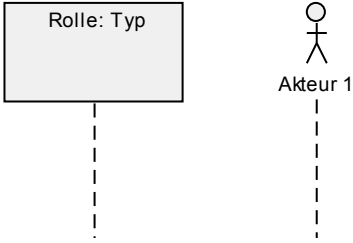
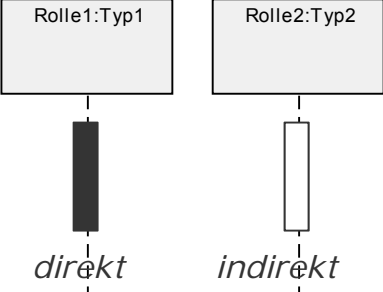


Filterungen und Zusicherungen: neg-Operator

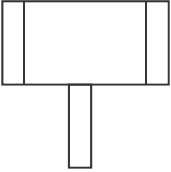

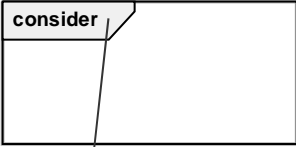
- Ungültige Interaktionen: Es darf nicht sein, dass sich ein Benutzer, der „logged out“ ist, zu einer Prüfung anmeldet



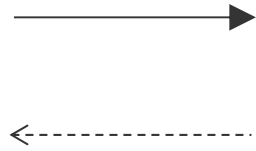

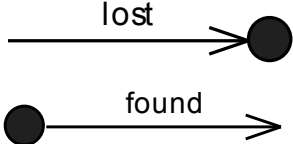
Basiselemente (1/3)

Name	Syntax	Beschreibung
Diagramm- rahmen		Begrenzung des Diagramms, Angabe von Parametern
Lebenslinie		Interaktionspartner
Ausführungs- spezifikation (direkt/indirekt)		Periode, in der ein Interaktionspartner ein Verhalten (direkt/indirekt) ausführt

Basiselemente (2/3)

Name	Syntax	Beschreibung
aktives Objekt		Objekt mit eigenem Kontrollfluss
Löschsymbol		Zeitpunkt zu dem ein Objekt aus seiner Rolle gelöscht wird
Kombiniertes Fragment	 <p data-bbox="730 1174 1093 1283">alt, opt, break, loop seq, par, strict, critical ignore, assert, consider, neg</p>	Steuerung des Kontrollflusses

Basiselemente (3/3)

Name	Syntax	Beschreibung
Synchrone Kommunikation		Nachricht Antwort
Asynchrone Kommunikation		Nachricht
Gefundene/ Verlorene Nachricht		spezielle Nachrichten von oder an unbekannte Interaktionspartner (z.B. Modellierung von Fehlerfällen in der Kommunikation)

Zusammenfassung

- Sie haben diese Lektion verstanden, wenn Sie wissen ...
- wofür Interaktionsdiagramme verwendet werden.
- welche Arten von Interaktionsdiagrammen es gibt.
- aus welchen Komponenten ein Sequenzdiagramm besteht.
- was mit einer Lebenslinie gemeint ist.
- wie die Reihenfolge von Ereigniseintritten im Sequenzdiagramm definiert ist.
- was der Unterschied zwischen direkter und indirekter Ausführung, sowie aktiven und passiven Objekten ist.
- welche Operatoren im Sequenzdiagramm zur Verfügung stehen.
- wie Nebenläufigkeit ausgedrückt werden kann.