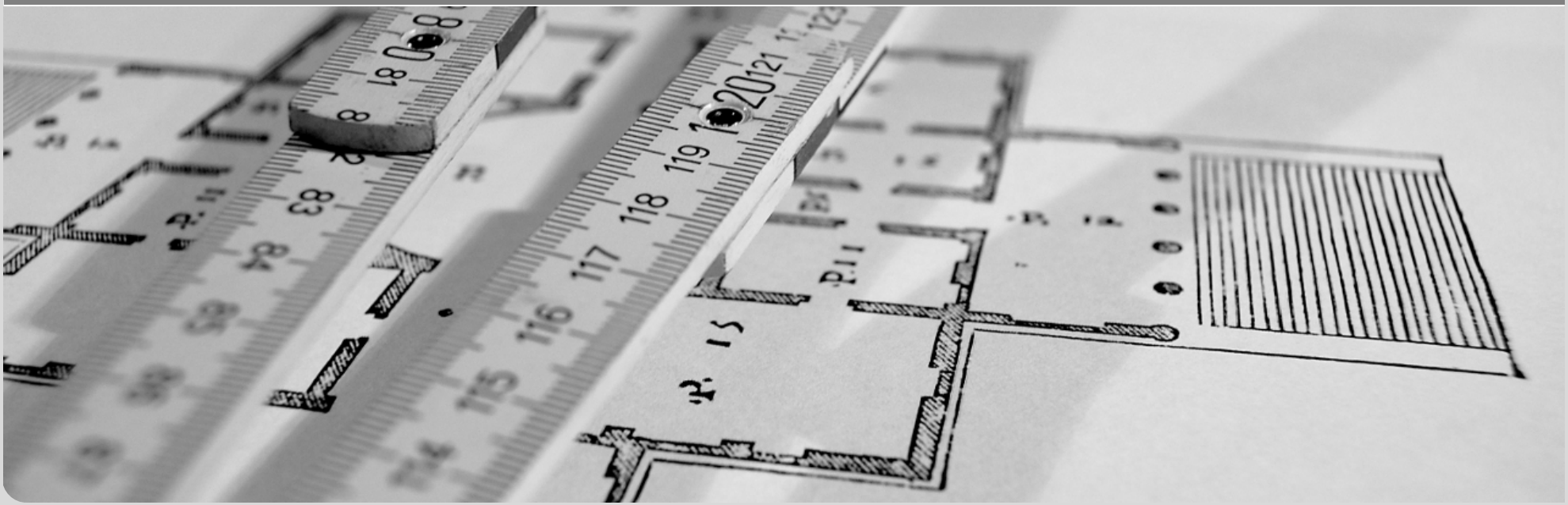
Karlsruher Institut für Technologie

Softwaretechnik 1 - 2. Tutorium

Tutorium 03

Felix Bachmann | 29.05.2017

KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)



2. Übungsblatt Statistik

Häufige Fehler

Allgemein

Häufige Fehler

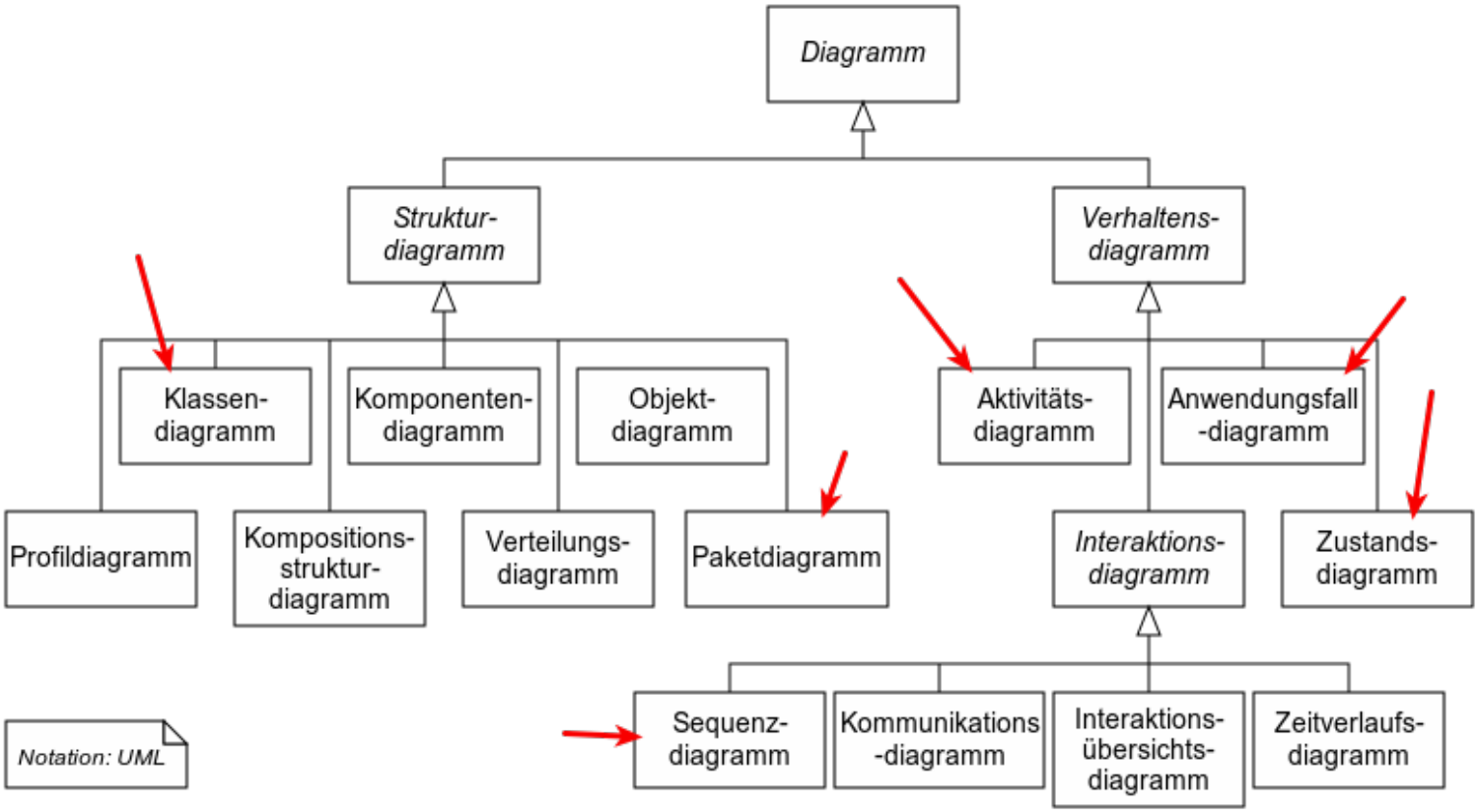
Aufgabe 1 (??)

Wo sind wir? Pflichtenheft!

1. Zielbestimmung
2. Produkteinsatz
3. Produktumgebung
4. Funktionale Anforderungen
5. Produktdaten
6. Nichtfunktionale Anforderungen
7. Globale Testfälle
8. Systemmodelle
   - Szenarien
   - Anwendungsfälle
   - Objektmodelle $\Longrightarrow$ UML-Klassendiagramme (letztes mal)
   - **Dynamische Modelle**
     - UML-Zustandsdiagramm
     - UML-Aktivitätsdiagramm
     - UML-Sequenzdiagramm

     } Heute!
   - Benutzerschnittstelle $\Longrightarrow$ Zeichnungen/Screenshots
9. Glossar

Begriffsklärung



Zustandsdiagramm - Allgemein

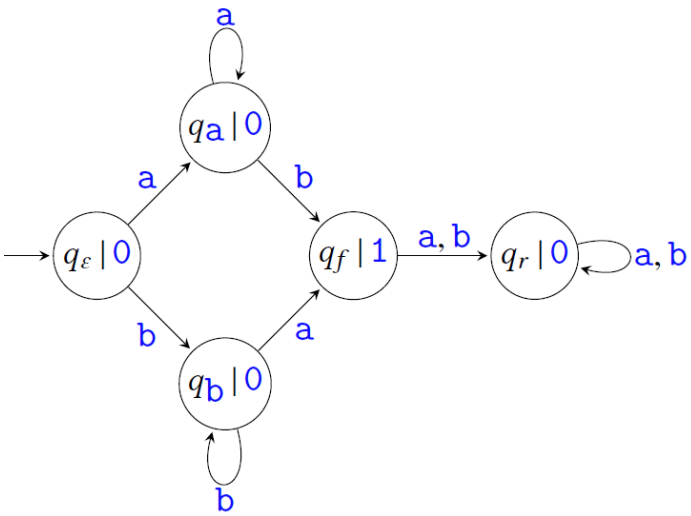
Wozu braucht man das?

Zustandsdiagramm - Allgemein

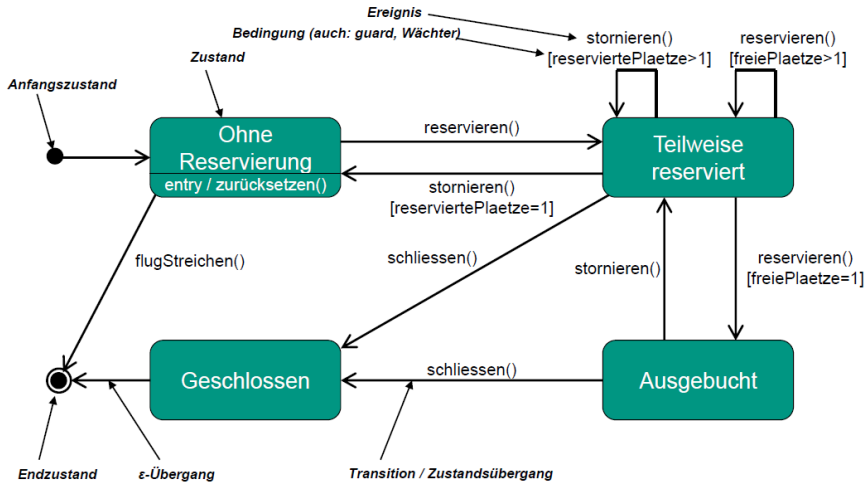
Wozu braucht man das?

- Zustand **eines Objektes** beschreiben
- Zustandsüberführungsfunktion?

Zustandsdiagramm ≈ endlicher Automat

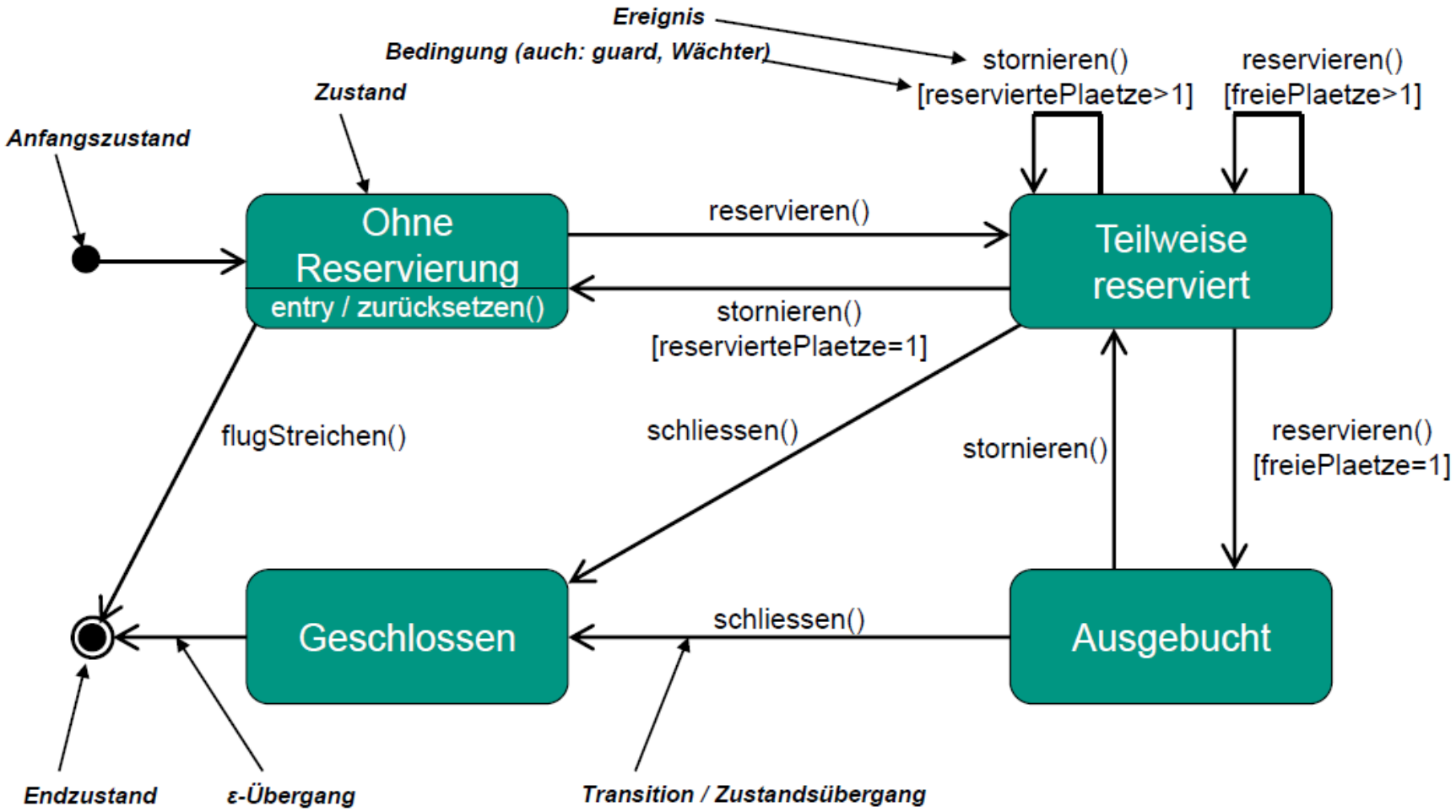


(a) GBI: DEA

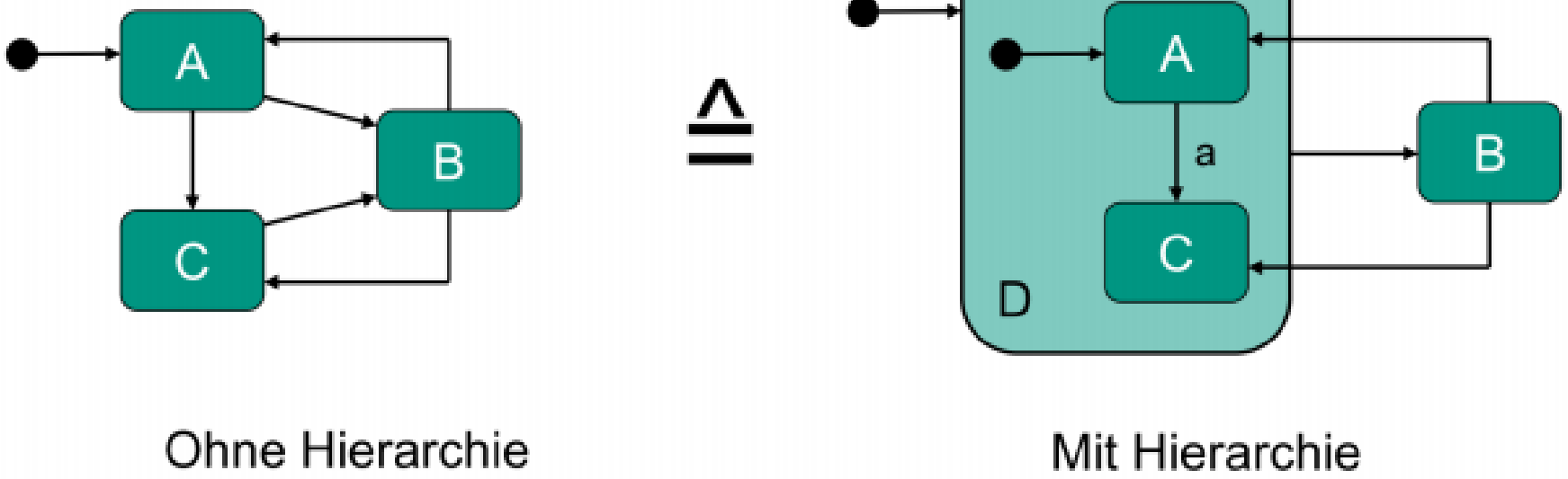


(b) SWT: Zustandsdiagramm

Zustandsdiagramm: Syntax

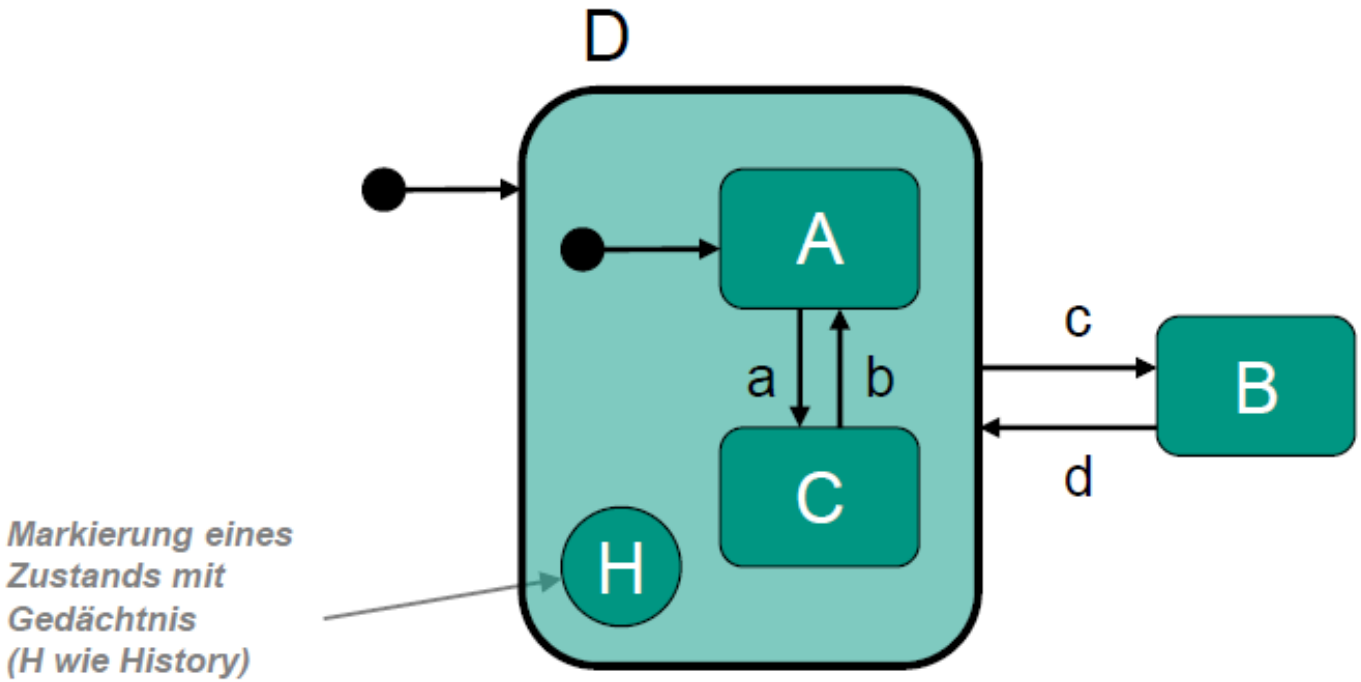


Zustandsdiagramm: Hierarchie



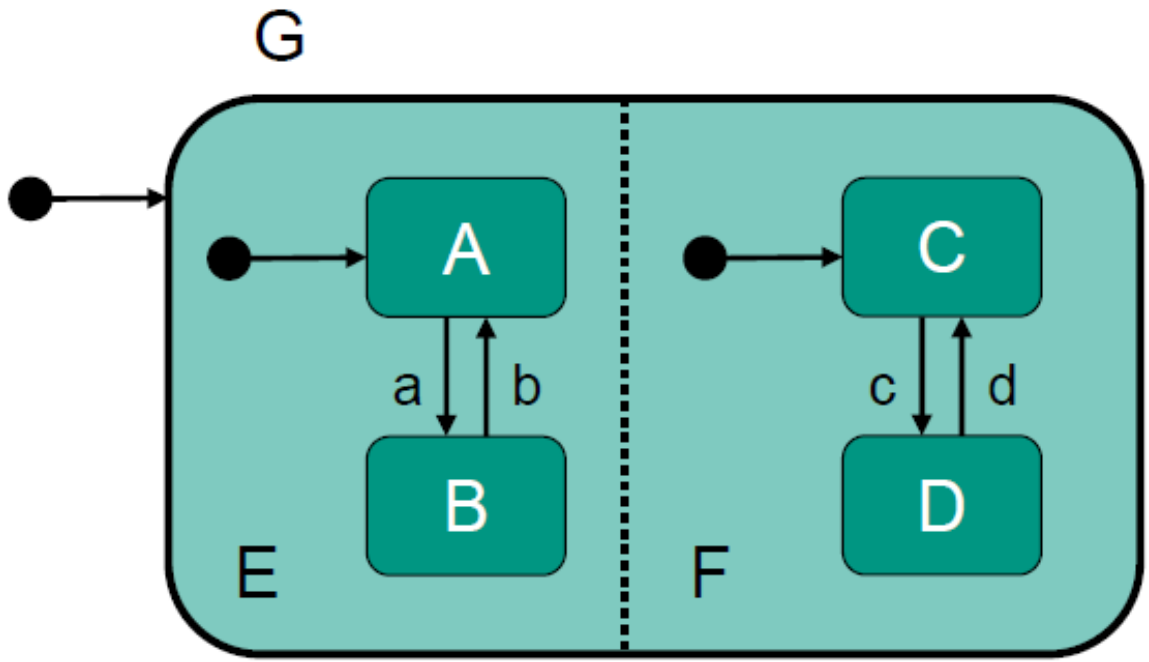
Zustandsdiagramm: Hierarchie - History

- History-Element, damit sich Hierarchie den letzten Zustand merkt



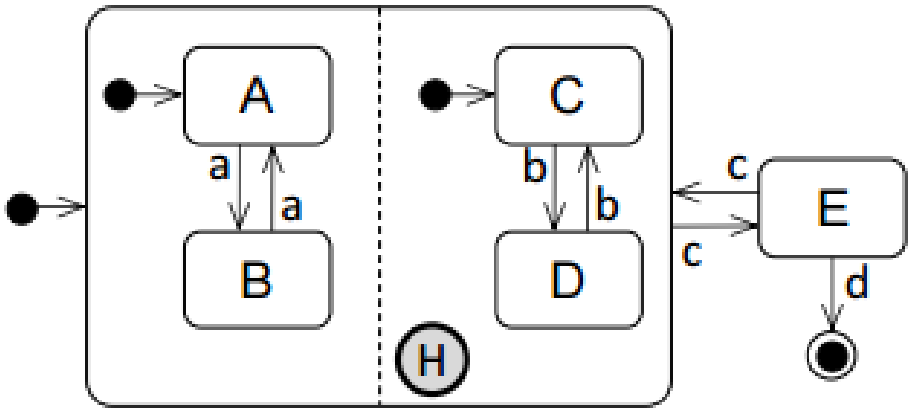
Zustandsdiagramm: Nebenläufigkeit

- mehrere Zustandsdiagramme in einem



Klausuraufgabe SS09

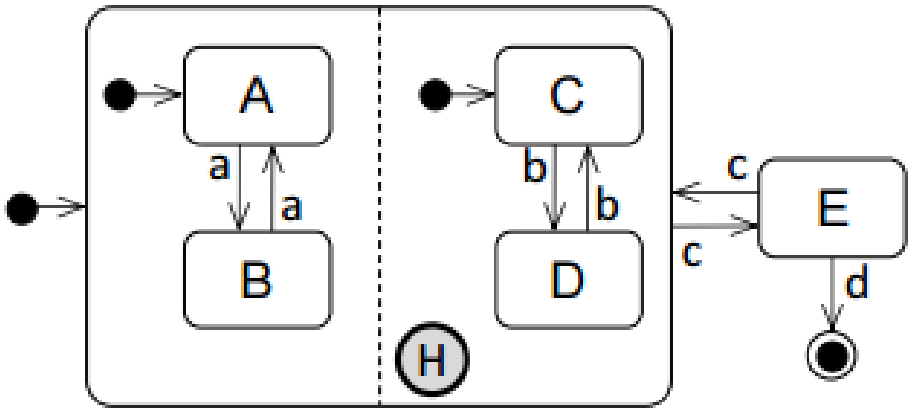
Gegeben ist der folgende UML-Zustandsautomat. Geben Sie an, in welcher Zustandskombination sich der Zustandsautomat, jeweils ausgehend vom Startzustand, nach den beiden Eingabefolgen befindet.



- a, b, c, c

Klausuraufgabe SS09

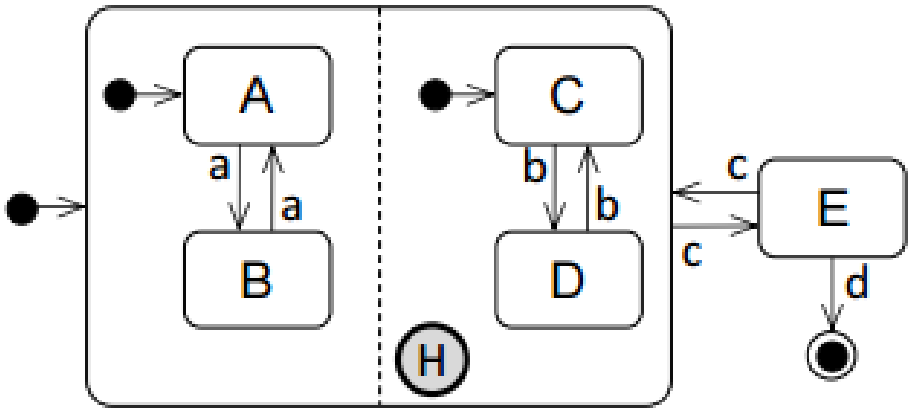
Gegeben ist der folgende UML-Zustandsautomat. Geben Sie an, in welcher Zustandskombination sich der Zustandsautomat, jeweils ausgehend vom Startzustand, nach den beiden Eingabefolgen befindet.



- a, b, c, c $\Longrightarrow$ AxD
- c, c, a, b, b, a, c, c, a

Klausuraufgabe SS09

Gegeben ist der folgende UML-Zustandsautomat. Geben Sie an, in welcher Zustandskombination sich der Zustandsautomat, jeweils ausgehend vom Startzustand, nach den beiden Eingabefolgen befindet.



- a, b, c, c $\Longrightarrow$ AxD
- c, c, a, b, b, a, c, c, a $\Longrightarrow$ BxC

Aktivitätsdiagramm - Allgemein

Wozu braucht man das?

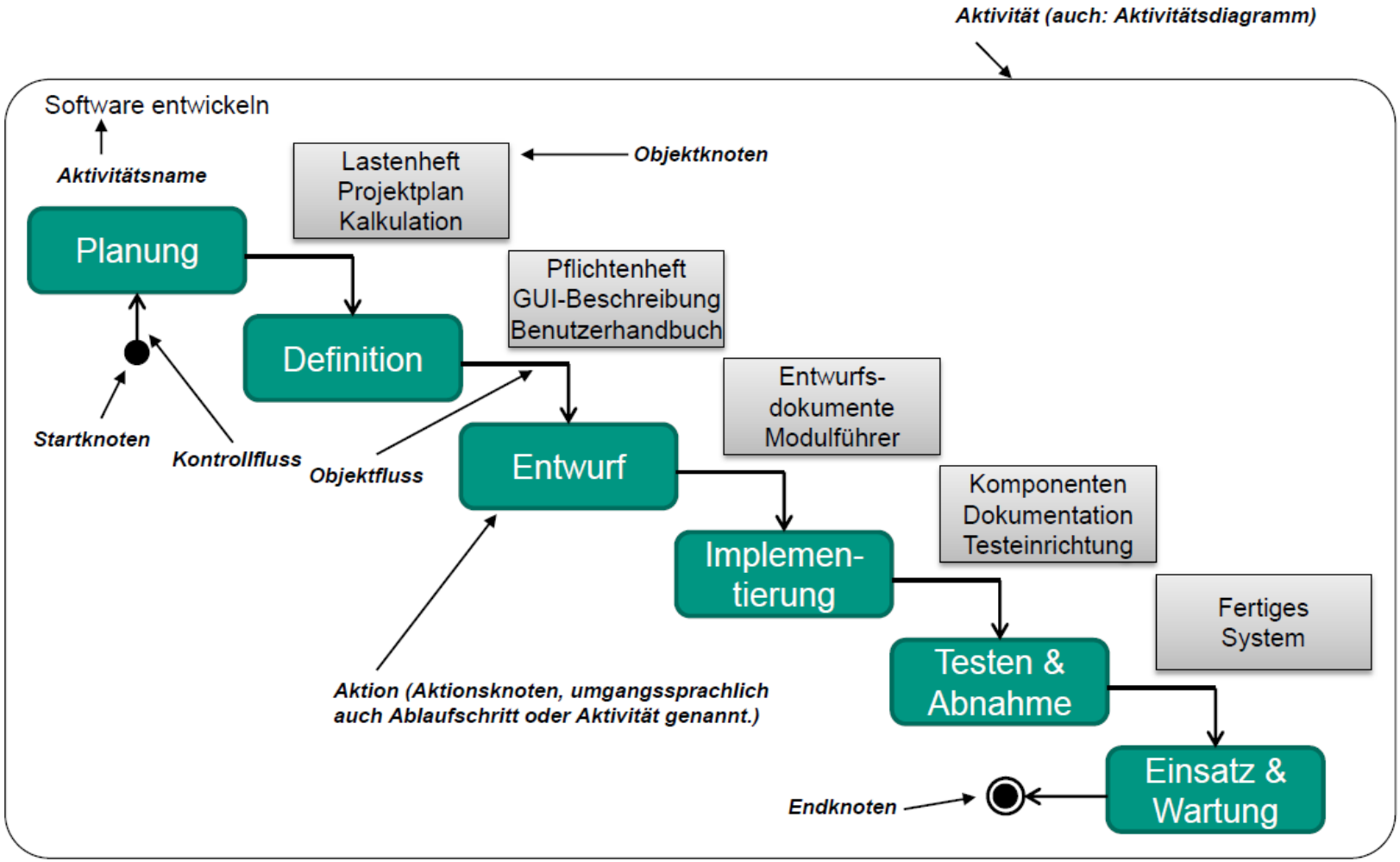
Aktivitätsdiagramm - Allgemein

Wozu braucht man das?

- Ablaufbeschreibungen (Kontrollfluss, Objektfluss)
- i.A. **mehrere verschiedene** Objekte

Aktivitätsdiagramm - Beispiel

- ist ebenfalls nicht neues!

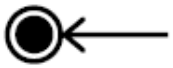


Aktivitätsdiagramm - Syntax

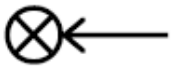
- Aktionen
  - Elementare Aktion
  - Verschachtelte Aktion

- Knoten
  - Startknoten
    - Startpunkt eines Ablaufs

  - Endknoten
    - Beendet alle Aktionen und Kontrollflüsse

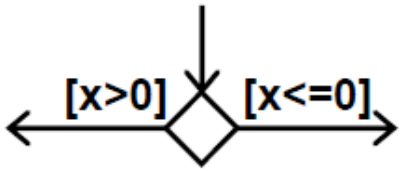


  - Ablaufende
    - Beendet einen einzelnen Objekt- und Kontrollfluss

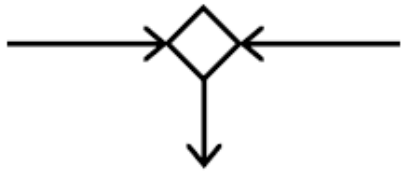


Aktivitätsdiagramm - Syntax

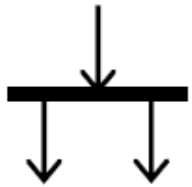
- Entscheidung
  - bedingte Verzweigung



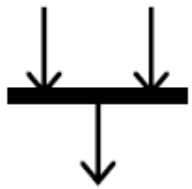
- Zusammenführung
  - „oder“-Verknüpfung



- Teilung
  - Aufteilung eines Kontrollflusses

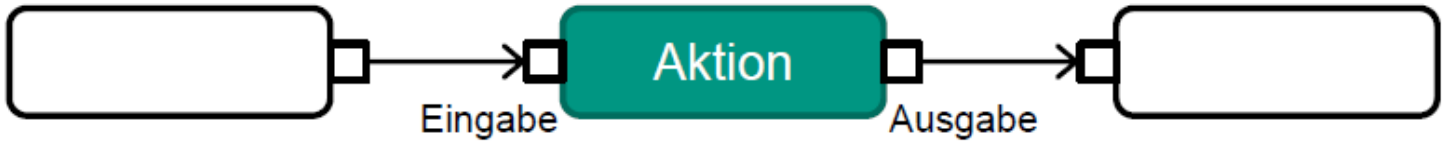


- Synchronisation
  - „und“-Verknüpfung

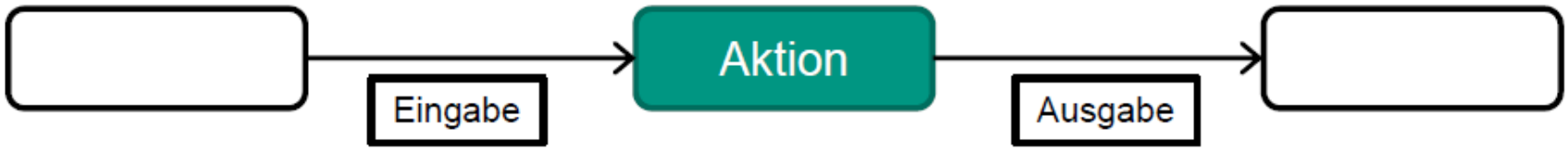


Aktivitätsdiagramm - Syntax

- Objektknoten
  - Eingabe- und Ausgabedaten einer Aktion
  - Darstellung durch Stecker (engl. *pin*)



  - Alternative Darstellungen

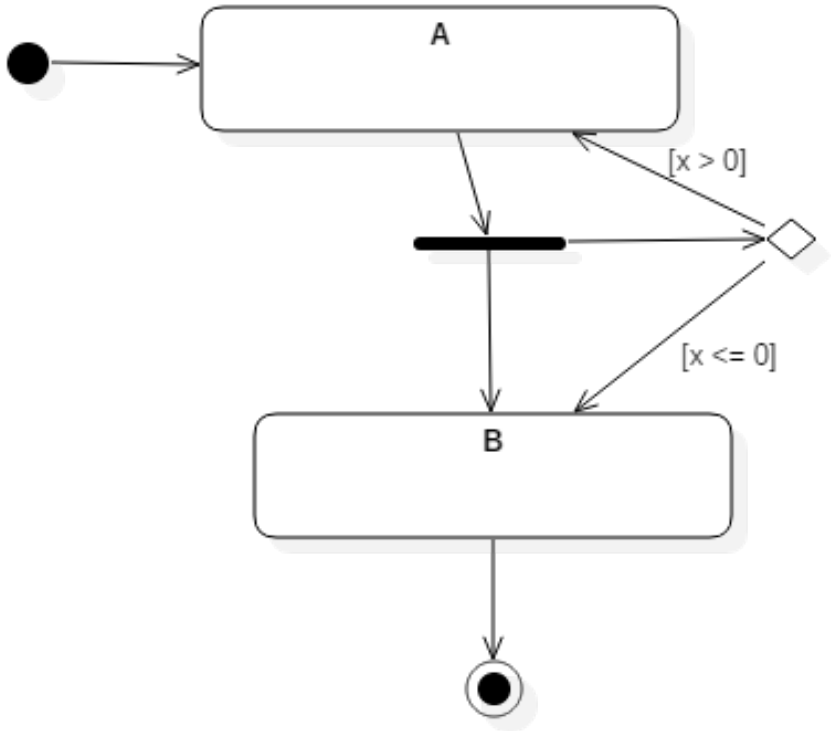


Aktivitätsdiagramm - Ablauf

- Start am Startknoten mit einer Marke
- Aktivitäten werden erst ausgeführt, wenn an jedem Eingang eine Marke anliegt
- wurde eine Aktivität ausgeführt, erscheinen an all ihren Ausgängen Marken

Aktivitätsdiagramm - Beispiel

Wie kommt man hier zum Endknoten?



Sequenzdiagramm - Allgemein

Wozu braucht man das?

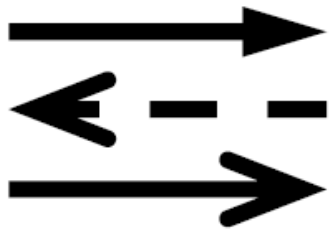
Sequenzdiagramm - Allgemein

Wozu braucht man das?

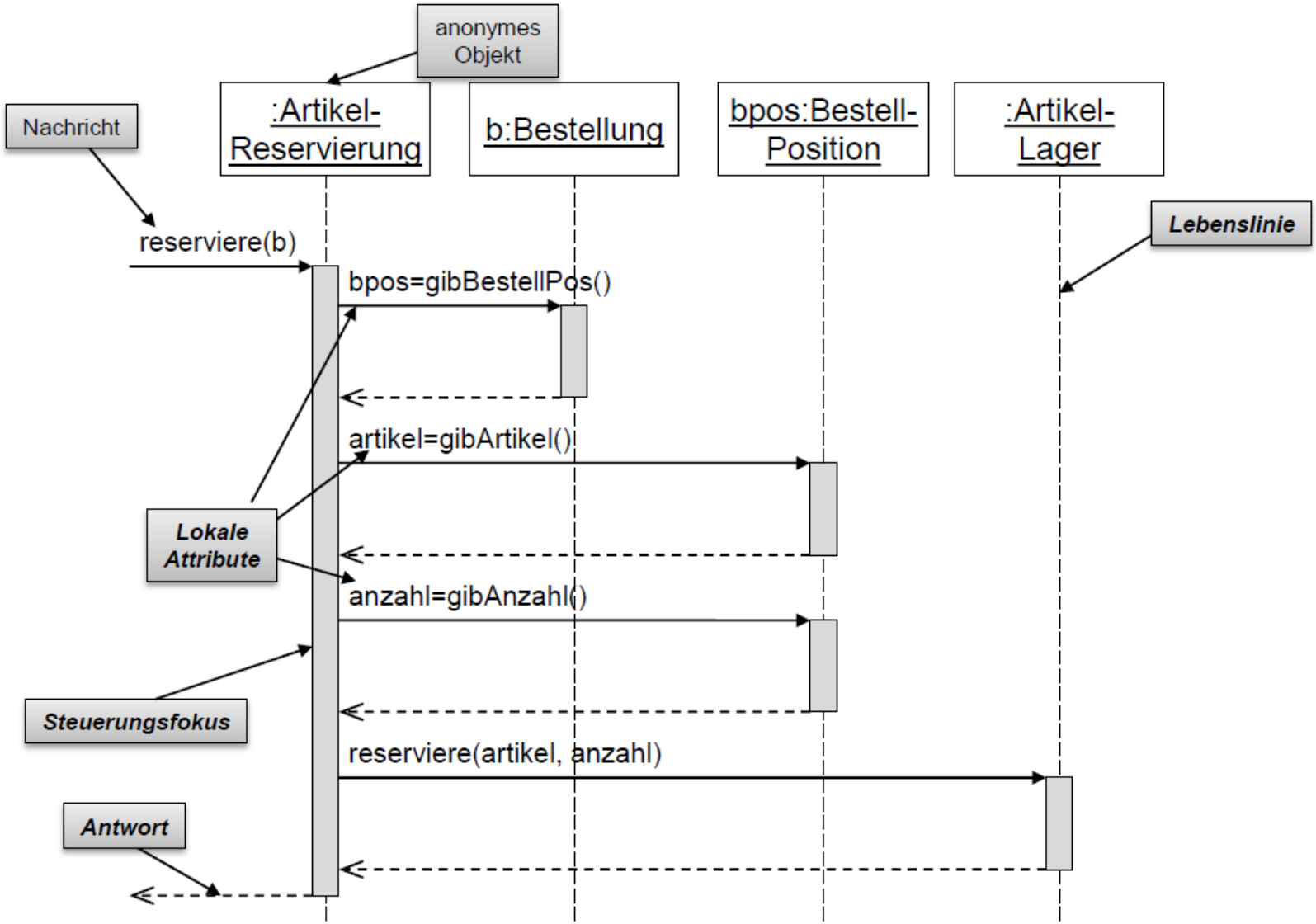
- stellt den möglichen Ablauf eines Anwendungsfalls dar
- **zeitlicher Verlauf** von Methodenaufrufen, Objekterstellung, Objektzerstörung

Sequenzdiagramm - Syntax

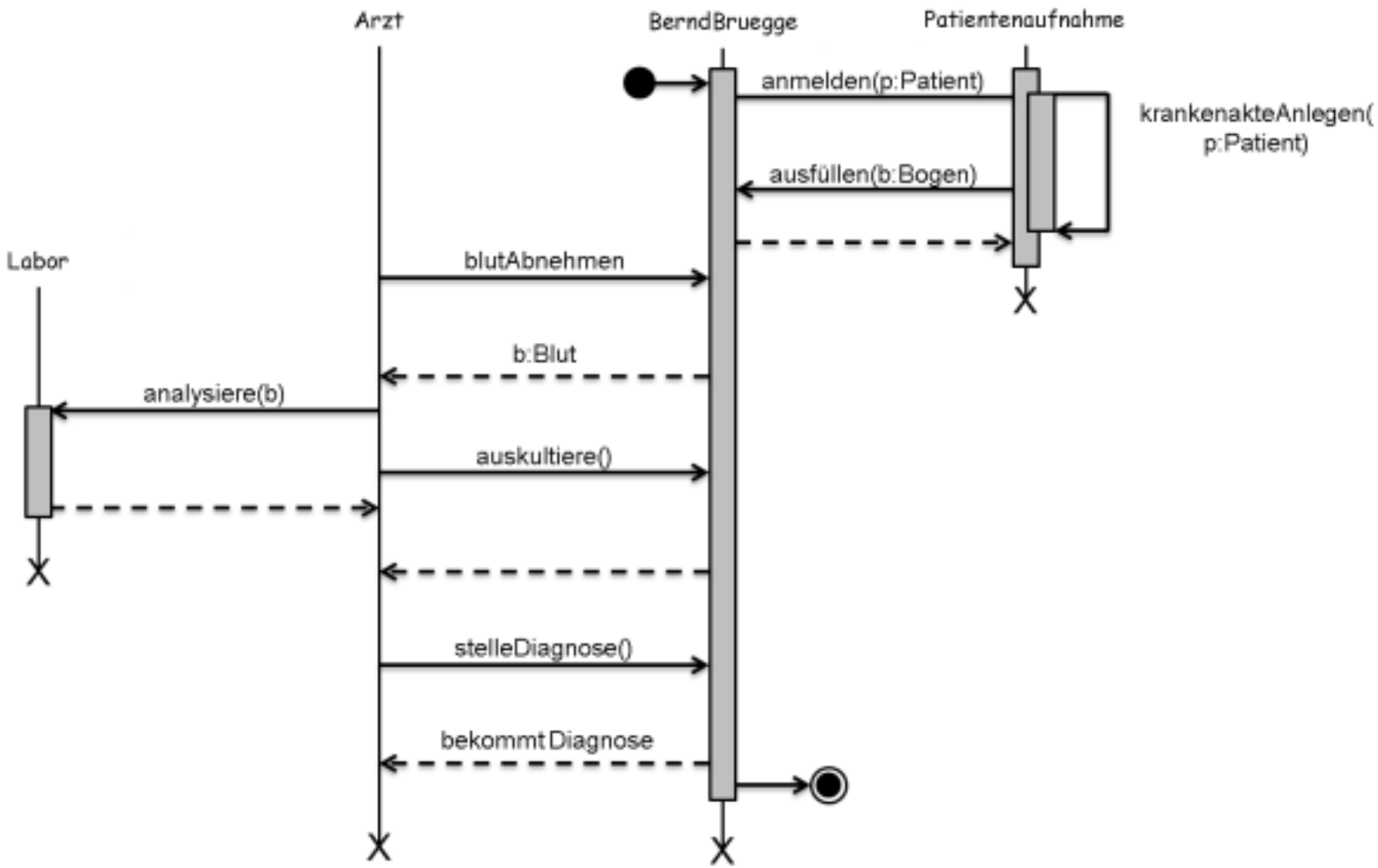
- Zeit verläuft von oben nach unten
- Lebenslinie
  - gestrichelte senkrechte Linie
  - eine pro Objekt
- Steuerungsfokus
  - dicker Balken über Lebenslinie
  - zeigt, dass Objekt gerade aktiv ist
- Nachrichtentypen
  - Synchrone Nachricht (blockierend)
  - Antwort (optional)
  - Asynchrone Nachricht



Sequenzdiagramm - Beispiel



Klausuraufgabe SS14



Hier stimmt was nicht…

Tipps - 3. Übungsblatt

Aufgabe 1-3: Plug-In programmieren

Tipps - 3. Übungsblatt

Aufgabe 1-3: Plug-In programmieren

- JavaDoc + CheckStyle . . .
- Falls ihr Tests schreibt, fügt junit in die jeweilige Untermodul-pom ein
- Java Swing benutzen (schaut euch die Java-Klassen JMenu und JMenuItem an)

Tipps - 3. Übungsblatt

Aufgabe 1-3: Plug-In programmieren

- JavaDoc + CheckStyle . . .
- Falls ihr Tests schreibt, fügt junit in die jeweilige Untermodul-pom ein
- Java Swing benutzen (schaut euch die Java-Klassen JMenu und JMenuItem an)

Aufgabe 4: Aktivitätsdiagramm

Tipps - 3. Übungsblatt

Aufgabe 1-3: Plug-In programmieren

- JavaDoc + CheckStyle . . .
- Falls ihr Tests schreibt, fügt junit in die jeweilige Untermodul-pom ein
- Java Swing benutzen (schaut euch die Java-Klassen JMenu und JMenuItem an)

Aufgabe 4: Aktivitätsdiagramm

- seperate Diagramme $\Longrightarrow$ verschachtelte Aktionen

Tipps - 3. Übungsblatt

Aufgabe 5: Sequenzdiagramm

Tipps - 3. Übungsblatt

Aufgabe 5: Sequenzdiagramm

- auf welchen Objekten/Klassen werden Methoden aufgerufen?
- auf Pfeile var=methode() schreiben, wenn Rückgabe von methode() in var gespeichert wird

Tipps - 3. Übungsblatt

Aufgabe 5: Sequenzdiagramm

- auf welchen Objekten/Klassen werden Methoden aufgerufen?
- auf Pfeile var=methode() schreiben, wenn Rückgabe von methode() in var gespeichert wird

Aufgabe 6: Substitutionsprinzip

Tipps - 3. Übungsblatt

Aufgabe 5: Sequenzdiagramm

- auf welchen Objekten/Klassen werden Methoden aufgerufen?
- auf Pfeile var=methode() schreiben, wenn Rückgabe von methode() in var gespeichert wird

Aufgabe 6: Substitutionsprinzip

- Folien "Folgerung aus dem Substitutionsprinzip" anschauen (Ko-/Kontravarianz)
- mal als Java-Programm hinschreiben und versuchen zu kompilieren

Denkt dran!

Abgabe

- Deadline am 7.6 um 12:00
- Aufgabe 4+5 handschriftlich (auf saubere Syntax achten!)
- an das Deckblatt denken!!

Bis dann! (dann := 12.06.17)