

# Softwaretechnik 1 - 2. Tutorium

Tutorium 03

Felix Bachmann | 29.05.2017

KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)



## 2. Übungsblatt Statistik

## Allgemein

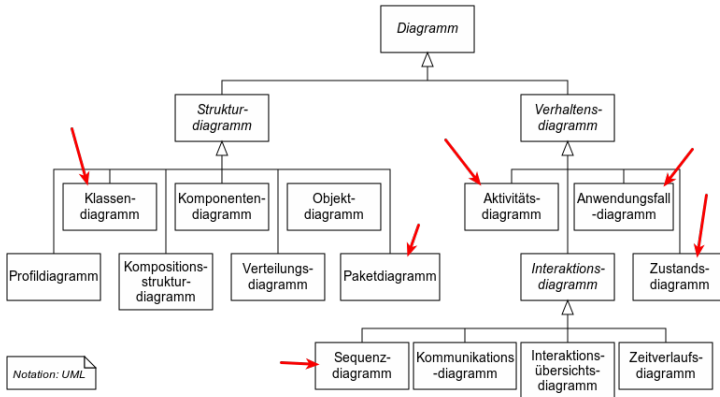
## Aufgabe 1 (??)

# 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  $\implies$  UML-Klassendiagramme (letztes mal)
  - **Dynamische Modelle**
    - UML-Zustandsdiagramm
    - UML-Aktivitätsdiagramm
    - UML-Sequenzdiagramm
- ⑨ Glossar

} Heute!

- Benutzerschnittstelle  $\implies$  Zeichnungen/Screenshots



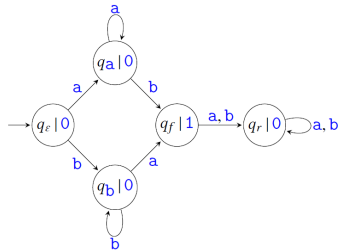
Wozu braucht man das?

## Wozu braucht man das?

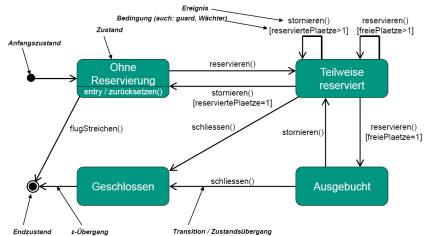
- Zustand **eines Objektes** beschreiben
- Zustandsüberföhrungsfunktion?



# Zustandsdiagramm $\approx$ endlicher Automat

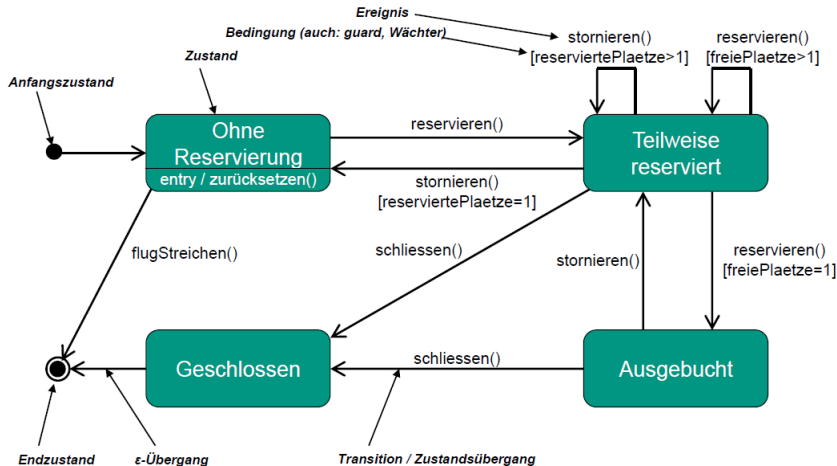


(a) GBI: DEA

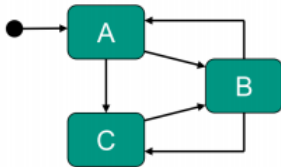


(b) SWT: Zustandsdiagramm

# Zustandsdiagramm: Syntax

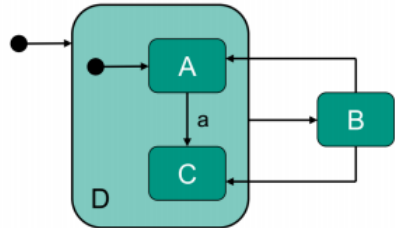


# Zustandsdiagramm: Hierarchie



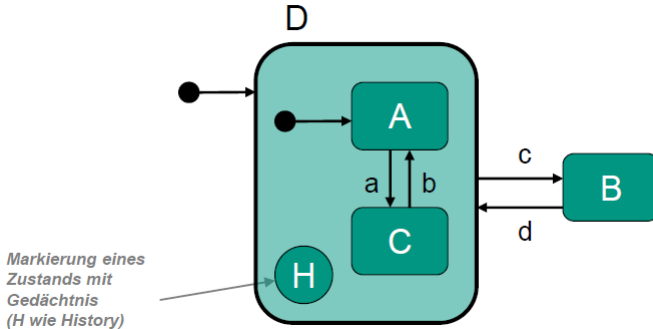
Ohne Hierarchie

$\cong$

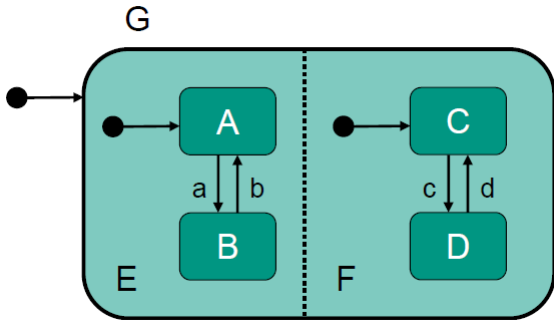


Mit Hierarchie

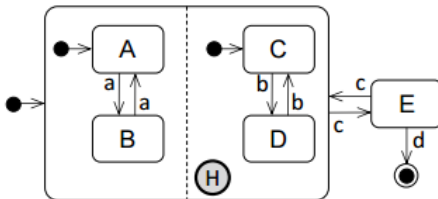
- History-Element, damit sich Hierarchie den letzten Zustand merkt



- mehrere Zustandsdiagramme in einem

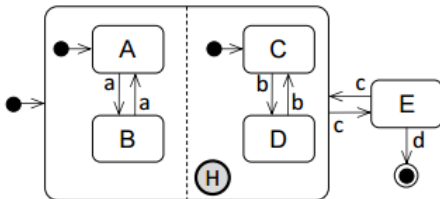


Gegeben ist der folgende UML-Zustandsautomat. Geben Sie an, in welcher Zustandskombination sich der Zustandsautomat, jeweils ausgehend vom Startzustand, nach den beiden Eingabefolgen befindet.



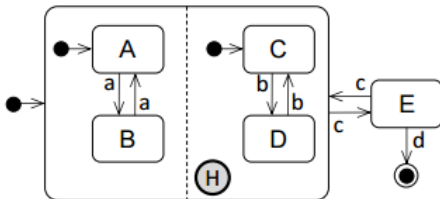
■ a, b, c, c

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

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



- $a, b, c, c \implies Ax D$
- $c, c, a, b, b, a, c, c, a \implies Bx C$



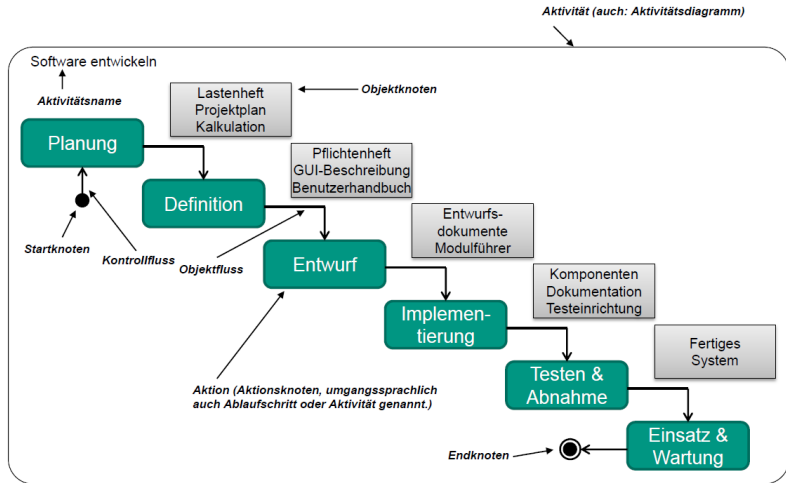
Wozu braucht man das?

## Wozu braucht man das?

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

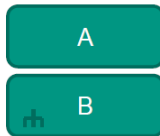
# Aktivitätsdiagramm - Beispiel

- ist ebenfalls nicht neues!



## ■ Aktionen

- Elementare Aktion
- Verschachtelte Aktion

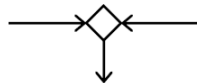
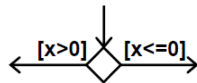


## ■ Knoten

- Startknoten
  - Startpunkt eines Ablaufs
- Endknoten
  - Beendet alle Aktionen und Kontrollflüsse
- Ablaufende
  - Beendet einen einzelnen Objekt- und Kontrollfluss



- Entscheidung
  - bedingte Verzweigung
- Zusammenführung
  - „oder“-Verknüpfung
- Teilung
  - Aufteilung eines Kontrollflusses
- Synchronisation
  - „und“-Verknüpfung



## ■ Objektknoten

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

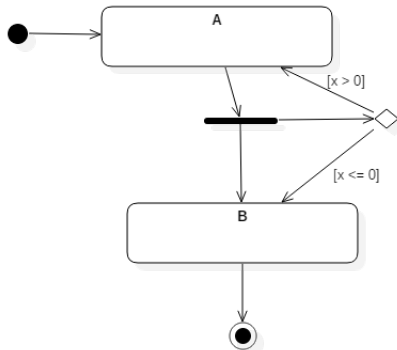


## ■ Alternative Darstellungen



- 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

Abbildung: Wie kommt man hier zum Endknoten?





## Aufgabe 1-3: Plug-In programmieren



## Aufgabe 1-3: Plug-In programmieren



## Aufgabe 4: Aktivitätsdiagramm



## Aufgabe 1-3: Plug-In programmieren



## Aufgabe 4: Aktivitätsdiagramm



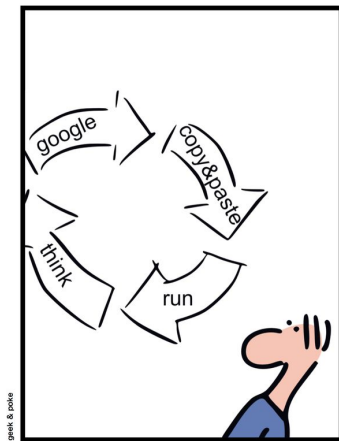
## Aufgabe 5: Sequenzdiagramm



## Abgabe

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

## SIMPLY EXPLAINED



DEVELOPMENT CYCLE