

# UML - Aktivitätsdiagramme

Berufsförderungswerk Berlin-Brandenburg e. V.  
2017

Dipl. Kfm. Patrick Lange (FH)



Berufsförderungswerk Berlin-Brandenburg e. V.

# Aktivität

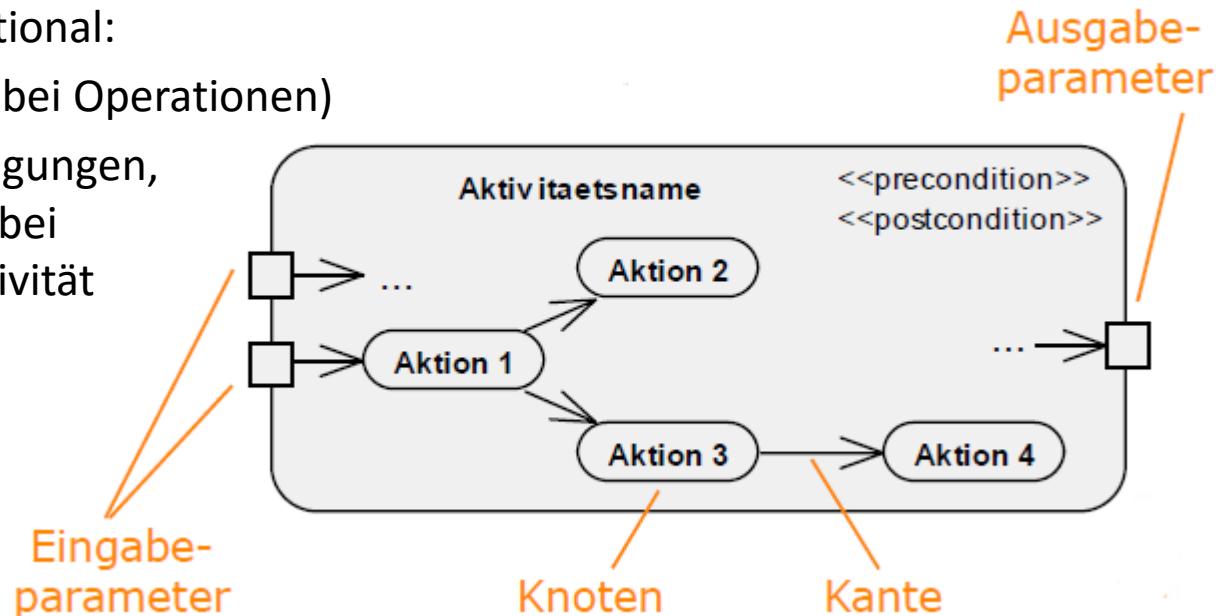
# Zweck

- Fokus: prozedurale Verarbeitung
- Spezifikation von Kontroll- und/oder Datenfluss zwischen Arbeitsschritten (Aktionen) zur Realisierung einer Aktivität
- Aktivitätsdiagramm in UML2:
  - ablauforientierte Sprachkonzepte
  - basierend u.a. auf Petri-Netzen und BPEL4WS
- Sprachkonzepte und Notationsvarianten decken ein breites Anwendungsgebiet ab
  - Modellierung objektorientierter und nichtobjektorientierter Systeme wird gleichermaßen unterstützt
  - Neben vorgeschlagener grafischer Notation sind auch beliebige andere Notationen (z.B. Pseudocode) erlaubt

Quelle: [http://www.uml.ac.at/wp-content/uploads/teaching/05\\_Aktivitaetsdiagramm\\_FolienMitText.pdf](http://www.uml.ac.at/wp-content/uploads/teaching/05_Aktivitaetsdiagramm_FolienMitText.pdf)

# Aktivität

- Eine Aktivität ist ein gerichteter Graph
  - Knoten: Aktionen
  - Kanten: Kontroll- und Datenflüsse
- Kontroll- und Datenflüsse legen potentielle »Abläufe« fest
- Spezifikation von benutzerdefiniertem Verhalten auf unterschiedlichen Granularitätsebenen optional:
  - Parameter (z.B. wie bei Operationen)
  - Vor- und Nachbedingungen, die bei Beginn bzw. bei Beendigung der Aktivität gelten müssen

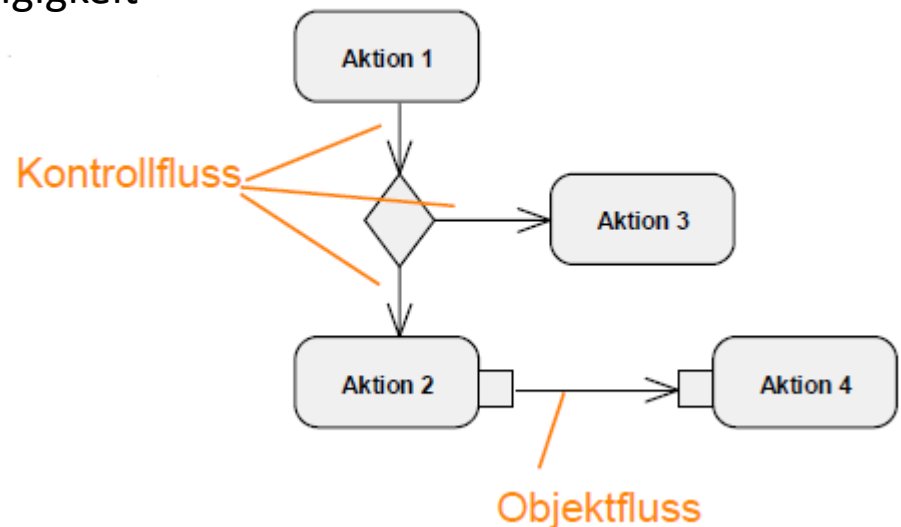


# Aktion

- Elementare Bausteine für beliebiges benutzerdefiniertes Verhalten
- Atomar, können aber abgebrochen werden
- Sprachunabhängig, allerdings Definition in beliebiger Programmiersprache möglich
- Aktionen können Eingabewerte zu Ausgabewerten verarbeiten
- Spezielle Notation für bestimmte Aktionsarten
- Kategorisierung der 44 in UML vordefinierten Aktionen:
  - Kommunikationsbezogene Aktionen (z.B. Signale und Ereignisse)
  - Objektbezogene Aktionen (z.B. Erzeugen und Löschen von Objekten)
  - Strukturmerkmals- und variablenbezogene Aktionen (z.B. Setzen und Löschen einzelner Werte von Variablen)
  - Linkbezogene Aktionen (z.B. Erzeugen und Löschen von Links zwischen Objekten sowie Navigation)

# Kanten

- Kanten verbinden Knoten und legen mögliche Abläufe einer Aktivität fest
  - Kontrollflusskanten
    - Drücken eine reine Kontrollabhängigkeit zwischen Vorgänger- und Nachfolgerknoten aus
  - Objektflusskanten
    - Transportieren zusätzlich Daten und drücken dadurch auch eine Datenabhängigkeit zwischen Vorgänger- und Nachfolgerknoten aus
- Überwachungsbedingung (guard)
  - Bestimmt, ob Kontroll- und Datenfluss weiterläuft oder nicht



# Start und Ende von Aktivitäten

- **Initialknoten**



- Beginn eines Aktivitätsablaufs
- Versorgt alle ausgehenden Kanten mit Kontrolltoken
- Aufbewahrung von Token erlaubt, da Überwachungsbedingungen die Weitergabe blockieren können
- Pro Aktivität keine oder mehrere Initialknoten erlaubt – letzteres ermöglicht Nebenläufigkeit

- **Aktivitätsendknoten**



- Beendet alle Abläufe einer Aktivität sowie den Lebenszyklus eines Objekts
- Der erste Token, der zu einem Endknoten gelangt, beendet die Aktivität (egal, wie viele Kanten in den Knoten führen)
- Keine Ausführung weiterer Aktionen
- Kontrolltoken werden gelöscht, Datentoken an Ausgabepins der Aktivität dagegen nicht
- Pro Aktivität mehrere Aktivitätsendknoten erlaubt

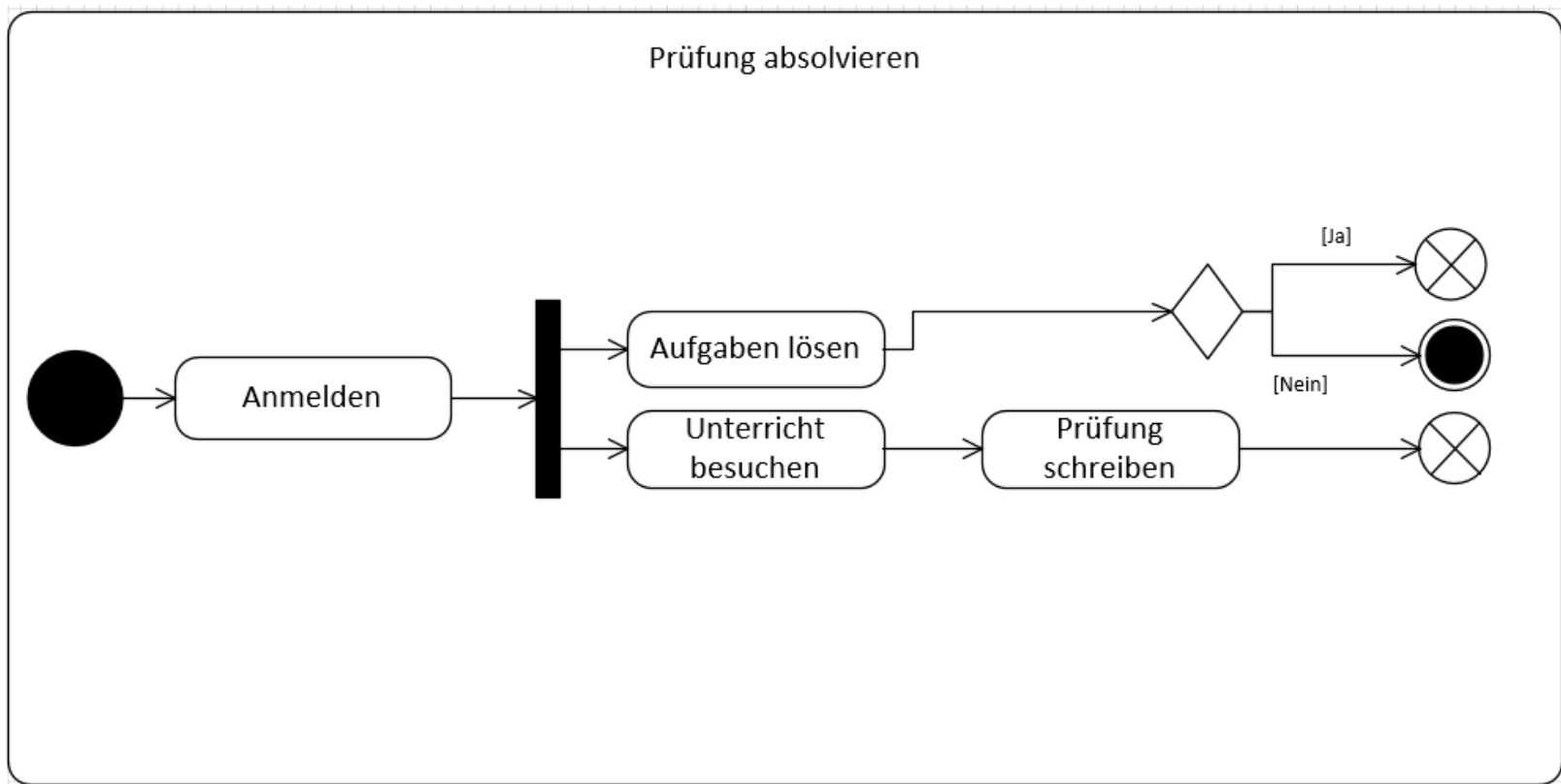
- **Ablaufendknoten**



- Beendet einen Ablauf einer Aktivität

# Beispiel

- Die Aktivität „Prüfung absolvieren“ besteht aus den Aktionen „Anmelden“, „Aufgaben lösen“, „Unterricht besuchen“ und „Prüfung schreiben“.



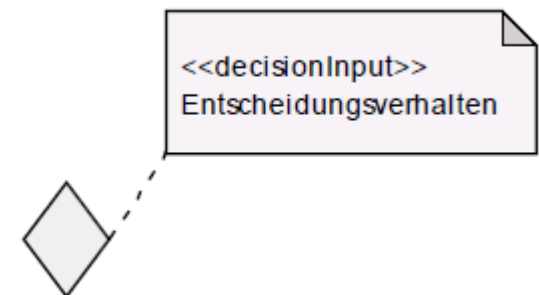
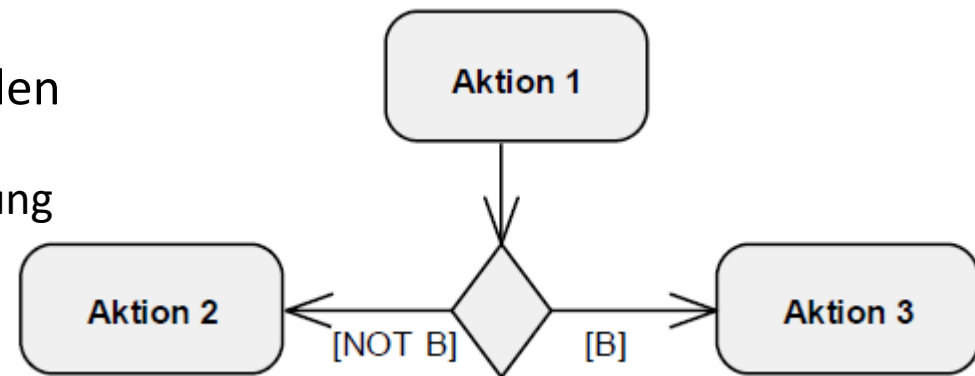


# Token

- „Virtueller Koordinationsmechanismus“ zur Beschreibung von Aktivitätsabläufen
- Vorgabe für die Implementierung einer Aktivität
- Token beschreibt möglichen Ablauf einer Aktivität zur Laufzeit
- Token fließen entlang der Kanten von Vorgänger- zu Nachfolgerknoten
  - Werden angeboten und aufbewahrt
  - Lösen Verarbeitung aus
- Unterscheidung in Kontroll- und Datentoken
  - Kontrolltoken: "Ausführungserlaubnis" für den Nachfolgeknoten
  - Datentoken: Transport von Datenwert oder Referenz auf Objekt
- Überwachungsbedingung kann Weitergabe von Token verhindern (Ansammlung mehrerer Token im Vorgängerknoten)

# Alternative Abläufe - Entscheidungsknoten

- Definiert alternative Zweige und repräsentiert eine »Weiche« für den Tokenfluß
  - Verwendung auch zur Modellierung von Schleifen
- Überwachungsbedingungen
  - Wählen den Zweig aus
  - Müssen wechselseitig ausschließend sein
  - [else] ist vordefiniert
- Entscheidungsverhalten
  - Ermöglicht detailliertere Spezifikation der Auswahlentscheidung an zentraler Stelle
  - Ankunft von Token startet das Entscheidungsverhalten – Datentoken fungieren als Parameter

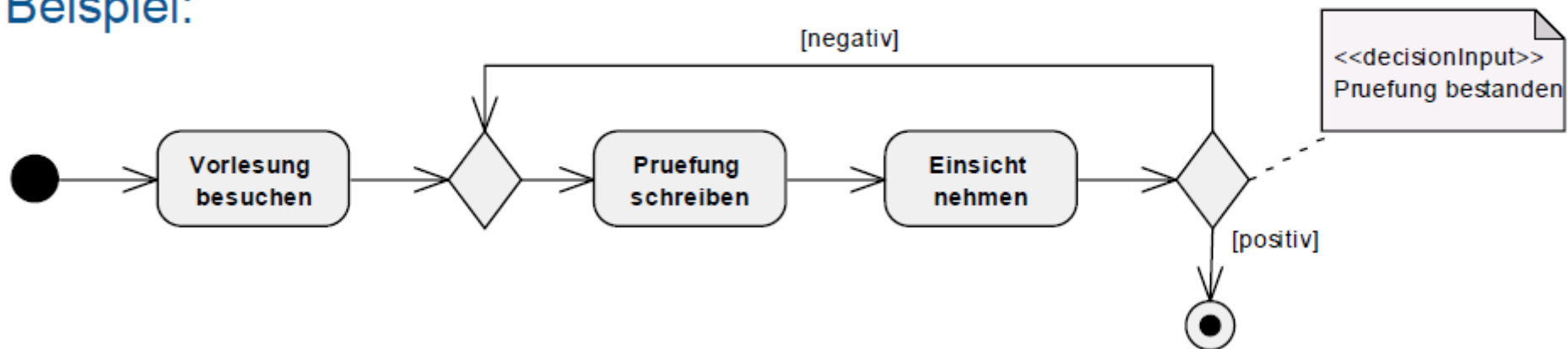


# Aufgabe 1

- Modellieren Sie ein Aktivitätsdiagramm mit folgenden Inhalten. Die Aktivität „Uni Prüfung absolvieren“ besteht aus den Aktionen „Vorlesung besuchen“, „Prüfung schreiben“ und „Einsicht nehmen“. Der Datentoken „Prüfung bestanden“ beschreibt bei „positivem“ Inhalt das Ende der Aktionen. Sollte dies „negativ“ sein, muss die Prüfung erneut geschrieben werden.

# Lösung

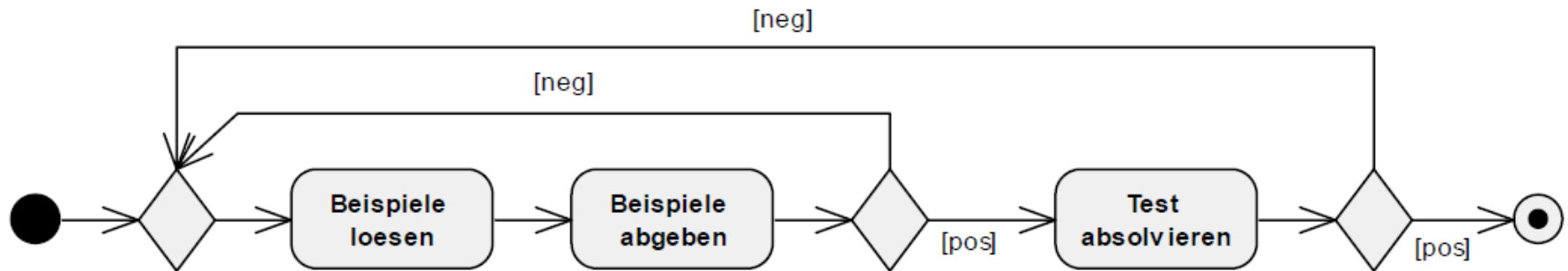
Beispiel:



# Aufgabe 2

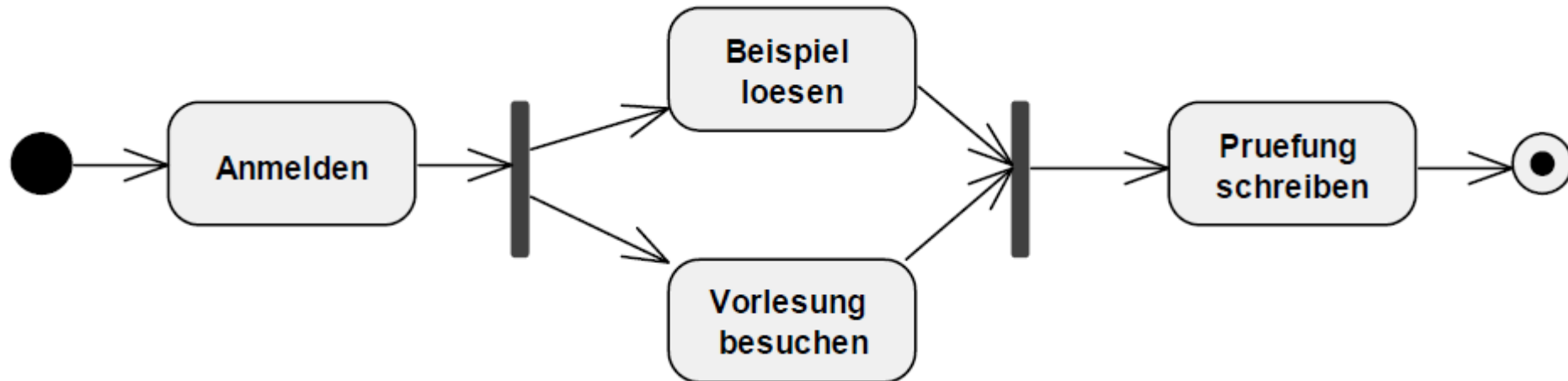
- Modellieren Sie ein Aktivitätsdiagramm mit folgenden Inhalten. Die Aktionen „Beispiele lösen“ um dann „Beispiele abgeben“ resultiert in einen „Test absolvieren“. Sollten die Beispiele nicht korrekt [negativ] gelöst sein, müssen weitere Beispiele gelöst werden bevor man wieder einen Test absolvieren kann. Ist der Test absolviert kann dieser positiv und negativ ausfallen. Bei positivem Ausgang ist die Aktivität abgeschlossen, bei negativem Ausgang müssen weitere Beispiele gelöst werden.

# Lösung



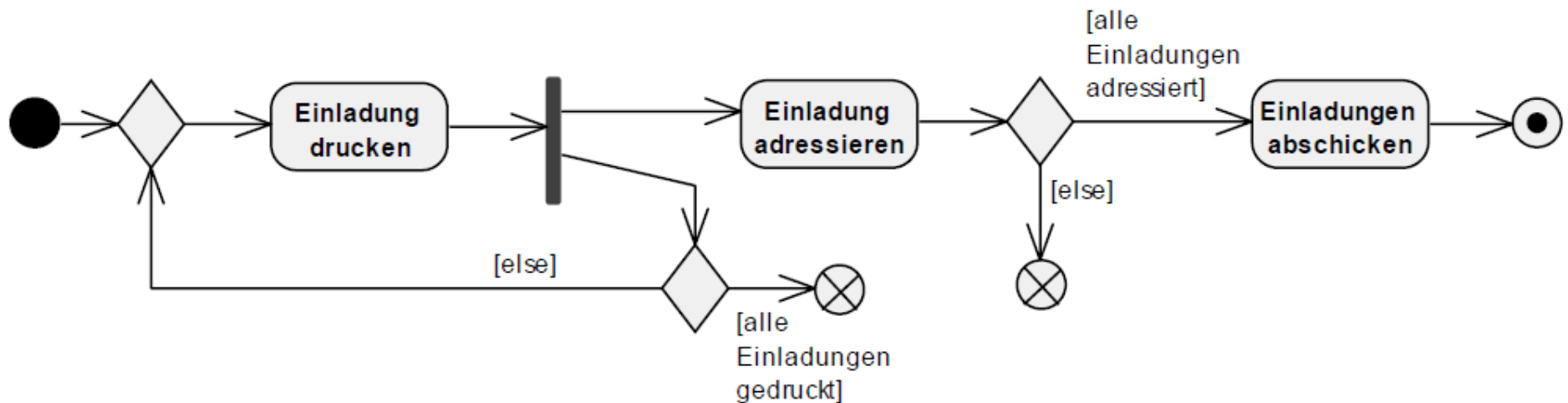
# Parallelisierungsknoten

- Modellierung der Aufspaltung von Abläufen ODER führt nebenläufige Abläufe zusammen (Split/Join)



# Beispiel

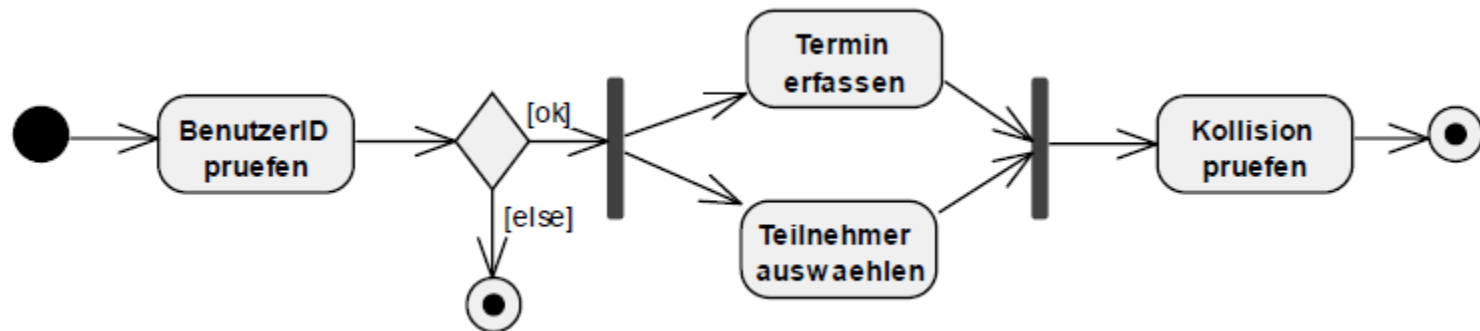
- Erstellen und Versenden von Einladungen zu einem Termin!





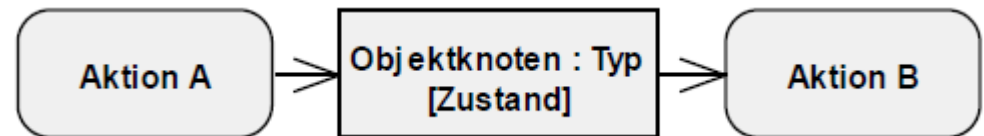
# Aufgabe

- Beschreiben Sie folgendes Aktivitätsdiagramm!

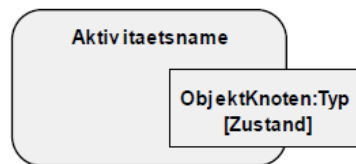


# Objektknoten

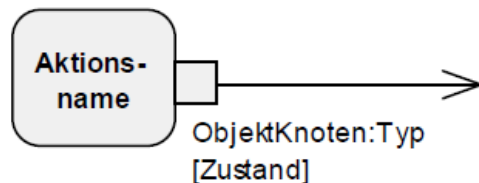
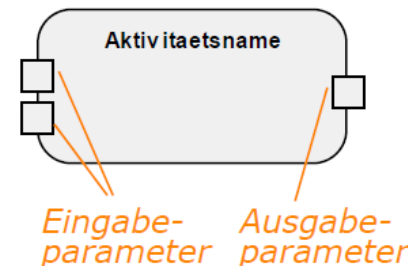
- Inhalt: Datentoken
- Objektknoten stehen durch Objektflüsse miteinander in Beziehung
- Inhalt ist Ergebnis einer Aktion und Eingabe für eine weitere Aktion
- Typangabe und Zustandseinschränkung sind optional
- Objektknoten als Ein-/Ausgabeparameter



– Aktivitäten

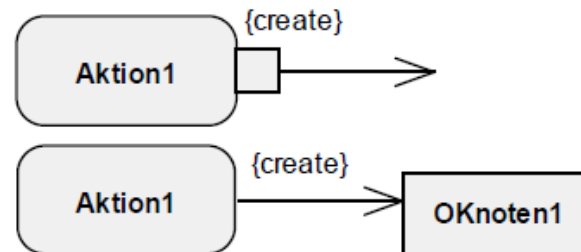


– Aktionen

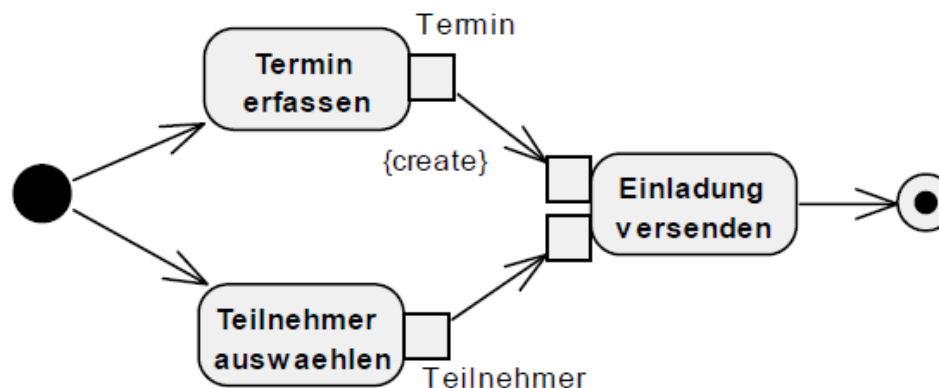


# Objektknoten

- Effekte einer Aktion auf z.B. Daten/Objekte
  - create, read, update und delete

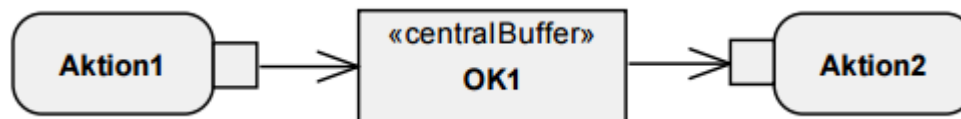


- Beispiel für Ein- und Ausgabepins und Effekte



# Objektknoten

- Konstanter Eingabewert – Wertepin
  - Zur Übergabe konstanter Werte
  - Startet nicht die Verarbeitung eines Knotens
- Pufferknoten
  - Zentrale Pufferung von Datentoken
  - Transienter Pufferknoten (central buffer node)
    - Löscht Datentoken, sobald er sie weitergegeben hat

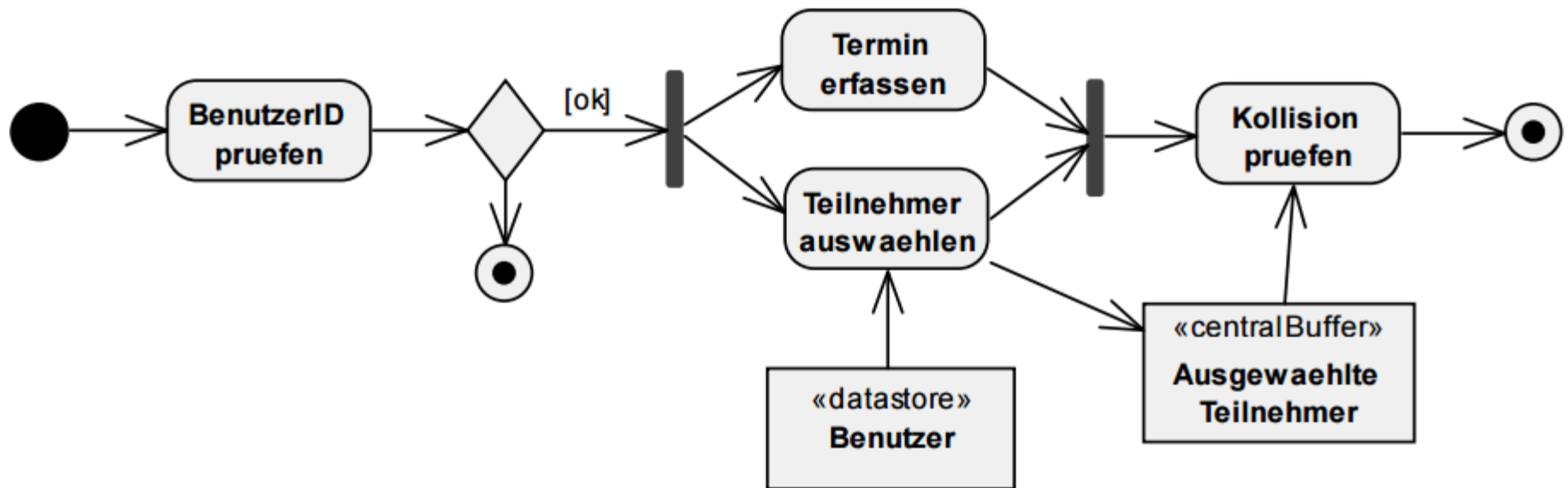


- Persistenter Pufferknoten (data store node)
  - Bewahrt Datentoken auf und gibt Duplikate weiter
  - Keine Mehrfachspeicherung identer Objekte
  - Explizites »Abholen« der Datentoken möglich



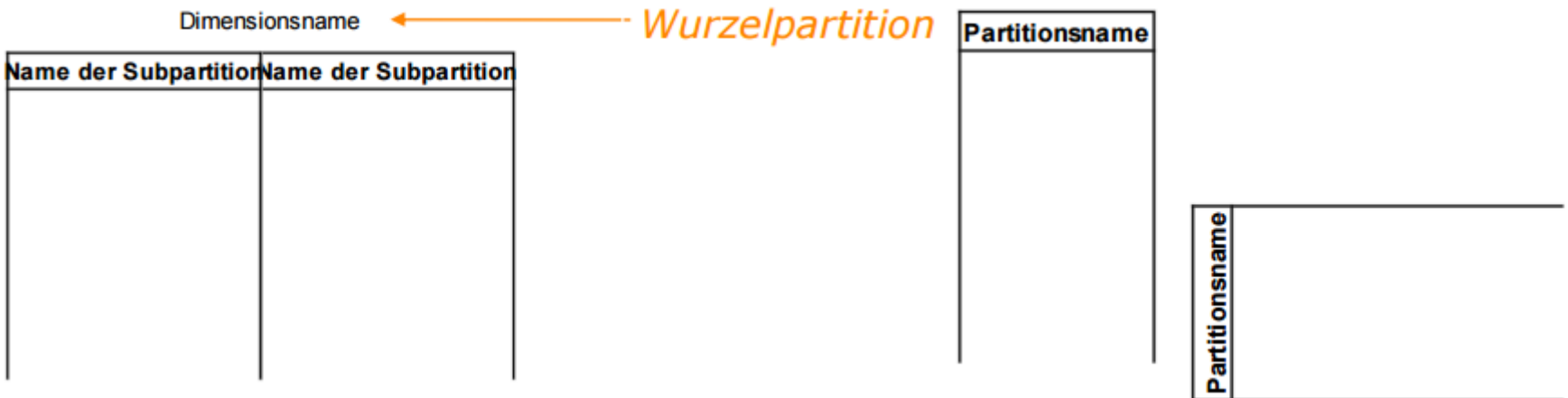
# Beispiel: Puffer

- alle Benutzer werden im Puffer "Benutzer" gespeichert (Datenbank)
- Ausgewählte Benutzer werden in einem transienten Puffer zwischengespeichert und erst für die Versendung von Einladungen wieder entnommen.

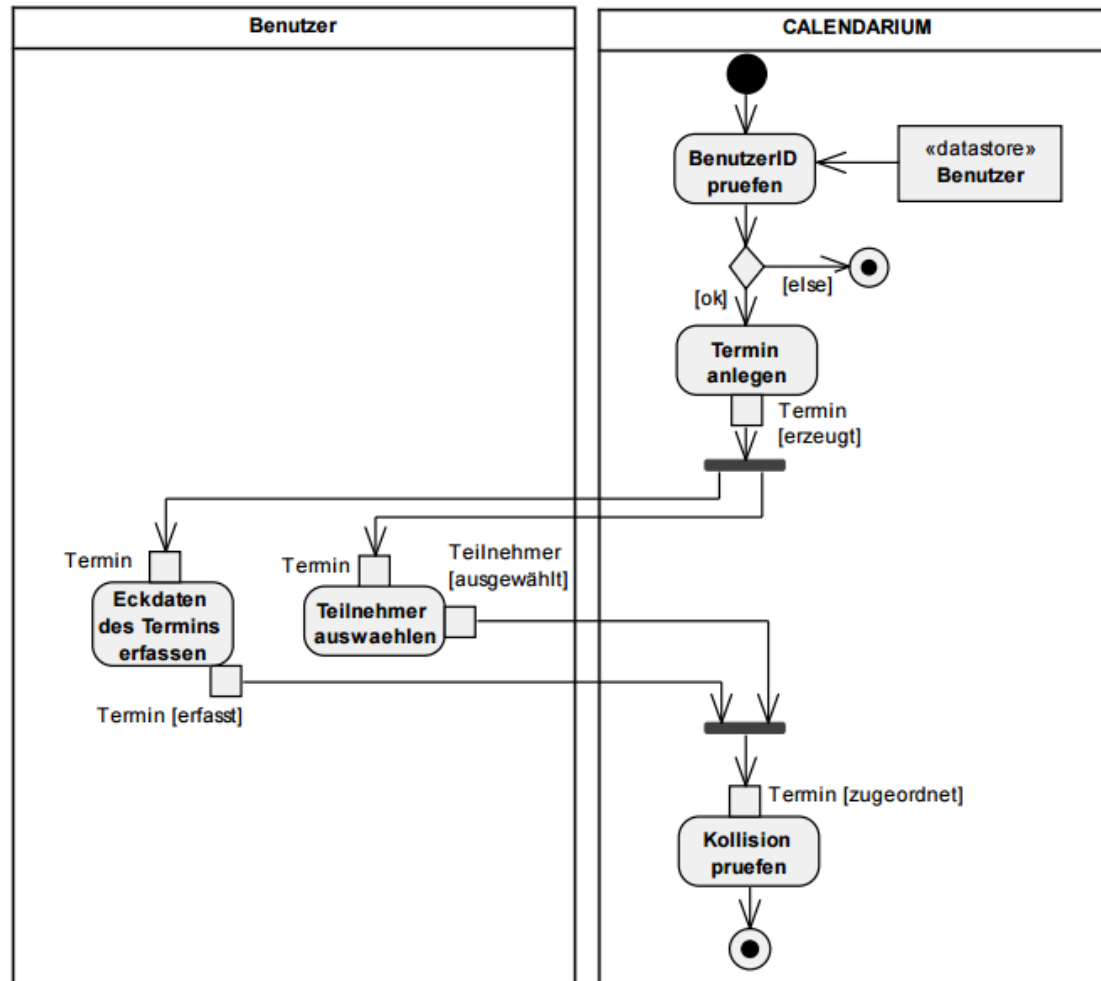


# Partitionen

- Erlauben die Gruppierung von Knoten und Kanten einer Aktivität nach bestimmten Kriterien
- Logische Sicht auf eine Aktivität zur Erhöhung der Übersichtlichkeit und Semantik des Modells
- »Schwimmbahnen«-Notation (partitions, swimlanes)
- Hierarchische Partitionen
  - Zur Schachtelung auf verschiedenen Hierarchieebenen



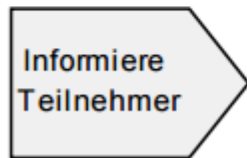
# Beispiel



# Signale und Ereignisse

## Sonderformen von Aktionen

### Senden von Signalen:

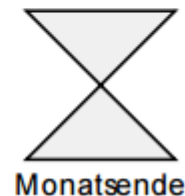


### Empfangen von Ereignissen:

#### Asynchrones Ereignis



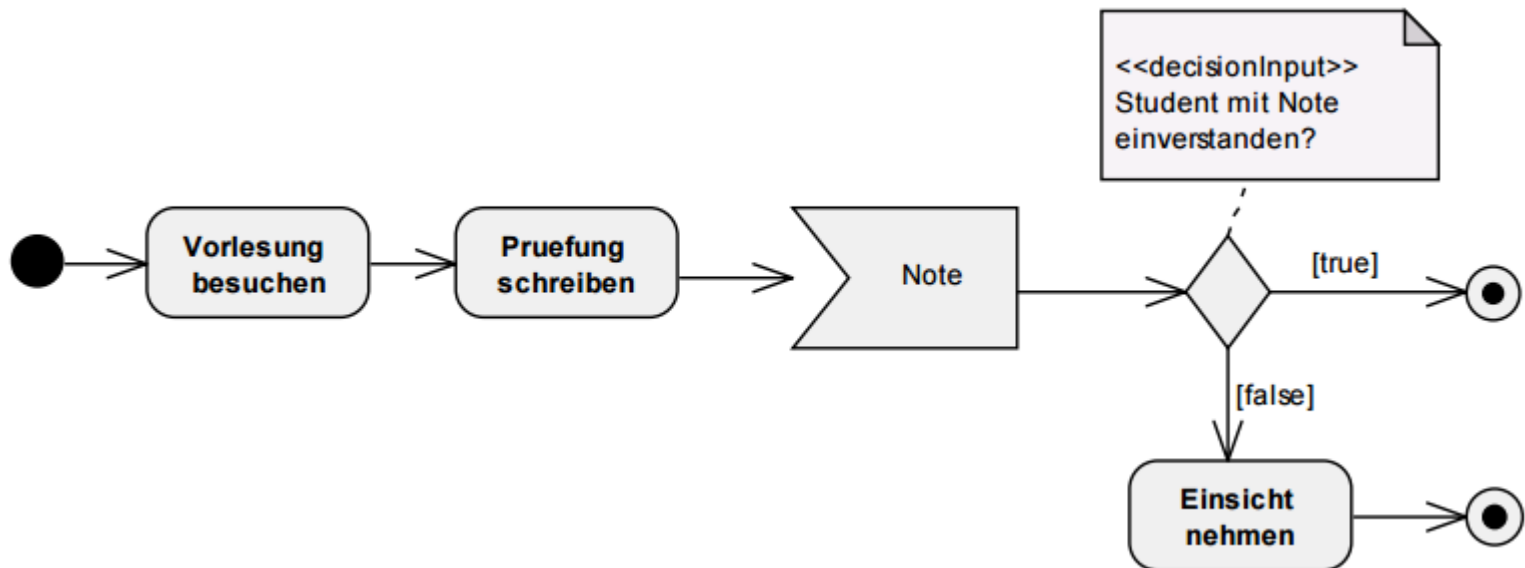
#### Asynchrones Zeitereignis





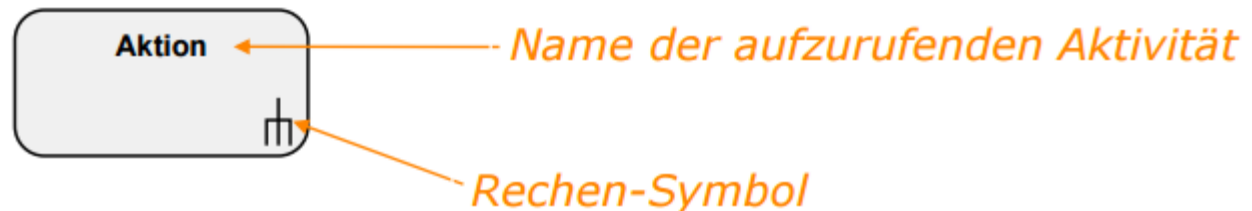
# Asynchrones Ereignis

- Um eine Vorlesung zu absolvieren, besucht ein Student zuerst die Vorlesung, dann schreibt er eine Prüfung und wartet auf die Note. Der Student wird informiert, sobald die Note verfügbar ist. Nachdem der Student die Note erfahren hat, besteht die Möglichkeit, Einsicht zu nehmen.



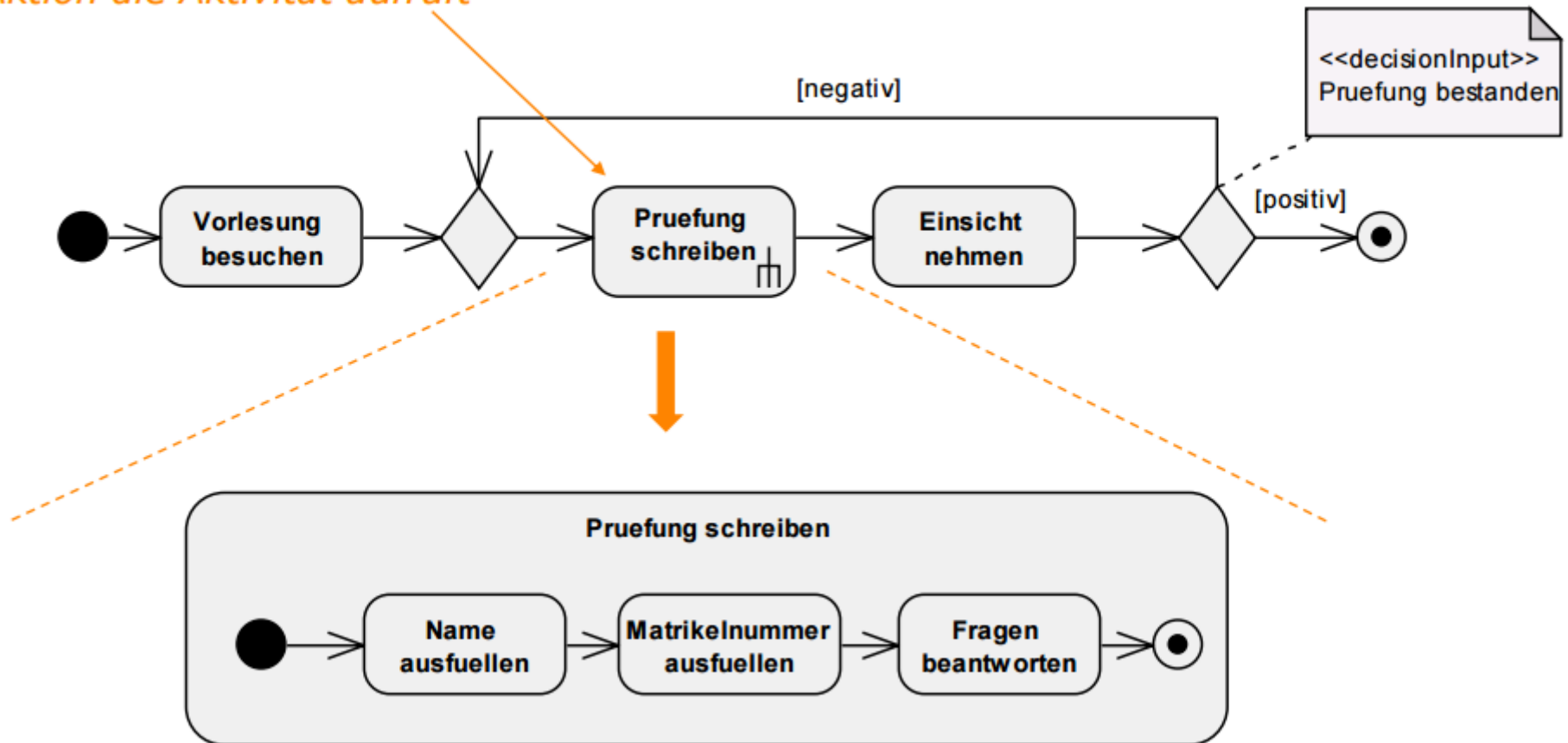
# Schachtelung von Aktivitäten

- Aktivitäten können wiederum Aktivitäten aufrufen
- So können Details in eine tiefere Ebene ausgelagert werden
- Vorteile:
  - Bessere Lesbarkeit
  - Wiederverwendung
- Notation:
  - In einer Aktion wird eine Aktivität aufgerufen



# Beispiel

*Aktion die Aktivität aufruft*



# Ausnahmebehandlung (Exception Handling)

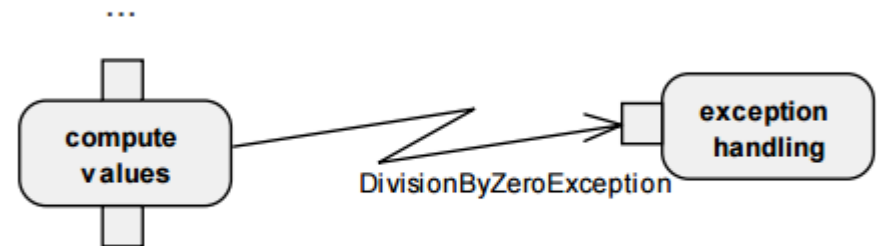
- Vordefinierte Ausnahmen, beispielsweise durch das Laufzeitsystem
- (z.B. Division durch 0)
- Benutzerdefinierte Ausnahmen
  - `RaiseExceptionAction`
- Behandlung einer Ausnahme durch dezidierten Ausnahmebehandlungsknoten – nach Abarbeitung der Ausnahme kann mit dem "normalen" Ablauf fortgefahren werden
- Der Ausnahmebehandlungsknoten substituiert den „geschützten“ Knoten und hat daher keine eigenständigen ausgehenden Kontroll- oder Objektflüsse



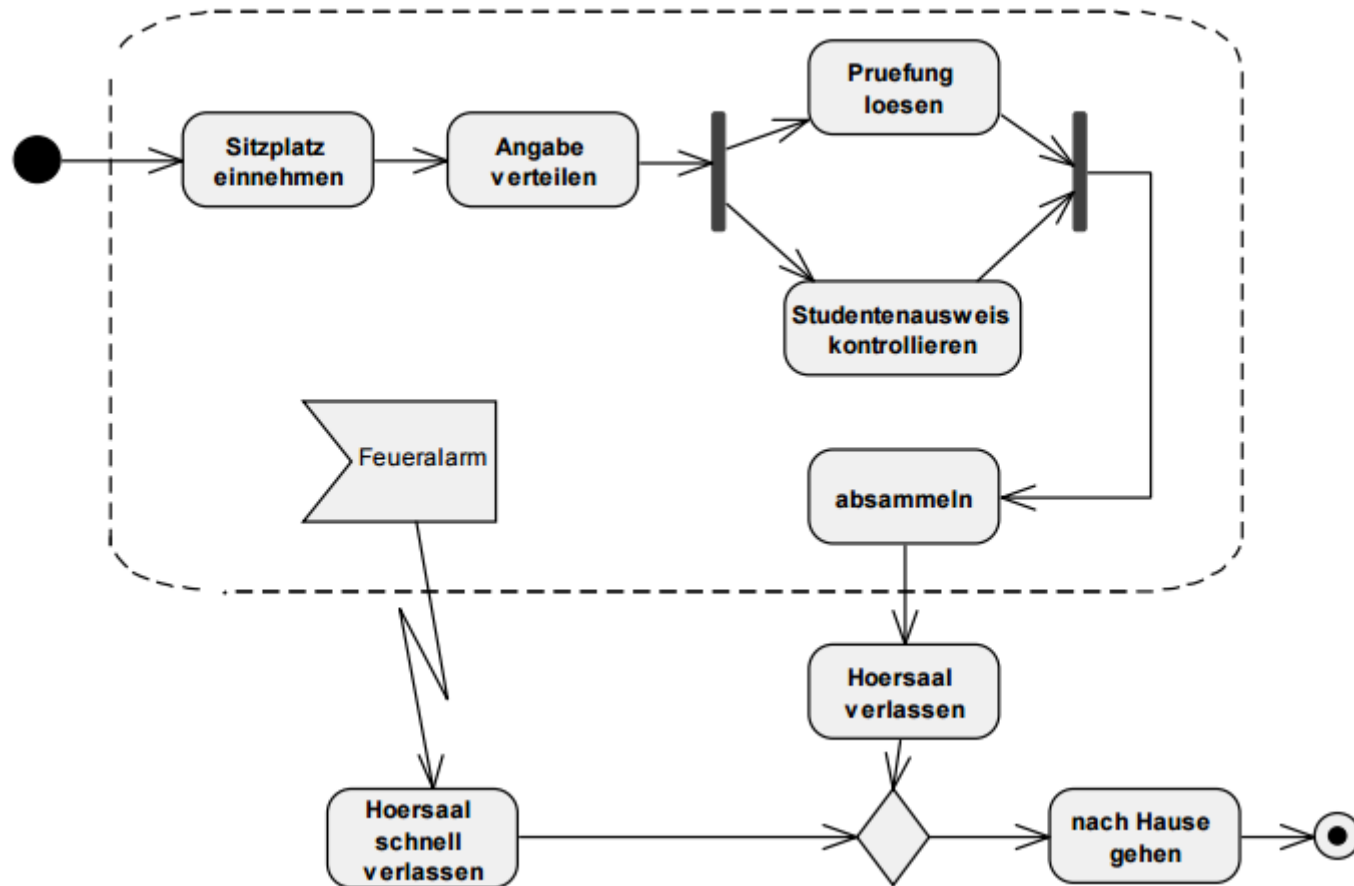
# Ausnahmebehandlung

- Existiert für einen Ausnahmetyp keine Ausnahmebehandlung, wird die betroffene Aktion beendet und die Ausnahme nach außen propagiert (d.h. es wird in der umgebenden Aktivität nach passender Ausnahmebehandlung gesucht)

```
try {  
    // compute values  
} catch (DivisionByZeroException) {  
    // exception handling  
}
```

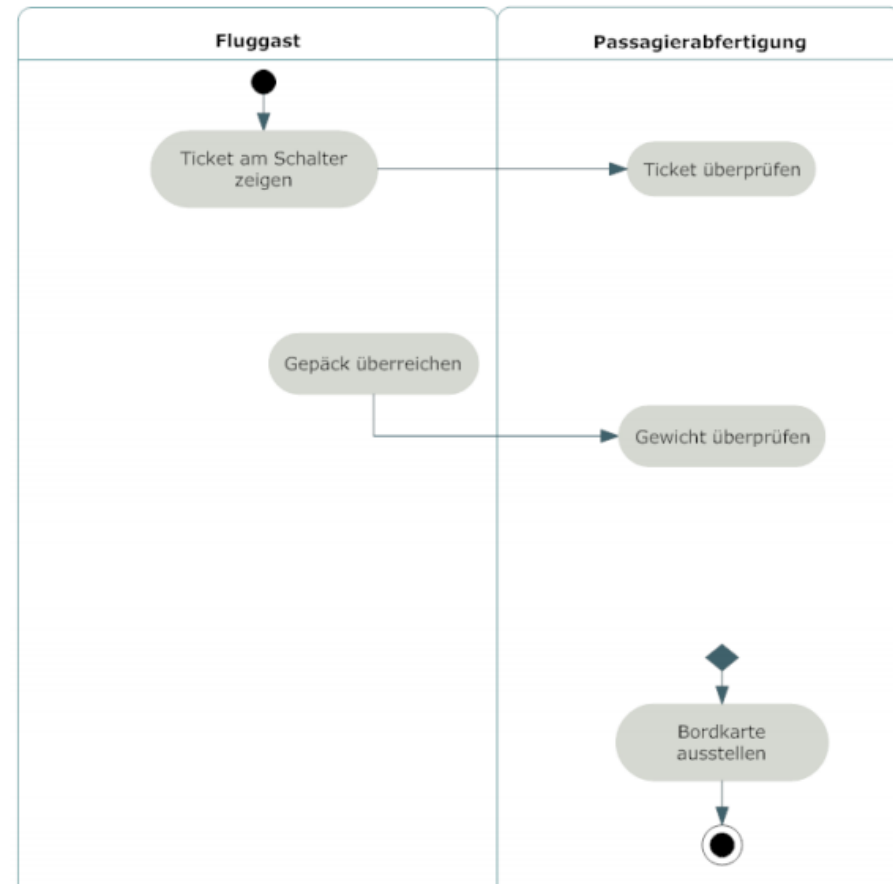


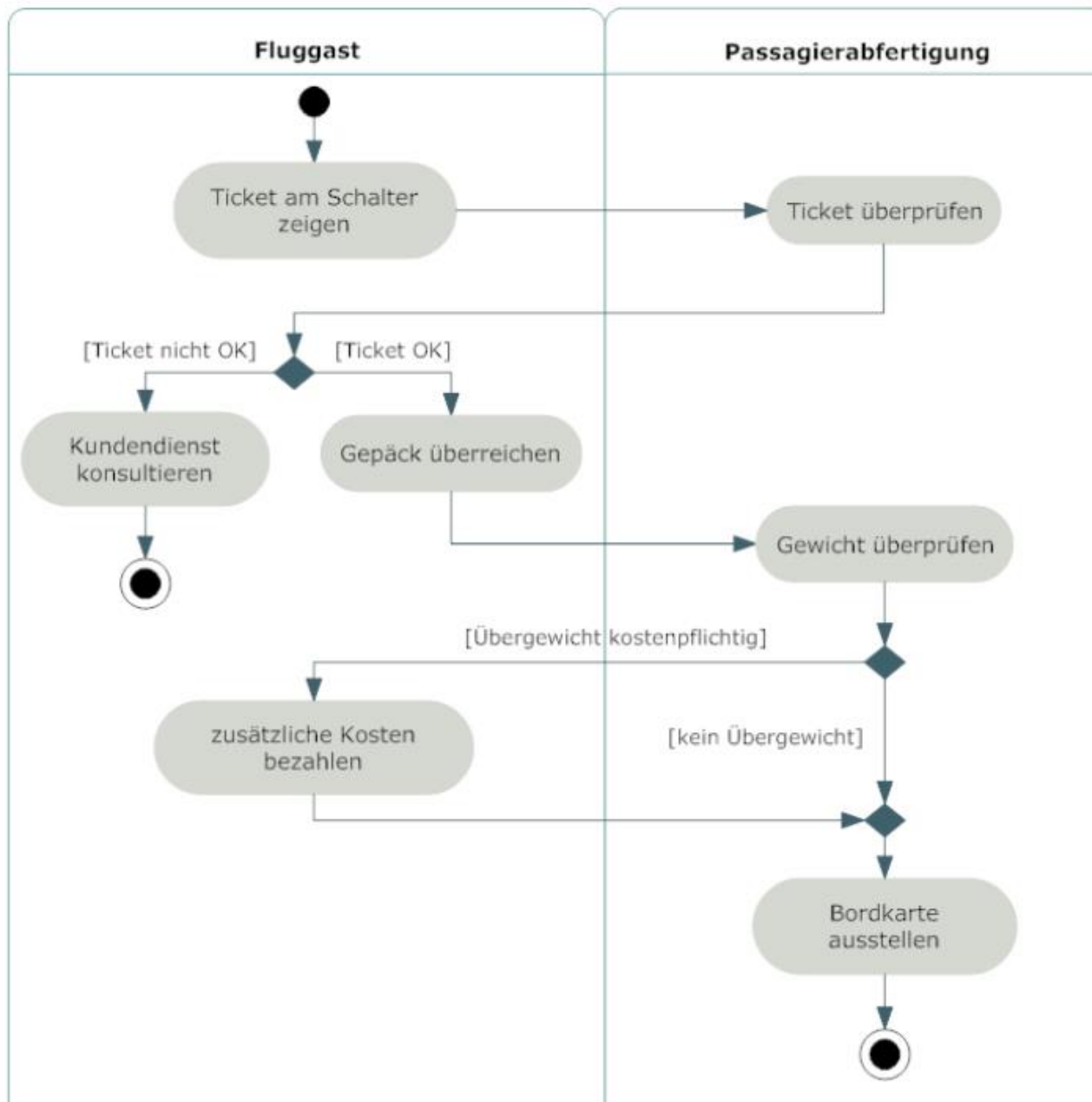
# Beispiel



# Aufgabe

- Ein Fluggast ist am Flughafen angekommen. Zur Überprüfung seines Tickets begibt er sich zum Schalter seiner Fluggesellschaft. Falls das Ticket in Ordnung ist, übergibt er am Schalter sein Gepäck. Falls mit dem Ticket etwas nicht stimmt, muss der Fluggast den Kundendienst konsultieren und er kann nicht mitfliegen. Das Gepäck wird zudem auf Übergewicht überprüft. Falls dem so ist, muss der Fluggast zusätzliche Kosten übernehmen. Falls aber das Gewicht in Ordnung ist, wird die Bordkarte ausgestellt. Modellieren Sie diesen Sachverhalt im folgenden unvollendeten UML-Aktivitätsdiagramm.







# Aufgabe

Zeichnen Sie das Aktivitätsdiagramm für die Organisation einer Klausur.  
Enthalten sein sollten mindestens:

- Die Teilnahme an den Übungen (erfolgreich / nicht erfolgreich)
- Die erfolgreiche Teilnahme führt zur Klausuranmeldung, dem Erfragen des Termins und sorgfältiger Vorbereitung in beliebiger Reihenfolge.
- Die Klausur selbst mit anschließender Abfrage der Ergebnisse im Internet.
- Eine einmalige Zulassung zur Nachklausur, wenn die Klausur im ersten Versuch nicht bestanden wurde.
- Die Nachholklausur erfordert eine erneute Terminabfrage und Vorbereitung.
- Ein Krankheitsfall kann ebenfalls eintreten. Mit einem Attest und einer anschließenden Genesung ist eine Wiederholung der jeweiligen Klausur möglich.

Beachten Sie, dass Aktivitäten auch parallel oder alternativ ablaufen können. Jeder der obigen Punkte kann dabei mehrere Aktionen umfassen, wenn es Ihnen angebracht erscheint.

# Lösung

