

# Programmiermethodik

## Übung 10

Sommersemester 2010  
Fachgebiet Software Engineering

Andreas Scharf  
andreas.scharf@cs.uni-kassel.de

# Agenda

- **Organisatorisches**
- **Vorstellung HA8**
- **Storydiagramme**
- **Praktische Übung: Castle**

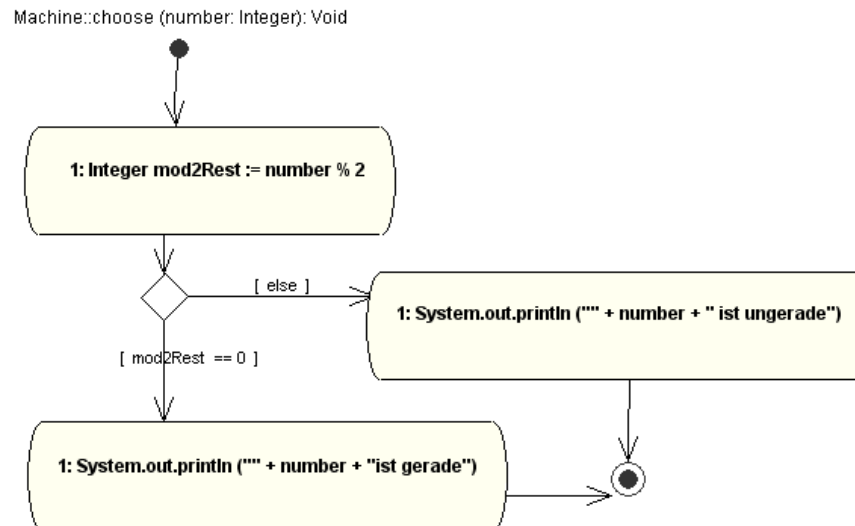
# Organisatorisches & Vorstellung HA 8

- Nächsten Dienstag (29.06.2010) Übung in HS 1332
- Hausaufgabentabelle aktualisiert
- **HA 8: Storyboards & Storydiagramme**
  - Aufgabe 1:
    - Storydiagramme von letzter HA abgeben
    - Ein neues zeichnen: Mancala::initGame(..) testen
  - Aufgabe 2-4: Storydiagramme zu
    - Mancala::initGame(...)
    - Mancala::checkEnd()
    - Pit::moveStones()

# Storydiagramme

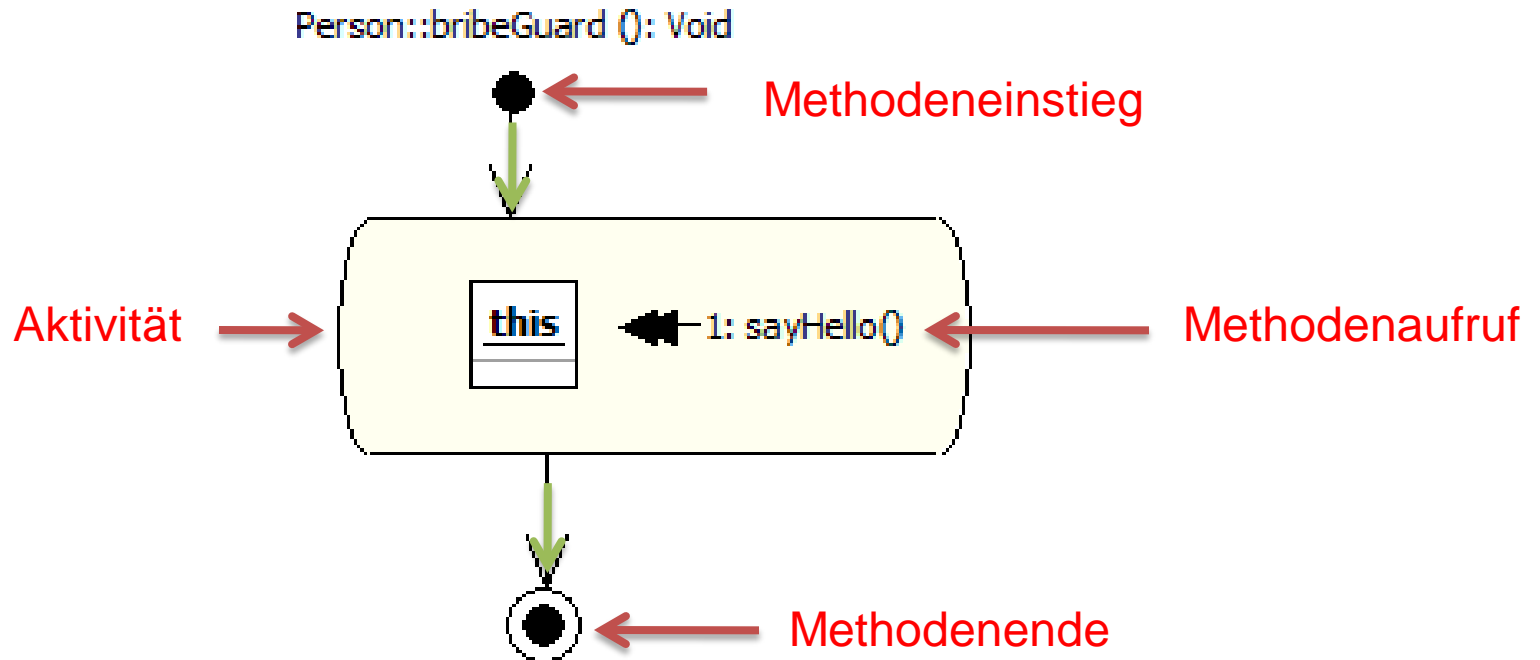
- **Warum?**

- Erhöhen das Abstraktionsniveau
- Man kann sich auf das Wesentliche konzentrieren
- Änderungen in Diagrammen sind meist leichter als im Code
- Nachvollziehbarkeit von Code geht in Diagrammen leichter
- ...



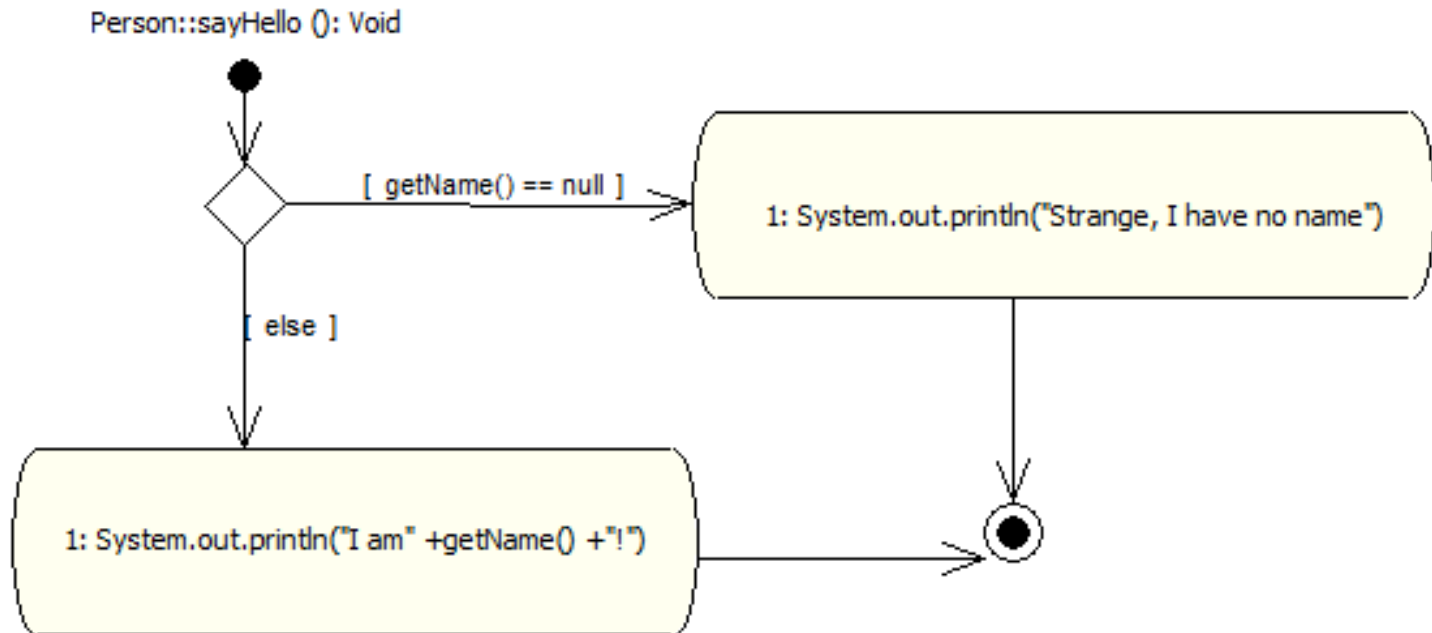
# Storydiagramme

- Kontrollfluss und Methodenaufrufe



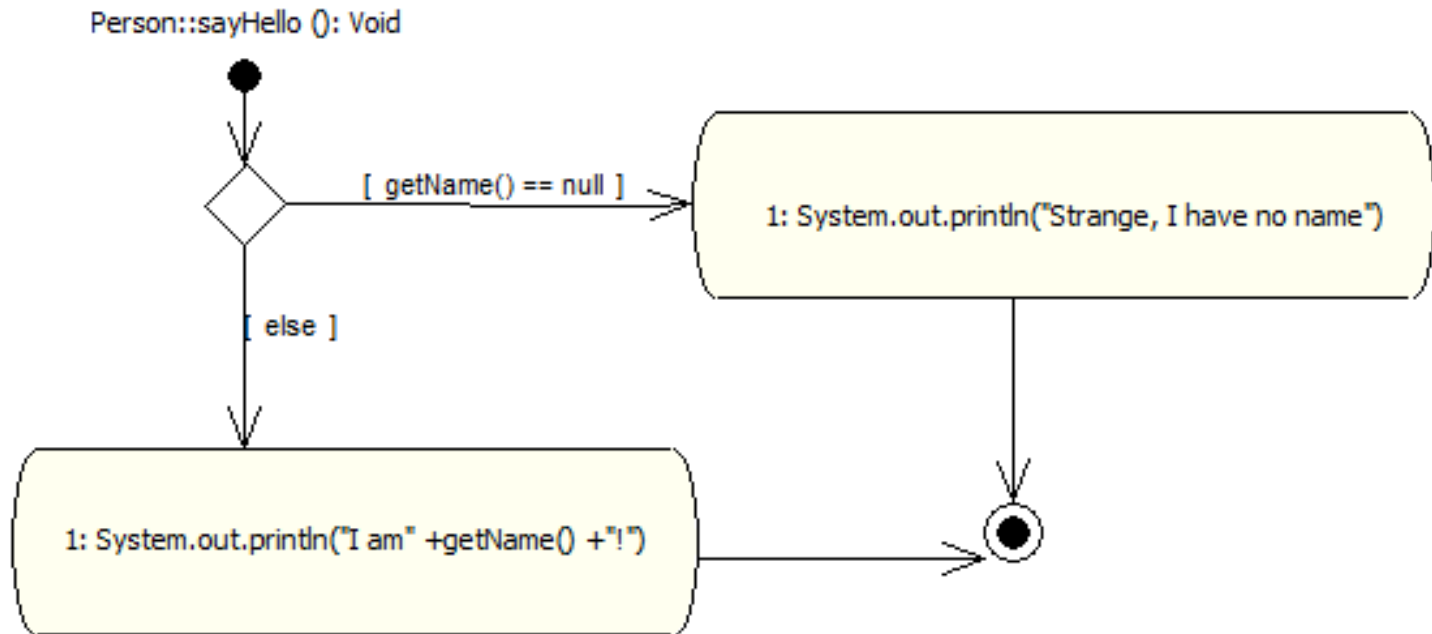
# Storydiagramme

- Fallunterscheidung



# Storydiagramme

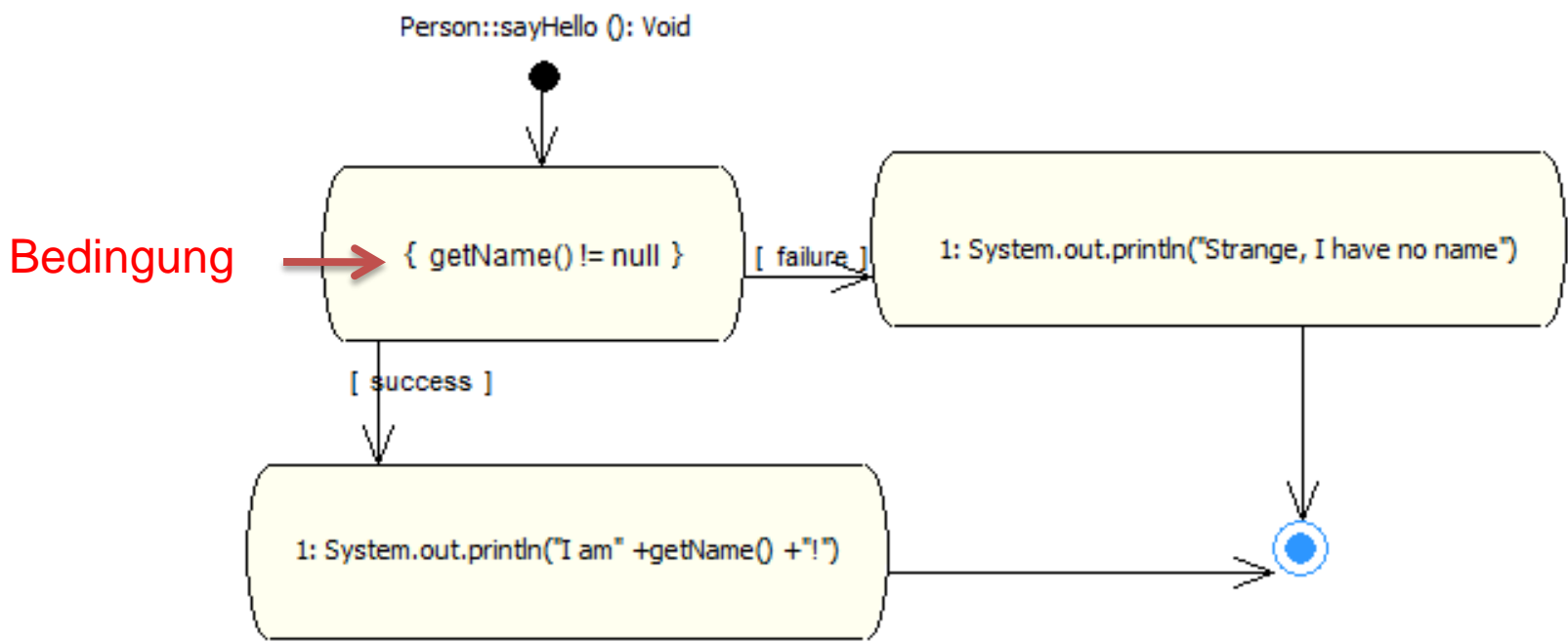
- Fallunterscheidung



- oder auch...

# Storydiagramme

- Fallunterscheidung

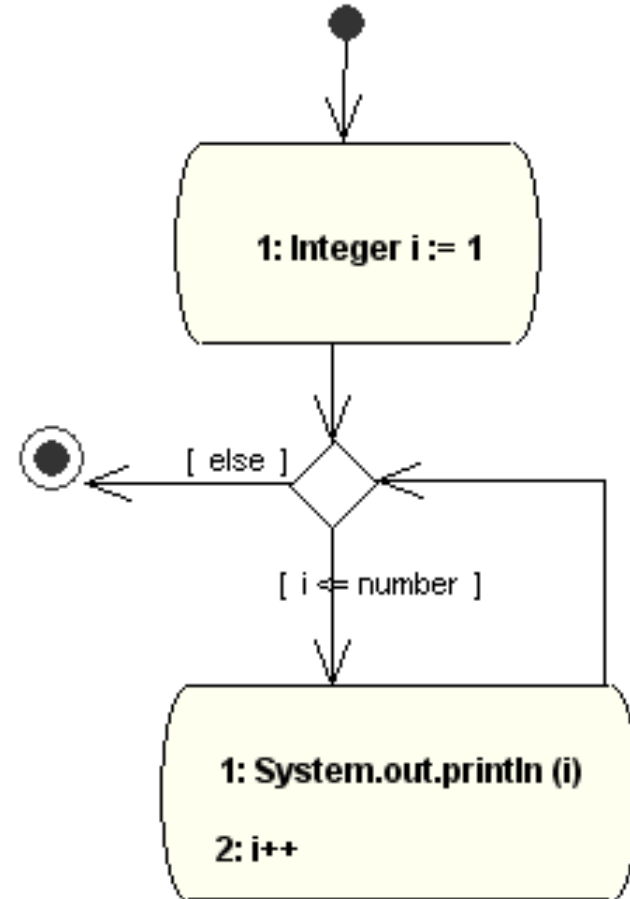




# Storydiagramme

- Zählschleife

Machine::countUp (number: Integer): Void

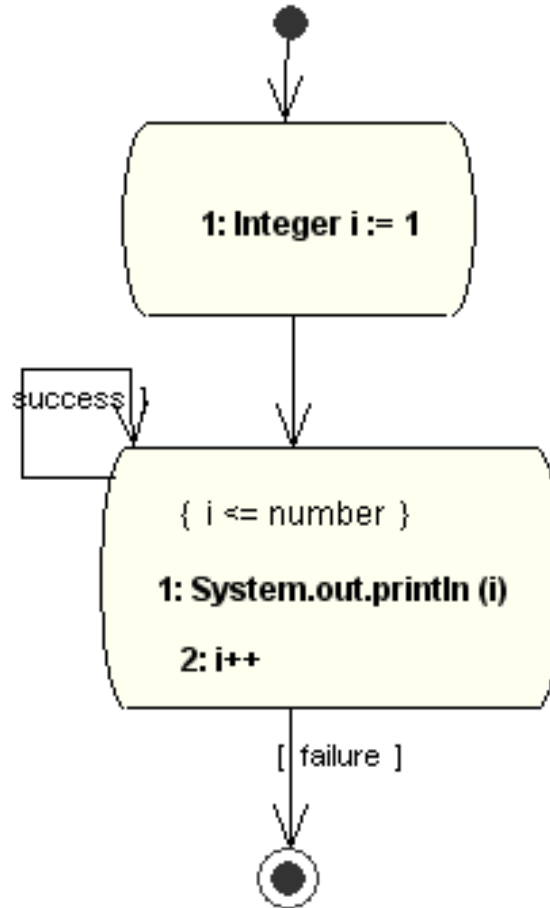


- oder auch...

# Storydiagramme

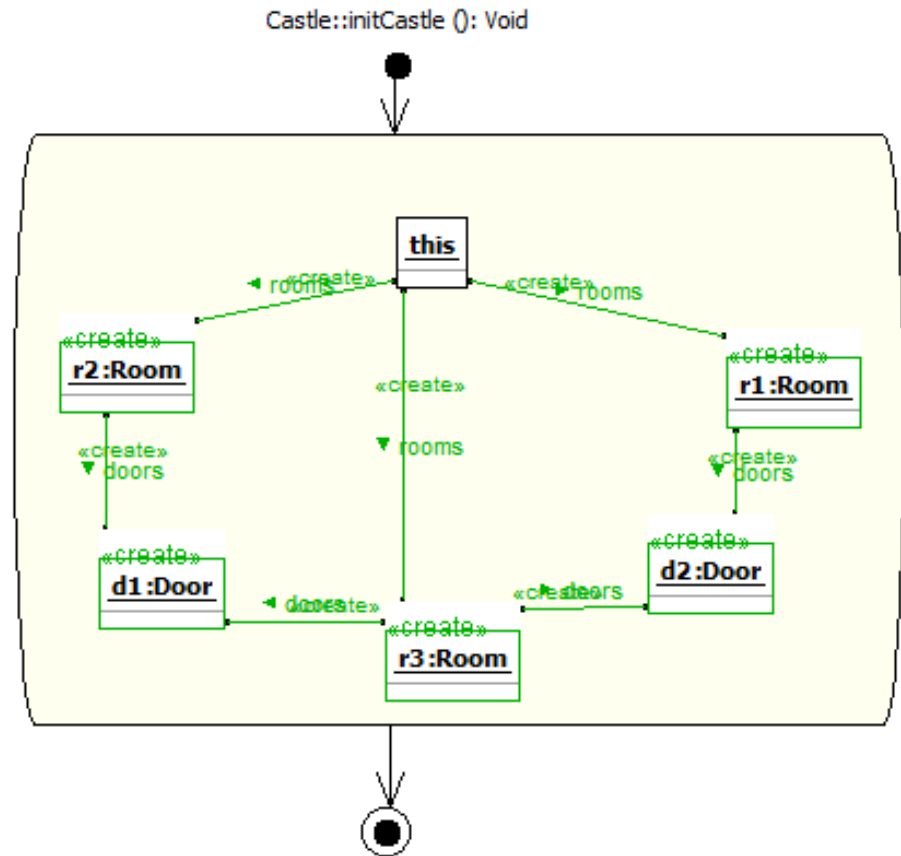
- Zählschleife

Machine::countUp (number: Integer): Void



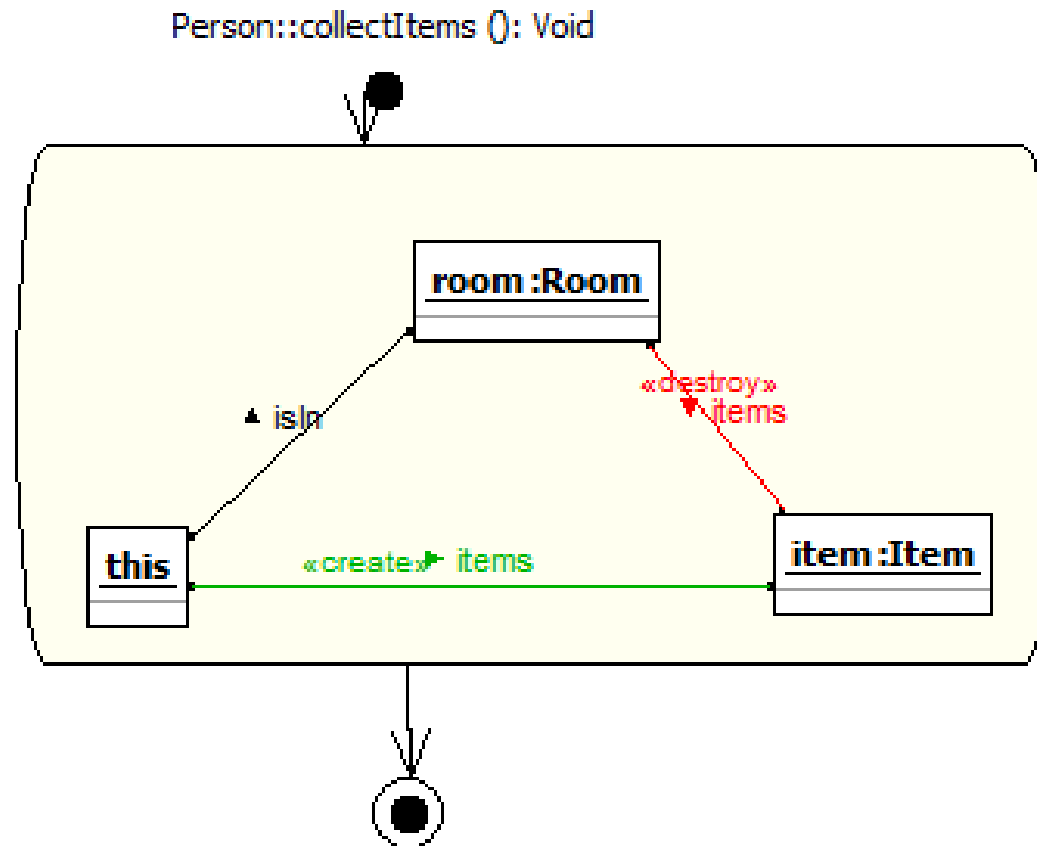
# Storydiagramme

- **Objekte/Links erzeugen**
  - Zuerst werden Objekte erzeugt
  - Dann Links



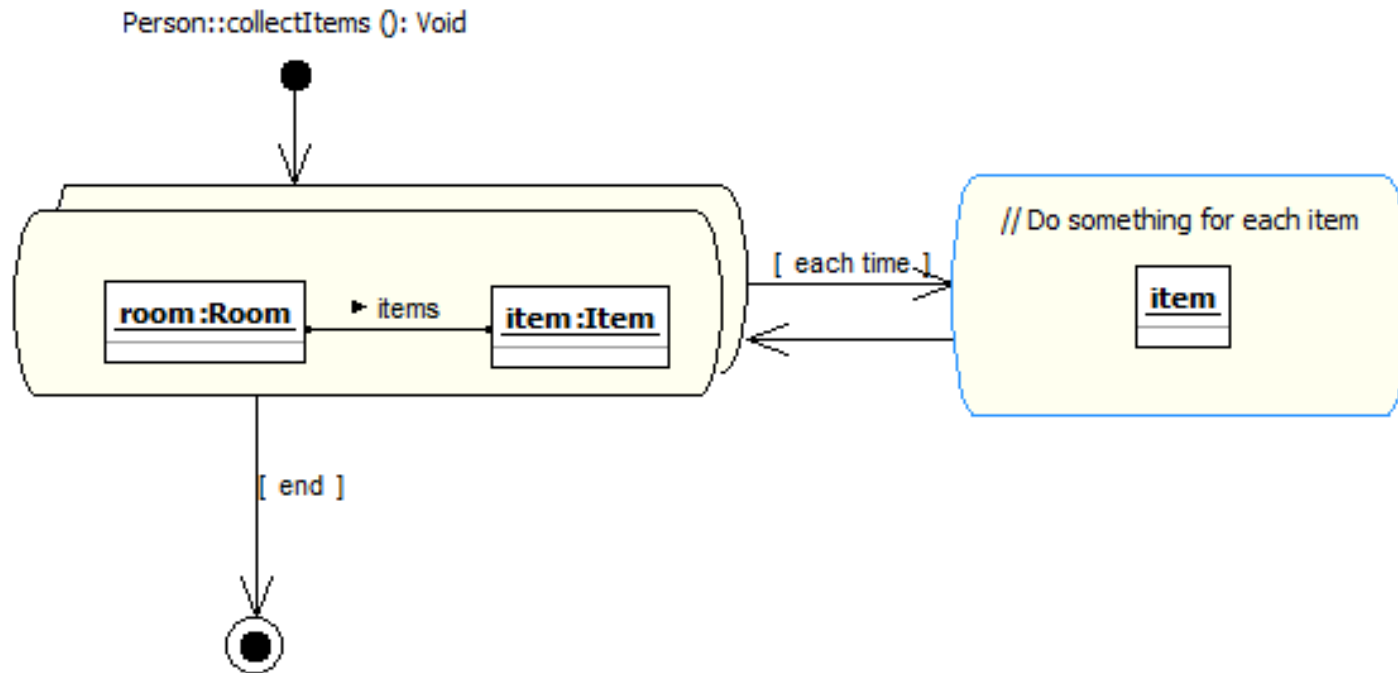
# Storydiagramme

- Objekte/Links zerstören



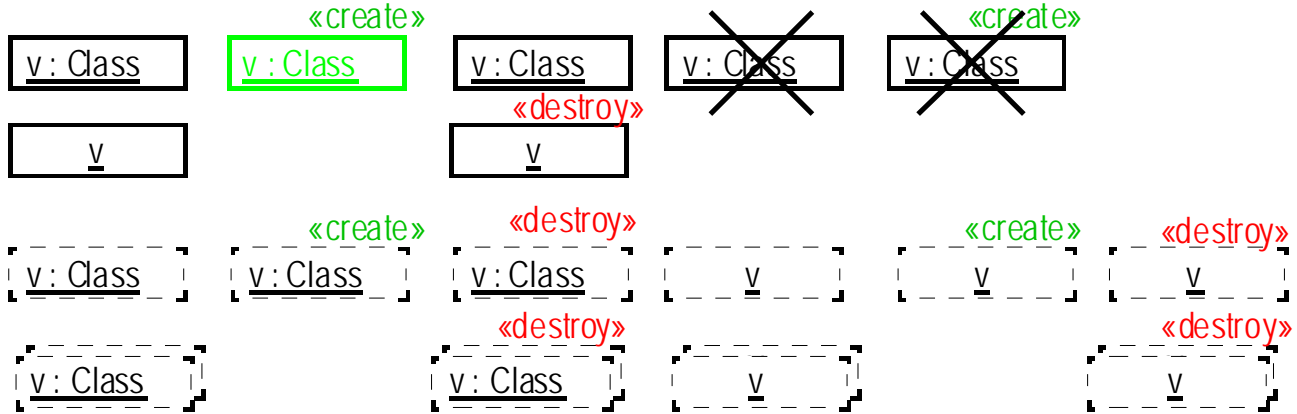
# Storydiagramme

- Mengen -> ForEach

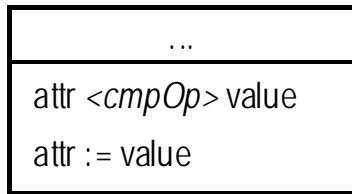


# Rule Syntax: Overview

Variables:



Attributes:



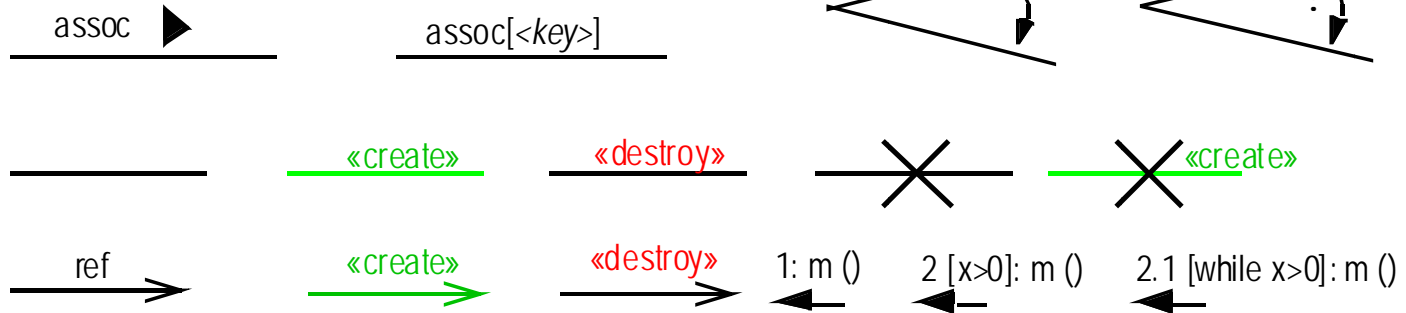
Constrains:

{ <boolExpr> }

{first}

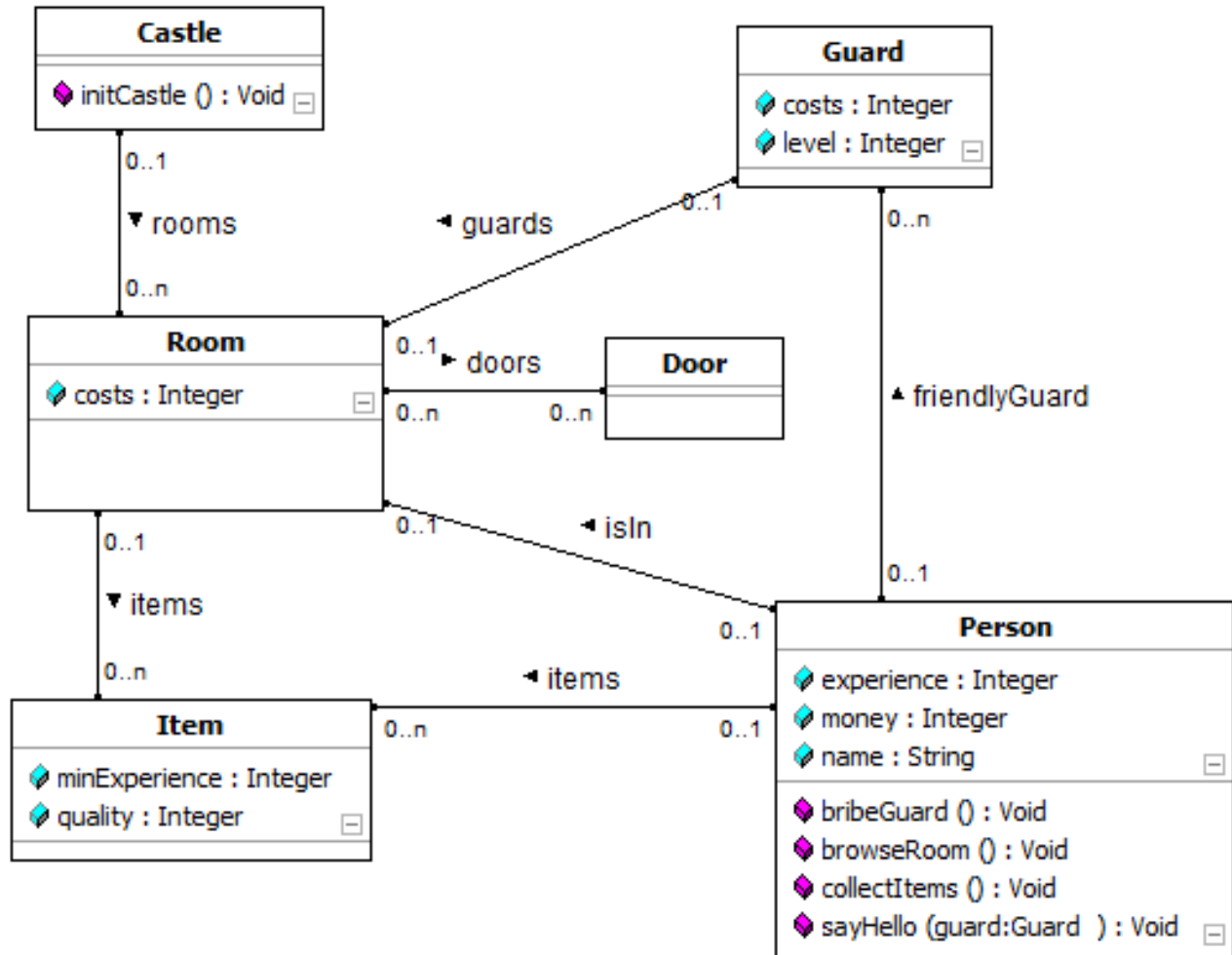
{last}

Links:



# Praktische Übung

- Castle



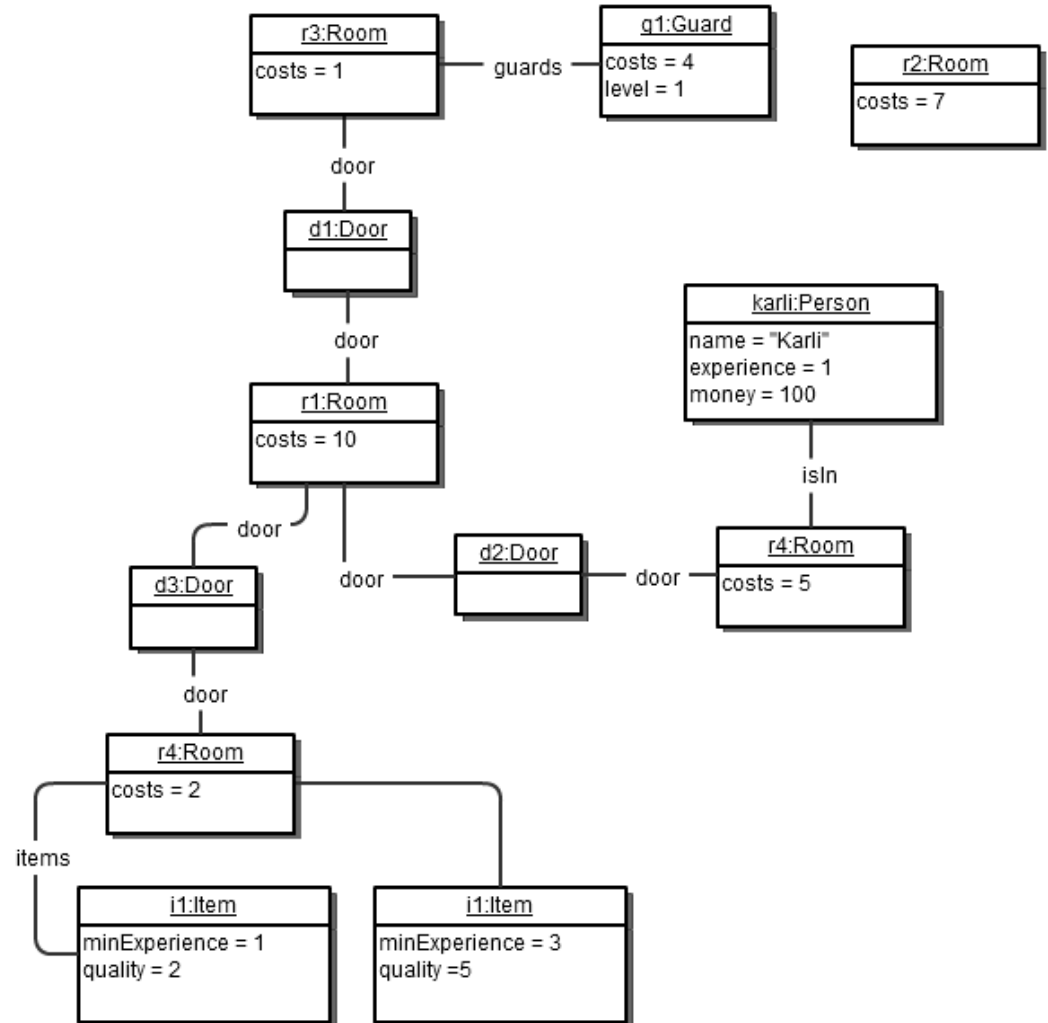
# Praktische Übung

- **Castle**
  - Das Schloss hat viele Räume, die mit Türen verbunden sind
  - Räume haben Kosten die bezahlt werden müssen
  - In Räumen können Items liegen
  - Items haben einen Wert (quality) und ein Level: Nur Personen mit mindestens dem gleichen Level können den Gegenstand aufnehmen
  - Räume können von Wächtern bewacht werden
  - Wächter müssen zunächst mit Geld bestochen werden, bevor der Raum geplündert werden kann.
  - Einmal bestochene Wächter sind einer Person freundlich gesinnt (friendlyGuard)
  - Personen haben einen Namen (name), Erfahrungspunkte (experience) und Geld (money)
  - Personen können Items besitzen



# Praktische Übung

- Beispiel Objektdiagramm



# Praktische Übung

- **Aufgabe:**

- Erstellt ein Storydiagramm für die Methode `Castle::initCastle()`, welche ein Schloss mit Räumen, Türen, Guards und Items erstellt.
- Erstellt ein Storyboard (Test) für die Methode `Person::collectItems()`
  - Nur Items mit gleichem oder niedrigerem Level dürfen eingesammelt werden
  - Eingesammelte Items werden der Person zugeordnet und aus dem Raum entfernt
- Erstellt ein Storydiagramm für die Methode `Person::collectItems()` und testet die Implementierung mit dem zuvor erstellten Storyboard

**Ende**

**Schönes WE!**