

---

**Software Engineering**

<http://proglang.informatik.uni-freiburg.de/teaching/swt/2005/>

---

**Übungsblatt 5 & 6**

Abgabe: 31. Mai 2005

**Aufgabe 1 – Activity Diagramm: (4 Punkte)**

Setzen Sie folgendes Rezept in ein Activity Diagram um:

Man schneidet 400g Hühnchen in kleine Würfel. Vermische 30ml Reiswein mit 10ml Sojasauce. Verrühr diese Sauce mit Speisestärke. Falls keine Speisestärke da ist, nimm Mehl. Vermische das Fleisch mit der Sauce. Schneide 10g Ingwer und 50g Lauch. Brate 150g Walnüsse kurz an. Brate das Fleisch scharf an. Gib Lauch und Ingwer hinzu und brate das Fleisch durch. Gib die Walnüsse und 10ml Sojasauce hinzu.

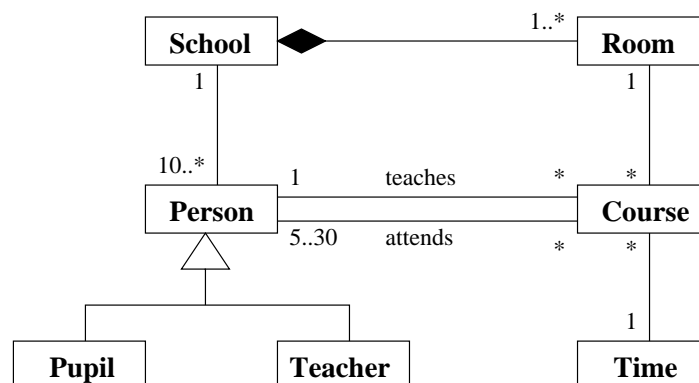
**Aufgabe 2 – Data Dictionary: (3 Punkte)**

Erstellen Sie ein Data Dictionary für deutsche Adressen.

Beachten Sie, dass es verschiedene Möglichkeiten gibt, wie z.B. Postfach vs. Straße, Zimmernummern, etc.

**Aufgabe 3 – Klassendiagramm: (5 Punkte)**

Analysieren Sie das folgende Klassendiagramm und beschreiben Sie detailliert, was es beschreibt. Gibt es Mängel in dem Design?


**Aufgabe 4 – Z Verstehen: (4 Punkte)**

Was wird durch die folgenden (generischen) axiomatischen Definitionen beschrieben und was sind die fehlenden Typen?

$$\frac{\text{walter} : ???}{\forall M : \mathbb{P}\mathbb{N}, x : \mathbb{N} \bullet \\ \text{walter}(\emptyset) = 0 \\ x \in M \Rightarrow \text{walter}(M) = \\ x + \text{walter}(M \setminus \{x\})}$$

$$\frac{[X]}{\frac{xaver : ???}{\forall s : \mathbb{F}X, n : \mathbb{N} \bullet \\ n = xaver \Leftrightarrow \exists f : (1..n) \twoheadrightarrow s \bullet true}}$$

$$\frac{\text{yolanda} : ???}{\forall m, n : \mathbb{N} \bullet \\ \text{yolanda}(m, n) = \{i : \mathbb{N} \mid m \leq i \leq n\}}$$

$$\frac{[X]}{\frac{zoe : ???}{\forall R : X \leftrightarrow X \bullet \\ zoe(R) = \bigcap \{T : X \leftrightarrow X \mid \\ (R \cup \text{id } X) \subseteq T \wedge \\ T \circ T \subseteq T\}}}$$

### Aufgabe 5 – Queue in Z: (4 Punkte)

Spezifizieren Sie eine Queue durch Z-Schemata. Die Queue soll Zahlen speichern. Die Funktionen *Init*, *Add* und *Next* sollen für die Queue definiert sein. Jede Funktion gibt eine Variable *result* vom Typ *REPORT* zurück, der aus *OK*, eine Fehlermeldung oder *Return i*, wobei *i* für eine beliebige Zahl steht, besteht.

Die Funktionen machen das folgende:

*Init* Instantiiert eine leere Queue.

*Add* Hängt eine gegebene Zahl an das Ende der Queue.

*Next* Gibt das erste Element aus der Queue zurück und entfernt es.