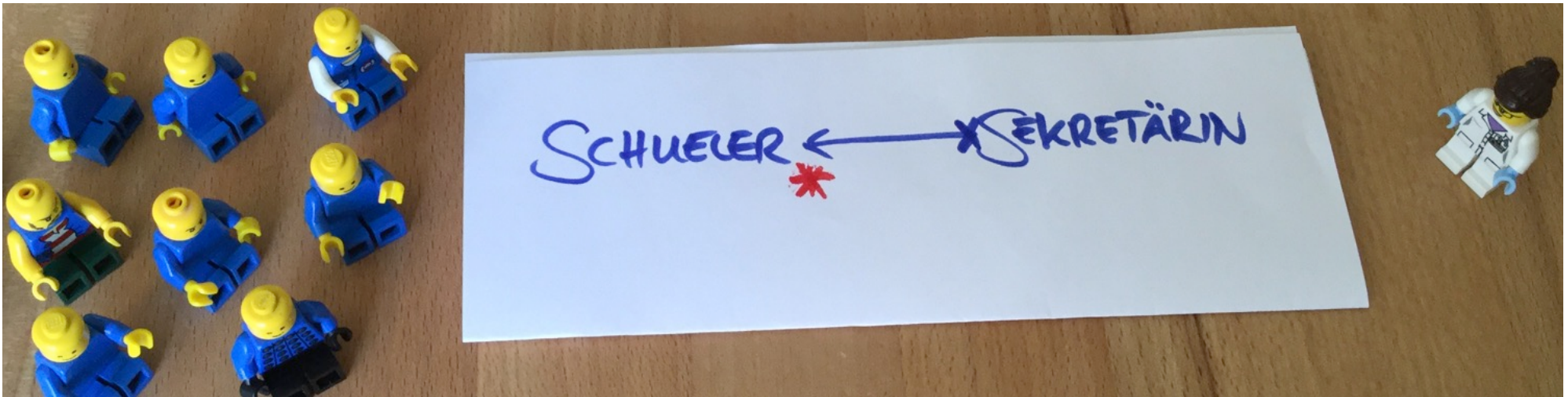


Objektorientierte Analyse / Objektorientiertes Design:

Assoziationen

**(= Beziehungen zw. Objekten/Klassen, auch:
"kennt"-Beziehung)**

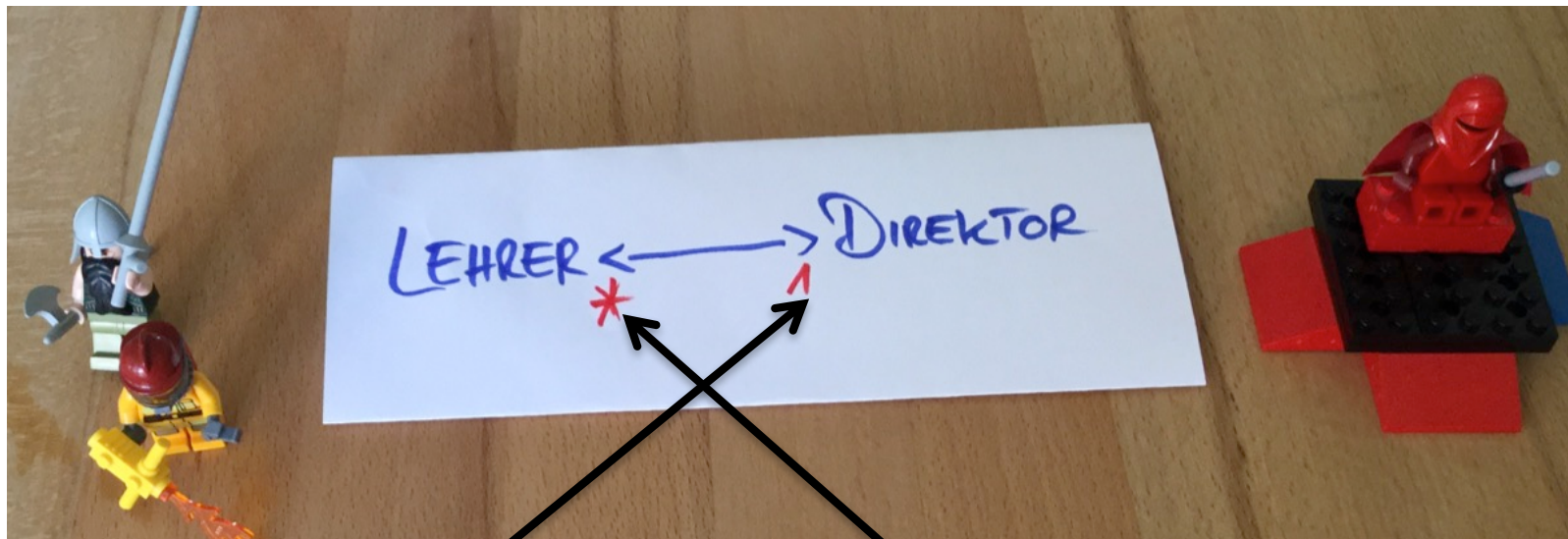


Ziel:

Wir modellieren die Beziehungen zwischen Klassen, um sie anschließend sinnvoll programmieren zu können.

- 1) **Multiplizitäten** – Mengenverhältnisse zwischen Klassen (0..1, 1, 1..*, *)
- 2) **Navigierbarkeit** – Wer kann auf wen zugreifen? (Pfeilspitze vs. X)
- 3) **Rollen** – Welche Rolle spielt eine Klasse für die andere?

1) Multiplizitäten



Jeder **Lehrer** hat genau einen Direktor (ohne Direktor kann die Schule nicht existieren)

Jeder **Direktor** hat keinen, einen oder mehrere Lehrer (er könnte in einer winzigen Landschule arbeiten, wo er der einzige Lehrer ist).

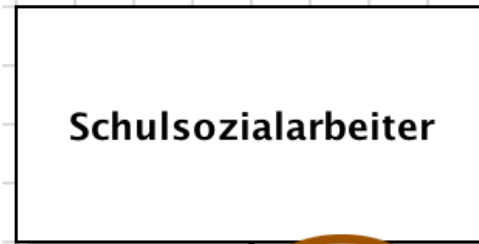
Multiplizitäten wie bei ER-Diagrammen

Multiplizität	Bedeutung	Beispiel
0..1	dem Objekt ist immer kein oder ein anderes Objekt zugeordnet	Ein Artikel befindet sich in keinem Einkaufswagen (= noch im Regal) oder in maximal einem.
1	dem Objekt ist immer genau ein anderes Objekt zugeordnet	Jede Laus sitzt immer auf genau einem Hund (ohne Hund kann sie nicht leben)
*	dem Objekt ist kein, ein oder mehrere andere Objekte zugeordnet	Auf jedem Hund sitzt keine, eine oder mehrere Läuse.
1..*	dem Objekt ist ein oder mehrere andere Objekte zugeordnet	Auf jeder Rechnung steht ein oder mehrere Artikel (eine Rechnung ohne Artikel ist sinnlos).

Multiplizitäten wie bei ER-Diagrammen

0...1

dem Objekt ist immer **kein**
oder ein anderes Objekt zugeordnet



0..1

1..*

Lehrer

1...*

dem Objekt sind **ein oder mehrere**
andere Objekte zugeordnet

*

dem Objekt sind **kein, ein oder**
mehrere andere Objekte zugeordnet



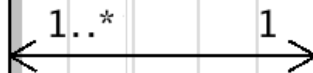
*

1

Direktor

1

dem Objekt ist immer **genau ein**
anderes Objekt zugeordnet



1..*

1

*

Jeder Direktor hat keine, eine oder mehrere Sekretärinnen.

1

Jede Sekretärin hat genau einen Direktor (niemals: keinen oder mehrere).

0...1

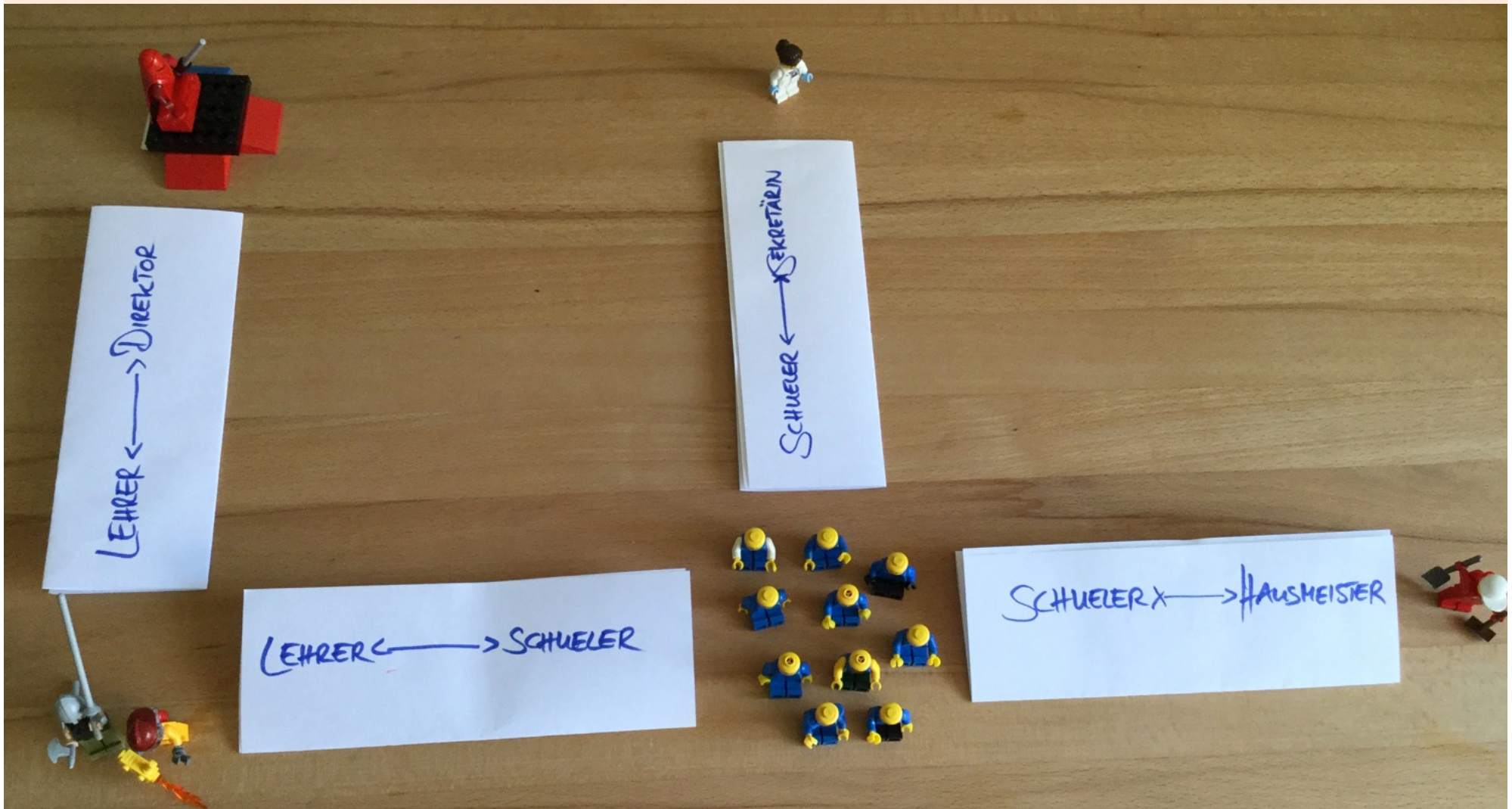
Jeder Lehrer arbeitet mit keinem oder einem Schulsozialarbeiter zugeordnet.

1...*

Jeder Schulsozialarbeiter arbeitet mit **mindestens** einem Lehrer.

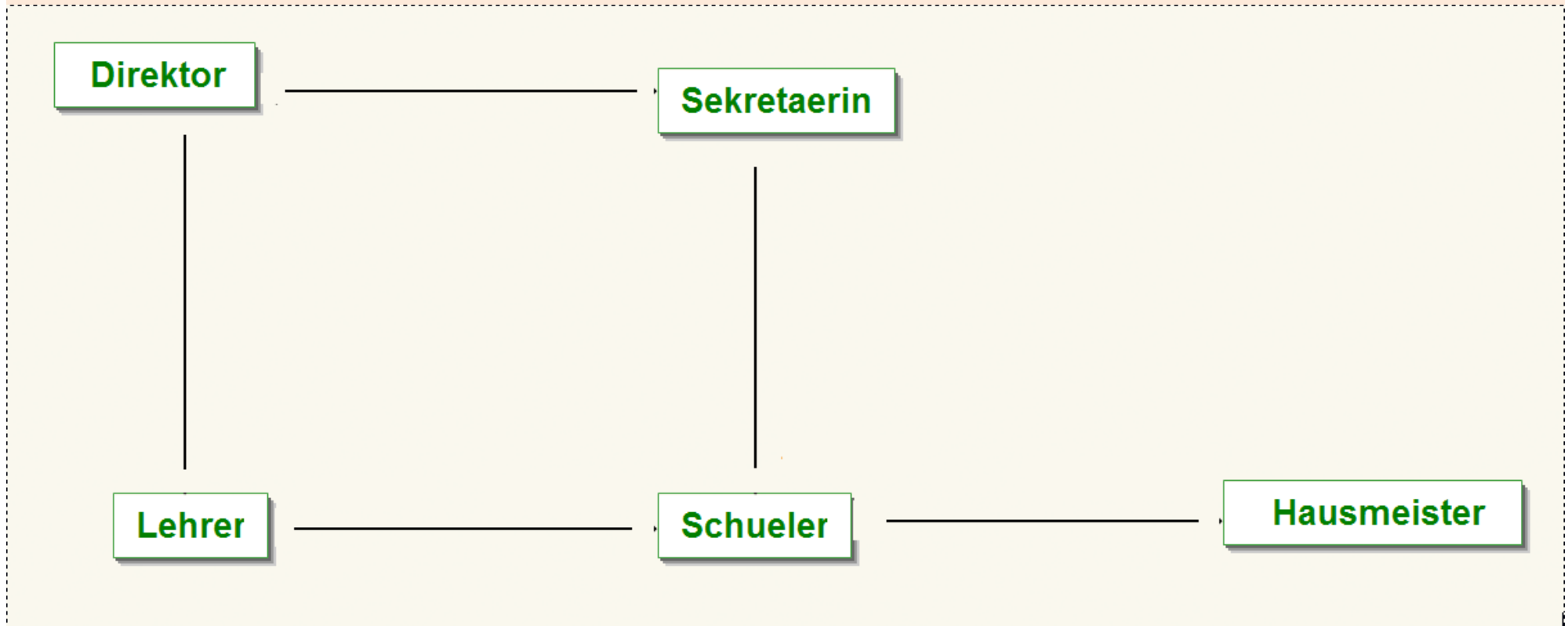
Übung Multiplizitäten

Wir bilden unsere Schule in einer Software ab. Wir wollen wissen, welche Klassen aufeinander zugreifen können müssen etc.



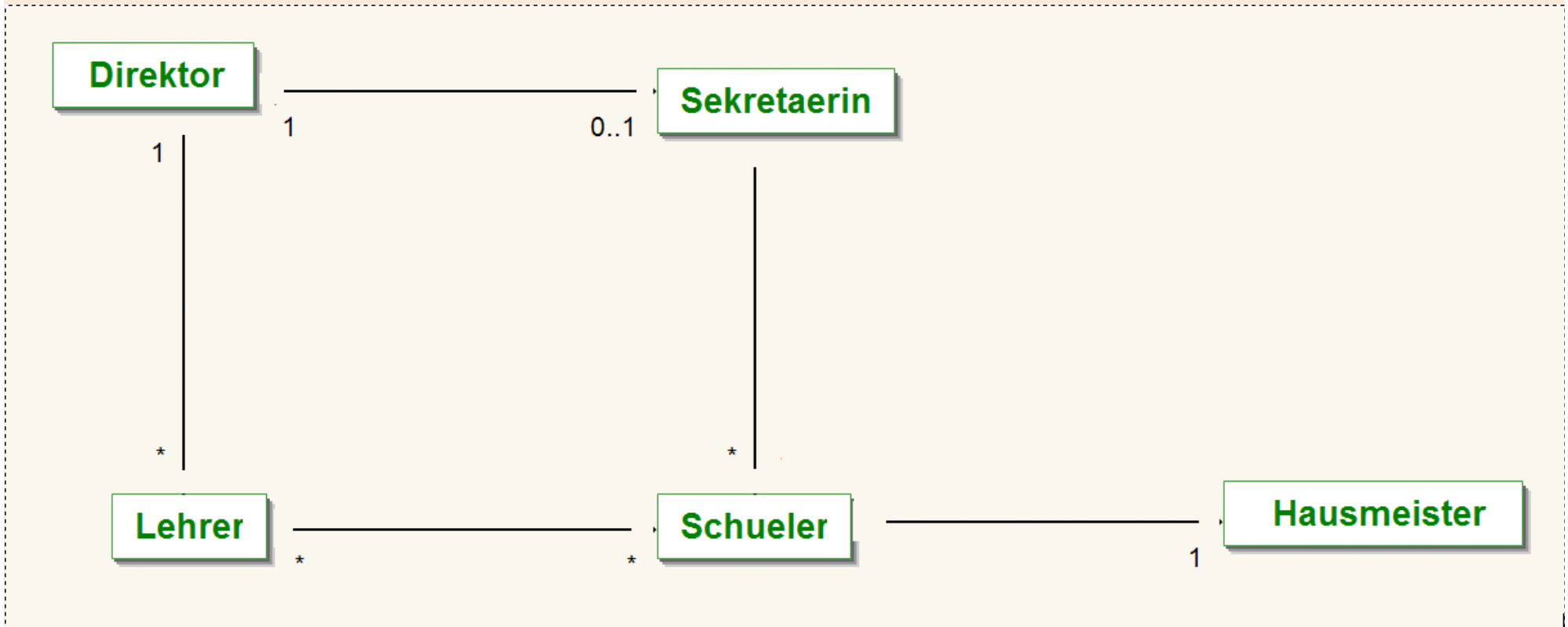
Übung Multiplizitäten

Tragen Sie die korrekten Multiplizitäten ein.



Übung Multiplizitäten

Tragen Sie die korrekten Multiplizitäten ein. **LÖSUNGSVORSCHLAG**



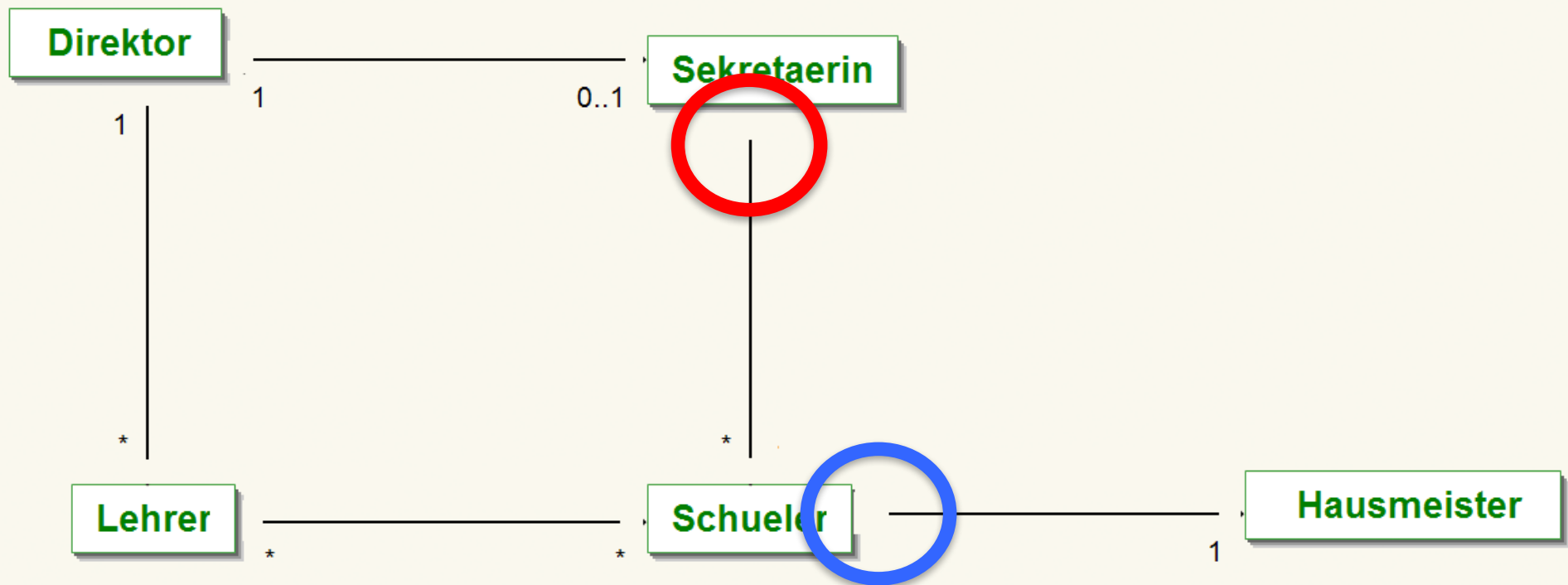
Muss der Schüler die Sekretärin "kennen"/auf diese Klasse Zugriff haben?

(z.B. ihr Gehalt ändern, sie zum Postamt schicken ...)

Muss der Hausmeister die Schüler "kennen"/auf diese Klasse Zugriff haben?

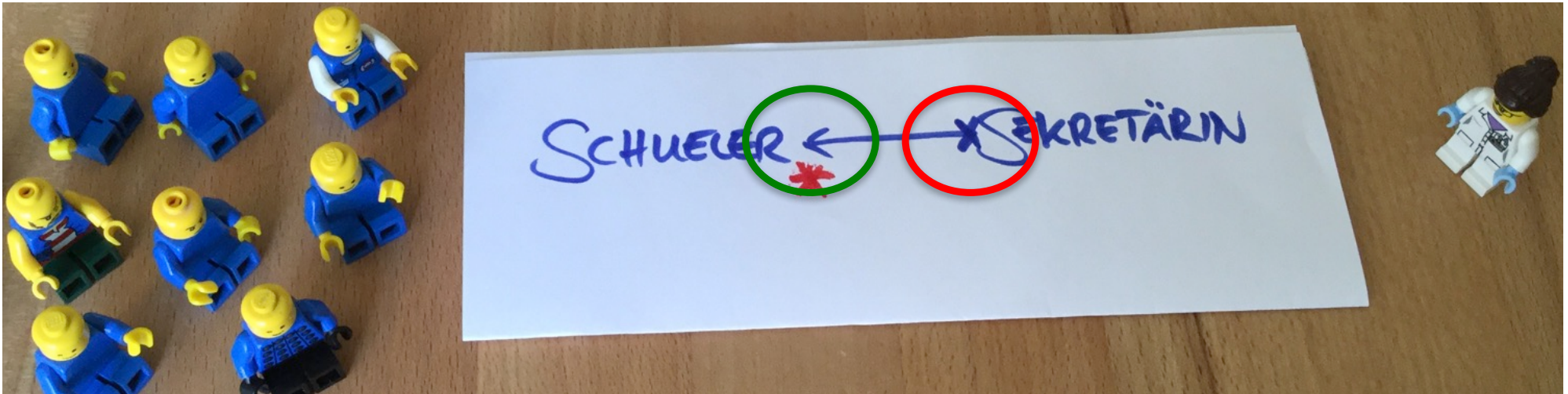
(z.B. ihren Namen, ihren Geburtsort, ihre Klasse...?)

Frage der "Navigierbarkeit"



2) Navigierbarkeit

Navigierbarkeit, Beispiel 1



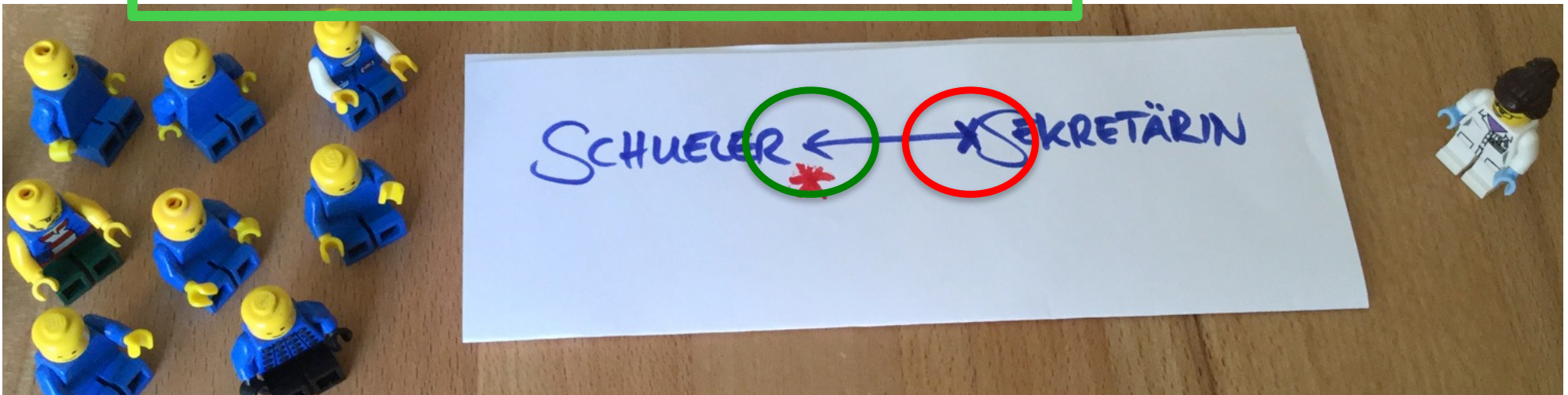
Pfeilspitze: Andere Klasse kann auf die Klasse zugreifen
X: Andere Klasse kann nicht auf die Klasse zugreifen
(nichts:) Keine Beziehung angegeben

Navigierbarkeit, Beispiel 1

Die Sekretärin kennt die Schüler, oder:

Die Sekretärin kann die Telefonnummer eines Schülers herausfinden, oder:

Die Sekretärin kann Schülerdaten verändern, z.B. Telefonnummer ändern usw. ...



Der Schüler weiß nichts über die Sekretärin, oder:

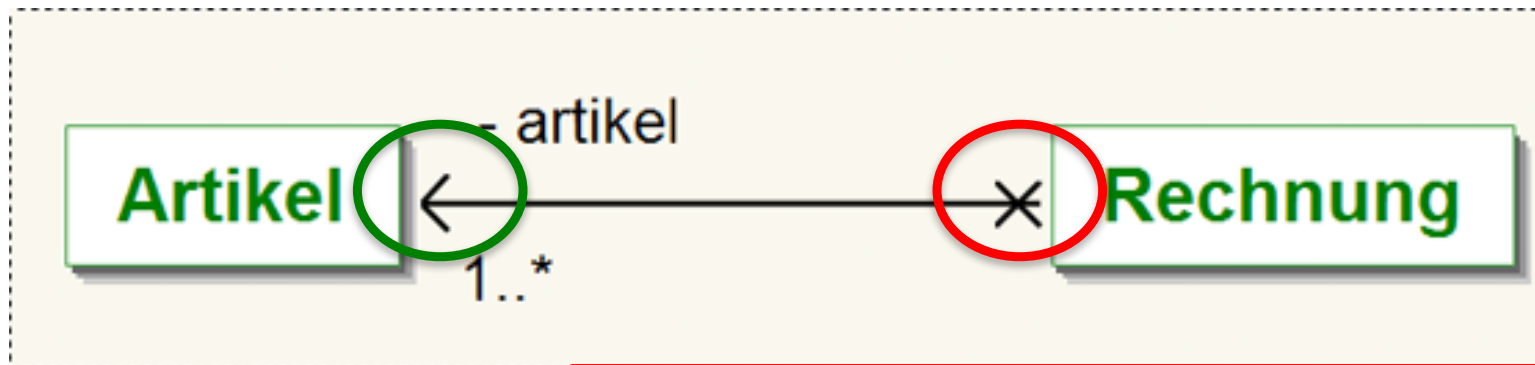
Es ist nicht notwendig, dass ein Schüler auf die Sekretärin zugreift (z.B. ihre Telefonnummer, ihre Bankverbindung sieht, verändert, sie zum Postamt schickt ...)

Navigierbarkeit, Beispiel 2

Die Rechnung "kennt" ihre Artikel; sie "weiß", welche Artikel draufstehen, oder:

Wenn ich eine Rechnung anschau, sehe ich, welche Artikel sie enthält, oder:

Ich möchte die Rechnung verändern können, indem ich Artikel hinzufüge, entferne ...



Wenn ich einen Artikel anschau, sehe ich nicht, auf welcher Rechnung er sich befindet, oder:

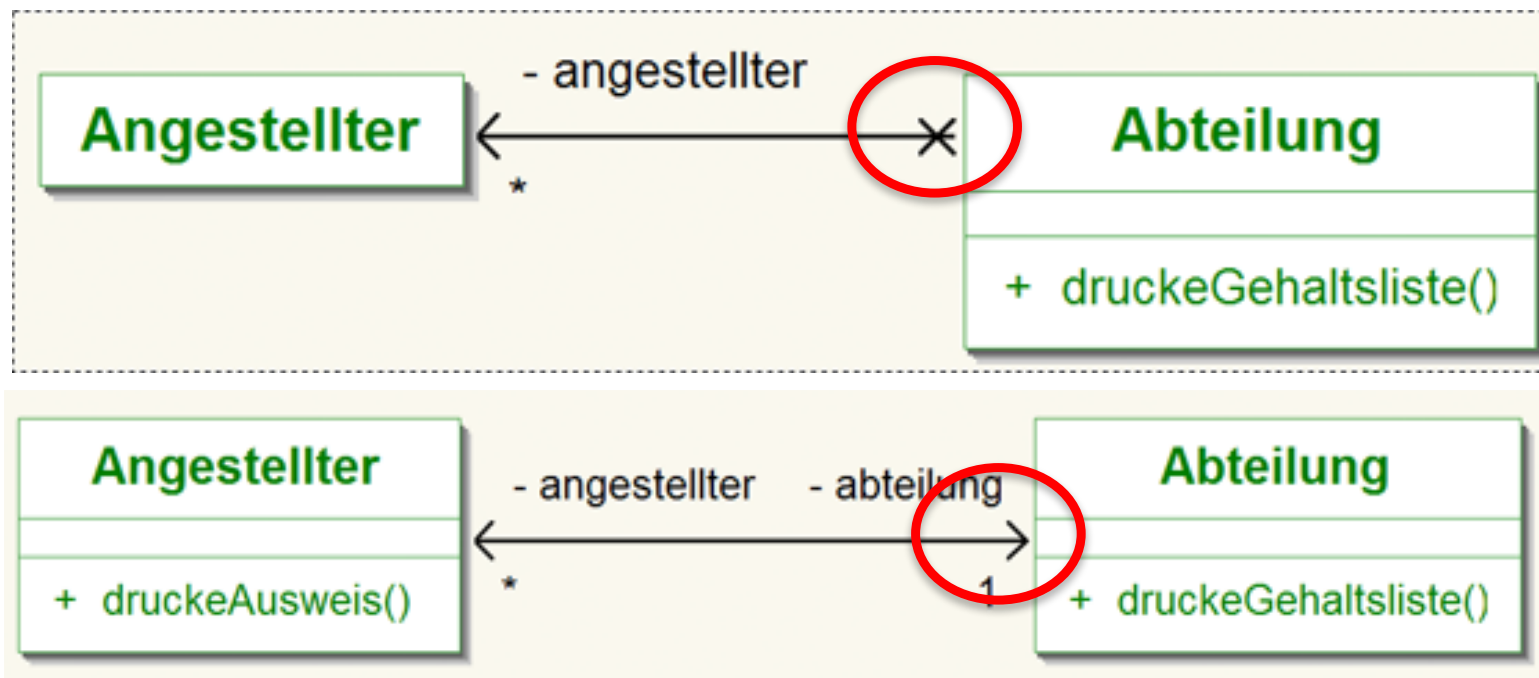
Es ist nicht notwendig, dass ich den Zustand einer Rechnung durch Manipulation des Artikels verändere

Navigierbarkeit, Beispiel 3

Die Navigierbarkeit hängt davon ab, welche Funktionalitäten man realisieren will.

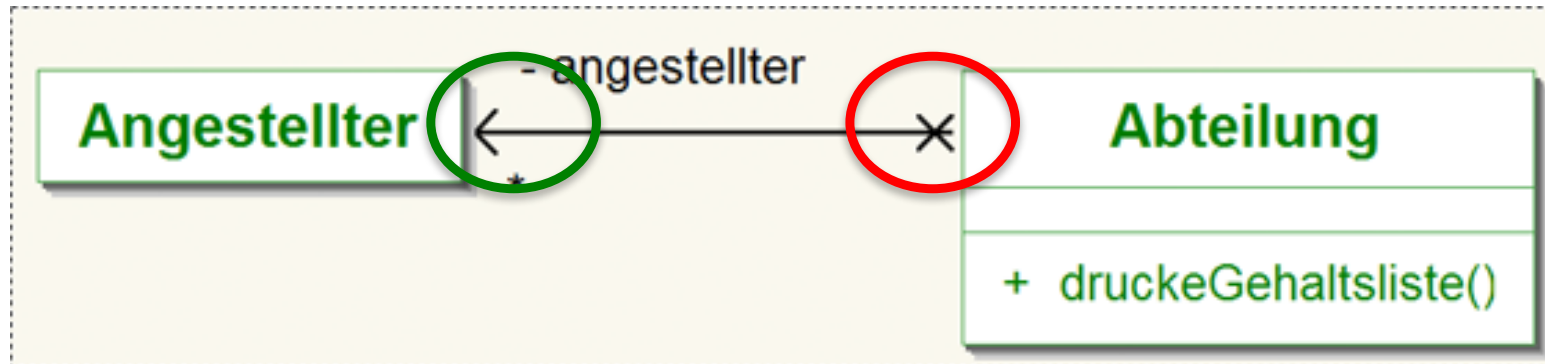
Beispiel:

Angestellter/Abteilung kann unterschiedlich navigierbar sein



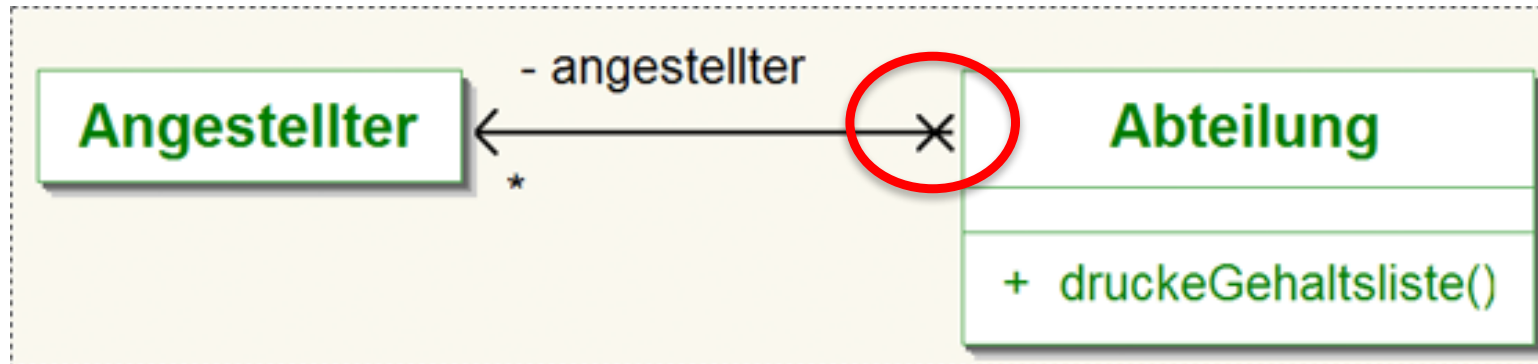
Navigierbarkeit, Beispiel 3 – Version 1

Die Abteilung muss auf die Angestellten zugreifen, um die Gehaltsliste drucken zu können (dazu braucht sie bspw. den Namen der Angestellten)

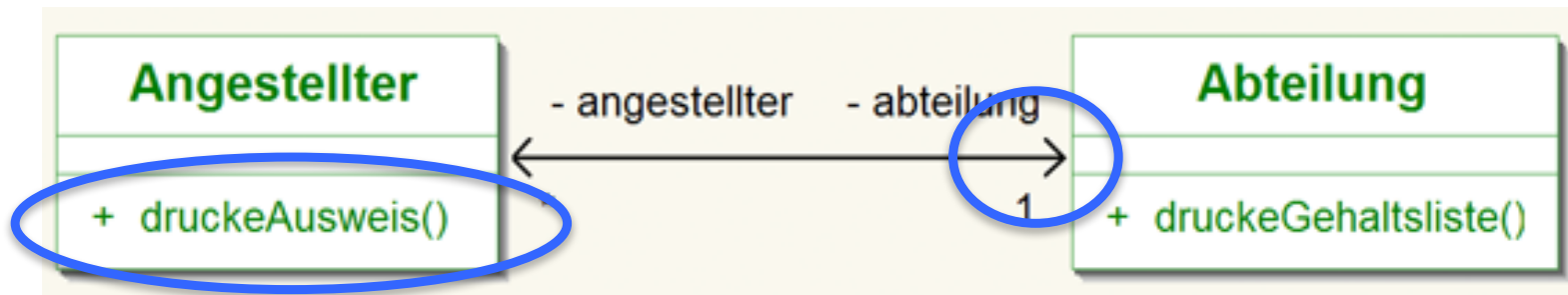


Der Angestellte greift nicht auf die Abteilung zu, da er für korrektes Funktionieren keine Informationen braucht.
(Welcher Angestellte wo arbeitet, kann man ja über die Abteilung herausbekommen)

Navigierbarkeit, Beispiel 3 – Version 2

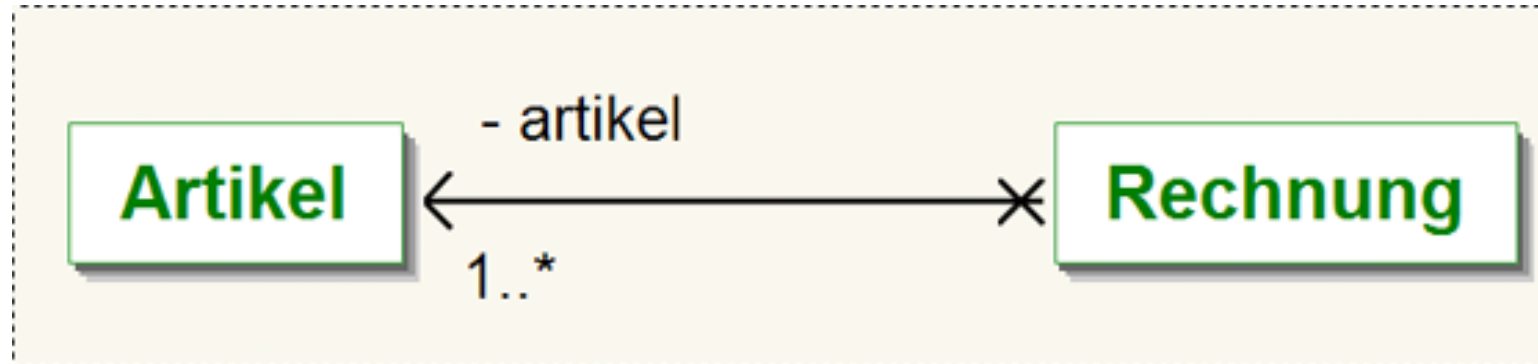


Wenn der Angestellte einen Ausweis drucken können muss, dann muss er wissen, in welcher Abteilung er arbeitet, wie die Abteilung heißt usw.
Deshalb muss die Assoziation nun auch in die andere Richtung navigierbar sein!

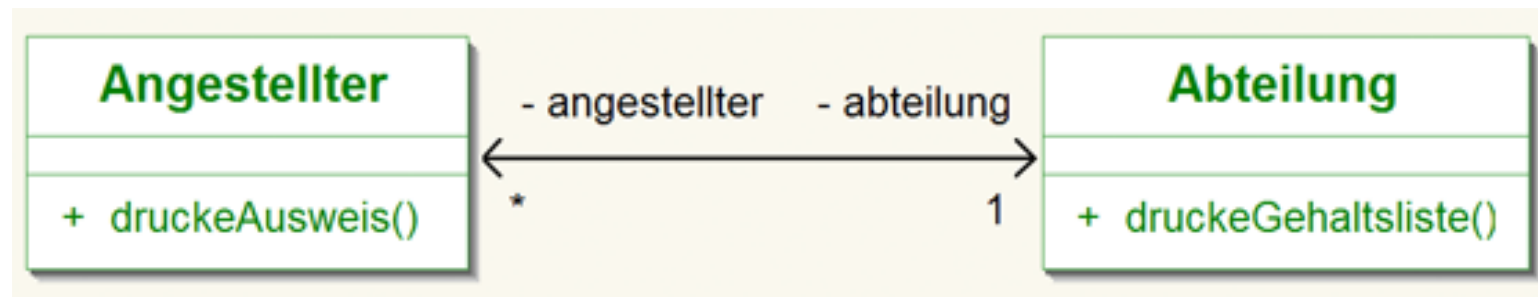


unidirektionale und bidirektionale Navigierbarkeit

unidirektional:
nur in eine Richtung navigierbar

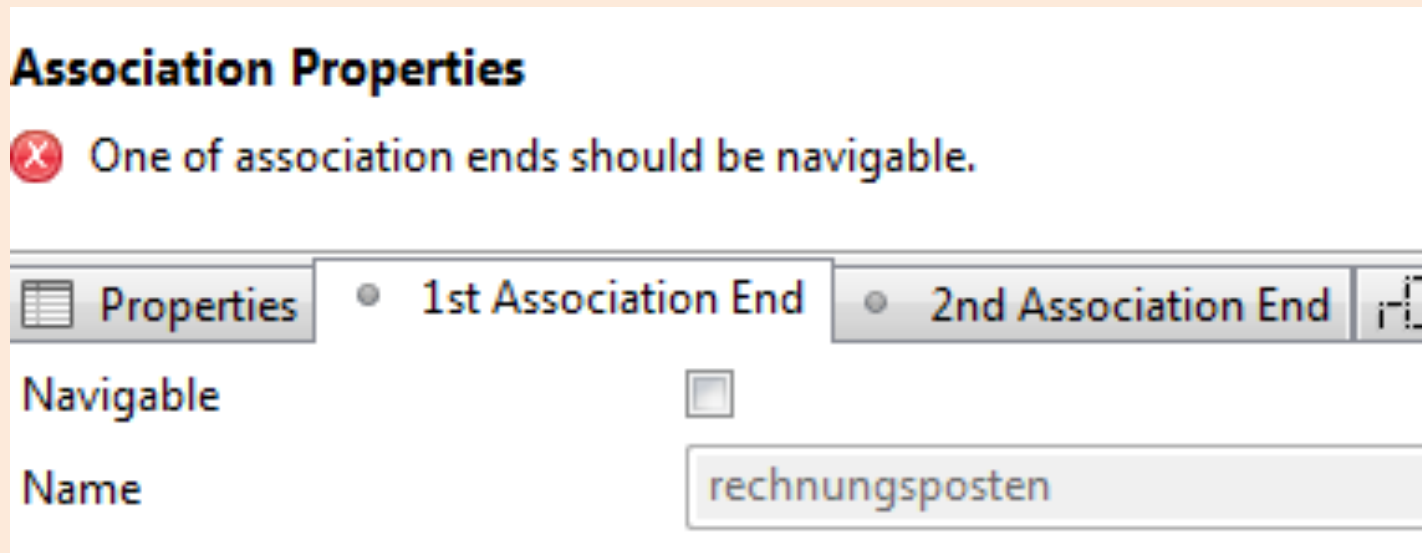


bidirektional:
in beide Richtungen navigierbar



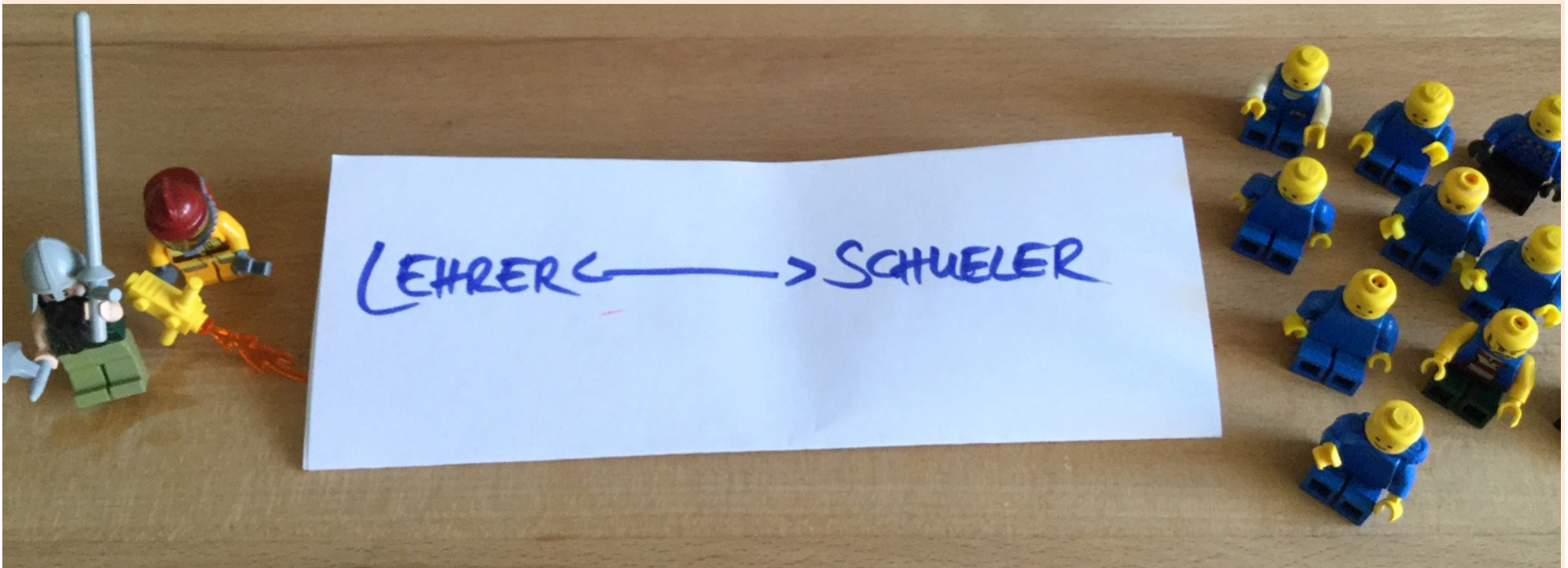
Übung Navigierbarkeit 1

Das eclipse-Plugin eUML2 weigert sich, Assoziationen anzulegen, die an beiden Seiten **nicht** navigierbar sind. Warum?



Übung Navigierbarkeit 2

Diskutieren Sie, ob die Navigierbarkeit hier korrekt angegeben ist:



Übung Navigierbarkeit 3

Ergänzen Sie die Navigierbarkeit:

Schüler

Direktor

Direktor

Hausmeister

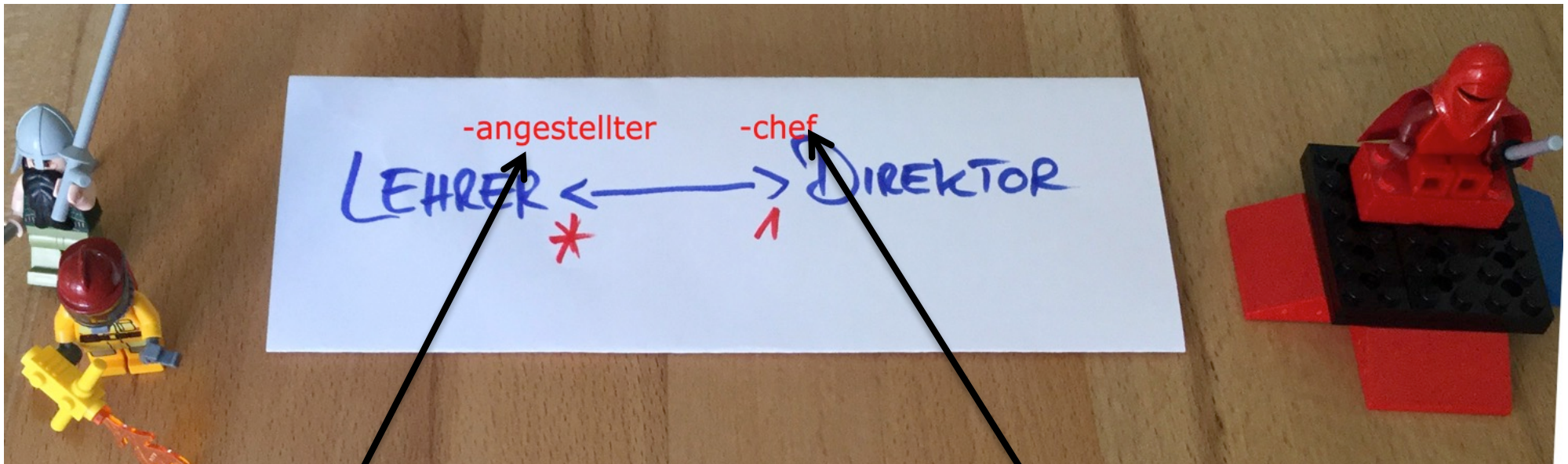
Sekretärin

Hausmeister

Übung Navigierbarkeit 4

Erstellen Sie zwei uni- und zwei bidirektionale Assoziationspaare. Das Thema ist Ihnen freigestellt.

3) Rollen



Der Lehrer hat in
Bezug auf den Direktor
die Rolle "Angestellter"

Der Direktor hat in
Bezug auf den Lehrer
die Rolle "Chef"

Frage, um zu korrekten Bezeichnungen zu kommen:
Welche Rolle spielt die eine Klasse für die andere?

Warum Rollenbezeichnungen wichtig sein können

Szenario 1 - Uns interessiert die **Verwaltungsstruktur**:

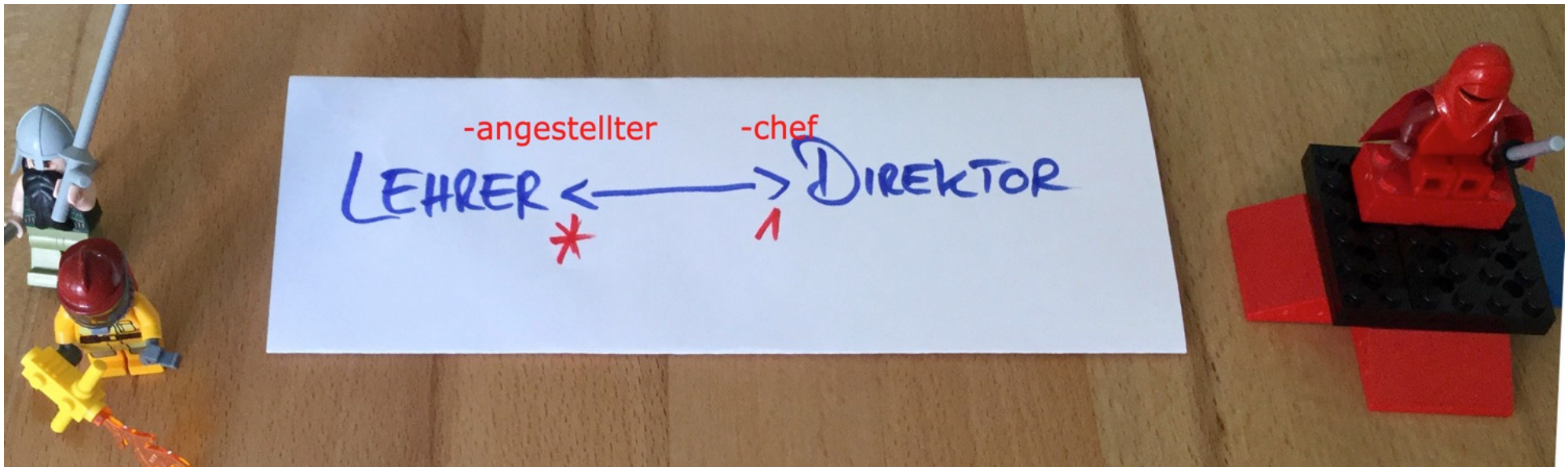
Der Direktor teilt den Lehrern Korrekturen zu, die Lehrer bitten den Direktor um Freistellung vom Unterricht etc.

Szenario 2 - Uns interessiert der **Ehestand**:

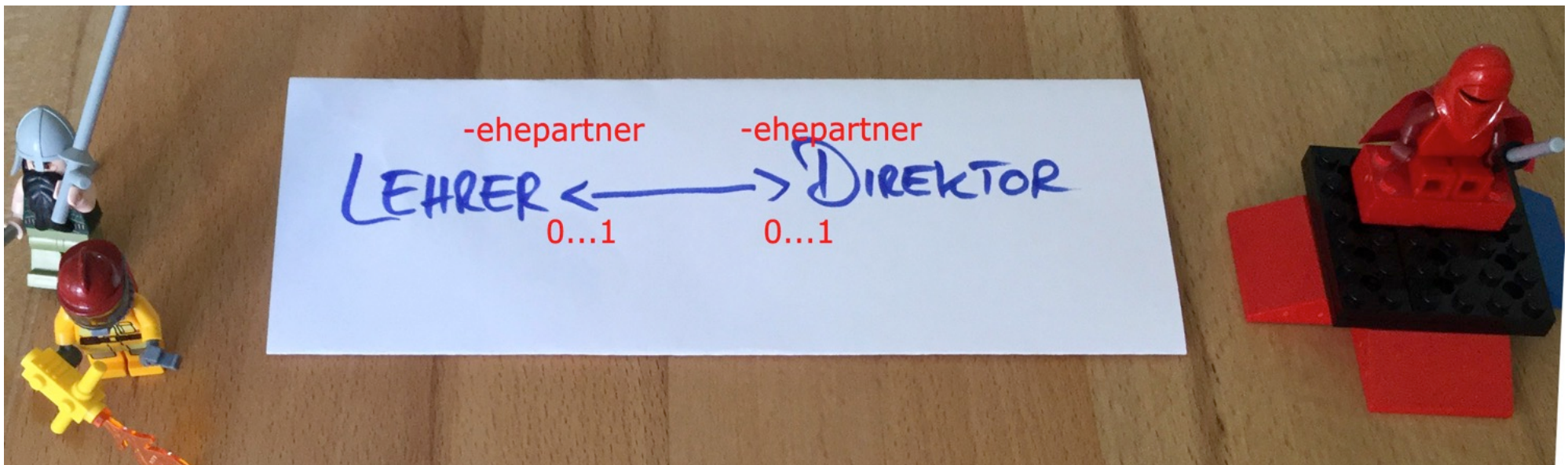
Der Direktor ist mit einem Lehrer verheiratet. Der Lehrer kennt das Gehalt des Direktors (wegen der gemeinsamen Steuererklärung), der Direktor kennt die Geheimzahl der EC-Karte des Lehrers usw.

Warum Rollenbezeichnungen wichtig sein können

Szenario 1 - Uns interessiert die **Verwaltungsstruktur**:

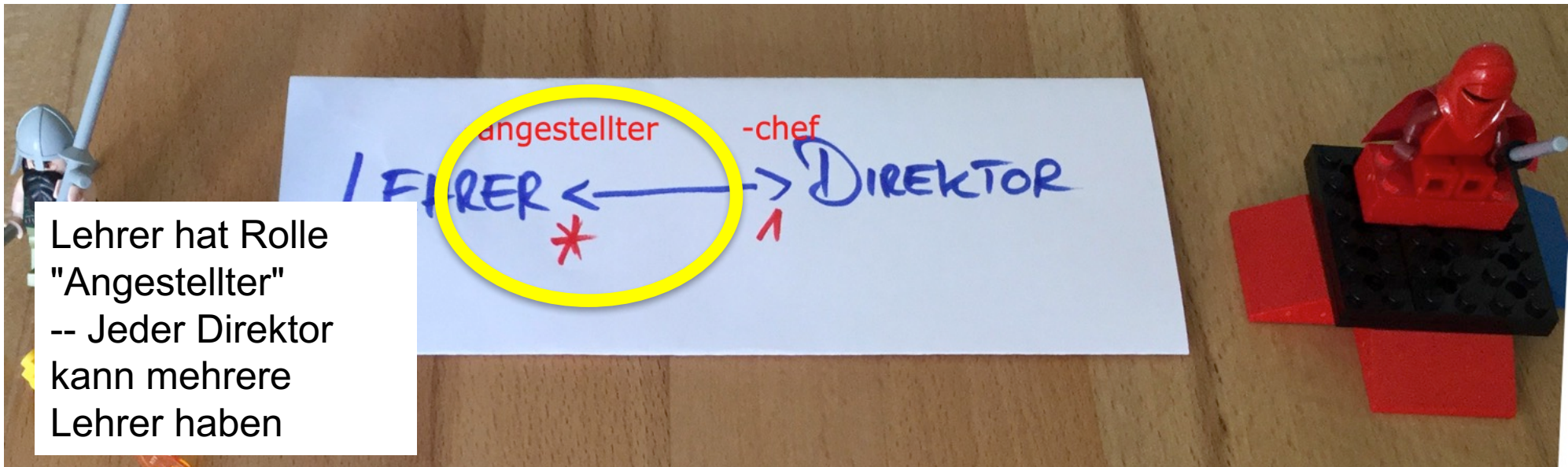


Szenario 2 - Uns interessiert der **Ehestand**:

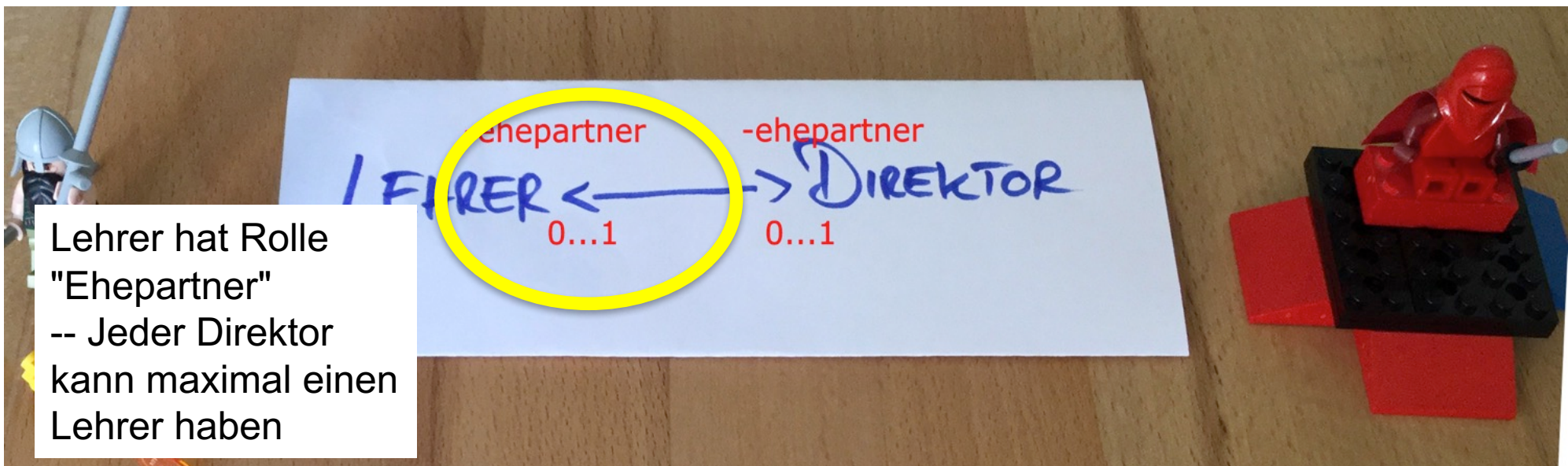


Warum Rollenbezeichnungen wichtig sein können

Szenario 1 - Uns interessiert die **Verwaltungsstruktur**:

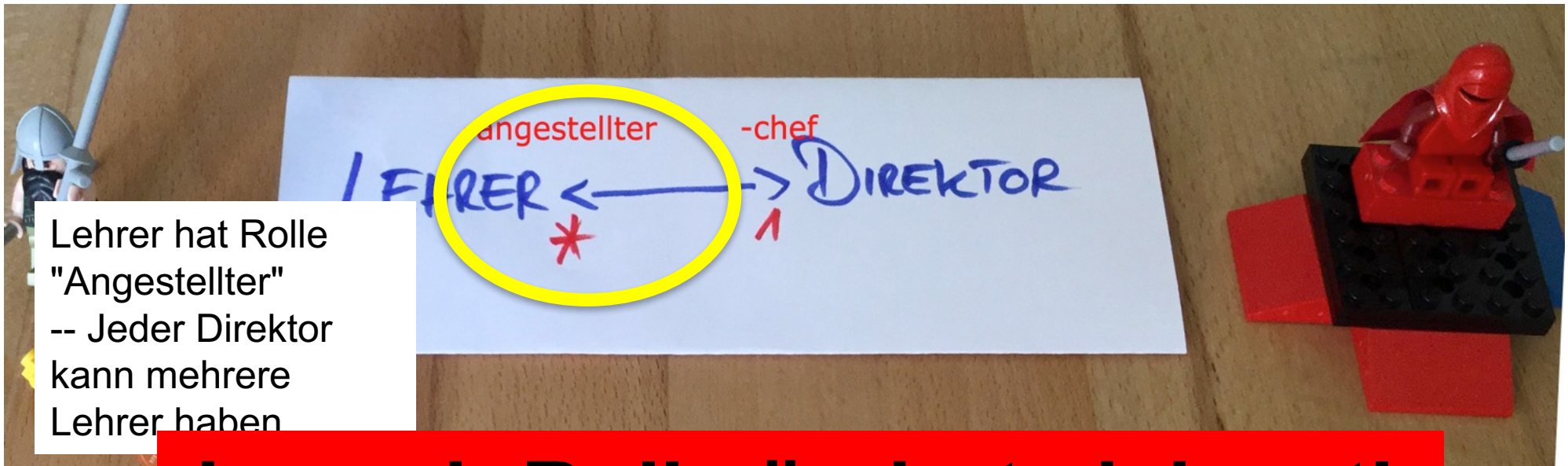


Szenario 2 - Uns interessiert der **Ehestand**:

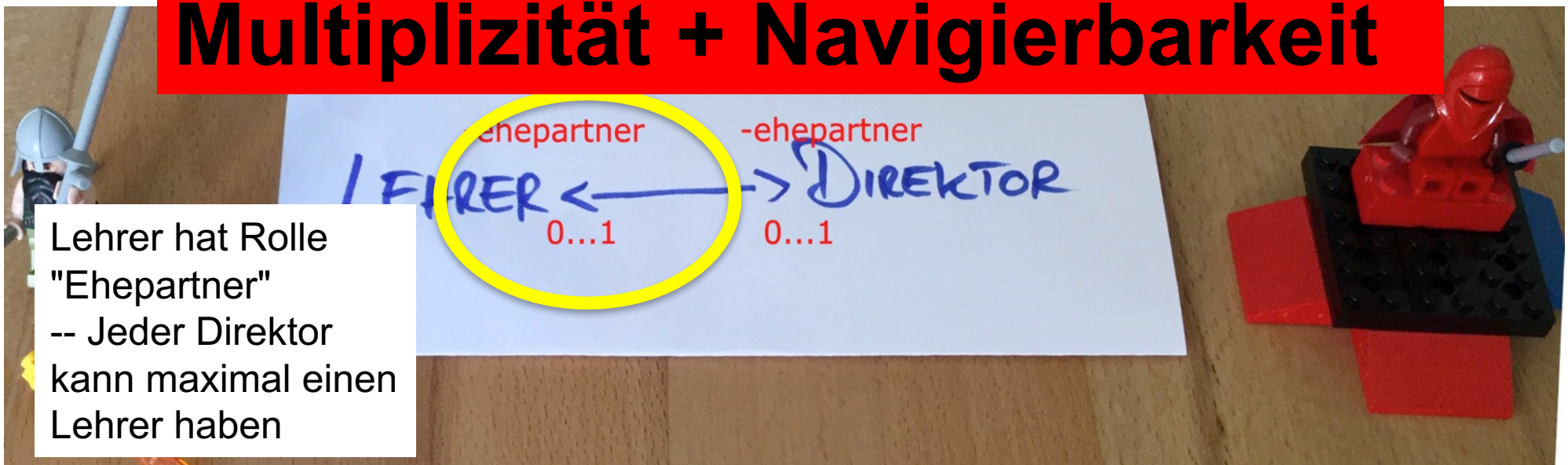


Warum Rollenbezeichnungen wichtig sein können

Szenario 1 - Uns interessiert die Verwaltungsstruktur:



Je nach Rolle ändert sich evtl. Multiplizität + Navigierbarkeit



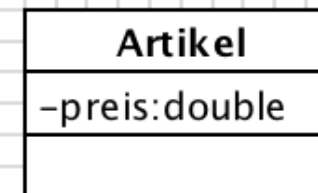
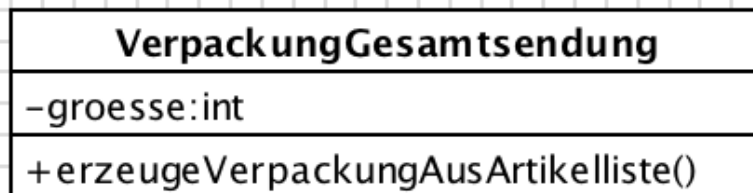
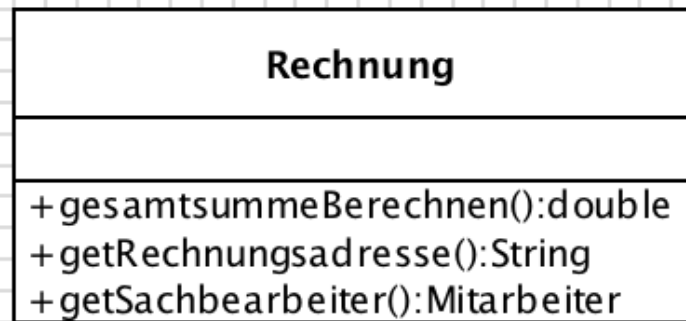
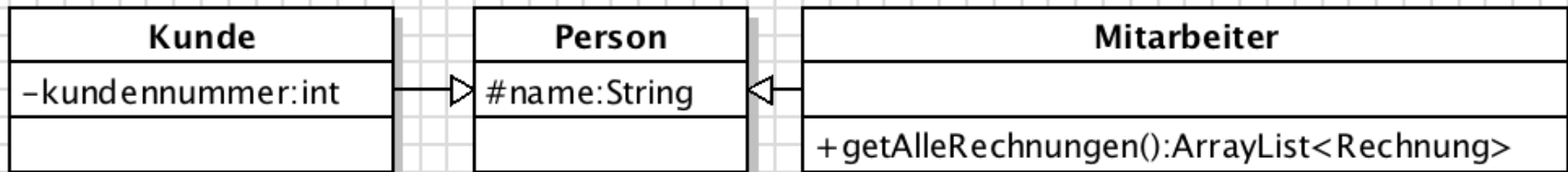
Allgemeines zu Rollen

- Zugriffsmodifikator ist grundsätzlich private (wie auch das Attribut, durch das die assoziierte Klasse repräsentiert wird – wird später erläutert)
- von eUML2 vorgeschlagene Defaultwerte sind oft in Ordnung
- Rollenbezeichnungen sind optional

Zusammenfassende Übung

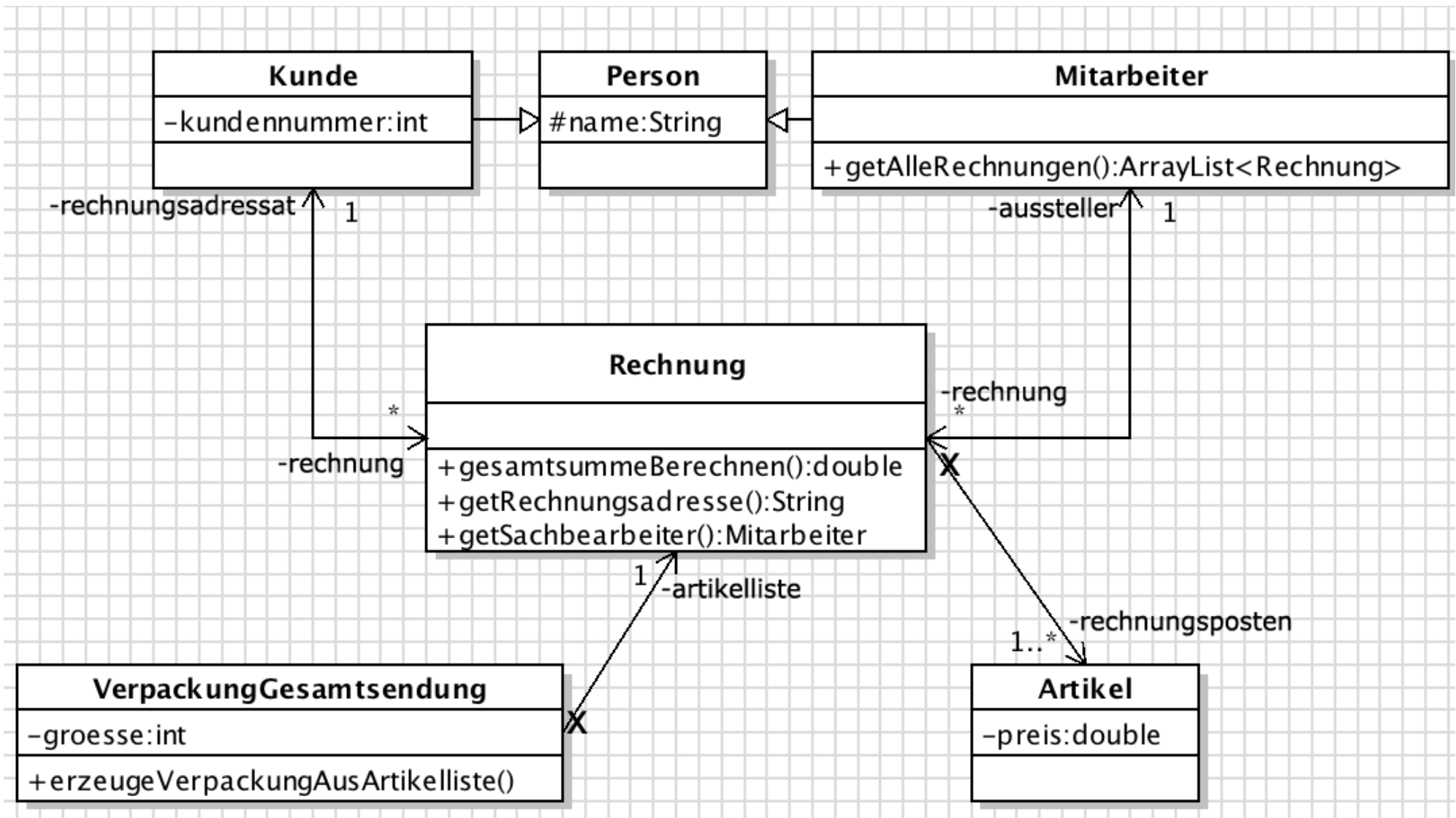
Übung: Assoziationen / UML

Zeichnen Sie korrekte
Assoziationen, Rollen,
Multiplizitäten und
Navigierbarkeiten ein.



Übung: Assoziationen / UML - Lösungsvorschlag

Zeichnen Sie korrekte Assoziationen, Rollen, Multiplizitäten und Navigierbarkeiten ein.



Übung: Assoziationen / UML - Lösungsvorschlag

mit Anzeige der assoziierten Attribute

(Achtung: Hier kleinere Fehler, z.B.

Sichtbarkeit des Attributs name bei Person)

