

Teil 5: Vererbung

Modul „Grundlagen der objektorientierten Programmierung mit Java“

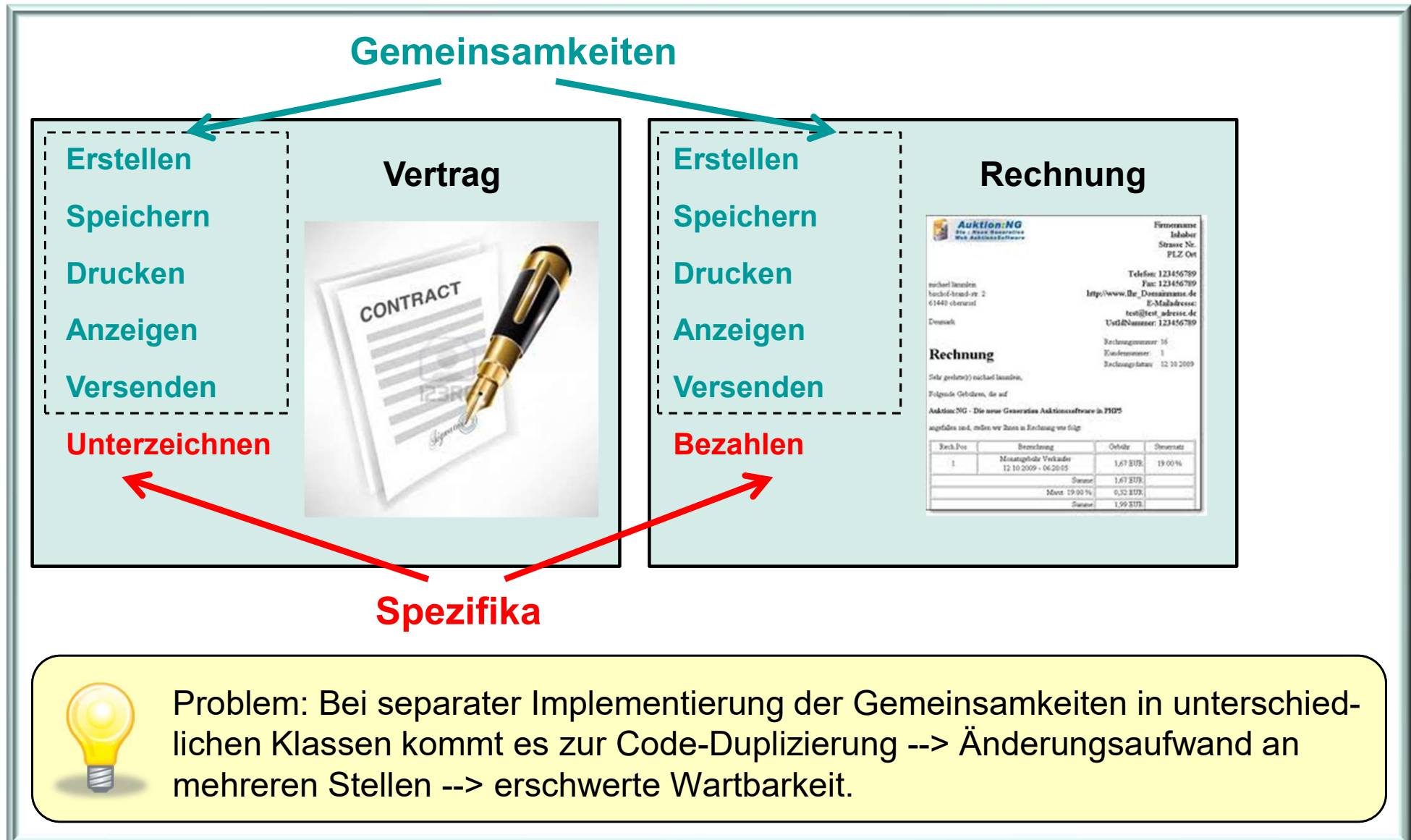
Prof. Dr. Cornelia Heinisch

Agenda

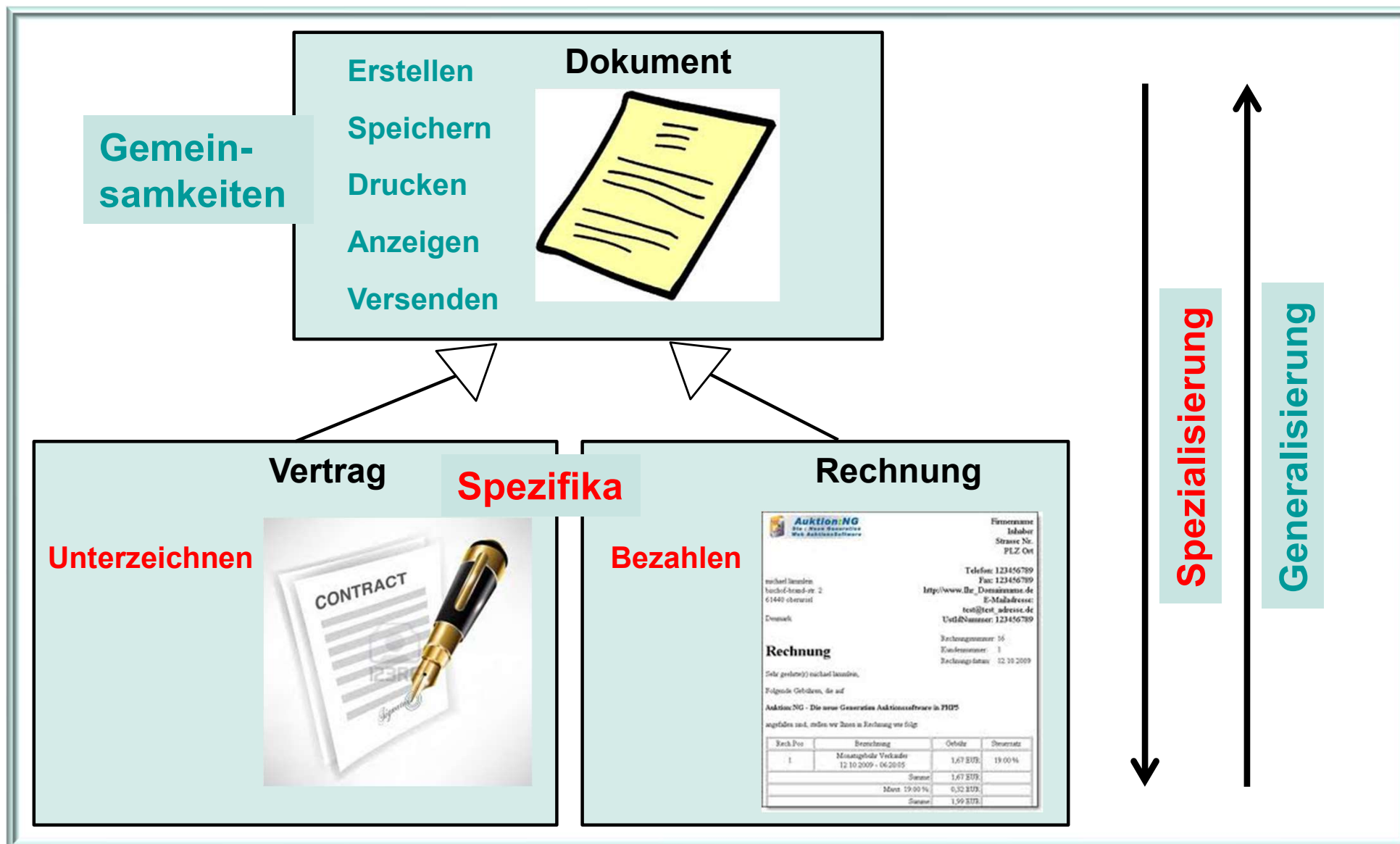
- Konzept der Vererbung
- Überladen, Erweitern, Überschreiben

Konzept der Vererbung

Vererbung – Ausgangslage: mehrere Klassen mit Gemeinsamkeiten

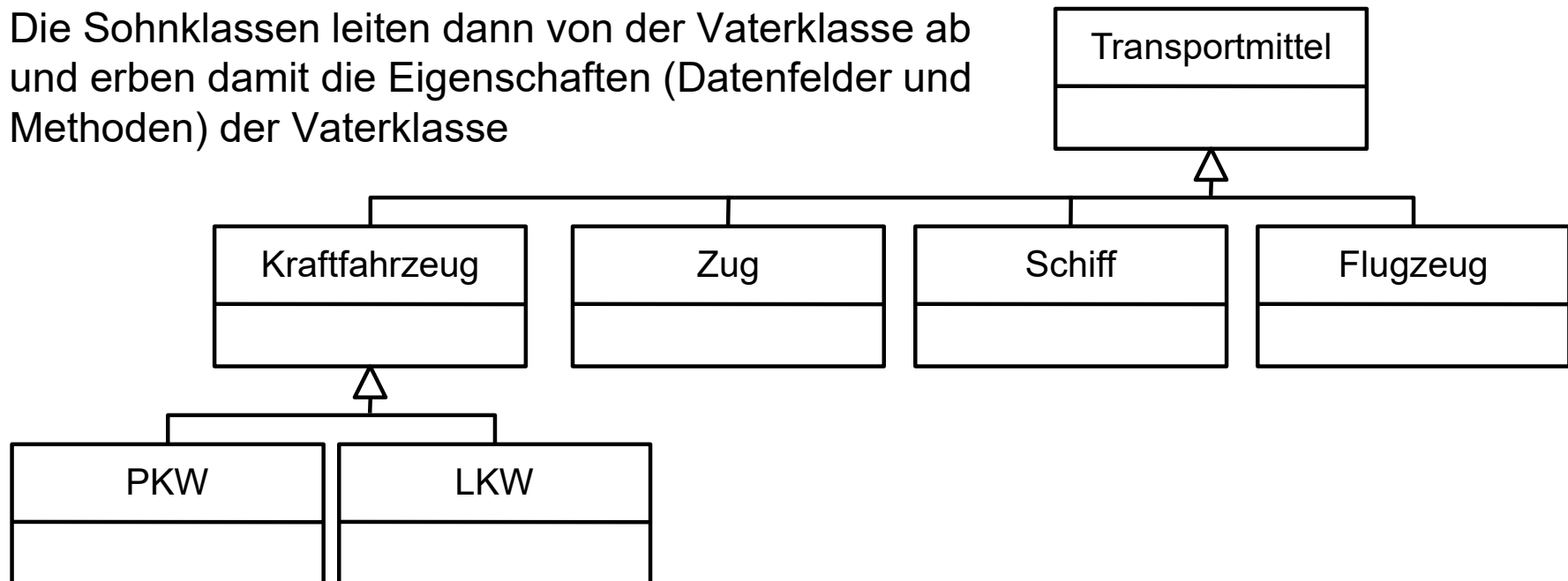


Vererbung – Lösung: Vererbung mit Variantenbildung



Vererbungshierarchie

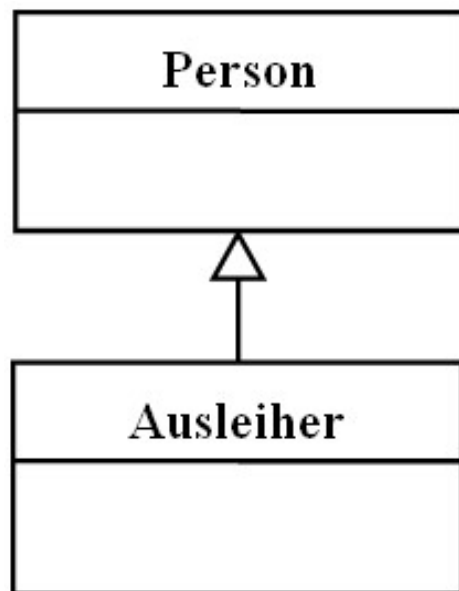
- Gemeinsame Eigenschaften (Datenfelder und Methoden) von Klassen können in einer Vaterklasse beschrieben werden.
- Die Sohnklassen leiten dann von der Vaterklasse ab und erben damit die Eigenschaften (Datenfelder und Methoden) der Vaterklasse



Vererbung ist in der Objektorientierung ein Mechanismus für die Wiederverwendung von Programmcode!

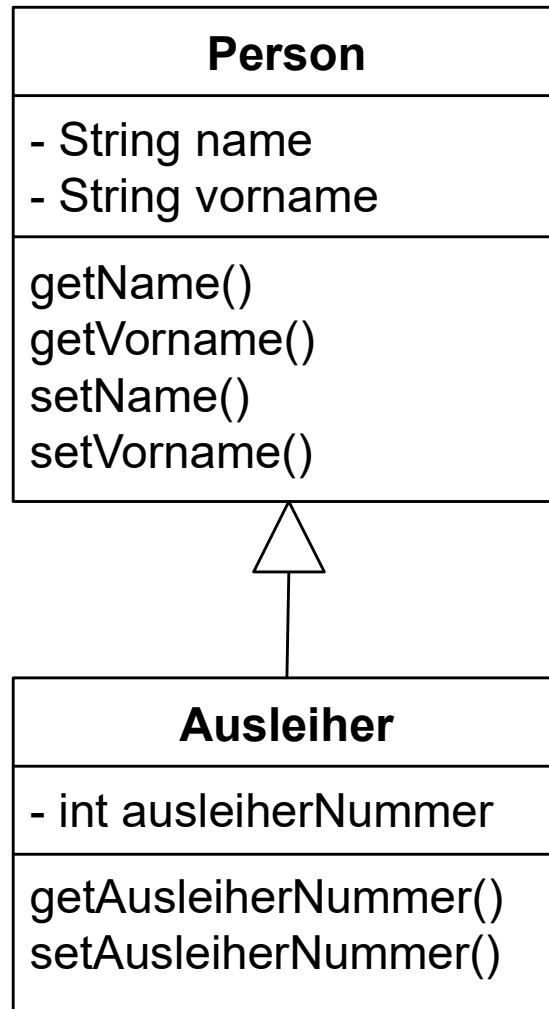
Einführung in die Vererbung

- Vererbung ist in der Objektorientierung ein Mechanismus für die Wiederverwendung von Programmcode.
- Vererbungsbeziehungen gibt es zwischen Klassen.
- Damit Programmcode-Wiederverwendung durch Vererbung möglich ist, müssen **zwei Klassen** gefunden werden, die **vieles gemeinsam** haben.



- Ein Ausleiher ist eine Person („is-a-Beziehung“), die einen Bibliotheksausweis besitzt.
- Gemeinsamkeiten: Ausleiher besitzt wie Person Name und Anschrift.
- Unterschiede: Ausleiher besitzt **zusätzlich** einen Bibliotheksausweis.

UML-Notation mit Datenfeldern und Methoden



- Gemeinsamkeiten finden sich in der Vaterklasse.
 - Alternative Begriffe:
 - Superklasse,
 - Basisklasse,
 - durch Ausleiher **abgeleitete** Klasse.
 - Die Vaterklasse ist eine Generalisierung der Sohnklasse(n)
-
- Spezifika finden sich in der Sohnklasse.
 - Alternative Begriffe:
 - Subklasse
 - **ableitende** Klasse von Person
 - Die Sohnklasse ist eine Spezialisierung der Vaterklasse.

Hinweise zur korrekten Verwendung der Vererbung



Ist es sinnvoll für die Bibliotheksverwaltung eine Klasse `Ausleiher` von der Klasse `Person` abzuleiten?

- Die Vererbungsbeziehung zwischen den Klassen `Ausleiher` und `Person` ist auf jeden Fall korrekt.
- Sie ist aber im Falle der Bibliotheksverwaltung nicht sinnvoll.



Warum?

- Die Vererbungsbeziehung ist nur dann sinnvoll, wenn beiden Typen (`Ausleiher` und `Person`) in der Bibliotheksverwaltung benötigt werden.
- Aktuell werden nur `Ausleiher` benötigt.



Wie würde die sinnvolle Implementierung aussehen?

Agenda

- Konzept der Vererbung
- Überladen, Erweitern, Überschreiben

Hinweis für Dozent: A5, A4, A6, A7, A8 und A9 mit Studenten live programmieren.

Was sind überladene Methoden?



Methoden in einer Klasse, die **denselben Methodennamen** aber **verschiedene Parameterlisten** besitzen, nennt man **überladene Methoden**.

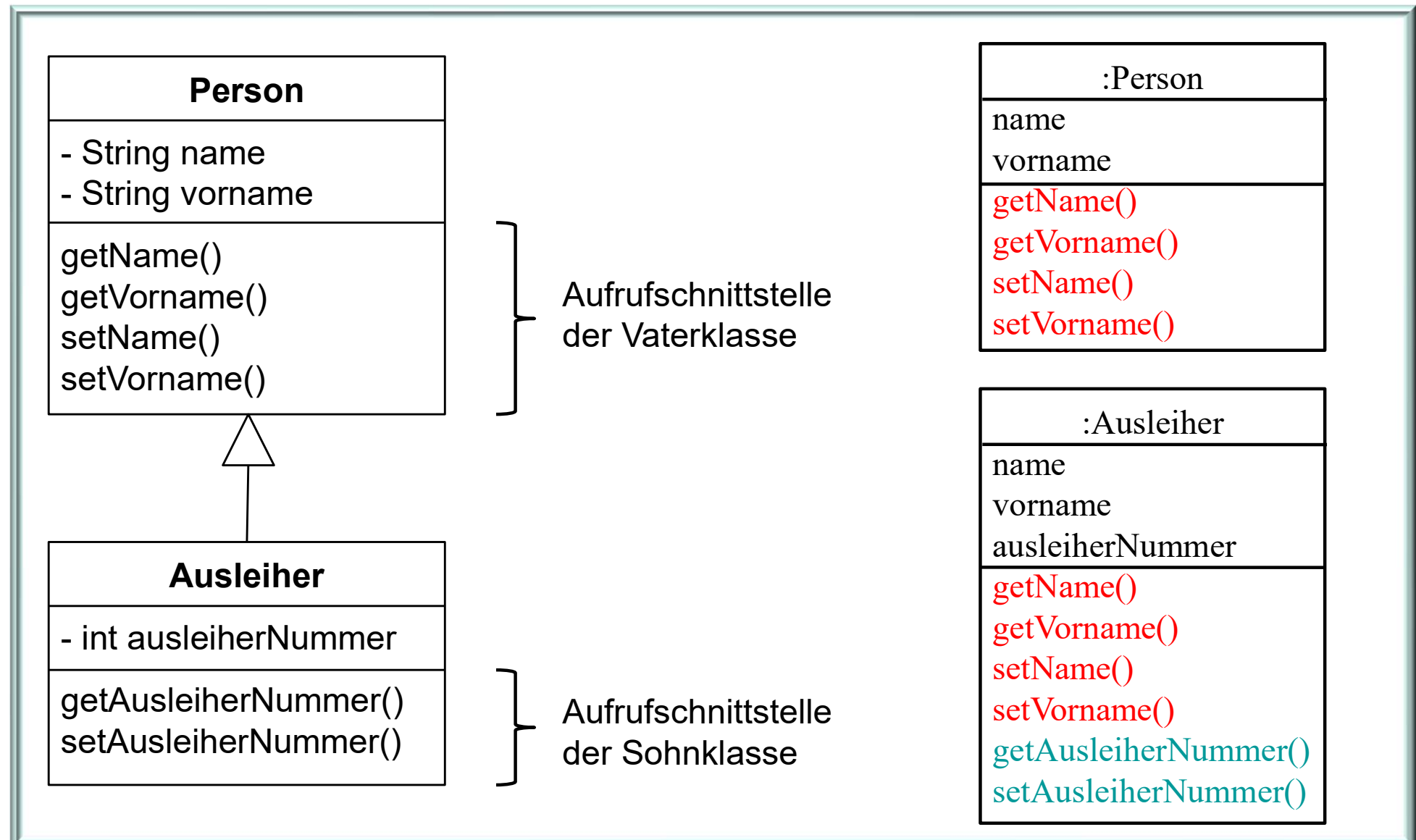
Beispiel: Bei der Methode `println()` der Klasse `PrintStream` aus der Java-Klassenbibliothek handelt es sich um eine „vielfach“ überladene Methode:

■	<code>void</code>	<code>println</code>	<code>()</code>
■	<code>void</code>	<code>println</code>	<code>(boolean x)</code>
■	<code>void</code>	<code>println</code>	<code>(int x)</code>
■	<code>void</code>	<code>println</code>	<code>(String x)</code>
■	gleicher Methodenname	verschiedene Parameterlisten



Es ist nicht zulässig, zwei Methoden zu überladen, die sich nur durch den Rückgabotyp unterscheiden.

Erweitern der Aufrufschnittstelle der Vaterklasse in der Sohnklasse



Was versteht man unter Erweitern?

- Eine Sohnklasse kann die Aufrufschnittstelle der Vaterklasse erweitern.



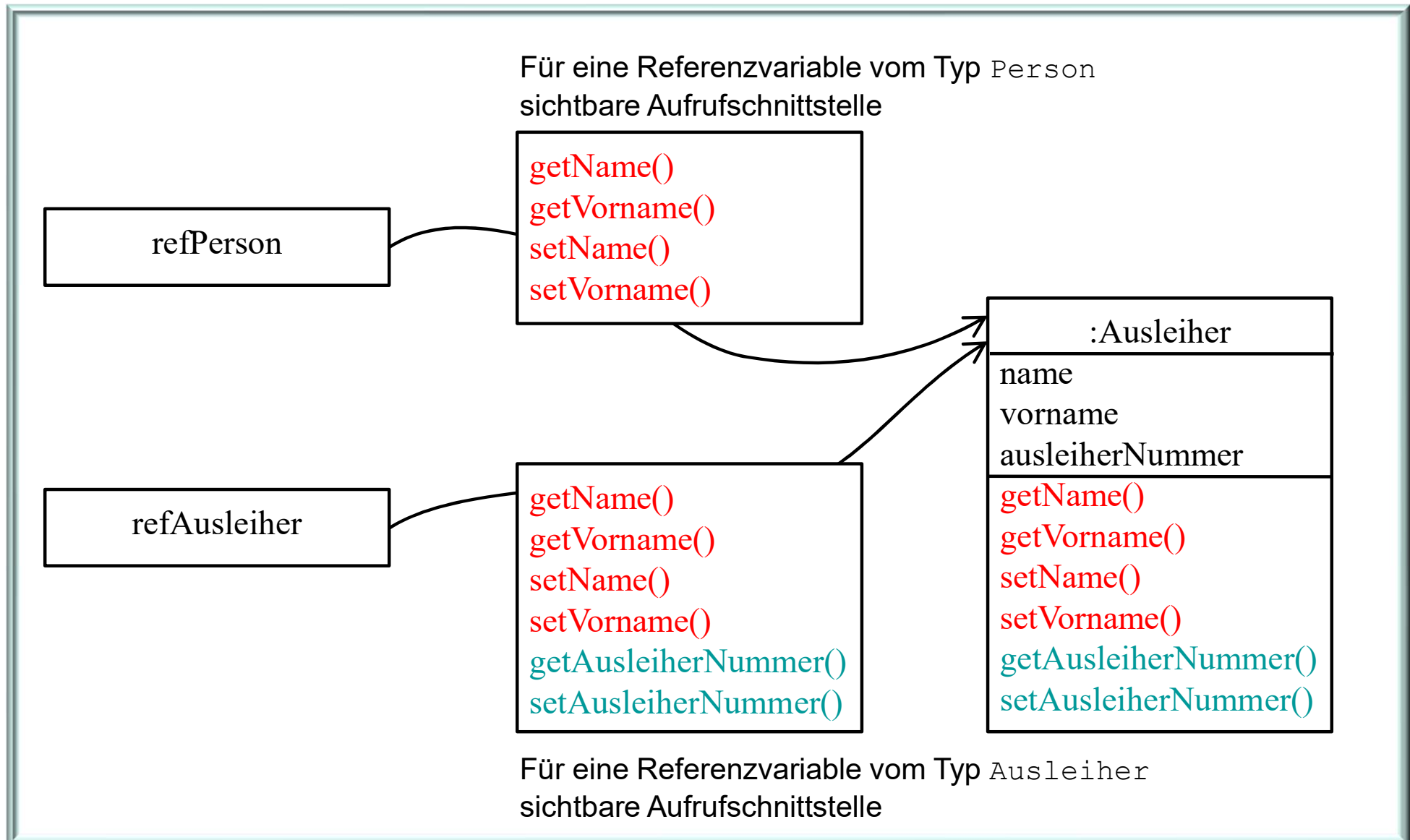
Die **öffentlichen** (`public`) **Methoden** einer Klasse werden als **Aufrufschnittstelle** bezeichnet.



Beim **Erweitern** werden in der Sohnklasse ein oder mehrere öffentliche Methoden definiert, welche die Aufrufschnittstelle der Vaterklasse erweitern (ergänzen).

- Die Klasse `Ausleiher` erweitert die Aufrufschnittstelle der Klasse `Person` durch die Instanzmethoden
 - `getAusleiherNummer()`
 - und `setAusleiherNummer()`.
- Zu einem Objekt der Klasse `Ausleiher` können die öffentlichen Methoden
 - der Klasse `Person`
 - und der Klasse `Ausleiher` aufgerufen werden.

Aufruf von Methoden mit Hilfe einer Referenzvariablen



Zuweisungskompatibilität



Welcher Codeausschnitt ist korrekt?

1

```
Person refPerson = new Ausleiher("Mustermann", "Max", 100);  
refPerson.getAusleiherNummer();
```

2

```
Ausleiher refAusleiher = new Ausleiher("Mustermann", "Max", 100);  
refAusleiher.getAusleiherNummer();
```

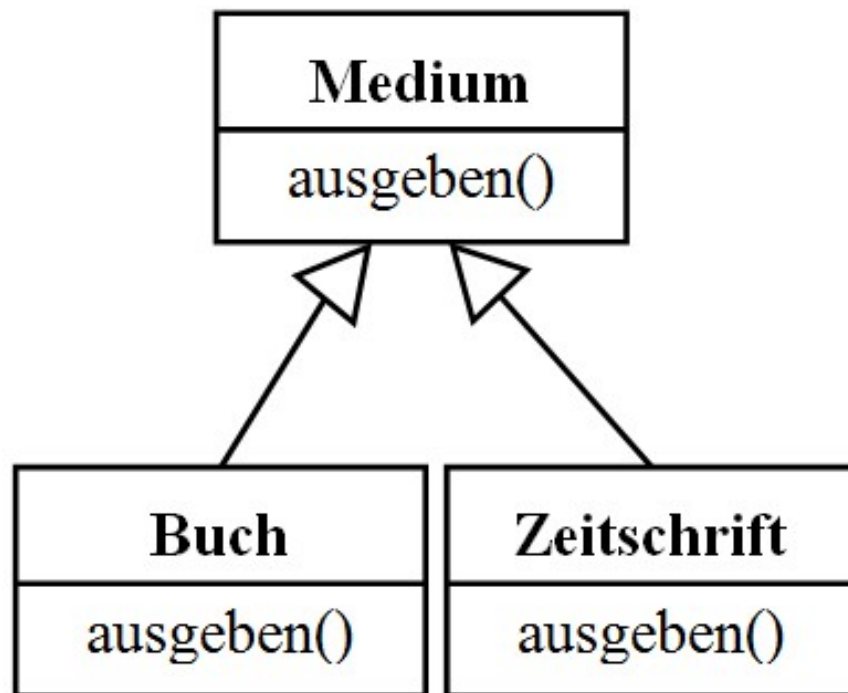


Der Typ der Referenzvariablen bestimmt, welche Methoden aufgerufen werden können.

Was ist Überschreiben?



Beim Überschreiben wird eine Instanzmethode, die zur Aufrufschnittstelle der Vaterklasse gehört, in der Sohnklasse nochmals definiert und mit einer für die Sohnklasse angepassten Implementierung versehen.



- In der Klasse `Buch` wird die Methode `ausgeben()` der Klasse `Medium` überschrieben.
- In der Klasse `Zeitschrift` wird die Methode `ausgeben()` der Klasse `Medium` ebenfalls überschrieben.