

Objektorientierung I

((itanius informatik))

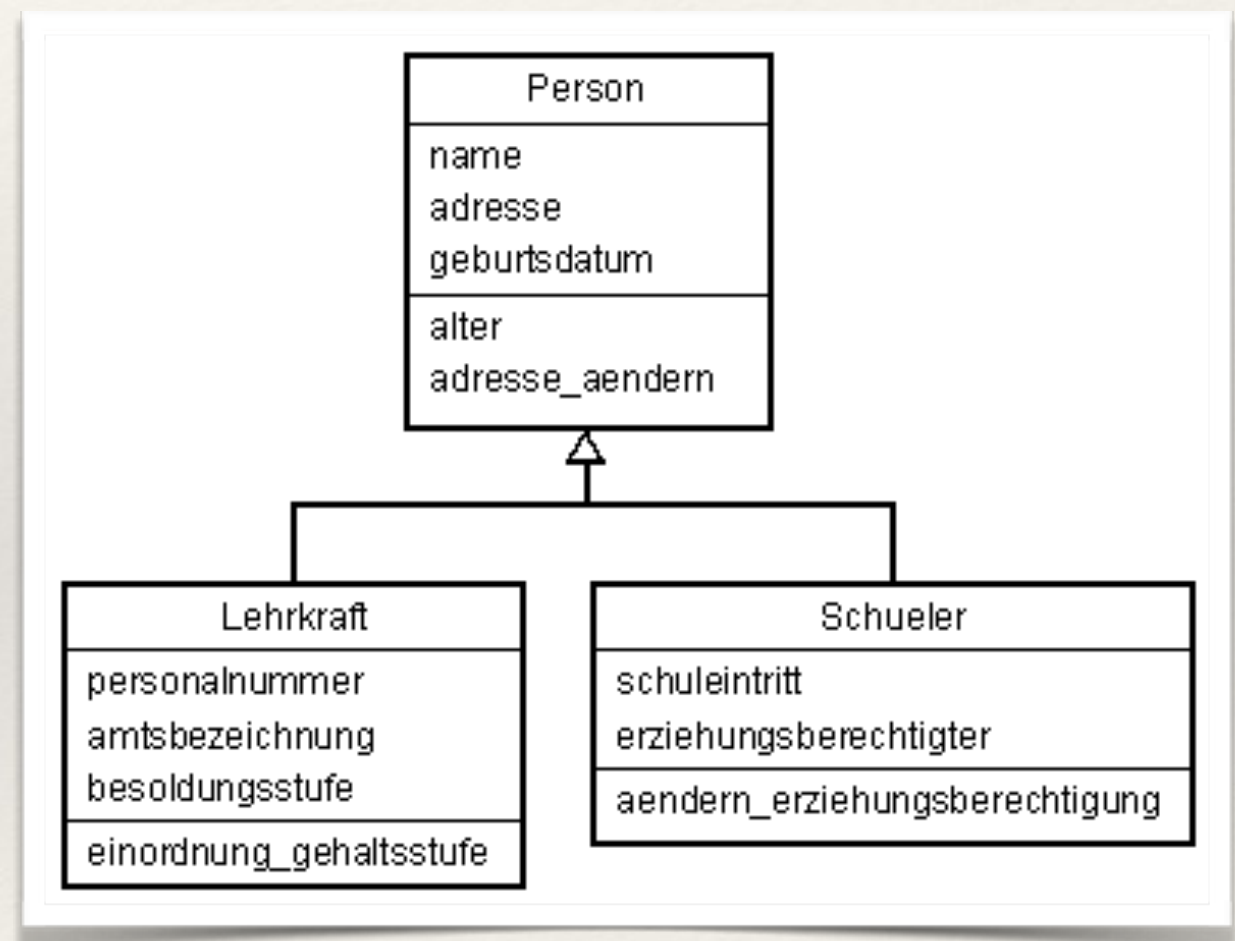
Java Foundation Track

Objektorientierung I

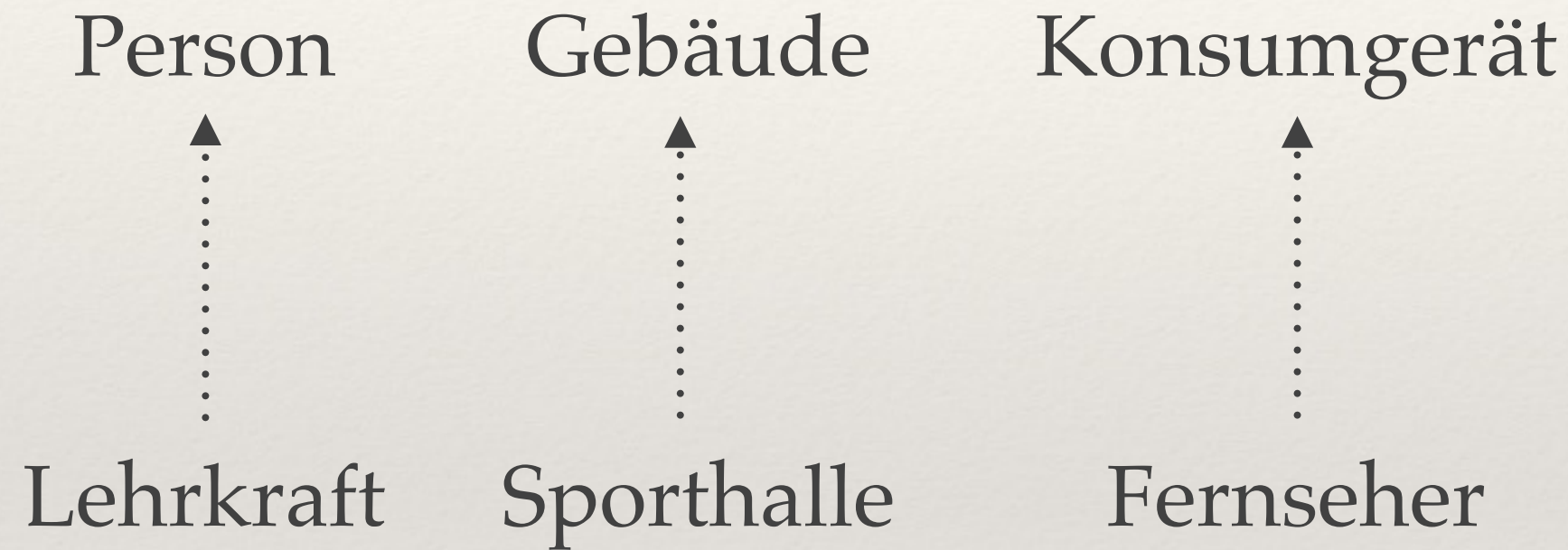
- Generalisierung
- Vererbung
- Kapselung
- Polymorphismus

Generalisierung / Spezialisierung

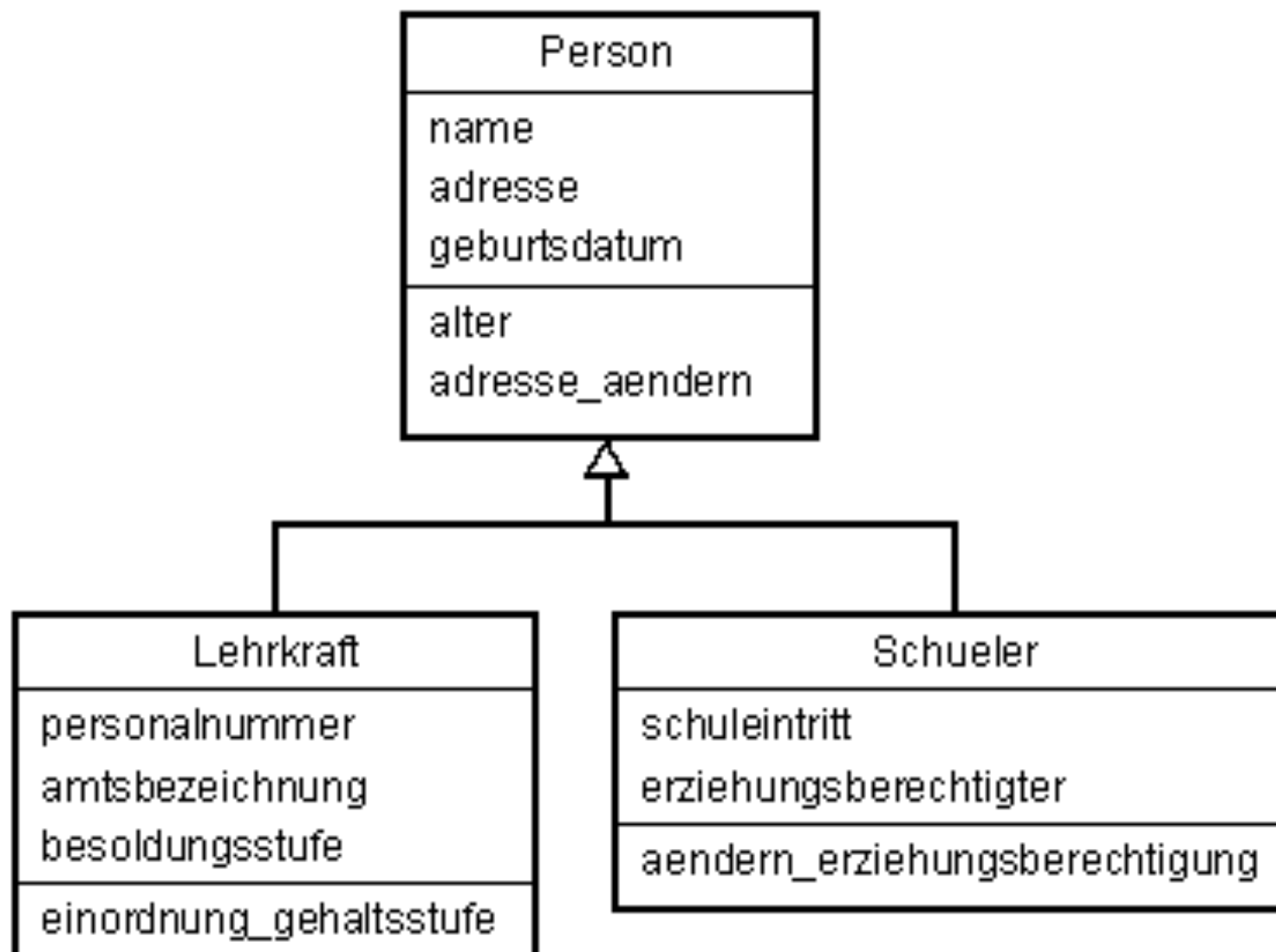
- ❖ ist-ein Beziehung und meint die Beziehung zwischen ähnlichen Klassen
- ❖ werden durch Vererbung ausgedrückt
- ❖ Klasse wird nicht komplett neu definiert, sondern von anderer Klasse abgeleitet



Vererbung



Vererbung



Kapselung

- ❖ In der Objektorientierung gilt das **Geheimnisprinzip**.
- ❖ Für das Verwenden der Klasse muss ich so **wenig** wie möglich wissen.
- ❖ Der Zugriff von außen funktioniert einzig und allein über **Schnittstellen**

Kapselung

- ❖ Verbindet Datenstrukturen und auf ihnen operierende Methoden
- ❖ Versteckt Implementierungsdetails

Mögliche Zugriffsarten bei der Kapselung

public (+) - Zugriff von außerhalb und innerhalb der Klasse möglich

private (-) - Zugriff nur innerhalb der Klasse möglich

protected (#) - Zugriff nur innerhalb der Klasse und von Spezialisierungen derselben

package (~) - Zugriff für alle Elemente innerhalb des eigenen Pakets

Typisches Beispiel Datenkapselung

```
private int myAttribute;
```

```
public void setMyAttribute(int myAttribute) {  
    this.myAttribute = myAttribute;  
}
```

```
public int getMyAttribute() {  
    return this.myAttribute;  
}
```

Polymorphismus

- ❖ Die Polymorphie der objektorientierten Programmierung ist eine Eigenschaft, die immer im Zusammenhang mit Vererbung und Schnittstellen (Interfaces) auftritt.
- ❖ Eine Methode ist polymorph, wenn sie in verschiedenen Klassen die gleiche Signatur hat, jedoch **erneut** implementiert ist.

Beziehungen

- ❖ Man unterscheidet drei Arten von Beziehungen
 - ❖ Beziehungen zwischen Klassen und Objekten
 - ❖ Beziehungen zwischen Klassen
 - ❖ Beziehungen zwischen Objekten

Beziehungen

- ❖ Objekte und Klassen haben für gewöhnlich Beziehungen zueinander, ein Fahrrad ähnelt einem Motorrad, ein Auto hat Reifen etc.
- ❖ in der Objektorientierung reduzieren sich die Beziehungen auf wenige Grundtypen
 - ❖ is-a Beziehungen (Generalisierung, Spezialisierung)
 - ❖ part-of Beziehungen (Aggregation, Komposition)
 - ❖ Verwendungs- oder Aufrufbeziehungen

Im Buch gibt es mehr Infos unter Kapitel 3.1

[http://openbook.galileocomputing.de/javainsel/
javainsel_03_001.html#dodtp394ff2f8-9134-4eb3-a6b4-469ed4572a74](http://openbook.galileocomputing.de/javainsel/javainsel_03_001.html#dodtp394ff2f8-9134-4eb3-a6b4-469ed4572a74)

Übung

Erstellen Sie ein Klassendiagramm, in dem folgende Klassen vertreten sind(Vererbung)

Mensch

Sportler,

Fußballer,

Priester,

Torwart,

Stürmer,

Büroangestellter

Finden Sie jeweils zwei sinnvolle Attribute und eine sinnvolle Methode! Verwenden Sie Zugriffsmodifikatoren.

Übung

- ❖ Versuchen Sie ihr Klassengerüst zu implementieren.
- ❖ Kleiner Hinweis. Das Wort extends erweitert eine Klasse(bildet Vererbung ab)