Höhere objektorientierte Konzepte

Küchlin/Weber: Einführung in die Informatik

Vererbung und abgeleitete Klassen

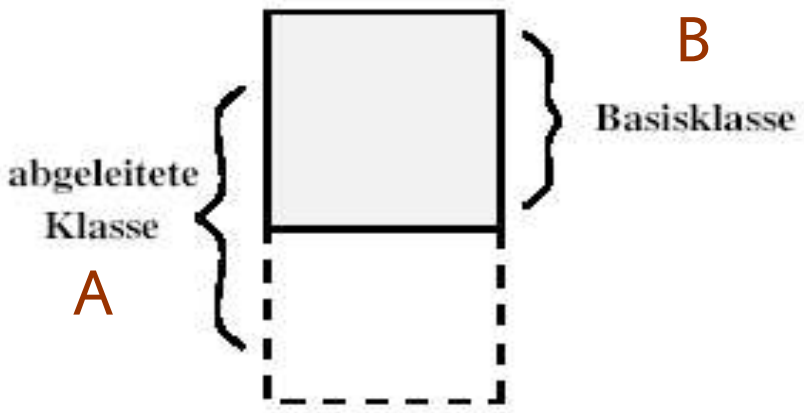
- **Vererbung** (*inheritance*) ist ein Mechanismus, mit dem die „**ist-ein**“ (*is-a*)-Beziehung zwischen Datentypen
  - dargestellt und
  - ausgenutzt

  werden kann

Vererbung und abgeleitete Klassen

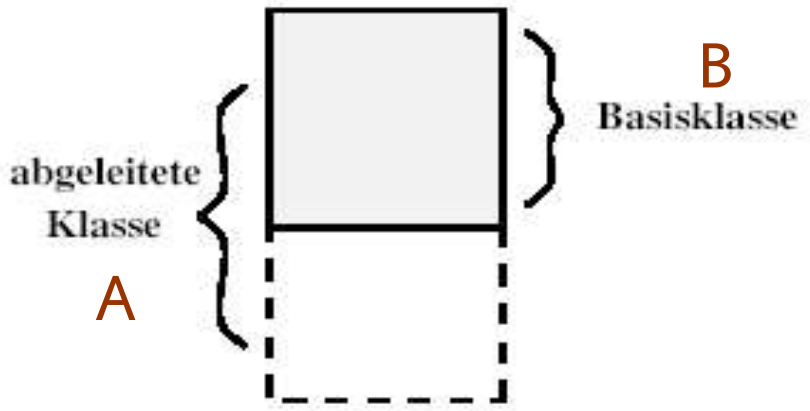
- Datentyp A ein
  - **Untertyp** (*subtype*) bzw.
  - **abgeleitete Klasse**

  eines
  - **Obertyps** (*supertype*)
  - **Basistyps** B



Vererbung und abgeleitete Klassen

- Verschiedene Sichtweisen (je nach konkretem Fall)
  - Jedes a aus A **ist** (auch) **ein** B
  - Die Klasse A ist eine **Erweiterung** der Klasse B
  - Die Klasse A **hat alle Eigenschaften** von B (und eventuell noch weitere)



Vererbung und abgeleitete Klassen

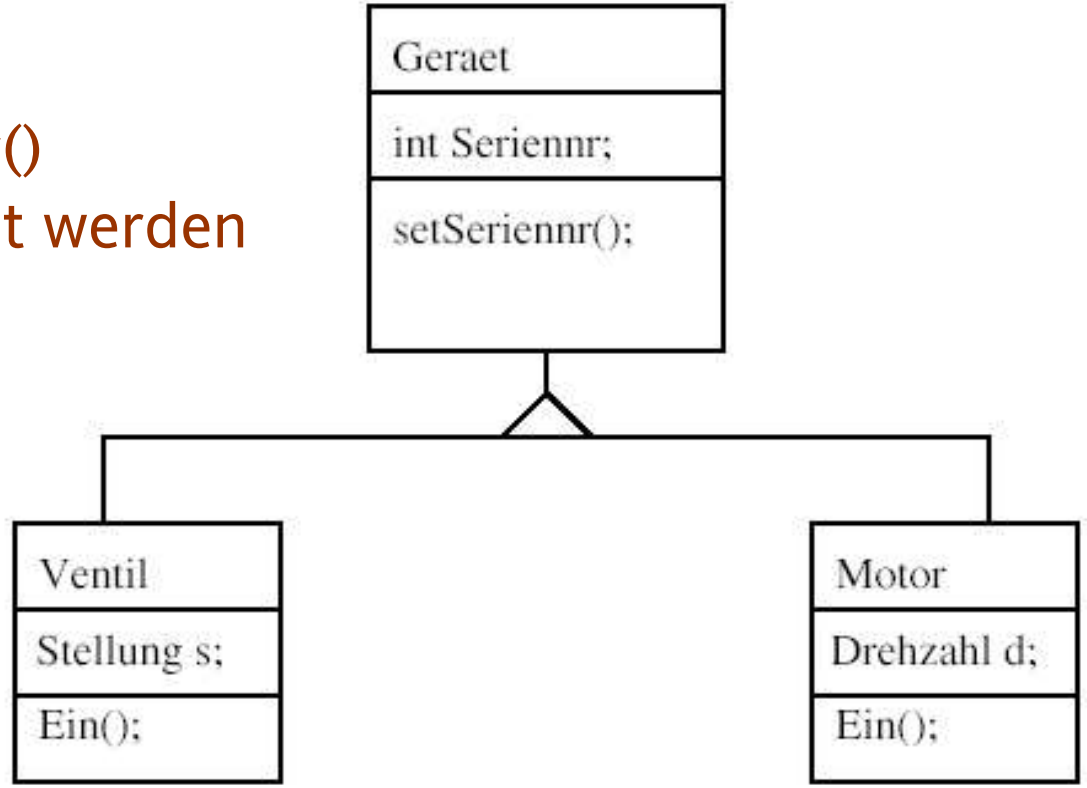
- Methoden von B können in A wieder verwendet werden
  - **Reale Vererbung:**
    A verwendet tatsächlich die Implementierung einer Methode aus B wieder
  - **Virtuelle Vererbung:**
    A verwendet die Methoden-Schnittstelle wieder, überschreibt aber die Implementierung.

Vererbung und abgeleitete Klassen

- Beispiel

Ventil erbt
- erbt Attribut seriennr
- erbt Methode setSeriennr()
- kann als Geraet verwendet werden

```
Ventil v = new Ventil();
v.setSeriennr(1508);
Geraet g = new Ventil();
```

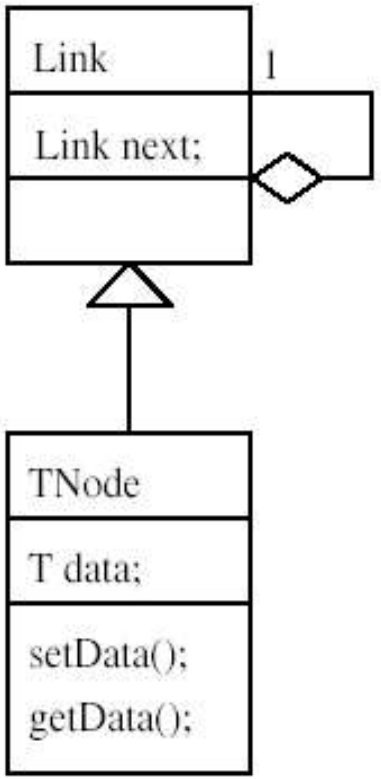


Vererbung und abgeleitete Klassen

- Beispiel: Extraktion von gemeinsamen Teilen

```
class Link {
    Link next;
}

class TNode extends Link {
    T data;
    public void setData(T d){
        data = d;
    }
    public T getData() {
        return data;
    }
}
```



Konstruktoren in Klassen-Hierarchien

- In einer abgeleiteten Klasse A wird der Konstruktor der Basisklasse B mit

  super()

  super(arg)

  etc. bezeichent

- Methode m() der Basisklasse kann von der abgeleiteten Klasse aus mit

  super.m()

  aufgerufen werden

Das Kreis-Ellipse-Problem

- **Was ist die Basisklasse: Kreis oder Ellipse?**
- 1. Möglichkeit

```
class Ellipse {
    Punkt mittelpunkt;
    int kleinerHalbmesser, grosserHalbmesser;
    void zeichne() { ... }
    int berechneFlaeche() { ... }
}
```

  - Kreis kann problemlos von Ellipse erben
  - Kreis ist Ellipse mit zusätzlicher Invariante

```
kleinerHalbmesser == grosserHalbmesser
```

Das Kreis-Ellipse-Problem

- **Was ist die Basisklasse: Kreis oder Ellipse?**
- 2. Möglichkeit

```
class Ellipse {
    Punkt mittelpunkt;
    int kleinerHalbmesser, grosserHalbmesser;
    void zeichne() { ... }
    void verzerre(int xFaktor, int yFaktor) { ... }
}
```

  - Kreis kann **nicht** von Ellipse erben
  - Aufruf von verzerre kann Kreiseigenschaft (die Invariante) zerstören

Das Kreis-Ellipse-Problem

- **Was ist die Basisklasse: Kreis oder Ellipse?**
- 3. Möglichkeit

```
class Kreis {
    Punkt mittelpunkt;
    int flaeche;
}
class Ellipse extends Kreis {
    Punkt zweiterPunkt;
    void verzerre(int xFaktor, int yFaktor) { ... }
}
```

Das Kreis-Ellipse-Problem

- **Was ist die Basisklasse: Kreis oder Ellipse?**
- 3. Möglichkeit
  Dann
  - Nicht: Ellipse „ist ein“ Kreis
  - Sondern: Ellipse „ist ein“ Ding mit
    - Mittelpunkt und Fläche (wie Kreis)
    - Außerdem zweitem Punkt und Methode zum verzerren

Das Kreis-Ellipse-Problem

- **Was ist die Basisklasse: Kreis oder Ellipse?**
- Keine klare Antwort möglich
- Hängt von der konkreten Modellierung ab