

Informatik II Übung, Woche 17

Giuseppe Accaputo

28. April, 2016

Plan für heute

1. Vererbung
2. Vorbesprechung Übung 8
3. Erweiterung des Vorlesungsverzeichnis (VVZ)
(Live-Programmierung)

Vererbung I

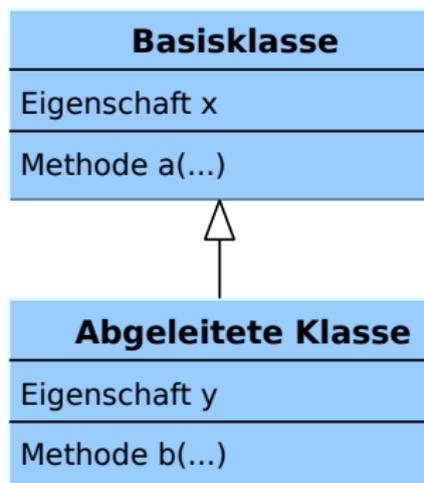


Figure: Vererbung dargestellt mittels UML. Die abgeleitete Klasse hat die Attribute x und y und verfügt über die Methoden a und b (Quelle: Wikipedia)

Vererbung II

- ▶ Die vererbende Klasse wird meist *Basisklasse* genannt
- ▶ Die erbende Klasse wird meist *abgeleitete Klasse* genannt
- ▶ Abgeleitete Klasse und Basisklasse stehen in einer "ist-ein"-Beziehung untereinander
- ▶ Gemeinsame Eigenschaften und Operationen können also zusammengefasst werden

Vererbung: Beispiel mit Tieren

Klassenaufgabe: Zeichne ein passendes Klassendiagramm (inkl. Vererbung) bestehend aus den folgenden Klassen:

- ▶ Papagei
- ▶ Katze
- ▶ Säugetier
- ▶ Haustier
- ▶ Hund
- ▶ Vogel
- ▶ Schäferhund

Übung 8: Geometrische Figuren

In Übung 8 wird verlangt, geometrische Figuren zu implementieren.

- ▶ Gemeinsame Eigenschaften (Variablen):
Position und Farbe
- ▶ Gemeinsame Operationen (Funktionen):
Berechnung der Fläche, Zeichnen der Figur, Rückgabe der Farbe
- ▶ Gemeinsame Operationen, die jedoch Figur-spezifisch sind:
Berechnung der Fläche und Zeichnen der Figur

⇒ Perfekte Situation, um eine abstrakte Klasse zu verwenden!

Die Shape Klasse

Implementiere die geometrischen Figuren Rechteck, Dreieck und Ellipse, welche von der Klasse Shape ableiten:

```
class Triangle extends Shape
```

Die Shape Klasse selbst ist eine abstrakte Klasse (`abstract`):

```
abstract class Shape
```

- ▶ Shape ist eine abstrakte Klasse, weil Figuren gemeinsame Eigenschaften und Operationen besitzen, jedoch einzelne gemeinsame Operationen Figur-spezifisch sind, also je nach Figur anders implementiert werden müssen (z.B. Berechnung der Fläche)

Das `abstract` Keyword: Abstrakte Klasse

Abstrakte Klassen sind spezielle Klassen, welche nicht instanziiert werden können, d.h. im Falle der abstrakten Klasse `Shape` würde folgendes zu einem Kompilierfehler führen:

```
Shape s = new Shape(); // Kompilierfehler!
```

- ▶ Abstrakte Klassen können grundlegende Methoden implementieren und Variablen deklarieren
- ▶ Abstrakte Klassen können einen Konstruktor definieren
- ▶ Innerhalb von abstrakten Klassen können abstrakte Methoden deklariert werden
- ▶ Eine Klasse muss als abstrakt markiert werden, sobald sie eine abstrakte Methode deklariert

Das `abstract` Keyword: Abstrakte Methoden

Abstrakte Methoden sind Methoden, welche in einer abstrakten Basisklasse (z.B. `Shape`) definiert, jedoch erst in der vererbenden Klasse (z.B. `Rectangle`) implementiert werden, d.h. die abstrakte Klasse enthält nur die Methodensignatur:

```
abstract class Shape {  
    // Returns area of the shape  
    abstract public double getArea();  
    // Draws the shape on the ImageViewer  
    abstract public void draw(ImageViewer  
        ↪ viewer);  
    ...  
}
```

Das `super` Keyword

Um beispielsweise den Konstruktor einer abstrakten Basisklasse aufzurufen, wird die Methode `super()` verwendet.

- ▶ Da die Shape Klasse als `abstract` definiert ist und einen parametrisierten Konstruktor anbietet, muss dieser mittels `super(...)` in den einzelnen Figuren-Konstruktoren aufgerufen werden:

```
public Triangle(..., boolean isPaintable)
{
    super(isPaintable);
    ... // Weiterer Code
}
```

Erweiterung des Vorlesungsverzeichnis

Klassenaufgabe: Welche Klassen könnten von einer abstrakten Klasse profitieren?

- ▶ Dozent
- ▶ Student
- ▶ Vorlesung
- ▶ Vorlesungsverzeichnis

Neue Operation: Lohnauszahlung

Wir definieren nun eine neue Operation, nämlich die Lohn-Berechnung

- ▶ Jeder Dozent verdient einen fixen Jahreslohn
- ▶ Studenten können nun auch Hilfsassistenten sein, verdienen jedoch einen fixen Monatslohn (der für alle Studenten gleich ist) während eines gegebenen Semesters

Definition einer Person

- ▶ Gemeinsame Eigenschaften (Variablen):
Name, Vorname und Emailkürzel
- ▶ Personen-spezifische Eigenschaften: Departement (Dozent),
Legi Nr. und Studiengang (Student)
- ▶ Gemeinsame Operationen (Funktionen):
Berechnung des Lohns, Rückgabe der einzelnen Eigenschaften
- ▶ Gemeinsame Operationen, die jedoch Person-spezifisch sind:
Berechnung des Lohns

Klassenaufgabe: Definition der abstrakten Klasse Person (Code)

Erweiterung des Vorlesungsverzeichnis

Live-Programmierung