

Kapitel 2: Vererbung

Lernziele

- [LZ 2.1] Das Java-Vererbungskonzept erläutern können
- [LZ 2.2] Das Prinzip der Aggregation erläutern können
- [LZ 2.3] Die für Vererbung benötigten Sichtbarkeitsstufen anwenden können
- [LZ 2.4] UML Klassenmodelle mit Vererbungsbeziehungen erstellen können
- [LZ 2.5] Verstehen, wo geerbte Eigenschaften genutzt werden können
- [LZ 2.6] Die Schlüsselworte „this“ und „super“ anwenden können
- [LZ 2.7] Konstruktorenverkettung verstehen und anwenden können
- [LZ 2.8] Die Klasse Object und ihre Methoden kennen
- [LZ 2.9] Den Operator „instanceof“ kennen und anwenden können
- [LZ 2.10] Das Schlüsselwort „final“ und seine Wirkung kennen
- [LZ 2.11] Abstrakte Klassen und Methoden kennen und ihren Einsatz motivieren können
- [LZ 2.12] Wissen, wann Aggregation und wann Vererbung genutzt werden sollte

2. Vererbung

[LZ 2.1] Das Java-Vererbungskonzept erläutern können

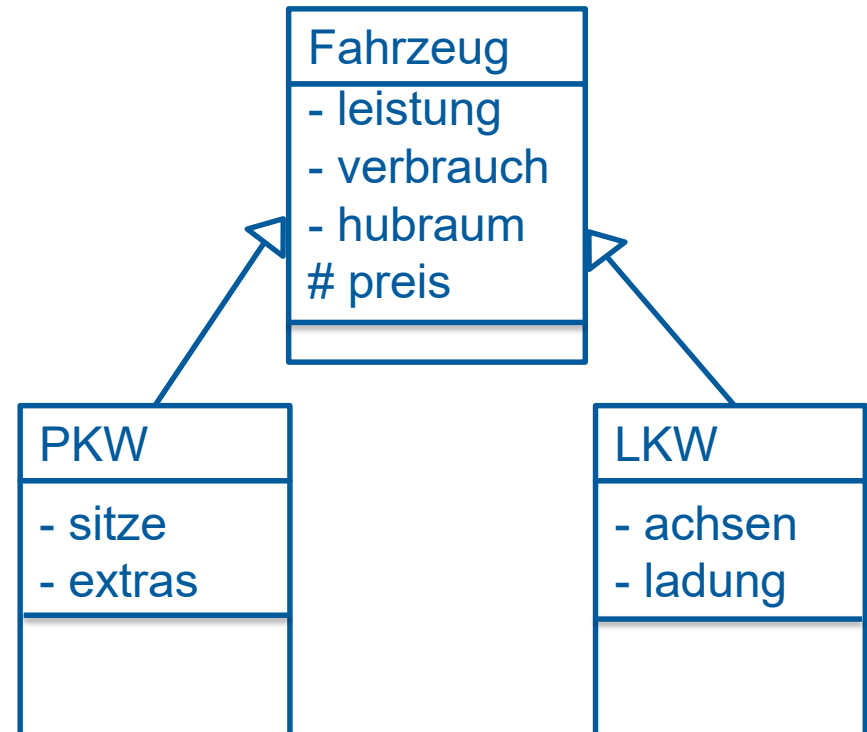
[LZ 2.2] Das Prinzip der Aggregation erläutern können

- Was versteht man unter Aggregation?
- Was versteht man unter Vererbung?
- Aggregation und Vererbung ermöglichen, ein in der Software-Entwicklung wünschenswertes Ziel zu erreichen – welches?
- Wie erkennt man sprachlich Vererbung, wie Aggregation?

2. Vererbung

[LZ 2.3] Die für Vererbung benötigten Sichtbarkeitsstufen anwenden können

- Welche Sichtbarkeitsstufen werden für die Vererbung neu eingeführt, welche Bedeutung haben diese?
- Wie kann man erreichen, dass eine Methode zwar vererbt wird, aber nicht von außen aufrufbar ist?
- [*] Welche Eigenschaften haben PKW- und LKW-Objekte und wo (außen/innen) kann man diese nutzen?



2. Vererbung

[LZ 2.4] UML Klassenmodelle mit Vererbungsbeziehungen erstellen können

Gegeben seien Bruchstücke der folgenden Klassen:

```
class Punkt { // Punkt in der Ebene
    private double x, y;
    public double getX() { ... } public double getY() { ... }
}
abstract class Flaeche { // Geometrische Fläche
    protected double abstandZweierPunkte(Punkt p1, Punkt p2) { ... }
    abstract public void verschiebe (double verschiebungInXRichtung,
        double verschiebungInYRichtung);
    abstract public double umfang();
}
class Kreis extends Flaeche {
    private Punkt mittelpunkt; private double radius;
    ... }
```

Erstellen Sie ein zu obigem Fragment passendes UML Klassenmodell aus dem auch die Vererbungsbeziehungen ersichtlich sind. Für jede Klasse sollte neben ihrem Namen auch Bezeichnung und Typ der Attribute sowie Bezeichnung, Ergebnistyp und Parameterliste der Methoden erkennbar sein. Ein abstraktes Element wird in UML durch ein vorangestelltes <<abstract>> gekennzeichnet.

2. Vererbung

[LZ 2.5] Verstehen, wo geerbte Eigenschaften genutzt werden können

Geben Sie jeweils an, welche der Anweisungen in der Methode f der Klasse Ableitung erlaubt sind.
Welche Regel gilt in diesem Zusammenhang?

```
public class Basis {
    public int a;
    protected int b;
    int c;
    private int d;

    public void f() { ... };
    protected void g() { ... };
    void h() { ... };
    private void k() { ... };
}

public class Ableitung extends Basis {
    public void f() {           // korrekt ?
        a = 1;                 // [ ]
        b = 2;                 // [ ]
        c = 3;                 // [ ]
        d = 4;                 // [ ]
        f();                   // [ ]
        g();                   // [ ]
        h();                   // [ ]
        k();                   // [ ]
    }
}
```

2. Vererbung

[LZ 2.6] Die Schlüsselworte „this“ und „super“ anwenden können

- Erläutern Sie anhand von Beispielen verschiedene Anwendungsmöglichkeiten von „this“.
- Erläutern Sie anhand von Beispielen verschiedene Anwendungsmöglichkeiten von „super“.

2. Vererbung

[LZ 2.7] Konstruktorenverkettung verstehen und anwenden können

- Was ist Konstruktorenverkettung und in welchen Situationen ist sie notwendig?
- Was tut der Java-Compiler im Zusammenhang mit Konstruktorenverkettung?
- Gegeben sei folgendes Programm. Ergänzen Sie die Lücken im Konstruktor.

```
public class Quadrat {
    private double seitenLaenge;
    public Quadrat(double seitenlaenge) {
        this.seitenLaenge = seitenlaenge;
    }
}
class BuntesQuadrat extends Quadrat {
    private String farbe;
    public BuntesQuadrat(           ) {

    }
}
```

In welcher Reihenfolge werden die Konstruktoren beim Erzeugen eines bunten Quadrats durchlaufen?



2. Vererbung

[LZ 2.8] Die Klasse Object und ihre Methoden kennen

- Welche Methoden bietet Object und wozu werden diese genutzt?
- Was müssen Sie tun, damit Quadrat-Objekte bzgl. einer von Ihnen entwickelten Quadrat-Klasse inhaltlich vergleichbar werden?
- Wie muss die Quadrat-Klasse ergänzt werden, so dass Quadrat-Objekte eine sinnvolle Ausgabe auf der Konsole erhalten?



2. Vererbung

[LZ 2.9] Den Operator „instanceof“ kennen und anwenden können

- Welche Funktion hat der Operator instanceof?
- Wie kann instanceof verwendet werden, um einen sicheren Downcast zu realisieren? Erklären Sie anhand eines Beispiels.



2. Vererbung

[LZ 2.10] Das Schlüsselwort „final“ und seine Wirkung kennen

- Was bedeutet final bei Klassendefinitionen?
- Was bedeutet final bei Methodendefinitionen?
- Was bedeutet final bei Attributdefinitionen?

2. Vererbung

[LZ 2.11] Abstrakte Klassen u. Methoden kennen und ihren Einsatz motivieren können

- Was ist eine abstrakte Klasse?
- Was ist eine abstrakte Methode?
- Wann verwendet man abstrakte Klassen und Methoden?
- Kann es eine abstrakte Methode in einer nicht-abstrakten Klasse geben?



2. Vererbung

[LZ 2.12] Wissen, wann Aggregation und wann Vererbung genutzt werden sollte

Nennen Sie die grundlegenden Regeln, die bei der Designentscheidung zwischen Vererbung und Aggregation helfen können.

2. Vererbung

[*] Übungsaufgabe

- Erstellen Sie ein Java-Programm für Wertpapiere
 - Ein Wertpapier kann eine Aktie oder eine festverzinsliche Anlage sein
 - Jedes Wertpapier hat einen Namen, eine Kennung und eine Anlagesumme
 - Ein Wertpapier kann eine Verzinsung für eine Anzahl von Jahren (int) errechnen, wobei
 - die Verzinsung bei festverzinslichen Anlagen anhand eines Zinssatzes für die jeweilige Anlage (Eigenschaft der Anlage!) errechenbar ist
 - Die Verzinsung für eine Aktie anhand der jährlichen Renditen (rückblickend!) erfolgt
- Stellen Sie obige Aussagen in einem UML-Klassenmodell dar
- Realisieren Sie das Klassenmodell mit Java und testen Sie die Anwendung, indem Sie mehrere Wertpapiere anlegen und deren Verzinsungen ausgeben.