

Vererbung

Javakurs 2013

Arne Kappen

www.freitagrunde.org

08.03.2013



Inhalt

1. Wiederholung: Objekte
2. Wozu Vererbung?
3. Abstrakte Klassen
4. Interfaces
5. Beispiel



Inhalt

1. Wiederholung: Objekte
2. Wozu Vererbung?
3. Abstrakte Klassen
4. Interfaces
5. Beispiel



Wiederholung: Objekte

- Sinnvolle Bündelung von Daten
 - Geben Struktur durch
 - Members
 - Methoden

Beispiel: Fahrzeug

```
public class Fahrzeug {  
    private int raeder;  
    private int baujahr;  
    private String farbe;  
  
    public void fahren() {}  
}
```

Inhalt

1. Wiederholung: Objekte
2. Was ist Vererbung?
3. Abstrakte Klassen
4. Interfaces
5. Beispiel



Was ist Vererbung?

- Hierarchie von Klassen
- Unterklassen “erben” Member & Methoden
 - Methoden können überschrieben werden

Wozu Vererbung?

- Ähnliche Idee wie Objekte “im Kleinen”
- Klassenhierarchien strukturieren große Programme
- Wesentlicher Bestandteil der Architektur und der Planung

Beispiel: Auto

```
public class Auto extends Fahrzeug {  
    // implizit alle Member von Fahrzeug  
    // zusätzliche eigene Member  
    private String[] gepaeck;  
  
    // alle Methoden von Fahrzeug  
    // eigene/überschriebene Methoden  
}
```

Inhalt

1. Wiederholung: Objekte
2. Was ist Vererbung?
3. Abstrakte Klassen
4. Interfaces
5. Beispiel



Abstrakte Klassen

- Erweiterung des Oberklassen-Konzepts
- Abstrakte Klasse kann nicht instanziiert werden
 - Deklaration geht nach wie vor!
- Dient nur noch der Strukturierung

Beispiel: Fahrzeug

```
public abstract class Fahrzeug {  
    private int raeder;  
    private int baujahr;  
    private String farbe;  
  
    public void fahren() {}  
}
```

- Denn es gibt DAS Fahrzeug nicht, nur die Kategorie

Inhalt

1. Wiederholung: Objekte
2. Was ist Vererbung?
3. Abstrakte Klassen
- 4. Interfaces**
5. Beispiel



Interfaces

- Keine Klasse, sondern Methodensammlung
- Klassen implementieren Interfaces
- Interfaces sichern Funktionalität zu

Beispiel: Fahrbar

```
public interface Fahrbar {  
    public void gasGeben(int geschwindigkeit);  
  
    public void bremsen(int geschwindigkeit);  
  
}
```

Beispiel: Auto

```
public class Auto implements Fahrbar {  
    public void gasGeben(int geschwindigkeit) {  
        ...  
    }  
    public void bremsen(int geschwindigkeit) {  
        ...  
    }  
}
```

Beispiel: Auto

- Jetzt ist klar: Auto ist fahrbar
- Ich kann gasGeben() und bremsen() auf Auto aufrufen
- Interface-Namen verdeutlichen Funktion
 - Comparable, Runnable...
- “sprechende” Methoden-Namen helfen!

Inhalt

1. Wiederholung: Objekte
2. Was ist Vererbung?
3. Abstrakte Klassen
4. Interfaces
5. Beispiel



