

Kapitel 10 – Vererbung:

Zusammenfassung + Übungen

In diesem Kapitel lernst du das Konzept der Vererbung kennen. Durch Vererbung kann eine Klasse Eigenschaften und Verhalten einer anderen Klasse übernehmen. Das spart Code und macht Programme flexibler erweiterbar.

Themenübersicht

1. Basisklasse und Subklasse

``class A {}`` ist die Basisklasse, ``class B extends A {}`` ist eine Subklasse, die von A erbt.

2. Vererbung von Attributen & Methoden

Eine Subklasse erbt alle ``public`` und ``protected`` Elemente der Oberklasse.

3. Konstruktoren und Vererbung

Der Konstruktor der Superklasse wird mit ``super(...)`` aufgerufen.

4. Überschreiben von Methoden

Subklassen können geerbte Methoden mit ``@Override`` neu definieren.

5. Polymorphie

Ein Objekt der Subklasse kann als Objekt der Oberklasse behandelt werden:

```
`Mitarbeiter m = new Abteilungsleiter();`
```

6. Zugriff auf geerbte Methoden

Auch ohne eigene Definition kann die Subklasse Methoden der Oberklasse nutzen.

7. Das ``super``-Schlüsselwort

Zugriff auf Konstruktoren oder Methoden der Superklasse mit ``super``.

Übungsaufgaben

1. Erstelle eine Klasse ``Tier`` mit den Attributen ``name`` und ``gewicht``.
2. Leite eine Klasse ``Hund`` von ``Tier`` ab. Füge ein Attribut ``rasse`` hinzu.
3. Überschreibe in ``Hund`` eine Methode ``lautGeben()``, die "Wuff" ausgibt.
4. Rufe im Konstruktor von ``Hund`` den Konstruktor der Oberklasse mit ``super(...)`` auf.
5. Erkläre, warum Polymorphie nützlich ist.
6. Schreibe eine Methode ``public void beschreibeTier(Tier t)``, die ``name`` und ``gewicht`` ausgibt.

7. 7. Was passiert, wenn du eine Methode in der Subklasse nicht überschreibst?
8. 8. Erstelle eine Klasse `Katze`, die auch von `Tier` erbt. Gib ihr eigene Methode `schnurren()`.