



Interfaces und Vererbung

Lösungen zu den
Übungsaufgaben

```
/**
 * Interface für Vögel die fliegen und singen.
 */
public interface Vogel {

    /**
     * Der Vogel fliegt.
     */
    public void flieg();

    /**
     * Der Vogel "singt" einen Text.
     */
    public void flieg(String text);
}
```

```
/**
 * Repräsentation eines Vermögenswerts vom Typ Oldtimer.
 */
public class Oldtimer implements Vermoegenswert {

    /**
     * Modell-Bezeichnung des Fahrzeugs
     */
    private String modell;

    /**
     * Adresse des Grundstücks.
     */
    private double basisPreis;

    /**
     * Alter in Jahren.
     */
    private int alter;

    /**
     * Konstruktor.
     */
    public Oldtimer(String modell, double basisPreis, int alter) {
        this.modell = modell;
        this.basisPreis = basisPreis;
        this.alter = alter;
    }

    @Override
    public String getName() {
        return modell;
    }

    @Override
    public double getEuroWert() {
        return basisPreis * alter;
    }

    @Override
    public Risiko getRisiko() {
        return Risiko.MITTEL;
    }
}
```

- Konsolenausgabe
 - NullPointerException beim ersten `ausgabe.ausgeben()`;
 - dann
 - Ich bin ein A.*
 - Ich bin ein B.*
- Typen
 - statischer Typ: `ISelbstausgabe`
 - dynamischer Typ
 - zunächst `null`
 - A (nach erstem `new`)
 - B (nach zweitem `new`)

```
/**
 * Generisches Interface für ein Tier, das Geräusche machen kann.
 */
public interface Tier {

    /**
     * Gibt ein Geräusch von sich.
     */
    public void machGeraeusch();

}

/**
 * Hunde machen "Wuff".
 */
public class Hund implements Tier {

    @Override
    public void machGeraeusch() {
        System.out.println("Wuff");
    }

}

/**
 * Ein Rettungshund ist ein Hund, der zusätzlich Menschen retten kann.
 */
public class Rettungshund extends Hund {

    /**
     * Rettet einen Menschen.
     */
    public void retteMensch() {
        System.out.println("Mensch erfolgreich gerettet!");
    }

}
```