



# Vererbung

Lösungen zu den  
Übungsaufgaben

Was ist die Ausgabe von ...?

```
Bird bird1 = new Penguin();  
bird1.fly();  
bird1.sing();
```

```
Bird bird2 = new Duck();  
bird2.sing();
```

```
Bird bird3 = new RubberDuck();  
bird3.sing();
```

```
Can't fly :-(  
tröt, tröt
```

```
I am duck!  
flap, flap  
quak, quak
```

```
I am duck!  
flap, flap  
Oh, I forgot, can't fly  
I am rubber duck!  
quitsch
```

# Übung: KaffeeMaschine



```
public class EspressoMaschine extends KaffeeMaschine {
    /**
     * Macht Kaffee.
     *
     * @param mengeKaffee
     *         Kaffeepulver in Gramm
     */
    public void kaffeeMachen(int mengeKaffee) {
        System.out.format("Lecker Espresso aus %d Gramm Kaffeepulver.\n",
            mengeKaffee);
    }

    /**
     * Macht Cappuccino.
     *
     * @param mengeKaffee
     *         Kaffeepulver in Gramm.
     *
     * @param milchMenge
     *         MilchMenge in Milliliter.
     */
    public void kaffeeMachen(int mengeKaffee, int milchMenge) {
        System.out
            .format(
                "Lecker Cappuccino aus %d Gramm Kaffeepulver und %d Milliliter Milch machen.
\n",
                mengeKaffee, milchMenge);
    }
}
```

## Anwendung:

```
KaffeeMaschine kaffeeMaschine =
    new KaffeeMaschine();
kaffeeMaschine.kaffeeMachen(100);
kaffeeMaschine = new EspressoMaschine();
kaffeeMaschine.kaffeeMachen(60);
((EspressoMaschine) kaffeeMaschine)
    .kaffeeMachen(60, 120);
```

## Ausgabe:

Lecker Kaffee aus 100 Gramm Kaffeepulver.  
Lecker Espresso aus 60 Gramm Kaffeepulver.  
Lecker Cappuccino aus 60 Gramm Kaffeepulver  
und 120 Milliliter Milch machen.

```
/**
 * Ein DoppelZaehler erhöht seinen Wert immer in Zweierschritten.
 */
public class DoppelZaehler extends Zaehler {

    @Override
    public void erhoehen() {
        super.erhoehen();
        super.erhoehen();
    }

    /**
     * Erhöht den Wert um 2 und liefert sich selbst zurück (Verkettung
     * von
     * Aufrufen möglich).
     *
     * @return
     */
    public DoppelZaehler doppeltErhoehen() {
        erhoehen();
        return this;
    }
}
```

# Übung: RollenspielCharakter



```
/**
 * Elternklasse für alle Rollenspiel-Charaktere
 */
public abstract class RollenspielCharakter {

    /**
     * Names des Charakters
     */
    private String name;

    /**
     * Konstruktor.
     */
    public RollenspielCharakter(String name) {
        this.name = name;
    }

    /**
     * Getter.
     */
    public String getName() {
        return name;
    }

    /**
     * Der Charakter kämpft.
     */
    public abstract void kaempfen();
}
```

```
/**
 * Ein Elf kämpft mit dem Bogen und kann Zauber sprechen.
 */
public class Elf extends RollenspielCharakter {

    /**
     * Konstruktor.
     */
    public Elf(String name) {
        super(name);
    }

    @Override
    public void kaempfen() {
        System.out.println("Schuß mit dem Bogen!");
    }

    /**
     * Ein Elf kann auch einen Zauber sprechen.
     */
    public void zauberSprechen() {
        System.out
            .println("Nio o Chitaeglr, lasto beth daer: Rimmo nim
Bruinnen Dann in Ulaer.");
    }

    /**
     * Programmeinstieg.
     */
    public static void main(String[] args) {
        RollenspielCharakter charakter = new Elf("Legolas");
        charakter.kaempfen();
        if (charakter instanceof Elf) {
            ((Elf) charakter).zauberSprechen();
        }
    }
}
```

