



Konstruktor, Sichtbarkeit, Unveränderliche Klassen

Lösungen zu den
Übungsaufgaben

Übung: Auto



```
/**
 * Repräsentation eines Auto, dessen Tankinhalt bei Fahrten simuliert wird.
 *
 * @author Philipp Jenke
 */
public class Auto {
    /**
     * Fahrzeugtyp
     */
    private String name = "Stadtflitzer";

    /**
     * Fassungsvermögen des Tanks in Liter.
     */
    private double tankkapazitaet = 32;

    /**
     * Aktueller Inhalt des Tanks in Liter.
     */
    private double tankinhalt = 32;

    /**
     * Verbr auch in Liter auf 100 km.
     */
    private double verbrauch = 8;

    /**
     * Gefahrene Strecke seit dem letzten Vulltanken.
     */
    private double gefahreneStrecke = 0;

    /**
     * Konstruktor ohne Parameter (Umsetzung des Defaults-Konstruktors).
     */
    private Auto() {
        this("Stadtflitzer", 32, 32, 8, 0);
    }

    /**
     * Konstruktor zur Initialisierung aller Objektvariablen.
     */
    public Auto(String name, double tankkapazitaet, double tankinhalt,
        double verbrauch, double gefahreneStrecke) {
        this.name = name;
        this.tankkapazitaet = tankkapazitaet;
        this.tankinhalt = tankinhalt;
        this.verbrauch = verbrauch;
        this.gefahreneStrecke = gefahreneStrecke;
    }

    /**
     * Fahren einer vorgegebenen Strecke, anpassen des Tankinhalts.
     *
     * @param entfernung
     *      Gefahrene Strecke.
     */
    public void fahren(double entfernung) {
        gefahreneStrecke += entfernung;
        tankinhalt -= entfernung * verbrauch / 100.0;
    }

    /**
     * Vulltanken des Tanks, zurücksetzen der gefahrenen Strecke.
     */
    public void vollTanken() {
        tankinhalt = tankkapazitaet;
        gefahreneStrecke = 0;
    }

    /**
     * Erzeugt eine "Selbstdarstellung" des Objektzustands als Zeichenkette.
     */
    public String toString() {
        String beschreibung = "";
        beschreibung += "name: " + name + "\n";
        beschreibung += "tankkapazitaet: " + tankkapazitaet + "\n";
        beschreibung += "tankinhalt: " + tankinhalt + "\n";
        beschreibung += "verbrauch: " + verbrauch + "\n";
        beschreibung += "gefahreneStrecke: " + gefahreneStrecke + "\n";
        return beschreibung;
    }

    /**
     * Gibt die "Selbstdarstellung" des Objektes auf der Konsole aus.
     */
    public void gibInformationenAus() {
        System.out.println(toString());
    }

    /**
     * Programmeinstieg.
     *
     * @param args
     *      Kommandozeilenargumente.
     */
    public static void main(String[] args) {
        Auto auto = new Auto();
        auto.fahren(330);
        auto.fahren(330);
        auto.gibInformationenAus();
        auto.vollTanken();
        auto.gibInformationenAus();
    }
}
```

```
/**  
 * Konstruktor zur Initialisierung einer Ganzzahl.  
 */  
Bruch(int zahl) {  
    this(zahl, 1);  
}
```

- **private** int zaehler; // Objektvariablen immer private
- **private** int nenner; // Objektvariablen immer private
- **public** Bruch()
- **public** Bruch(int zaehler, int nenner)
- **public** Bruch(Bruch andererBruch)
- **public** void initialisiere(int zaehler, int nenner)
- **public** void print()
- **public** void vereinfache()
- **public** void erweitere(int faktor)
- **private** int berechneGgt(int zahl1, int zahl2) // Interne Hilfsmethode
- **public** void addiereDazu(Bruch andererBruch)
- **public** double getWert()
- **public** void subtrahiereDavon(Bruch andererBruch)
- **public** int getZaehler()
- **public** int getNenner()
- **public** void setZaehler(int neuerWert)
- **public** void setNenner(int neuerWert)

Übung: Unveränderliche Klassen



```
/**
 * Alternative Version für erweitere in einer unveränderlichen
 * Klasse.
 */
public Bruch erweitereFinal(int faktor) {
    return new Bruch(zaehler * 2, nenner * 2);
}

/**
 * Alternative Version für vereinfache in einer unveränderlichen
 * Klasse.
 */
public Bruch vereinfacheFinal() {
    int gcd = berechneGgt(zaehler, nenner);
    return new Bruch(zaehler / gcd, nenner / gcd);
}
```