



## Übungsaufgabe – Programmierung einer Klasse in Java

Öffne BlueJ und lege die folgende Klasse an:



1. Lege die Klasse **Auto** mit den folgenden Instanzvariablen an:

- Fahrgestellnummer
- Autokennzeichen
- Farbe
- Tankvolumen
- Tankinhalt
- VerbrauchPro100km
- kmStand

2. Programmieren nun folgende Methoden

1. **AutoAnzeigen** (Es werden alle Eigenschaften der Klasse am Bildschirm ausgegeben)
2. **ummelden** (Es bekommt ein neues Kfz-Kennzeichen)
3. **umlackieren**
4. **fahren** (import: Anzahl gef. Km; ändern des km-Standes und des Tankinhaltes)
5. **tanken** (hier: volltanken, ausrechnen wieviel Liter reinpasst und Tankinhalt ändern, getankte Liter ausgeben)
6. **ReichweiteBerechnen** (in Abhängigkeit des aktuellen Tankinhaltes und des Verbrauchs wird ausgerechnet, wie viele Kilometer noch gefahren werden kann.)
7. **tanken2** (wie tanken, jedoch wird hier der Benzinpreis importiert und zusätzlich der Rechnungsbetrag ausgegeben).
8. **bewertenVerbrauch**, bei einem Verbrauch von weniger als 5 L/100km ausgibt (auf den Bildschirm druckt): *Der Verbrauch ist gering*; bzw. einem Verbrauch zwischen 5 und 10 Litern: *Der Verbrauch ist durchschnittlich* und sonst: *Das Fahrzeug hat einen überdurchschnittlichen Verbrauch*.
9. **ändernDaten**, die den Verbrauch und Tankinhalt ändert – zum Beispiel, weil die vorherigen Eingaben fehlerhaft waren.
10. **tankenJN** (hier wird ausgerechnet wieviel Kilometer man noch fahren kann und wenn die möglichen Kilometer kleiner als 100 sind, dann soll noch ausgegeben werden, das getankt werden soll)

Lege nun eine Instanz an und teste die Methoden.

IF- Beispiele

```
if (20 > 18) {
    System.out.println("20 is greater than 18");
}
if (time < 18) {
    System.out.println("Good day.");
} else {
    System.out.println("Good evening.");
}
if (time < 10) {
    System.out.println("Good morning.");
} else if (time < 18) {
    System.out.println("Good day.");
} else {
    System.out.println("Good evening.");
}
```

**Lösung Teil 1**



```
/**
 * Beschreiben Sie hier die Klasse auto.
 * Dies ist ein mehrzeiliger Kommentar
 * @author (xx)
 * @version (12.12.2024)
 */
public class auto
// - hier werden die Instanzvariablen definiert (=festgelegt)
{
    // Instanzvariablen – Datentyp & Name der Variablen
    String fahrgestellNr;
    String farbe;
    double tankvolum;
    String autokennzeichen;
    double tankinhalt;
    double verbrauchPro100km;
    double kmStand;

/**
 * Konstruktor für Objekte der Klasse auto
 * - Der Konstruktor legt fest, wie eine Instanz dieser Klasse erzeugt wird.
 */
    public auto(String inFarbe,String inFahrgestellNr,
double inTankvolum, String inAutokennzeichen,double inTankinhalt, double inVerbrauchPro100km,
double inKmStand)
    { // hinter dem Methodennamen stehen die Werte, die die Methode braucht um
//ausgeführt zu werden. In der Regel erfolgt dies durch Benutzereingaben.
        // Instanzvariable initialisieren
        fahrgestellNr = inFahrgestellNr;
        farbe = inFarbe;
        tankvolum = inTankvolum;
        autokennzeichen = inAutokennzeichen;
        tankinhalt = inTankinhalt;
        verbrauchPro100km = inVerbrauchPro100km;
        kmStand = inKmStand; }

public void umlackieren(String neueFarbe)
    { //void bedeutet, dass die Methode nichts zurückliefert.
        // hier gibt der Benutzer die neue Farbe ein
        // tragen Sie hier den Code ein - Dies ist ein einzeliger Kommentar
        farbe = neueFarbe;
        // = ist hier kein Gleichheitszeichen, sondern eine Zuweisung,
        //d.h. farbe wird neu gesetzt.
        System.out.println ("Die neue Farbe ist jetzt " + farbe);
//druckt auf der Konsole aus. }
        public void autoAnzeigen ()
        { // tragen Sie hier den Code ein
            System.out.println ("Die Farbe meines Autos ist " + farbe + ".");
            System.out.println ("Die Fahrgestellnummer lautet " + fahrgestellNr+ ".");
            System.out.println ("Das Tankvolumen beträgt " + tankvolum+ ".");
            System.out.println ("Der Tankinhalt ist " + tankinhalt+ ".");
            System.out.println ("Der Verbrauch pro 100 km beträgt " + verbrauchPro100km + "+" + ".
Der Kilometerstand beträgt " + kmStand + ".");}

public void fahren(double inKm)
    {
        // tragen Sie hier den Code ein
        kmStand = kmStand + inKm;
        tankinhalt = tankinhalt - verbrauchPro100km * inKm / 100;
        System.out.println ("Der Tankinhalt beträgt jetzt " + tankinhalt + ".");
    }
    public void reichweiteBestimmen()
    { // tragen Sie hier den Code ein
        double reichweite;
        double maxReichweite;        reichweite = (tankinhalt / verbrauchPro100km)* 100;
        maxReichweite = (tankvolum / verbrauchPro100km)* 100;        System.out.println ("Die
```



```
maximale Reichweite des Autos beträgt " + maxReichweite + "." );  
    System.out.println ("Die aktuelle Reichweite des Autos beträgt " + reichweite + "." );  
    }  
}
```



Öffne BlueJ und lege die folgende Klasse artikel mit ihren Methoden an:

### 1. Instanzvariablen

- EAN-Code
- bezeichnung
- einkaufspreis
- lagerbestand
- mindestbestellmenge
- verbrauchProMonat
- bestellmenge

### 2. Methoden

- **artikel** (Artikel erfassen, die Daten werden alle vom Benutzer eingegeben; Konstruktor)
- **artikelAusgeben** (Es werden alle Eigenschaften der Klasse am Bildschirm ausgegeben)
- **wertErmitteln** (Wert des Lagerbestandes berechnen und auf dem Bildschirm ausgeben)
- **verbrauchen** (import : verbrauchte Menge; Lagerbestand wird entsprechend reduziert)
- **anliefern** (der Lagerbestand wird um die bestellte Menge erhöht.)
- **anzMonatezurBestellungBerechnen** (hier wird ausgerechnet, wie viele Monate es noch dauert, bis die Mindestbestellmenge erreicht ist.)
- **setEinkaufspreis** (der Einkaufspreis hat sich geändert und wird aktualisiert)
- **optimaleBestellmenge** (import: bestellfixeKosten und LagerkostenInProzent; Ausgabe: optimaleBestellmenge)



### Optimale Bestellmenge:

Die optimale Bestellmenge wird mit der Andlerschen Formel berechnet. Diese Formel hilft dir, die Bestellmenge zu ermitteln, bei der die Gesamtkosten aus Bestell- und Lagerkosten minimiert werden. Die Formel lautet:  $q = \sqrt{(2 \cdot \text{Jahresbedarf} \cdot \text{bestellfixeKosten}) / (\text{Einkaufspreis} \cdot \text{Lagerkostensatz})}$ .

Die Wurzel kann durch hoch 0,5 gerechnet werden. Dazu kann die Funktion `math.pow` benutzt werden.

Bsp. `math.pow(2,3) = 23 = 8`