



## Übungsaufgaben

### Aufgabe 1

1. Dokumentiere diesen Code. Was passiert hier in jeder einzelnen Programmzeile?
2. Was wird auf der Konsole bei Ausführung der Methoden erzeugt, wenn der Benutzer folgende Zahl eingibt:
  - `multiply(int inpNumber)`-> 10
  - `add (int inpNumber)`-> 5
  - `count(int inpNumberUntil, int inpNumberTo)` -> 10, 20
  - `potentiate(int inpNumber)` -> 2

➔ *Anm: Die Zahlen sind so ausgewählt, dass alles im Kopf gerechnet werden kann.*

```
public class Number
{
    // Instanzvariablen - ersetzen Sie das folgende Beispiel mit Ihren Variablen
    int number;

    public Number ()
    {number = 2;
    }
    public int setNumber(int inpNumber)
    {
        number = inpNumber;
        return number;
    }
    public void multiply(int inpNumber)
    {int k;
        System.out.println("Die Zahl wird mit " + inpNumber + "... und .....");
        for (k = 0; k <= 8; k = k + 1)
            System.out.println((number+k)* inpNumber);
    }
    public static void add (int inpNumber)
    {int i;
        System.out.println("Hier wird was addiert ..... ");
        for ( i = 1; i <= inpNumber; i = i + 1)
            System.out.println(i+ inpNumber);
    }
    public static void count(int inpNumberUntil, int inpNumberTo)
    {int i;
        System.out.println("Zählen von bis ....");
        for ( i = inpNumberUntil; i <= inpNumberTo; i = i + 2)
            System.out.println(i);
    }
    public static void potentiate(int inpNumber)
    {int i;
        System.out.println("Potenz errechnen ....");
        for ( i = 0; i <= 5; i = i + 1)
            // Anm: Math.pow(a,b) berechnet die Potenz; hier ab.
            System.out.println(Math.pow(inpNumber,i));
    }
}
```



## 1. Aufgabe

Es soll eine Software für einen Autoverleiher entwickelt werden. Dazu werden einige Klassen und Methoden programmiert.

### Klasse Kunde

#### Instanzvariablen

- Name (String)
- Alter (integer)
- AnzahlUnfall (int)
- vermietungJN (boolean); default: false

#### Methoden

- kundeAnlegen  
Alle Attribute, bis auf *vermietungJN* werden durch den Benutzer eingegeben und gesetzt.
- pruefenVermietung  
Ist der Kunde unter 21 oder über 75 so soll an ihn kein Auto vermietet werden. (*vermietungJN=false*), ebenso wenn er/sie mehr als 2 Unfälle hatte. Ansonsten ist eine Vermietung möglich. (*vermietungJN=true*). Es ist die Instanzvariable zu setzen und eine entsprechende Meldung auszugeben.

### Klasse Vermietung

#### Instanzvariablen

- AnzahlTage (int)
- AnzahlKm (int)
- PreisProTag (Double)
- PreisProKm (Double)
- Gesamtpreis (Double)

#### Methoden

- **vermietungErstellen**  
Die Variablen *AnzahlTage* und *AnzahlKm* werden durch den Benutzer eingegeben und gesetzt. Die anderen drei Variablen werden durch Methodenaufruf berechnet.

*PreisProTag*, *PreisProKm* und *Gesamtpreis* sind zu berechnen und durch eine entsprechende Meldung mit allen drei Informationen auszugeben.

- **preisProTagBerechnen**  
Bei einer Leihdauer bis 10 Tagen beträgt die Leihgebühr pro Tag 80 EUR, bis 20 Tagen 70 EUR pro Tag und ab 21 Tagen 65 EUR pro Tag.

- **preisProKmBerechnen**  
Aus der folgenden Übersicht gehen die Kosten je km hervor:

Kilometeranzahl	Kosten je Kilometer und Tag
Bis 100 km	Kostenfrei
Über 100 km	0,20 EUR
Über 200 km	0,35 EUR
Über 300 km	0,50 EUR

- **preisGesamtpreisBerechnen**  
Der Gesamtpreis ist zu berechnen und durch eine entsprechende Meldung mit allen drei Informationen auszugeben.
- **PreisUebersichtErstellen**  
Der Anwender gibt den Zeitraum ein (*Anzahl Tag von* und *Anzahl Tage bis*) und es wird eine Tabelle erzeugt mit den drei Spalten *Anzahl Tage*, *Preis bei unter 100km* und *Preis bei über 200Km*.



## JAVA Info

### Vergleichsoperatoren

Um Bedingungen zu formulieren werden Vergleichoperatoren benötigt. Ein Vergleich liefert **true** oder **false** zurück.

< kleiner  
<= kleiner oder gleich  
> größer  
>= größer oder gleich  
== gleich  
!= ungleich

### UND / ODER

Vergleichsoperationen können durch **und** bzw. durch **oder** verknüpft werden.

oder **||** ist dann true, wenn mind. eine Bedingung true ist.

und **&&** ist dann true, wenn beide Bedingungen erfüllt sind.

Java has the following conditional statements:

- Use **if** to specify a block of code to be executed, if a specified condition is true
- Use **else** to specify a block of code to be executed, if the same condition is false
- Use **else if** to specify a new condition to test, if the first condition is false
- Use **switch** to specify many alternative blocks of code to be executed.

```
switch(expression) {  
  case x:  
    // code block  
    break;  
  case y:  
    // code block  
    break;  
  default:  
    // code block  
}  
} Beispiel:  
int day = 4;  
switch (day) {  
  case 6:  
    System.out.println("Today is Saturday");  
    break;  
  case 7:  
    System.out.println("Today is Sunday");  
    break;  
  default:  
    System.out.println("Looking forward to the Weekend");  
}
```

```
if (condition1) {  
  // block of code to be executed if condition1 is true  
} else if (condition2) {  
  // block of code to be executed if the condition1 is false and condition2 is true  
} else {  
  // block of code to be executed if the condition1 is false and condition2 is false  
}  
} Beispiel:  
int time = 22;  
if (time < 10) {  
  System.out.println("Good morning.");  
} else if (time < 18) {  
  System.out.println("Good day.");  
} else {  
  System.out.println("Good evening.");  
}
```

```
}
```

## LOOPS

### Beispiele für eine FOR-Schleife

```
for (int i = 0; i < 5; i++) {  
    System.out.println(i);  
}
```

```
public static void zaehlen(int inpZahlBis, int inpZahlVon)  
{  
    int i;  
    System.out.println("Zählen von bis ....");  
    for (i = inpZahlVon; i <= inpZahlBis; i = i + 1)  
        System.out.println(i);  
}
```

Das Schlüsselwort **static** bedeutet, dass für die Ausführung dieser Methode keine Instanz angelegt werden muss.

