

Was bisher geschah...

Bisher haben wir bei der Erstellung einer neuen Klasse ausschließlich die `public static void main(String[] args)`-Methode verwendet. Diese `main` entspricht unserem Hauptprogramm. Das heißt, in dem Moment, in dem wir unseren Quellcode ausführen, werden die Anweisungen Zeile für Zeile ausgeführt.

1 Das Methodenkonzept

Aus Effizienzgründen, wie auch aus dem Wunsch heraus, so wenig Quellcode wie möglich im Hauptprogramm zu verwenden, gibt es die Möglichkeit, Anweisungen, die nicht zwangsläufig immer ausgeführt werden müssen, auszulagern. Dies tun wir, indem wir die entsprechenden Anweisungen in eine sogenannte **Methode** packen.

1.1 *Wie funktioniert das?* - Methodendeklaration

Innerhalb der Klasse deklarieren wir eine Methode nach folgender allgemeiner Syntax:

```
<Zugriffsart> Rückgabetyt Methoden_Name (Liste der Parameter) {  
    Lokale Variablen;  
    Anweisungen;  
}
```

Wir schauen uns kurz ein **kleines Beispiel** an.

```
public static int berechneVielfaches(int zahl, int multiplikator) {  
    int vielfaches;  
    vielfaches = zahl*multiplikator;  
  
    return vielfaches;  
}
```

Allgemeines zur Methodendefinition:

Die Übergabe von Werten "von außen" an die Methode erfolgt über sogenannte **Parameter**. Bei der Definition kann eine Liste von beliebig vielen formalen Parametern, in der Form `Datentyp <Parameterbezeichnung>`, deklariert werden. Möchte man eine Methode verwenden, so müssen alle aufgelisteten Parameter mit Werten besetzt werden. Die übergebenen Parameter können dann innerhalb der Methode wie eine lokale Variable verwendet werden.

Es ist auch möglich eine leere Parameterliste (also keine Parameter) anzugeben.

In Java werden alle Parameter „by value“ übergeben. Das heißt, rufen wir eine Methode auf, so wird der übergebene Parameterwert in der entsprechenden Parametervariable gespeichert.

Bei der Methodendefinition muss der Typ des Rückgabewertes der Methode angegeben werden. Erlaubt sind hier grundsätzlich alle Datentypen.

Um innerhalb eine Methode einen Wert zurückzugeben, muss in der Methode die Anweisung `return` verwendet werden. Mit diesem Befehl wird die Methode auch beendet und verlassen.

Das heißt natürlich, jegliche Anweisungen, die nach *return* folgen, werden **nie** ausgeführt.

Die mit Lokale Variablen repräsentierten Variablen innerhalb der Methode können ausschließlich innerhalb der Methode verwendet werden, in welcher sie deklariert wurden.

Beispiel: Methode mit Rückgabewert String

```
public String erzeugeName(String  
    vorname, String nachname){  
    String name;  
    name = vorname + " " + nachname;  
  
    return name;  
}
```

Beispiel: Methode mit Rückgabewert int

```
public int berechneSumme(int a, int b,  
    int c){  
    int summe;  
    summe = a + b + c;  
  
    return summe;  
}
```

Falls eine Methode keinen Wert zurückgeben soll, wird als Rückgabetyt **void** (quasi "nichts") angegeben. Eine Methode **ohne** Rückgabewert wird auch *Prozedur* genannt.

Solche Prozeduren verändern entweder die Werte der globalen Variablen oder sie machen eine Konsolenausgabe.

Beispiel: Prozedur ohne Rückgabewert (mit Konsolenausgabe)

```
public void addiere(int summand1, int summand2){  
    System.out.println("Die Summe von " + summand1 + " und " + summand2 + " ist " +  
        (summand1+summand2));  
}
```

1.2 Wie nutzt man eigentlich eine Methode bzw. Prozedur? - Aufruf

Jetzt haben wir gewisse Funktionalitäten unseres Quellcodes in eine Methode ausgelagert. Um diese auch zu benutzen, müssen wir unserem Programm sagen, dass es dies tun soll. Hierfür **rufen wir die Methode auf**.

Dabei müssen wir unterscheiden zwischen einem **Methodenaufruf** und einem **Prozeduraufruf**.

1.2.1 Was heißt eigentlich Rückgabewert und was kann man damit machen? - Methodenaufruf

Wir haben zuvor bereits erwähnt, dass eine Methode mit der Anweisung *return* einen Variablenwert zurückgibt.

Das bedeutet, der Wert, welcher durch den Aufruf der Methode generiert wurde, kann im restlichen Quellcode verwendet werden. Um aber damit zu arbeiten, müssen wir den Rückgabewert irgendwo (Variable) speichern.

Beispiel: Prozedur mit Rückgabewert (mit Konsolenausgabe)

```
public static void main(String[] args){
    String vollerName = "";
    vollerName = erzeugeName("Petra", "Meier");
    System.out.println("Hallo " + vollerName);
}

public String erzeugeName(String vorname, String nachname){
    String name;
    name = vorname + " " + nachname;

    return name;
}
```

1.2.2 *Wie kann ich eine Prozedur nutzen?* - Prozeduraufruf

Bei der Verwendung einer Prozedur haben wir eben schon gesagt, dass es keinen Rückgabewert gibt. Das bedeutet für uns, wenn wir eine Prozedur aufrufen können wir dies einfach so tun, ohne auf etwas spezielle achten zu müssen.

Beispiel: Prozedur ohne Rückgabewert (mit Konsolenausgabe)

```
public static void main(String[] args){
    addiere(5,7);
}

public void addiere(int summand1, int summand2){
    System.out.println("Die Summe von " + summand1 + " und " + summand2 + " ist " +
        (summand1+summand2));
}
```