Leonard Berresheim

07 Methoden

M21 - Grundlagen der Programmierung

Hochschule für Technik und Wirtschaft Berlin
University of Applied Sciences

Methoden
Erklärung

Eine **Methode** ist eine Art **Unterprogramm**, dass irgendwo definiert ist und jederzeit **aufgerufen** werden kann.

Diesem Programm können **Parameter** übergeben werden und es kann einen **Wert** zurückgeben.

Eine Methodendefinition sieht wie folgt aus:

```
public static Rueckgabetyp Methodenname([Parameter-Liste]){
    Anweisung
}
```

Methoden
Erklärung

Folgende **Methode** addiert zwei *Integer* miteinander und gibt einen *Integer zurück.*

```
public static int addition(int zahlA,int zahlB){
    return zahlA + zahlB;
}
```

Diese Methode kann dann im Hauptprogramm wie folgt aufgerufen werden:

```
public static int ergebnis = addition(5, 4);
```

Anstatt von **5** und **4** können nun alle möglichen *Integer* **Werte** eingegeben werden.
Diese werden an die **Methode** *addition* **weitergegeben**.

Welche dann die Addition beider Zahlen **zurückgibt**.

Methoden

Aufgabe

Entwerfe eine Methode mit dem namen *multiplication*, die drei **Gleitkommazahlen** miteinander **multipliziert**.

Methoden
Aufgabe

Entwerfe eine Methode mit dem namen *multiplication*, die drei **Gleitkommazahlen** miteinander **multipliziert** und das **Ergebnis** als Gleitkommazahlen zurück gibt.

```
public static float multiplication(float zahlA, float zahlB, float zahlC){
    return zahlA * zahlB * zahlC;
}
```

Methoden
void

Eine **Methode** muss nicht unbedingt **Parameter** haben. In diesem Fall kann man die Parameterliste **leer** lassen.

```
public static int gibMirEineFuenf(){
    return 5;
}
```

Eine Methode muss auch keinen **Rückgabewert** haben. In diesem Fall muss man vor den Methoden das **Schlüsselwort** *void* schreiben.

```
public static void sayHelloTo(String name){
    System.out.println("Hello " + name);
}
```

Hier wird auch der *return* / die Rückgabe weggelassen.

Methoden
Aufgabe

Entwerfe eine Methode die ein **Array** *name* entgegennimmt und zu **jedem** Namen in dem Array “Hello <name>” **ausgibt**.

```
public static void sayHelloToMany(String[] name){
    for(String vorname : name){
        System.out.println(“Hello “ + vorname);
    }
}
```

Methoden
Aufgabe

Erstelle ein Array mit drei Namen und rufe die Methode in der main Schleife auf.

```
public static void sayHelloToMany(String[] name){
    for(String vorname : name){
        System.out.println("Hello " + vorname);
    }
}
```

Methoden
Aufgabe

Geben ist folgende **Mathematische** Funktion:

$$f(x, n) = x^2 + n^2 + nx$$

Setzte diese Funktion als Methode um.

Methoden
Aufgabe

Geben ist folgende **Mathematische** Funktion:

$$f(x, n) = x^2 + n^2 - nx$$

Setzte diese Funktion als Methode um.

```
public static float f(float x, float n){
    return x*x + n*n - n*x;
}
```

Methoden
Scope - Geltungsbereich

Variablen die **innerhalb** der Methode erstellt werden, **existieren** nur innerhalb der Methode.

Variablen von **außen**, die in der Methode **benutzt** werden sollen, müssen über die **Parameterliste** weitergegeben werden.

```
public static int add(int zahlA, int zahlB){
    int ergebnis = zahlA + zahlB;
    return ergebnis;
}

public static void main(String[] args){
    int ergebnis = 5;
    int summe = add(3+9);
    System.out.println(ergebnis);
}
```

Ausgabe:
5

Methoden
Scope - Geltungsbereich

Parameter werden vor dem benutzten **dupliziert** und behalten somit in der **Ursprungsfunktion** ihren Wert - auch wenn sie in der Methode **verändert** werden.

```
public static void swap(int zahlA, int zahlB){
    int tmp = zahlA;
    zahlA = zahlB;
    zahlB = tmp;
}

public static void main(String[] args){
    int zahlA = 5, zahlB = 10;
    swap(zahlA, zahlB);
    System.out.println(zahlA + "  " + zahlB);
}
```

Ausgabe:
```
5  10
```

Nur was **zurückgegeben** (*return)* wird gelangt nach **außen**. Alles andere geht nach **beenden** der Methode verloren.

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

www.htw-berlin.de

Quellen

[1] Grundkurs JAVA Von den Grundlagen bis zu Datenbank- und Netzanwendungen by Dietmar Abts