



# Methoden

**Eine Funktionalität unter einem Namen umsetzen**

## Brief verschicken

**Methodenname:** verschickeBrief

**Parameter:**

- Adresse, Briefmarke, Inhalt

**Methodenkörper:**

- Die Post verschickt den Brief

**Rückgabebetyp:**

- `void` (senden und vergessen)
- alternativ einen Beleg



# Pizzabacken

**Methodenname:** pizzaSalamiMachen

**Parameter:**

- Teig, Käse, Tomaten, Salami, Zeit

**Methodenkörper:**

- Ofen backt die Pizza

**Rückgabetyp:**

- Pizza Salami 🍕

---

Quelle: Kompetenznachweis Thema 3 einer Lernenden,  
<https://www.youtube.com/watch?v=oSDtCcDXcTM>



# Verantwortung einer Sache

- Jemand (Methode), macht **eine Sache gut** und übernimmt die Arbeit für alle
- Der **Methodennamen ist klar**, das jeder weiss was geschieht

## Beispiel Post

- Die Post ist gut in Briefe verschicken
- Sie Verschickt die Briefe für alle

## Beispiel Ofen

- Der Ofen kann gut backen
- Backt die Pizza

# Nutzen

- Eine Programmlogik wird nur einmal programmiert
- Eine Programmlogik kann an verschiedenen Orten verwendet werden
- Ändert sich die Logik wird diese überall geändert
- Der Methodennamen beschreibt direkt den Quellcode
- Kleine Methoden machen den Quellcode übersichtlicher

# Code-Beispiel / Struktur

```
import mytools.StdInput;

public class Beispiel {

    public static void main(String[] args) {
        int userInput = readNumberInRange(40, 60); // Verwendung
        System.out.println("Sie haben " + userInput + " eingegeben!");
    }

    private static int readNumberInRange(int min, int max) // Kopf / Signatur
    {
        // Methodenkörper / Methoden-Body
        int userInput;
        do {
            System.out.print("Geben Sie eine Zahl zwischen " + min + " und " + max + " ein: ");
            userInput = StdInput.readInt();
        } while (min > userInput || userInput > max);
        return userInput;
    }
}
```

## Verwendung (Gelb)

Die Methode wird in einer Anweisung aufgerufen

## Methoden Kopf / Signatur (Blau)

Definiert **wie** die Methode aufgerufen werden muss

## Methodenkörper (Grün)

Definiert **was** die Methode genau macht

# Code-Beispiel / Flow

```

1  import mytools.StdInput;
2
3  public class Beispiel {
4      public static void main(String[] args) {
5          int userInput = readNumberInRange(40, 60);
6          System.out.println("Sie haben " + userInput + " eingegeben!");
7      }
8
9      private static int readNumberInRange(int min, int max) // Kopf / Signatur
10     {
11         // Methodenkörper / Methoden-Body
12         int userInput;
13         do {
14             System.out.print("Geben Sie eine Zahl zwischen " + min + " und " + max + " ein: ");
15             userInput = StdInput.readInt();
16         } while (min > userInput || userInput > max);
17         return userInput;
18     }
19 }
20
21 }

```

der Zurückgegebene Wert vom Methodenaufruf wird in der Variable `userInput` gespeichert.

Methodenaufruf

Übergebene Parameter: min = 40, max = 60

Kopf / Signatur

Methodenkörper / Methoden-Body

Konsolen-Eingabe vom User

Solange bis der eingegebene Wert zwischen 40 und 60 ist

der `userInput` wird zurückgegeben "return"

der "Flow" beim Programmieren ist von Rechts nach Links!

1. Methodenaufruf, springt in die Methode
2. Schleife wird mehrfach wiederholt
3. `return` Wert wird ausgegeben
4. Der `return` Wert wird der Variable zugewiesen
5. Die Variable wird ausgegeben