

# Methoden

## Einführung

Methoden dienen dazu, wiederkehrende Befehle bzw. Befehlsabfolgen zu *kapseln* um diese nicht mehrfach programmieren zu müssen.

Man kann dadurch die eigentliche Hauptmethode `main` übersichtlicher halten und die Fehlersuche wird vereinfacht.

Eigene Methoden erstellt man **innerhalb** der Klasse und **vor** der `main`-Methode. (*Anmerkung: Prinzipiell lassen sich Methoden auch nach der `main`-Methode anlegen, in der Schule einigen wir uns darauf, dass eigene Methoden davor programmiert werden.*)

Eine Methode hat folgenden Aufbau:

```
public static void NAME(DATENTYP1 PARAMETER1, DT2 P2 [ ,...]) {  
    ...      // Befehle  
}
```

Listing 1: Aufbau einer Methode

`public static` gehört zu Beginn immer dazu. Die genaue Bedeutung davon wird erst bei der *objekt-orientierten Programmierung* wichtig.

`void` ist der *Rückgabotyp*, näheres dazu auf dem nächsten Arbeitsblatt.

Der Name der Methode ist frei wählbar, muss aber innerhalb einer Klasse eindeutig sein.

Eine Methode kann außerdem (beliebig viele) Parameter annehmen. Auch hier gilt wiederum: jeder Parameter ist innerhalb der Methode identisch zu einer Variablen. Deshalb müssen wir für jeden Parameter ebenfalls dessen Datentyp mit angeben.

Soll eine solche Methode nun aufgerufen werden, so geschieht das mit dem Namen der Methode. Für jeden Parameter den die Methode erwartet müssen wir den *Wert* angeben. Ein Beispiel steht unter Aufgabe 1.

## 1. Aufgabe: Potenz

Erstelle ein neues Paket `ab3` mit einer Klasse `Methoden`.

Programmiert werden soll eine Methode `Pot`, die die Potenz berechnet. Diese muss natürlich zwei Parameter annehmen:

- die Basis vom Typ `double`
- den Exponent vom Typ `int`

Nach der Berechnung der Potenz soll – ebenfalls in der Methode – das Ergebnis in der Konsole ausgegeben werden.

```
Pot( 2.5 , 3 );
```

Listing 2: Beispielaufruf der Methode

Dieser Aufruf soll folgende Ausgabe erzeugen:

**Die Potenz von 2.5 hoch 3 ist 15.625**

Rufe zunächst die Methode mit fest in den Quellcode einprogrammierten Zahlen auf. Wenn alles funktioniert erweitere das Programm so, dass die Basis und der Exponent vom Benutzer über die Konsole eingegeben werden können.

## 2. Zusatzaufgabe: Rückgabewert

Bisher wird das Ergebnis der Rechnung lediglich auf der Konsole ausgegeben, wir können damit innerhalb unseres Programms noch nicht weiterrechnen.

Informiere dich über den *Rückgabewert* einer Methode, d. h. wie ein bestimmter Wert als Ergebnis der Methode zur weiteren Rechnung/Bearbeitung genutzt werden kann.

Beschreibe alle Änderungen am bisherigen Code so, dass du es allen deinen Mitschülern erklären könntest.

---

---

---

---

---

---

---

---

---

---

## 3. Zusatzaufgabe: Rekursion

Informiere dich, was *Rekursion* bedeutet und beschreibe:

---

---

---

---

---

---

---

---

---

---

Auch die Potenzberechnung lässt sich rekursiv durchführen. Ändere deine Methode dahingehend ab dass die Potenz rekursiv berechnet wird.