



Programmierung 9: Methoden mit Parametern und Rückgabe

Implementieren Sie alle Methoden in einem gemeinsamen Programm namens `AB9`. Bei jeder Aufgabe ist (mindestens) ein Test angegeben. Schreiben Sie diese Test-Anweisungen in die `onStart`-Methode und überprüfen Sie die Ausgabe Ihres Programms.

- 1** In einer Bäckerei kann man normale Brötchen für jeweils 30 ct und Körnerbrötchen für jeweils 50 ct sowie Brot für 3,50€ kaufen. Ab einem Einkaufswert von 6€ erhält man 10% Rabatt.
 Implementieren Sie die Methode `double baecker(int n, int k, int b)`, die den Preis (in Euro) für `n` normale Brötchen, `k` Körnerbrötchen und `b` Brote berechnet.
 Test 1: `System.out.println(baecker(2, 4, 0));` →1.6
 Test 2: `System.out.println(baecker(10, 2, 1));` →6.75
- 2** Der Eintritt ins Informatik-Museum kostet für eine erwachsene Person 5,50€ und für Kinder 2€. Besteht eine Gruppe aus mindestens 8 Personen, so erhält die »teuerste Person« freien Eintritt.
 Implementieren Sie die Methode `double eintritt(int e, int k)`, die den Eintrittspreis für eine Gruppe aus `e` Erwachsenen und `k` Kindern berechnet.
 Test 1: `System.out.println(eintritt(3, 6));` →23.0
 Test 2: `System.out.println(eintritt(0, 10));` →18.0
 Test 3: `System.out.println(eintritt(3, 2));` →20.5
- 3** Implementieren Sie die Methode `double widerstand(double spannung, double stromstaerke)`: Diese berechnet den elektrischen Widerstand bei gegebener spannung und stromstaerke nach dem ohm'schen Gesetz.
 Test: `System.out.println(widerstand(133.5, 2.2));` →60.681818
- 4** Implementieren Sie die Methode `double bmi(double gewicht, double groesse)`: Diese Methode berechnet den Body-Maß-Index (BMI) für eine Person, die `gewicht` kg wiegt und `groesse` Meter groß ist.
 Test: `System.out.println(bmi(79, 1.88));` →22.3517429
- 5** Implementieren Sie die Methode `String striche(int n)`: Diese Methode soll einen String aus `n` Strichen zurückgeben.
 Test: `System.out.println(striche(6));` →"IIIIII"
- 6** Implementieren Sie die Methode `int quadratsumme(int n)`: Diese Methode bildet die Summe der ersten `n` Quadratzahlen, also

$$1^2 + 2^2 + 3^2 + 4^2 + \dots + (n-1)^2 + n^2$$

Test: `System.out.println(quadratsumme(10));` →385



- 7** Implementieren Sie die Methode `int[] simulationWuerfel(int anzahl)`: Diese Methode soll das `anzahl`-malige Werfen eines Würfels simulieren und in einem Array der Länge 6 zurückgeben, wie oft die Zahlen gewürfelt worden sind.
Test: `System.out.println(simulationWuerfel(20));` → Zufällig! Es könnte so etwas wie `[3, 5, 0, 4, 2, 6]` herauskommen.
- 8** Implementieren Sie die Methode `int simulationAugensumme(int maximal)`: Diese Methode soll 1000-mal zwei Würfel werfen und die Augensumme bilden. Die Methode soll zurückgeben, wie oft dabei die Augensumme kleiner oder gleich `maximal` war.
Test: `System.out.println(simulationAugensumme(5));` → zufällig! Es sollte aber etwa 280 dabei herauskommen.
- 9** Implementieren Sie die Methode `double wurzel(double x)`: Diese Methode soll eine Näherung für \sqrt{x} mit Hilfe des Heron-Verfahrens berechnen. Das Heron-Verfahren funktioniert folgendermaßen:
- (1) Setze $a = x$ und $b = 1$.
 - (2) Wiederhole die nachfolgenden Schritte 20 mal:
 - (1) Setze a auf $\frac{a+b}{2}$.
 - (2) Setze b auf $\frac{x}{a}$.
 - (3) Das Ergebnis steht am Ende in a .
- Test: `System.out.println(wurzel(5.7));` → 2.38767
- 10** Implementieren Sie die Methode `int[] teiler(int n)`: Diese Methode soll alle Teiler der Zahl `n` in einem Array zurückgeben.
Test: `System.out.println(teiler(30));` → `[1, 2, 3, 5, 6, 10, 15, 30]`
Hinweis: Erstellen Sie zunächst ein Array der Länge `n`, in das Sie die Teiler nacheinander eintragen.
Nachdem alle Teiler eingetragen wurden, können Sie ein neues Array der passenden Länge erstellen und die gefundenen Teiler in dieses kopieren.
- 11*** Die Strichlisten-Methode aus Aufgabe 1 soll so modifiziert werden, dass nach jeweils 5 Strichen ein Leerzeichen eingefügt wird. Implementieren Sie diese Anforderung.
Test: `System.out.println(striche(12));` → "IIIII IIIII II"