

1 Funktionen

1.1 Funktionen mit einem Parameter

1.1.1 Aufgabe

Implementiere eine Funktion `square`, die eine ganze Zahl übergeben bekommt und deren Quadrat zurückgibt.

```
1 square(4)
```

16

```
1 square(7)
```

49

<https://www.codewars.com/kata/523b623152af8a30c6000027/train/kotlin>

1.1.2 Aufgabe

Implementiere eine Funktion `doubleInteger`, die das Doppelte ihres Arguments zurückgibt.

```
1 doubleInteger(3)
```

6

```
1 doubleInteger(10)
```

20

<https://www.codewars.com/kata/53ee5429ba190077850011d4/train/kotlin>

1.1.3 Aufgabe

Implementiere eine Funktion `opposite`, die die Gegenzahl ihres Arguments zurückgibt. Die Gegenzahl einer Zahl ist genauso weit von der Null entfernt, wie die Zahl selbst. Die Gegenzahl einer positiven Zahl ist negativ und die Gegenzahl einer negativen Zahl ist positiv.

<https://www.codewars.com/kata/56dec885c54a926dcd001095/train/kotlin>

```
1 opposite(2)
```

-2

```
1 opposite(5)
```

-5

1.1.4 Aufgabe

Implementiere eine Funktion `nthEven`, die eine Zahl n übergeben bekommt und die n -te gerade Zahl zurückgibt. Die Zahl n ist mindestens eins.

```
1 nthEven(2)
```

2

```
1 nthEven(1)
```

0

```
1 nthEven(5)
```

8

<https://www.codewars.com/kata/5933a1f8552bc2750a0000ed/train/kotlin>

1.2 Mehrere Parameter

1.2.1 Aufgabe

Implementiere eine Funktion `multiply`, die zwei ganze Zahlen übergeben bekommt und deren Produkt zurückgibt.

```
1 multiply(2, 3)
```

6

```
1 multiply(9, 8)
```

72

<https://www.codewars.com/kata/50654ddff44f800200000004/train/kotlin>

1.2.2 Aufgabe

Implementiere eine Funktion `otherAngle`, die aus den Größen von zwei Winkeln eines Dreiecks in Grad die Größe des dritten Winkels berechnet.

```
1 otherAngle(60,30)
```

90

<https://www.codewars.com/kata/5a023c426975981341000014/train/kotlin>

1.2.3 Aufgabe

Kopiere und korrigiere den folgenden Code:

```
1 fun add_five(num: Int):Int{  
2     val total = num + 5  
3     return num  
4 }
```

<https://www.codewars.com/kata/56200d610758762fb0000002/train/kotlin>

1.2.4 Aufgabe

Implementiere eine Funktion `past`, die die Stunden Minuten und Sekunden seit Mitternacht als Argumente übergeben bekommt und die Zeit seit Mitternacht in Millisekunden zurückgibt. Seit Mitternacht sind höchstens 23 Stunden und 59 Minuten bzw. Sekunden vergangen.

Lege um den Code einfacher lesbar zu machen, die lokalen Variablen `passedHoursInSeconds`, `passedMinutesInSeconds` und `passedTimeInSeconds` an. Implementiere außerdem eine Hilfsfunktion `seconsToMilliseconds` für die Umrechnung von Sekunden in Millisekunden.

```
1 past(0,1,1)
```

61000

```
1 past(1,1,1)
```

3661000

```
1 past(1,1,1)
```

3661000

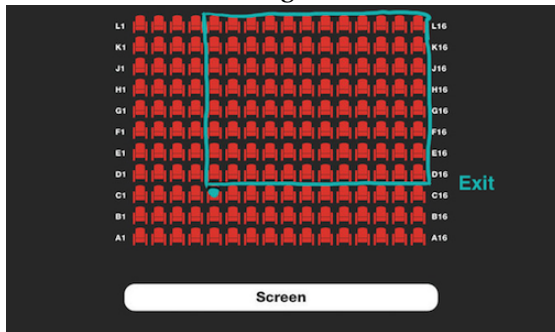
<https://www.codewars.com/kata/55f9bca8ecaa9eac7100004a/train/kotlin>

1.2.5 Aufgabe

Implementiere eine Funktion `seatsInTheater`. Diese hilft dir zu berechnen wie vielen Leuten du die Sicht versperren wirst, wenn du während einer Kinovorstellung auf die Toilette musst. Der Funktion werden die Anzahl der Spalten und Zeilen im Kino übergeben, sowie die Spalte und die Zeile in der du sitzt.

Im folgenden Bild ist für ein Kino mit mit 16 Spalten und 11 Zeilen und einen Kinobesucher in Spalte 5 und Zeile 3 zu sehen, wie vielen Zuschauern er die Sicht verdeckt.

Nutze für deine Lösung mindestens zwei lokale Variablen.



```
1 seatsInTheater(16, 11, 5, 3)
```

96

<https://www.codewars.com/kata/588417e576933b0ec9000045/solutions/kotlin>

1.2.6 Aufgabe

Schreibe Tests für die Aufgaben in diesem Kapitel.