

1 Booleans

1.1 Vergleiche

1.1.1 Aufgabe

Ein Held ist auf dem Weg zu einem Schatz, der von Drachen geschützt wird. Um einen Drachen zu besiegen braucht er zwei Kugeln. Hat er zu wenige Kugeln wird er von den Drachen gefressen. Implementiere eine Funktion `hero`, die zwei Parameter vom Typ *Integer* hat. Der erste Parameter ist die Anzahl der Kugeln, die der Held dabei hat und der zweite Parameter ist die Anzahl der Drachen. Die Funktion soll zurückgeben, ob der Held überlebt.

```
1 hero(10, 5)
```

true

```
1 hero(7, 4)
```

false

<https://www.codewars.com/kata/59ca8246d751df55cc00014c/train/kotlin>

1.1.2 Aufgabe

Implementiere eine Funktion `zeroFuel`, die dir hilft, zu berechnen, ob du es mit einem Auto und dem übrigen Benzin bis zur nächsten Tankstelle schaffst. Diese Funktion hat drei Parameter vom Typ *Integer*. Dabei handelt es sich um die Entfernung zur Tankstelle in Kilometern, die Kilometeranzahl, die mit einem Liter Benzin gefahren werden können und die übrige Tankfüllung in Litern. Sie gibt zurück, ob du es bis zur nächsten Tankstelle schaffst.

```
1 zeroFuel(50, 25, 2)
```

true

```
1 zeroFuel(100, 50, 1)
```

false

1.2 Verknüpfungen

1.2.1 Aufgabe

Implementiere eine Funktion `setAlarm`, die zurückgibt, ob ein Wecker gestellt werden muss. Die Funktion hat zwei Parameter vom Typ *Bool* (die Typannotation ist `bool`). Der erste Parameter gibt an ob eine Person angestellt ist. Der zweite Parameter gibt an, ob die Person gerade Urlaub hat. Die Funktion soll genau dann `true` zurückgeben, wenn die Person angestellt ist und gerade keinen Urlaub hat. Ansonsten muss Sie sich keinen Wecker stellen.

```
1 setAlarm(true, false)
```

`true`

```
1 setAlarm(true, true)
```

`false`

1.2.2 Aufgabe

Implementiere eine Funktion `isTriangle` die drei Parameter vom Typ *Integer* hat und zurückgibt, ob es ein Dreieck mit diesen Seitenlängen gibt.

```
1 isTriangle(1, 1, 30)
```

`false`

```
1 isTriangle(1, 2, 3)
```

`false`

```
1 isTriangle(5, 5, 7)
```

`true`

Es gibt genau dann ein Dreieck mit den angegebenen Seitenlängen, wenn jede Seite kürzer ist als die Summe der beiden anderen Seiten.

1.2.3 Aufgabe

Implementiere eine Funktion `xor` die zwei Parameter vom Typ *Bool* hat und zurückgibt, ob genau ein Argument `true` ist.

```
1 xor(true, true)
```

false

```
1 xor(true, false)
```

true

```
1 xor(false, false)
```

false

1.2.4 Aufgabe

Implementiere eine Funktion `checkForFactor`, die zwei ganze Zahl übergeben bekommt und zurückgibt, ob die zweite Zahl ein Teiler der ersten Zahl ist.

```
1 checkForFactor(12, 3)
```

true

```
1 checkForFactor(11, 2)
```

false

<https://www.codewars.com/kata/55cbc3586671f6aa070000fb/train/kotlin>

1.2.5 Aufgabe

Schreibe eine Funktion `loveFun`, die zwei ganze Zahl übergeben bekommt und prüft, ob genau eine der Zahlen gerade ist.

```
1 loveFun(4, 5)
```

true

```
1 loveFun(3, 5)
```

false

```
1 loveFun(4,6)
```

false

<https://www.codewars.com/kata/555086d53eac039a2a000083/train/kotlin>

1.2.6 Aufgabe

Schreibe eine Funktion `leapYear`, die prüft ob ein gegebenes Jahr ein Schaltjahr ist. - Alle durch 4 teilbaren Jahre sind Schaltjahre, bis auf die folgende Ausnahme - Ein durch 100 teilbares Jahr ist kein Schaltjahr, bis auf die folgende Ausnahme - Ein durch 400 teilbares Jahr ist ein Schaltjahr

```
1 isLeapYear(95)
```

false

```
1 isLeapYear(96)
```

true

```
1 isLeapYear(100)
```

false

```
1 isLeapYear(400)
```

true

<https://www.codewars.com/kata/526c7363236867513f0005ca/train/kotlin>