

Aufgabe A2

Grundsätzliches

- In dieser Aufgabe übst du den Umgang mit Schleifen und Arrays.
- Eine Aufgabe zählt nur als vollständig, wenn sie Testcode aufweist.
- Passe auf die Formulierungen auf: 'zurückgeben' heisst als `return`-Wert, 'ausgeben' heisst auf der Konsole mit `print()`-Befehlen.

Schleifen

Ressourcen

- `LoopExamples.java`
- [W3Schools](#): Seiten "Java While Loop" bis und mit "Java Break/ Continue".

Aufgabenstellungen

Schreibe deine Lösungen in die Klasse `LoopExercises.java`.

a) Schreibe eine Methode `crossSum()`, die die Quersumme eines `int`-Arguments `n` zurückgibt.

b) Schreibe eine Methode `primeFactorisation()`, die eine positive `int`-Zahl `n` in ihre Primfaktoren zerlegt. Die Methode soll die Zerlegung auf der Konsole ausgeben, z.B. für das Argument `12` soll auf der Konsole die Zeile `Prime facorisation of 12 = 2 * 2 * 3` ausgegeben werden.

x) Schreibe eine Funktion `countOnes()`, die eine `int`-Zahl `n` entgegennimmt und zählt, wie häufig darin die Ziffer 1 vorkommt.

y) Schreibe eine Funktion `factorial()`, die eine `int`-Zahl `n` entgegennimmt und ihre Fakultät $n!$ berechnet.

c) Ausprobieren statt denken:

Computer sind so schnell, dass sie gewisse Dinge einfach ausprobieren können. Password-Knackprogramme funktionieren nach diesem Prinzip.

Captain CiaoCiao blättert in der "Pirates Daily" und findet eine Denksportaufgabe: Er muss für die Buchstaben L, O, T und X jeweils eine Ziffer finden, damit die Rechnung $XOL + LXX = TLT$ stimmt. Der Preis für das Rätsel ist ein alter Kompass, den will Captain CiaoCiao unbedingt gewinnen. Doch leider fehlt ihm die Lust zu denken.

Entwickle ein Programm `tryAll()`, das durch Ausprobieren aller Möglichkeiten eine Lösung findet.

Gib alle Lösungen aus und markiere die Lösungen, in denen X, O, L und T alle unterschiedlich sind.

*d) Berechnung von Pi:

Schreibe eine Methode `pi()`, die einen Integer `digits` entgegennimmt und Pi auf mindestens `digits` Stellen nach dem Komma angenähert zurückgibt. Dabei soll die [Madhava-Leibniz-Reihe](#) gebraucht werden.

Tipp: Mit `Math.abs()` berechnest du den Betrag einer Zahl.

Zusatzaufgabe: Wie viele Stellen sind möglich? Was sind die limitierenden Faktoren, und kannst du sie umgehen/ ausschalten?

Arrays

Ressourcen

- `ArrayExamples.java`
- `ArrayTests.java`: Variablen, die für Arrays stehen, sind Objektvariablen (d.h. Zeigervariablen). Für den Unterschied siehe [GeeksForGeeks: Primitive data type vs. Object data type](#).
- [W3Schools](#): "Java Arrays"

Aufgabenstellungen

e) Bonnie Brain hat die Einträge in der Piraten-Buchhaltung umgekehrt chronologisch statt chronologisch sortiert.

Schreibe eine Methode `invert()`, die die Reihenfolge der Elemente in einem `double`-Array input umdreht.

f) Schreibe ein Programm `find(int n, int[] numbers)`, das eine Ganzzahl `n` in einem `int`-Array `numbers` sucht und den Index des ersten Fundes zurückgibt. Bei Nichtfund soll -1 zurückgegeben werden.

g) Ein magisches Quadrat ist ein Zahlenquadrat, in dem die Summe jeder Zeile, jeder Spalte und der beiden Diagonalen jeweils den gleichen Wert ergibt. Schreibe eine Methode `isMagicSquare()`, die quadratisches Array von ganzen Zahlen als Argument entgegennimmt und zurückgibt, ob das Array ein magisches Quadrat ist.

```
{ { 12, 6, 15, 1 },  
  { 13, 3, 10, 8 },  
  { 2, 16, 5, 11 },  
  { 7, 9, 4, 14 }  
}
```

ist ein magisches Quadrat.