

Übung 1: Iterative und rekursive Lösung

In dieser Übung erstellen sie zwei Programme aufgrund eines beschriebenen Algorithmus. Sie üben dabei die beiden Sichtweisen einzunehmen und diskutieren die Lösungen mit ihrem Team-Partner

1. Schritte

1. Suchen sie sich einen Tandem-Partner
2. Ihnen wird eine der folgenden - ihnen sicherlich bekannten - Algorithmen/Formeln zugewiesen.
3. Erstellen sie die notwendigen Methoden für
 1. Eine iterative Lösung
 2. Eine rekursive Lösung
4. Senden sie ihre Java-Datei an yves.nussle@tbz.ch.
5. Sie werden ihre Lösung kurz präsentieren
 1. Erklären des Algorithmus
 2. Erklären der wichtigen Elemente ihres Codes, z.B. welche Variablen welche Daten speichern, was übergeben wird, Rekursionsbasis, etc

2. Algorithmen/Formeln

2.1 Fakultät

$$n! = 1 \cdot 2 \cdot 3 \cdots n = \prod_{k=1}^n k$$

Die Fakultät ist beschrieben über die obige Formel. Erstellen sie Methoden um die n-te Fakultät zu berechnen. Verwenden sie **nicht** die Math-Bibliothek, sondern rechnen sie selbst iterativ und rekursiv.

Überprüfen sie ihr Resultat mit ihrem Taschenrechner.

2.2 Fibonacci

Die Fibonacci-Folge f_1, f_2, f_3, \dots ist durch das folgende rekursive Bildungsgesetz definiert:

$$f_n = f_{n-1} + f_{n-2} \text{ für } n \geq 3$$

Anfangswerte sind dabei:

$$f_1 = f_2 = 1$$

Erstellen sie Methoden, um die n-te Fibonacci-Zahl zu berechnen. Erstellen sie eine eigene Logik sowohl iterativ als auch rekursiv.

Überprüfen sie ihr Resultat mit einem Online-Rechner, z.B. [hier](#).

2.3 Grösster Gemeinsamer Teiler

Es gibt verschiedene Arten den ggT zu berechnen. Für eine rekursive Methode eignet sich folgende Logik:

Der ggT von zwei positiven ganzen Zahlen a und b ist:

- b , wenn bei der Division von a durch b kein Divisionsrest auftritt bzw.
- $\text{ggT}(b, a \bmod b)$ im anderen Fall.

Erstellen sie Methoden zur Berechnung des ggT iterativ und rekursiv.