

5 Potenzrechner

Der Begriff „implementieren“ steht in der Programmierung für die Umsetzung eines Algorithmus am Computer in einer Programmiersprache.

Mit Variablen vom Typ `double` kann man Fließkommazahlen speichern. Bsp.:

```
double pi = 3.1415;
```

- Beschreibe in Worten einen Algorithmus, um Potenzen mit natürlichen Hochzahlen zu berechnen.
Erkläre auch welche Variablen du benötigst und von welchem Typ sie sind.
Begründe, ob eine `for`- oder eine `while`-Schleife hier besser geeignet ist.
- Kopiere dir das Projekt „Kap1A5_Potenzen_Vorlage“ auf dein Home-Verzeichnis und öffne das Projekt in Processing.
Implementiere das Unterprogramm `berechnePotenz`.
- Verbessere dein Unterprogramm so, dass es auch Potenzen mit negativen Exponenten berechnen kann.
Bsp.: $4^1=0,25 \quad 10^3=0,001$

6 Fehlersuche

Sketch Kap1_Jans_Programm

```
void jan(double zahl) {  
    zahl = zahl * 0.3 + 7;  
}  
  
void setup() {  
    double zahl = 50;  
    jan(zahl);  
    println(zahl);  
}
```

Jan hat ein Programm erstellt, um eine Zahl zuerst mit 0,3 zu multiplizieren und anschließend 7 zu addieren. Diese Rechenoperation hat er „jan“ genannt.

- Erkläre welchen Fehler Jan gemacht hat.
- Korrigiere das Programm von Jan.
- Jan behauptet: „Egal mit welcher Startzahl gilt: Wenn man die Jan-Rechnung häufig ausführt, erhält man etwa den Wert 10“. Überprüfe seine Behauptung.

7 „Einzeiler“

Die folgenden Unterprogramme kann man in einer Zeile lösen. Ergänze die passende Zeile und passende Datentypen. Recherchiere für Rechnungen ggf. im Internet.

```
rechteckUmfang(_____, _____ a, _____ b) {  
}
```

```
darfBierTrinken(_____, _____ alter) {  
}
```

```
fahrenheitNachCelsius(_____, _____ fahrenheit) {  
}
```

8 Fibonacci-Zahlen und Wertetausch

Vorüberlegung:

- Amelie möchte die Werte ihrer Variablen a und b tauschen. Erkläre, warum ihre Idee (links) nicht funktioniert.
- Korrigiere den Programmcode für den Tausch der Variablen.
Tipp: Du benötigst eine weitere Variable als Zwischenspeicher.

Die Fibonacci-Zahlenfolge startet mit den Zahlen 1, 1, 2, 3, 5, 8, 13, 21, ... Dabei ist jede Zahl die Summe ihrer beiden Vorgänger.

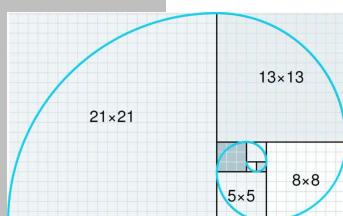
- Implementiere ein Programm, um die Fibonacci-Zahlen auf der Konsole auszugeben, bis die Obergrenze 1000 erreicht ist.

Erstelle dazu ein Unterprogramm, das man auch mit anderen Obergrenzen verwenden kann.

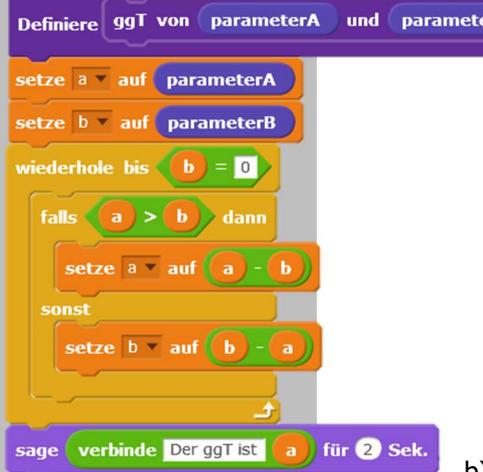
sehr schwierige Bonusaufgabe:

Erstelle einen Sketch, um die Fibonacci-Spirale zu zeichnen. Du kannst mit den Quadraten beginnen.

Für die Viertelkreise kannst du die Anweisung `arc` nachschlagen.



9 Größter gemeinsamer Teiler (ggT)



Du kennst bereits den euklidischen Algorithmus zur Berechnung des ggT von zwei Zahlen. Links siehst du eine Implementierung des Algorithmus in Scratch.

- a) Implementiere den Algorithmus in Java als Unterprogramm. Verwende diese Vorlage:

```
 Sketch Kap1_A9_ggT
ggT( _____ ) {

}

void setup() {
    println(ggT(570, 2415)); // Lösung: 15
}
```

- b) Um den Ablauf einer Schleife besser nachvollziehen zu können haben wir in Klasse 8 Ablauftabellen erstellt.

Ergänze in deinem Unterprogramm für die Berechnung des ggT `println`-Anweisungen, dass es eine Ablauftabelle wie rechts automatisch erstellt.

```
Start: a=570 b= 2415
a=570 b= 1845
a=570 b= 1275
a=570 b= 705
a=570 b= 135
a=435 b= 135
a=300 b= 135
a=165 b= 135
a=30 b= 135
a=30 b= 105
a=30 b= 75
a=30 b= 45
a=30 b= 15
a=15 b= 15
a=15 b= 0
15
```

10 Intervallschachtelung

```
 Sketch Kap1_A2_Intervallschachtelung
```

```
double mysterium(double x) {
    double untereGrenze = 0;
    double obereGrenze = x;
    double mitte = (untereGrenze + obereGrenze) / 2;
    for (int i=0; i<15; i++) {
        if (mitte * mitte > x) {
            obereGrenze = mitte;
        }
        else {
            untereGrenze = mitte;
        }
        mitte = (untereGrenze + obereGrenze) / 2;
    }
    return mitte;
}

void setup() {
    println(mysterium(25.0));
}
```

- a) Finde heraus, was das Unterprogramm `mysterium` berechnet indem du verschiedene Werte für den Parameter ausprobierst.
- b) Ergänze in dem Unterprogramm `println`-Anweisungen um eine Ablauftabelle zu erstellen.
- c) Erkläre den Algorithmus in eigenen Worten.

