

**Datum:** \_\_\_\_\_

Realisieren Sie die nachfolgenden Aufgaben mit Hilfe von `for`, `while` oder `do-while`.

### Satz des Pythagoras

Der Satz des Pythagoras definiert die Beziehungen der Seitenlängen in einem rechtwinkligen Dreieck zu  $a^2 + b^2 = c^2$ . Die Formel für die Berechnung der Hypotenuse bei gegebenen Katheten ist demnach  $c = \text{Math.sqrt}((a^2 + b^2))$ . Es ist eine bemerkenswerte Tatsache, dass die Hypotenusenlänge auch berechnet werden kann, ohne explizit die Wurzel zu ziehen.

Das folgende Verfahren wurde 1983 von C.Moler und D.Morrison beschrieben.

Zunächst berechnen Sie:

```
c = maximum(a,b);  
q= minimum(a,b);
```

Dann führen Sie drei Iterationen der folgenden Anweisungen aus:

```
r = (q/c)^2;  
s = r/ (4.0+r);  
c = c+(2.0*s*c);  
q = s*q;
```

Bereits nach nur drei Iterationen findet sich eine sehr gute Annäherung der Hypotenuse c.

**Schreiben Sie ein Programm, das dieses Verfahren implementiert.**

(Hinweis: Für die Bestimmung des Maximums der Zahlen a und b können Sie die Anweisung `Math.max(a,b)` und für das Minimum die Anweisung `Math.min(a,b)` nutzen.)

### Summe von Zahlen

Schreiben Sie ein Programm zur Berechnung der Summe der Zahlen von 1 bis n.

Ermitteln Sie das Resultat durch Summierung innerhalb einer Schleife.

### Tabelle Quadrat- und Kubikzahlen

Schreiben Sie ein Programm, das eine Tabelle von den Quadrat- und Kubikzahlen der Zahlen 1 bis n ausgibt.

| n   | $n^2$ | $n^3$ |
|-----|-------|-------|
| 1   | 1     | 1     |
| 2   | 4     | 8     |
| 3   | 9     | 27    |
| ... | ...   | ...   |

(Hinweis: Nutzen Sie für die Berechnung der Quadrat- bzw. Kubikzahlen die Anweisung `Math.pow(Basis,Exponent)`. Basis = Zahl - z.B.: 1, 2 oder 3 - und Exponent = Potenz - z.B. 2 für Quadrat und 3 für Kubik.)

Für die einheitliche Ausgabe als Tabelle (Spalten) können Sie `"\t"` als Tabstopp nutzen.

### **Quadratwurzel - do-while-Schleife**

Schreiben Sie ein Program, welches die Quadratwurzel einer natürlichen Zahl annähert.

Benutzen Sie dazu die *Folge von Heron*:

$$X_{n+1} = 1/2 * (X_n + \text{Zahl}/X_n);$$

Das Programm soll abbrechen, wenn der Unterschied zum vorherigen Näherungswert kleiner als 0.000 000 000 000 001 wird. Der Wert  $X_0$  ist die Zahl selbst.

### **Teiler einer Zahl**

Geben Sie zu einer einzugebenden Zahl alle Teiler dieser Zahl aus.

Realisieren Sie die Lösung sowohl mit einer **for**- als auch mit einer **while**-Schleife.