

Java Grundlagen - Optionale Aufgaben

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhbw.de, Raum
149B
Version: 1.0.1

1. Algorithmisches Denken

Aufgabe

1.1. Berechnung der Quadratwurzel

Bearbeitungszeit: ca. 10 Minuten für erste Lösung; ca. 15 Minuten für beide Lösungen.

Implementieren Sie eine Methode `sqrt(double x, double epsilon)`, die die Quadratwurzel von `x` mit einer Genauigkeit von `epsilon` berechnet. Verwenden Sie dazu das Newton-Raphson-Verfahren: $y_{n+1} = \frac{1}{2} \left(y_n + \frac{x}{y_n} \right)$. Implementieren Sie die Methode einmal rekursiv und einmal iterativ. Der Abbruch soll erfolgen, wenn $|y_{n+1} - y_n| < \epsilon$.

※ Hinweis

Für die rekursive Variante kann es sinnvoll sein neben der Hauptmethode eine zweite Hilfsmethode zu implementieren.

↩ Beispiel

```
# java SQRT.java
Enter number to compute SQRT for: 9
Enter epsilon: 0.0001
9.0 => 5.0
...
3.00009155413138 => 3.0000000001396984
The SQRT of 9.0 is 3.0000000001396984
```

Aufgabe

1.2. Rekursive Berechnung von E

Berechnen Sie den Wert von e auf eine vom Nutzer festgelegte Anzahl Stellen. Nutzen Sie dafür die Formel: $e = \sum_{i=0}^{\infty} \frac{1}{i!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \dots$. Implementieren Sie die Funktion rekursiv. Die Signatur der Hauptfunktion muss wie folgt sein: `double e(double precision)`.

※ Hinweis

Sie brauchen ggf. Hilfsfunktionen.

↪ Beispiel

```
# java E.java  
Enter number of decimal places of accuracy (> 0):10  
E = 2.71828182845823
```