

3.2 Aufgaben

3.2.1 Aufgabe 1: Fibonacci (★★☆☆☆)

Aufgabe 1a: Fibonacci rekursiv (★☆☆☆☆)

Schreiben Sie eine Methode `long fibRec(int)`, die die Fibonacci-Zahlen rekursiv basierend auf folgender Definition berechnet:

$$fib(n) = \begin{cases} 1, & n = 1 \\ 1, & n = 2 \\ fib(n-1) + fib(n-2), & \forall n > 2 \end{cases}$$

Beispiel Prüfen Sie die Implementierung beispielsweise mit folgendem Werteverlauf:

Eingabe	1	2	3	4	5	6	7	8
fib(n)	1	1	2	3	5	8	13	21

Aufgabe 1b: Fibonacci iterativ (★★☆☆☆)

Die rekursive Berechnung der Fibonacci-Zahlen ist nicht effizient und die Laufzeit nimmt ab etwa der 40. – 50. Fibonacci-Zahl enorm zu. Schreiben Sie eine iterative Variante zur Berechnung: Was muss man dabei für die 1000. Fibonacci-Zahl beachten?

3.2.2 Aufgabe 2: Ziffern verarbeiten (★★☆☆☆)

Aufgabe 2a: Ziffern zählen (★★☆☆☆)

Schreiben Sie eine rekursive Methode `int calcDigits(int)`, die die Anzahl an Ziffern in einer positiven natürlichen Zahl ermittelt.

Aufgabe 2b: Quersumme (★★☆☆☆)

Berechnen Sie rekursiv die Quersumme einer Zahl. Diese ist definiert als die Summe ihrer Ziffern. Schreiben Sie dazu eine rekursive Methode `int calcSumOfDigits(int)`.

Beispiele

Eingabe	Anzahl Ziffern	Quersumme
1234	4	1 + 2 + 3 + 4 = 10
1234567	7	1 + 2 + 3 + 4 + 5 + 6 + 7 = 28