

Übungen 3: Aufrufbäume



(A1) Pascalsches Dreieck



(A2) Ficonacci mit Baum

Die Fibonacci-Funktion (auch als Fibonacci-Folge bezeichnet) ist definiert als:

$$f(n) = \begin{cases} 0 & \text{wenn } n = 0 \\ 1 & \text{wenn } n = 1 \\ f(n-1) + f(n-2) & \text{sonst} \end{cases}$$

Die ersten Werte der Fibonacci-Funktion sind 0, 1, 1, 2, 3, 5, 8, 13, ... Die Definition kann einfach in eine rekursive Methode übersetzt werden:

```
fib(n: int): int
    wenn n ≤ 1:
        gib n zurück
    sonst:
        gib fib(n-1) + fib(n-2) zurück
```

- (A) Gib die vier Werte der Fibonacci-Folge an, die auf das Folgenglied 13 folgen.
- (B) Stelle die ausgeführten Methodenaufrufe bei der Ausführung von `fib(4)` als Baum dar.
- (C) Begründe, warum die Anzahl der Methodenaufrufe für `fib(n)` weniger als 2^{n+1} beträgt.
- (D) Implementiere eine Methode `fibIterativ(n: int): int`, die dasselbe Ergebnis wie `fib` berechnet, die aber ohne Rekursion arbeitet.

From:
<https://www.info-bw.de/> -

Permanent link:
<https://www.info-bw.de/faecher:informatik:oberstufe:algorithmen:rekursion:uebungen03:start?rev=1675616136>

Last update: **05.02.2023 16:55**

