

Aufgabe 6: Rekursion

Für Binomialkoeffizienten $\binom{n}{k}$ gelten neben den grundlegenden Beziehungen $\binom{n}{0} = 1$ und $\binom{n}{n} = 1$ auch folgende Formeln:

$$\begin{aligned} \text{A) } \binom{n+1}{k} &= \binom{n}{k-1} + \binom{n}{k} \\ \text{B) } \binom{n}{k} &= \binom{n-1}{k-1} \cdot \frac{n}{k} \end{aligned}$$

- (a) Implementieren Sie unter Verwendung von Beziehung (A) eine rekursive Methode `binRek(n, k)` zur Berechnung des Binomialkoeffizienten in einer objektorientierten Programmiersprache oder entsprechendem Pseudocode!

Zuerst verwandeln wir die Beziehung (A) geringfügig um, indem wir n durch $n - 1$ ersetzen:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

```

8      /**
9      * Berechnet rekursiv den Binominalkoeffizienten „n über k“. Dabei
↪      muss gelten:
10     * n &#x3E;= 0, k &#x3E;= 0 und n &#x3E;= k.
11     *
12     * @param n Ganzzahl n
13     * @param k Ganzzahl k
14     *
15     * @return Eine Ganzzahl.
16     */
17     public static int binRek(int n, int k) {
18         if (k == 0 || k == n) {

```

Code-Beispiel auf Github ansehen:
[src/main/java/org/bschlangaul/examen/examen_46115/jahr_2014/fruehjahr/rekursion/Rekursion.java](https://github.com/bschlangaul/examen/examen_46115/jahr_2014/fruehjahr/rekursion/Rekursion.java)

- (b) Implementieren Sie unter Verwendung von Beziehung (B) eine iterative Methode `binIt(n, k)` zur Berechnung des Binomialkoeffizienten in einer objektorientierten Programmiersprache oder entsprechendem Pseudocode!

```

20         } else {
21             return binRek(n - 1, k - 1) + binRek(n - 1, k);
22         }
23     }
24
25     /**
26     * Berechnet iterativ den Binominalkoeffizienten „n über k“. Dabei
↪     muss gelten:
27     * n &#x3E;= 0, k &#x3E;= 0 und n &#x3E;= k.
28     *
29     * @param n Ganzzahl n
30     * @param k Ganzzahl k
31     *
32     * @return Eine Ganzzahl.
33     */
34     public static int binIt(int n, int k) {
35         // Das Ergebnis wird als Kommazahl deklariert, da nicht alle
36         // Zwischenergebnisse ganze Zahlen sind.

```

Code-Beispiel auf Github ansehen:
`src/main/java/org/bschlangaul/examen/examen_46115/jahr_2014/fruehjahr/rekursion/Rekursion.java`

- (c) Geben Sie die Laufzeitkomplexität der Methoden `binRek(n, k)` und `binIt(n, k)` aus den vorhergehenden beiden Teilaufgaben in O-Notation an!