

Binomialkoeffizient.java

```

1 /**
2  * Eine Instanz enthält für einen Parameter {@code n} alle Werte von
3  * {@code n} über k, k = 0, ..., n}.
4  */
5 public class Binomialkoeffizient {
6
7     /**
8      * Enthält den Wert {@code n}.
9      */
10    private int n;
11
12    /**
13     * Enthält die berechneten Werte.
14     */
15    private int[] binom;
16
17    /**
18     * Erzeugt eine Instanz, die alle entsprechenden Werte
19     * {@code n} über k, k = 0, ..., n} vorhält.
20     */
21    public Binomialkoeffizient(int n) {
22        this.n = n;
23        binom = new int[n+1];
24
25        /**
26         * Bei der Berechnung der Werte wird die Symmetrie des
27         * Binomialkoeffizienten ausgenutzt und für {@code k > n - k} der
28         * bereits berechnete Wert für {@code n - k} genutzt.
29         */
30        for (int i = 0; i < binom.length; i++) {
31            if (i > n - i)
32                binom[i] = binom[n - i];
33            else
34                binom[i] = berechneNUeberK(i);
35        }
36    }
37
38    /**
39     * Berechnet die Werte des Binomialkoeffizienten mit der Variablen
40     * {@code n} und dem übergebenen Parameter.
41     * @param k für das {@code n} über k} berechnet werden soll.
42     * @return {@code n} über k}
43     */
44    private int berechneNUeberK(int k) {
45        int b = 1;
46        if (k > n - k) k = n - k;
47        for (int j = 1, m = n; j <= k; j++, m--)
48            b = b * m / j;
49        return b;
50    }
51
52    /**
53     * Gibt den Wert {@code n} über k} zurück.
54     * @param k für das {@code n} über k} zurückgegeben werden soll.
55     * @return {@code n} über k}.
56     */
57    public int getUeber(int k) {
58        return binom[k];
59    }
60
61    /**
62     * Gibt alle Werte {@code n} über k, k = 0, ..., n} zurück.
63     * @param k für das {@code n} über k} zurückgegeben werden soll.
64     * @return {@code double[] (n über 0, ..., n über n)}.
```

Binomialkoeffizient.java

```
63     */
64     public int[] getBinom() {
65         return (int[]) binom.clone();
66     }
67
68     /**
69      * Gibt den Wert zurück, für den die Werte des Binomialkoeffizienten
70      * abgelegt sind.
71      * @return {@code n}, für das alle {@code n} über k, k = 0, ..., n} abgelegt
72      * sind.
73      */
74     public int getN() {
75         return n;
76     }
77 }
78
```