

```

1 import ij.process.ImageProcessor;
2
3 public class Moments {
4     static final int BACKGROUND = 0;
5
6     static double moment(ImageProcessor ip, int p, int q) {
7         double Mpq = 0.0;
8         for (int v = 0; v < ip.getHeight(); v++) {
9             for (int u = 0; u < ip.getWidth(); u++) {
10                if (ip.getPixel(u,v) != BACKGROUND) {
11                    Mpq += Math.pow(u, p) * Math.pow(v, q);
12                }
13            }
14        }
15        return Mpq;
16    }
17
18    static double centralMoment(ImageProcessor ip, int p, int q) {
19        double m00 = moment(ip, 0, 0); // region area
20        double xCtr = moment(ip, 1, 0) / m00;
21        double yCtr = moment(ip, 0, 1) / m00;
22        double cMpq = 0.0;
23        for (int v = 0; v < ip.getHeight(); v++) {
24            for (int u = 0; u < ip.getWidth(); u++) {
25                if (ip.getPixel(u,v) != BACKGROUND) {
26                    cMpq +=
27                        Math.pow(u - xCtr, p) *
28                        Math.pow(v - yCtr, q);
29                }
30            }
31        }
32        return cMpq;
33    }
34
35    static double normalCentralMoment
36        (ImageProcessor ip, int p, int q) {
37        double m00 = moment(ip, 0, 0);
38        double norm = Math.pow(m00, (double)(p + q + 2) / 2);
39        return centralMoment(ip, p, q) / norm;
40    }
41 }

```

Programm 11.2. Beispiel für die direkte Berechnung von Momenten in Java. Die Methoden `moment()`, `centralMoment()` und `normalCentralMoment()` berechnen für ein Binärbild die Momente m_{pq} , μ_{pq} bzw. $\bar{\mu}_{pq}$ (Gl. 11.14, 11.19, 11.21).