

```
/* ----- */
/* Die Klasse Point kann jetzt von der Klasse Figure erben oder das Interface */
/* FigureInterface implementieren oder (überflüssigerweise) sogar beides. */
/* */
/* Man beachte den Unterschied zu O'Caml bei der Sichtbarkeit von Parametern: */
/* In O'Caml hat jede Klasse nur einen Konstruktor, der implizit durch die De- */
/* finition der Klasse festgelegt ist. Insbesondere werden die Parameter die- */
/* ses Konstruktors einfach als Parameter der Klasse eingeführt, so dass sie */
/* im gesamten Rumpf der Klasse sichtbar sind. */
/* In Java kann eine Klasse mehrere Konstruktoren mit unterschiedlichen Para- */
/* metern haben. Diese Parameter sind dann natürlich nur im Rumpf des einzel- */
/* nen Konstruktors sichtbar, also muss man sie explizit in Instanzvariablen */
/* abspeichern, wenn man ihre Werte an anderer Stelle benötigt. */
/* ----- */

public class Point extends Figure implements FigureInterface {

    private int x0;

    private int y0;

    public int x;

    public int y;

    private int dx;

    private int dy;

    private String c;

/* ----- */
/* Die Namen von Instanzvariablen kann man als Parameternamen wiederverwenden. */
/* Zwar werden dadurch die Namen der Instanzvariablen verdeckt, aber durch ex- */
/* plizite Verwendung von "this" kann man immer noch auf sie zugreifen. */
/* ----- */

    public Point (int x, int y, String c, int dx, int dy) {
        this.x = x; this.y = y;
        x0 = x; y0 = y;
        this.dx = dx; this.dy = dy;
        this.c = c;
    }

    public void draw () {
        System.out.print("(" + x + "," + y + ") [" + c + "]\n");
    }

    public void move () {
        x = x + dx;
        y = y + dy;
    }

    public void reset () {
        x = x0;
        y = y0;
    }

    public static void main (String [] args) {
        Point p = new Point (2, 2, "blue", 1, 1);
        p.draw ();
        p.move ();
        p.draw ();
        p.reset ();
        p.draw ();
    }
}

/* ----- */
/* Ausgabe: */
/* (2,2) [blue] */
/* (3,3) [blue] */
/* (2,2) [blue] */
/* ----- */
```