Musterlösung – Objektorientierte Programmierung in Java

Lösung zur Klausur (Point-Klasse)

Vollständige Lösung: Klasse `Point`

```java
/**
* Die Klasse Point repräsentiert einen Punkt im
   zweidimensionalen
* Koordinatensystem. Der Punkt besitzt eine x- und eine y
   -Koordinate.
*/
public class Point {

    // 1. Attribute
    private int x;
    private int y;

    /**
    * Standardkonstruktor:
    * Erzeugt den Punkt (0, 0)
    */
    public Point() {
        this.x = 0;
        this.y = 0;
    }

    /**
    * Konstruktor mit Parametern
    * @param x x-Koordinate
    * @param y y-Koordinate
    */
    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }

    // Getter und Setter
    public int getX() {
        return x;
    }

    public void setX(int x) {
```

```
        this.x = x;
    }

    public int getY() {
        return y;
    }

    public void setY(int y) {
        this.y = y;
    }

    /**
    * Verschiebt den Punkt um dx und dy
    * @param dx Delta X
    * @param dy Delta Y
    */
    public void move(int dx, int dy) {
        this.x += dx;
        this.y += dy;
    }

    /**
    * Berechnet die Distanz zu einem anderen Punkt p
    * Formel: sqrt((x - p.x)^2 + (y - p.y)^2)
    * @param p anderer Punkt
    * @return Distanz als double
    */
    public double distance(Point p) {
        int dx = this.x - p.x;
        int dy = this.y - p.y;
        return Math.sqrt(dx * dx + dy * dy);
    }

    /**
    * Spiegelt den Punkt an der x-Achse.
    * Der y-Wert wird negiert.
    */
    public void mirrorX() {
        this.y = -this.y;
    }

    /**
    * Spiegelt den Punkt an der y-Achse.
    * Der x-Wert wird negiert.
    */
    public void mirrorY() {
        this.x = -this.x;
    }

    /**
    * Gibt den Punkt als String zurück,
```

```
     * z.B. (3, 5)
     */
    @Override
    public String toString() {
        return "(" + x + ", " + y + ")";
    }

    /**
    * BONUS:
    * Prüft, ob zwei Punkte gleich sind.
    * @param p Vergleichspunkt
    * @return true, wenn x- und y-Werte übereinstimmen
    */
    public boolean equals(Point p) {
        return this.x == p.x && this.y == p.y;
    }
}
```

Testprogramm (PointTest)

```
public class PointTest {

    public static void main(String[] args) {

        // Erzeugen zweier Punkte
        Point p1 = new Point(3, 4);
        Point p2 = new Point(0, 0);

        System.out.println("Punkt p1: " + p1);
        System.out.println("Punkt p2: " + p2);

        // Punkt verschieben
        p1.move(2, -1);
        System.out.println("p1 nach move(2, -1): " + p1);

        // Distanzberechnung
        double dist = p1.distance(p2);
        System.out.println("Distanz zwischen p1 und p2: "
            + dist);

        // Spiegelung an der X-Achse
        p1.mirrorX();
        System.out.println("p1 nach mirrorX(): " + p1);

        // Spiegelung an der Y-Achse
        p1.mirrorY();
        System.out.println("p1 nach mirrorY(): " + p1);
    }
}
```

Ende der Musterlösung