

Lösungsvorschlag:

```
/**
 * Die Klasse Koordinaten stellt Methoden zur Umrechnung
 * zwischen zweidimensionalen kartesischen Koordinaten
 * und zweidimensionalen Polarkoordinaten bereit.
 */
public class Koordinaten
{
    /**
     * Berechnet zu einem Paar kartesischer Koordinaten
     * die Winkelkoordinate im entsprechenden Polarkoordinatensystem.
     *
     * @param x x-Koordinate des Paares von kartesischen Koordinaten
     * @param y y-Koordinate des Paares von kartesischen Koordinaten
     * @return die entsprechende Winkelkoordinate (Bogenmass)
     */
    public static double kartesischZuPolarWinkel(double x, double y)
    {
        if(x > 0)
        {
            if (y >= 0)
            {
                return Math.atan(y / x);
            }
            else
            {
                return Math.atan(y / x) + 2 * Math.PI;
            }
        }
        else if(x < 0)
        {
            return Math.atan(y / x) + Math.PI;
        }
        else
        {
            if (y > 0)
            {
                return Math.PI / 2;
            }
            else if(y < 0)
            {
                return 3 * Math.PI / 2;
            }
            else
            {
                return 0;
            }
        }
    }
}
```

```

/**
 * Berechnet zu einem Paar kartesischer Koordinaten
 * die Radius-Koordinate im entsprechenden Polarkoordinatensystem.
 *
 * @param x x-Koordinate des Paares von kartesischen Koordinaten
 * @param y y-Koordinate des Paares von kartesischen Koordinaten
 * @return der Radius im Polarkoordinatensystem
 */
public static double kartesischZuPolarLaenge(double x, double y)
{
    return Math.sqrt(x*x + y*y);
}

/**
 * Berechnet zu gegebenen Polarkoordinaten die
 * x-Koordinate im entsprechenden kartesischen Koordinatensystem.
 *
 * @param alpha die Winkelkoordinate im Polarkoordinatensystem
 * @param delta der Radius im Polarkoordinatensystem
 * @return die x-Koordinate im entsprechenden kartesischen Koordinatensystem
 */
public static double polarZuKartesischX(double alpha, double delta)
{
    return delta * Math.cos(alpha);
}

/**
 * Berechnet zu gegebenen Polarkoordinaten die
 * y-Koordinate im entsprechenden kartesischen Koordinatensystem.
 *
 * @param alpha die Winkelkoordinate im Polarkoordinatensystem
 * @param delta der Radius im Polarkoordinatensystem
 * @return die y-Koordinate im entsprechenden kartesischen Koordinatensystem
 */
public static double polarZuKartesischY(double alpha, double delta)
{
    return delta * Math.sin(alpha);
}

/**
 * Gibt auf System.out einen String aus,
 * der den angegebenen kartesischen Koordinaten
 * die entsprechenden Polarkoordinaten gegenueberstellt.
 *
 * @param x x-Koordinate eines Paares von kartesischen Koordinaten
 * @param y y-Koordinate eines Paares von kartesischen Koordinaten
 */
public static void printErgebnisKartesischZuPolar(double x, double y)
{
    System.out.println("Die Polarkoordinaten fuer die kartesischen" +
        " Koordinaten\n x="+x+" und y="+y+" sind" +
        " alpha="+kartesischZuPolarWinkel(x, y)+
        " und delta="+kartesischZuPolarLaenge(x, y)+".");
}

```

```

/**
 * Gibt auf System.out einen String aus,
 * der den angegebenen Polarkoordinaten
 * die entsprechenden kartesischen Koordinaten gegenueberstellt.
 *
 * @param alpha die Winkelkoordinate im Polarkoordinatensystem
 * @param delta der Radius im Polarkoordinatensystem
 */
public static void printErgebnisPolarZuKartesisch(double alpha, double delta)
{
    System.out.println("Die kartesischen Koordinaten fuer die" +
        " Polarkoordinaten\n" +
        " alpha="+alpha+" und delta="+delta+" sind"+
        " x="+polarZuKartesischX(alpha, delta)+
        " und y="+polarZuKartesischY(alpha, delta)+".");
}

/**
 * Demonstriert die Methoden
 * {@link #printErgebnisKartesischZuPolar(double, double)}
 * printErgebnisKartesischZuPolar(double, double)}
 * und
 * {@link #printErgebnisPolarZuKartesisch(double, double)}
 * printErgebnisPolarZuKartesisch(double, double)}
 * mit jeweils vier Werten.
 *
 * @param args wird nicht verwendet
 */
public static void main(String[] args)
{
    printErgebnisKartesischZuPolar(3, 4);
    printErgebnisKartesischZuPolar(0, 4);
    printErgebnisKartesischZuPolar(0, -4);
    printErgebnisKartesischZuPolar(0, 0);

    printErgebnisPolarZuKartesisch(0.9272952180016122, 5.0);
    printErgebnisPolarZuKartesisch(1.5707963267948966, 4.0);
    printErgebnisPolarZuKartesisch(4.71238898038469, 4.0);
    printErgebnisPolarZuKartesisch(0.0, 0.0);
}
}

```