File - D:\Projects\Labo_08_Jeu_d_echecs\src\engine\utils\Coordinates.java

```java
package engine.utils;

public class Coordinates {
    int x;
    int y;

    /**
     * Constructor to initialize a Coordinates object with given x and y coordinates.
     *
     * @param x the x-coordinate
     * @param y the y-coordinate
     */
    public Coordinates(int x, int y) {
        this.x = x;
        this.y = y;
    }

    /**
     * Copy constructor to create a new Coordinates object with the same coordinates as the provided Coordinates object.
     *
     * @param coordinates the Coordinates object to copy from
     */
    public Coordinates(Coordinates coordinates) {
        copy(coordinates);
    }

    /**
     * Copy the coordinates from another Coordinates object to this object.
     *
     * @param coordinates the Coordinates object to copy from
     */
    public void copy(Coordinates coordinates) {
        this.x = coordinates.x;
        this.y = coordinates.y;
```

```
35      }
36
37      /**
38       * Set new coordinates for this object.
39       *
40       * @param x the new x-coordinate
41       * @param y the new y-coordinate
42       */
43      public void setCoordinates(int x, int y) {
44          this.x = x;
45          this.y = y;
46      }
47
48      /**
49       * Add the coordinates of another Coordinates object to this object.
50       *
51       * @param coordinates the Coordinates object to add
52       */
53      public void add(Coordinates coordinates) {
54          Coordinates c = addition(this, coordinates);
55          setCoordinates(c.getX(), c.getY());
56      }
57
58      /**
59       * Set a new value for the x-coordinate.
60       *
61       * @param x the new x-coordinate
62       */
63      public void setX(int x) {
64          this.x = x;
65      }
66
67      /**
68       * Get the x-coordinate of this object.
69       *
```

```
 70      * @return the x-coordinate
 71      */
 72     public int getX() {
 73         return this.x;
 74     }
 75
 76     /**
 77      * Set a new value for the y-coordinate.
 78      *
 79      * @param y the new y-coordinate
 80      */
 81     public void setY(int y) {
 82         this.y = y;
 83     }
 84
 85     /**
 86      * Get the y-coordinate of this object.
 87      *
 88      * @return the y-coordinate
 89      */
 90     public int getY() {
 91         return this.y;
 92     }
 93
 94     /**
 95      * Calculate the change in Y-coordinate between two Coordinates objects.
 96      *
 97      * @param positionInitial the initial position
 98      * @param positionFinal   the final position
 99      * @return the change in Y-coordinate
100      */
101     public static int deltaY(Coordinates positionInitial, Coordinates positionFinal) {
102         return positionFinal.y - positionInitial.y;
103     }
104
```

```java
    /**
     * Calculate the change in X-coordinate between two Coordinates objects.
     *
     * @param positionInitial the initial position
     * @param positionFinal   the final position
     * @return the change in X-coordinate
     */
    public static int deltaX(Coordinates positionInitial, Coordinates positionFinal) {
        return positionFinal.x - positionInitial.x;
    }

    /**
     * Calculate the angle in radians between two Coordinates objects.
     *
     * @param positionInitial the initial position
     * @param positionFinal   the final position
     * @return the angle in radians
     */
    public static double getAngle(Coordinates positionInitial, Coordinates positionFinal) {
        int deltaX = Coordinates.deltaX(positionInitial, positionFinal);
        int deltaY = Coordinates.deltaY(positionInitial, positionFinal);
        return Math.atan2(deltaY, deltaX);
    }

    /**
     * Calculate the angle in degrees between two Coordinates objects.
     *
     * @param positionInitial the initial position
     * @param positionFinal   the final position
     * @return the angle in degrees
     */
    public static double getAngleDegree(Coordinates positionInitial, Coordinates positionFinal) {
        return Math.toDegrees(getAngle(positionInitial, positionFinal));
    }

```

```
140     /**
141      * Perform vector addition of two Coordinates objects and return a new Coordinates object.
142      *
143      * @param c1 the first Coordinates object
144      * @param c2 the second Coordinates object
145      * @return a new Coordinates object representing the result of the addition
146      */
147     public static Coordinates addition(Coordinates c1, Coordinates c2) {
148         return new Coordinates(c1.getX() + c2.getX(), c1.getY() + c2.getY());
149     }
150 
151     /**
152      * Check if two Coordinates objects are equal in terms of their x and y coordinates.
153      *
154      * @param c1 the first Coordinates object
155      * @param c2 the second Coordinates object
156      * @return true if the coordinates are equal, false otherwise
157      */
158     public static boolean equal(Coordinates c1, Coordinates c2) {
159         return (c1.getX() == c2.getX()) && (c1.getY() == c2.getY());
160     }
161 }
162 
```