File - D:\Projects\Labo_08_Jeu_d_echecs\src\engine\movements\Step.java

```java
package engine.movements;

import engine.utils.Coordinates;

public class Step {
    private Coordinates coordinates;
    private Coordinates[] movement;
    private double stepAngleDegree;

    public Step(Coordinates step, int maxStep, boolean invertOrdinateAxis,
                boolean invertAbscissaAxis) {
        if (step == null)
            throw new IllegalArgumentException("step cannot be null");
        if (maxStep < 0)
            throw new IllegalArgumentException("maxStep cannot be negative");
        int x = invertAbscissaAxis ? (step.getX() * (-1)) : step.getX();
        int y = invertOrdinateAxis ? (step.getY() * (-1)) : step.getY();
        this.coordinates = new Coordinates(x, y);

        Coordinates originCoordinates = new Coordinates(0, 0);
        this.stepAngleDegree = Coordinates.getAngleDegree(originCoordinates,
                this.coordinates);

        this.movement = new Coordinates[maxStep];
        for (int i = 0; i < this.movement.length; ++i)
            this.movement[i] = nextStep(i);
    }

    /**
     * Get the next step for the number of step given
     *
     * @param nbStep the number of step
     * @return the next step
     */
    private Coordinates nextStep(int nbStep) {
```

```
36          int x = this.coordinates.getX();
37          int y = this.coordinates.getY();
38          x = x == 0 ? 0 : (x < 0 ? (x - nbStep) : (x + nbStep));
39          y = y == 0 ? 0 : (y < 0 ? (y - nbStep) : (y + nbStep));
40          return new Coordinates(x, y);
41      }
42
43      /**
44       * Check if the angle of the step is the same as the angle of the movement
45       *
46       * @param stepAngleDegree the angle of the step
47       * @return true if the angle of the step is the same as the angle of the
48       *         movement
49       */
50      public boolean stepAngleDegreeIsOk(double stepAngleDegree) {
51          return this.stepAngleDegree == stepAngleDegree;
52      }
53
54      /**
55       * Get the coordinates of the movement
56       *
57       * @return the coordinates
58       */
59      public Coordinates[] getMovement() {
60          return this.movement;
61      }
62  }
```