

```
1 package lineacalcoli;
2
3 /**
4  *
5  * @author radaelli11353
6  */
7 public class Line {
8     private double a;
9     private double b;
10    private double c;
11
12    /**
13     * Costruttore per linea con punto e pendenza
14     * @param x1 X del punto fornito
15     * @param y1 Y del punto fornito
16     * @param m Pendenza
17     */
18    public Line(double x1, double y1, double m) {
19        a = -m;
20        b = 1;
21        c = y1 - m * x1;
22    }
23
24    /**
25     * Costruttore per linea passante per due punti
26     * @param x1 X del primo punto
27     * @param y1 Y del primo punto
28     * @param x2 X del secondo punto
29     * @param y2 Y del secondo punto
30     */
31    public Line(double x1, double y1, double x2, double y2) {
32        a = (y2 - y1)/(x1 - x2);
33        b = 1;
34        c = (x2 * y1 - x1 * y2) / (x1 - x2);
35    }
36
37    /**
38     * Costruttore per linea con pendenza e intersezione con y
39     * @param m Pendenza
40     * @param q Intersezione con l'asse delle y
41     */
42    public Line(double m, double q) {
43        a = -m;
44        b = 1;
45        c = -q;
46    }
47
48    /**
49     * Costruttore per retta verticale
50     * @param k Costante che indica l'intersezione con x
51     */
52    public Line(double k) {
53        a = 1;
54        b = 0;
55        c = -k;
56    }
57
58    @Override
59    public String toString() {
60        return a + "," + b + "," + c;
61    }
62}
```

```
62
63  /**
64   * Metodo che controlla se c'è intersezione tra due rette (=non parallele)
65   * @param other Altra retta
66   * @return True se c'è intersezione, false se sono parallele
67   */
68  public boolean intersects(Line other) {
69      return !isParallelo(other);
70  }
71
72  /**
73   * Metodo che confronta l'uguaglianza tra due rette
74   * @param other Altra retta
75   * @return True se le rette sono uguali (sovrapposte)
76   */
77  public boolean equals(Line other) {
78      return a == other.a && b == other.b && c == other.c;
79  }
80
81  /**
82   * Metodo che controlla se le rette sono parallele
83   * @param other Altra retta
84   * @return True se sono parallele, false se c'è intersezione
85   */
86  public boolean isParallelo(Line other) {
87      double thisM = -a / b;
88      double otherM = -other.a / other.b;
89
90      return thisM == otherM;
91  }
92 }
93
```