

```

1 package Application;
2
3 public class Cromossomo implements Comparable<Cromossomo> {
4
5     //Palavra a ser transformada durante as gerações
6     private String axiom = "F[-F]F[+F]F";
7     private double eval = 0;
8     private int numberOfLeaves = 0;
9     private int sizeOfIndividual = 0;
10    private int height = 0;
11    public double p1, p2, p3;
12
13    public void applyMutation() {
14        int pos, posf, auxPos;
15
16        // Seleciona um bloco para substituir
17        pos = (int) Math.round(Math.random() * (axiom.length() - 1));
18        posf = calcPosf(pos, axiom);
19
20        String[] prodRuleMut = {"FF", "[+F]", "[-F]", "F+F", "F-F", "F"};
21        int block = (int) Math.round(Math.random() * 5);
22
23        if (posf < pos) {
24            auxPos = pos;
25            pos = posf;
26            posf = auxPos;
27        }
28
29        axiom = axiom.substring(0, pos)
30            + prodRuleMut[block]
31            + axiom.substring(posf + 1);
32    }
33
34    private int calcPosf(int pos, String axiom) {
35        int posf = pos, ck = 0;
36
37        if (axiom.charAt(pos) == '[') {
38            posf++;
39            while (axiom.charAt(posf) != ']' || ck != 0) {
40                if (axiom.charAt(posf) == '[') {
41                    ck++;
42                } else if (axiom.charAt(posf) == ']') {
43                    ck--;
44                }
45                posf++;
46            }
47        } else if (axiom.charAt(pos) == ']') {
48            posf--;
49            while (axiom.charAt(posf) != '[' || ck != 0) {
50                if (axiom.charAt(posf) == ']') {
51                    ck++;
52                } else if (axiom.charAt(posf) == '[') {
53                    ck--;
54                }
55            }

```