

```
1  package juego;
2
3  import java.awt.Color;
4  import entorno.Entorno;
5  import entorno.Herramientas;
6
7  public class Viga {
8
9      private int pos;
10     private double x;
11     private double y;
12     private double largo;
13     private double alto;
14
15     /*
16      * Este constructor, ya tiene definida de forma estricta y estática las
17      * posiciones de las vigas
18      */
19
20     public Viga(int pos) {
21
22         switch (pos) {
23             case 1:
24                 this.x = 400;
25                 this.y = 575;
26                 this.largo = 820;
27                 this.alto = 25;
28                 break;
29             case 2:
30                 this.x = 325;
31                 this.y = 475;
32                 this.largo = 700;
33                 this.alto = 25;
34                 break;
35             case 3:
36                 this.x = 475;
37                 this.y = 375;
38                 this.largo = 700;
39                 this.alto = 25;
40                 break;
41             case 4:
42                 this.x = 325;
43                 this.y = 275;
44                 this.largo = 700;
45                 this.alto = 25;
46                 break;
47             case 5:
48                 this.x = 475;
49                 this.y = 175;
50                 this.largo = 700;
51                 this.alto = 25;
52                 break;
53             case 6:
54                 this.x = 325;
55                 this.y = 75;
56                 this.largo = 700;
57                 this.alto = 25;
58                 break;
59         }
60
61         this.pos = pos;
62
63     }
```

```
64
65  /*
66  * Dibujar
67  *
68  * Esta función debe ser llamada en cada tick por cada viga que exista. Por
69  * cada
70  * tick dibuja la viga.
71  *
72  * La construcción gráfica de la viga tiene un condimento especial. Para dar
73  * la
74  * sensación de que es una estructura metálica, se dibuja un rectángulo de
75  * color
76  * rojo de fondo, y sobre el de forma estratégica, triángulos del mismo color
77  * que el fondo, en juegos de a dos. Cada uno invertido 90° con respecto al
78  * anterior.
79  */
80  public void dibujar(Entorno entorno) {
81
82      // Rectángulo básico de la viga, respetando los valores indicados por el
83      // constructor
84      entorno.dibujarRectangulo(this.x, this.y, this.largo, this.alto, 0.0,
85      Color.RED);
86
87      // El extremo izquierdo de la viga corrido 10px
88      double paso = this.x - (this.largo / 2) + 10;
89
90      // Se decide que la suma de la base de un triangulo, la punta del
91      // triángulo
92      // adyacente y un espacio
93      // extra sea la 25ava parte del ancho de la viga - 4 pixeles
94      double triangulos = (this.largo / 25) - 4;
95
96      // Indica la cantidad de parejas de triangulos dibujados. Una pareja es un
97      // triangulo con la punta hacia arriba
98      // y el otro con la punta hacia abajo.
99      int dibujados = 0;
100
101      // Este bucle dibuja la pareja de triángulos a lo largo de la viga.
102      while (dibujados <= triangulos) {
103
104          entorno.dibujarTriangulo(paso, this.y, 21, 21, Herramientas.radians(
105          90), java.awt.Color.BLACK);
106          paso += 14;
107          entorno.dibujarTriangulo(paso, this.y, 21, 21, Herramientas.radians(
108          270), java.awt.Color.BLACK);
109          paso += 14;
110          dibujados += 1;
111      }
112  }
113
114  // Devuelve la posX
115  public int getPosx() {
116      return (int) this.x;
117  }
118
119  // Devuelve la posY
120  public int getPosy() {
121      return (int) this.y;
122  }
123  }
```

```
120 // Devuelve el Ancho
121 public int getAncho() {
122     return (int) this.largo;
123 }
124
125 public int getPos() {
126     return pos;
127 }
128
129 /*
130  * DondeEmpiezaElSuelo
131  *
132  * Esta función devuelve el valor en el 'y' en el cual comienza la viga.
133  * Sabiendo que el 'y' se encuentra en el centro.
134  *
135  */
136 public double dondeEmpiezaElSuelo() {
137
138     return this.y - (this.alto / 2) - 1;
139 }
140
141 public double dondeTerminaElTecho() {
142
143     return this.y + (this.alto / 2) + 1;
144 }
145
146 /*
147  * Esta función indica donde comienza la viga en el eje X.
148  */
149 public int extremoIzquierdo() {
150     return this.getPosx() - this.getAncho() / 2;
151 }
152
153 /*
154  * Esta función indica donde termina la viga en el eje X.
155  */
156 public int extremoDerecho() {
157     return this.getPosx() + this.getAncho() / 2;
158 }
159
160 }
161
```