

ClaseNatacion.java

```

1 /* David López cu:173993
2  * 15/04/2018
3  * Ejercicio 5
4  */
5 public class ClaseNatacion {
6     private String nombre;
7     private String domicilio;
8     private int numAlbercas;
9     private double tamañoAlbercas;
10    private Rectangulo [] arrAlbercas;
11    private final int MAX = 100;
12
13    public ClaseNatacion() {
14        this.numAlbercas = 0;
15        this.arrAlbercas = new Rectangulo[MAX];
16    }
17
18    public ClaseNatacion(String nombre, String domicilio) {
19        this();
20        this.nombre = nombre;
21        this.domicilio = domicilio;
22    }
23
24    public String getNombre() {
25        return nombre;
26    }
27
28    public void setNombre(String nombre) {
29        this.nombre = nombre;
30    }
31
32    public int getNumAlbercas() {
33        return numAlbercas;
34    }
35
36    public void setNumAlbercas(int numAlbercas) {
37        this.numAlbercas = numAlbercas;
38    }
39
40    public double getTamañoAlbercas() {
41        return tamañoAlbercas;
42    }
43
44    public String getDomicilio() {
45        return domicilio;
46    }
47
48    public String toString(){
49        StringBuilder cad = new StringBuilder();
50
51        cad.append("Nombre: " + nombre);
52        cad.append("\nDomicilio: " + domicilio);
53        cad.append("\nNúmero de albercas: " + numAlbercas);
54        cad.append("\nTamaño de las albercas: " + tamañoAlbercas);
55        for (int i=0;i<=numAlbercas;i++){
56            cad.append("\nAlberca: " + i+1 + arrAlbercas[i]);
57        }
58        return cad.toString();
59    }
60
61    public int compareTo(ClaseNatacion otro){
62        return this.nombre.compareTo(otro.nombre);
63    }
64

```

ClaseNatacion.java

```

65     public boolean equals(Object obj) {
66         if (this == obj)
67             return true;
68         if (obj == null)
69             return false;
70         if (getClass() != obj.getClass())
71             return false;
72         ClaseNatacion other = (ClaseNatacion) obj;
73         if (nombre == null) {
74             if (other.nombre != null)
75                 return false;
76         } else if (!nombre.equals(other.nombre))
77             return false;
78         return true;
79     }
80
81     public boolean nuevaAlberca(double altura, double base) {
82         boolean resp = false;
83         Rectangulo alberca = new Rectangulo(altura, base);
84
85         if (this.numAlbercas < MAX) {
86             arrAlbercas[this.numAlbercas] = alberca;
87             numAlbercas++;
88         }
89         resp = true;
90         return resp;
91     }
92
93     public double calculaTamañoAlbercas() {
94         double resp = 0;
95
96         for (int i = 0; i < numAlbercas; i++) {
97             resp += arrAlbercas[i].calculaArea();
98             this.tamañoAlbercas = resp;
99         }
100        return resp;
101    }
102
103    public double calculaCostoTecho() {
104        double resp;
105
106        resp = calculaTamañoAlbercas() * 1000;
107        return resp;
108    }
109 }
110 }

```