

Folder src\JAAS270224

5 printable files

(file list disabled)

src\JAAS270224\Cyborg.java

```
1 package JAAS270224;
2
3 import java.util.concurrent.ThreadLocalRandom;
4 //Cyborg. Clase que hereda de SerVivo, no tendrá nuevos atributos y de ella no se podrá heredar.
5 public abstract class Cyborg extends SerVivo implements Guerrero{
6     //tendrá un constructor sin parámetros que establecerá el identificador del Cyborg de forma consecutiva y automática, empezando por
    el 1.
7     public Cyborg(int identificador) {
8         super(identificador);
9         for (int i = 1; i < 10; i++) {
10             identificador++;
11         }
12     }
13
14     @Override
15     public int dispara(int maximo) {
16         int daño = ThreadLocalRandom.current().nextInt(0, maximo);
17         System.out.println("Soy " + getClass() + identificador + " disparo a " + daño);
18         return daño;
19     }
20     @Override
21     public void alcanzado() {
22         System.out.println("Soy ciborg " + identificador + " estoy " + !vivo);
23     }
24     @Override
25     public boolean estoyVivo(){
26         return vivo;
27     }
28     @Override
29     public int identificador() {
```

```
30 |         // TODO Auto-generated method stub
31 |         throw new UnsupportedOperationException("Unimplemented method 'identificador'");
32 |     }
33 | }
34 |
```

src\JAAS270224\Guerrero.java

```
1 | package JAAS270224;
2 | //Guerrero. Interfaz con los siguientes métodos:
3 |
4 | public interface Guerrero {
5 |     // int dispara(int maximo). Lo implementarán Terricolas y cyborg. Generará un número aleatorio entre 0 y máximo, con máximo no
    incluido, y que interpretamos que estamos disparando a ese número, y es el valor que devolvemos. Mostrará un mensaje indicando quien es
    y a quien dispara.
6 |     // alcanzado(). Lo implementarán Terricolas y cyborg y mostrará un mensaje diciendo quien es, representa que lo han disparado.
    Mostrará un mensaje indicando que está muerto.
7 |     // boolean estoyVivo()
8 |     public int dispara(int maximo);
9 |     public void alcanzado();
10 |    public boolean estoyVivo();
11 |    public int identificador();
12 |
13 |
14 | }
15 |
```

src\JAAS270224\Nave.java

```
1 | package JAAS270224;
2 |
3 | public class Nave {
4 |     // Nave. Clase que va a contener un array de 10 Guerreros,
5 |     // ese atributo
6 |     // se llamará tripulacion,
7 |     // tendrá además
8 |     // un atributo
9 |     // para guardar
10 |    // el nombre
```

```
11 // de la
12 // nave que
13 // comenzará por"Nave Estelar"
14 // si es
15 // terricola o"Cubo Borg"
16 // si es
17 // cyborg.Los atributos
18 // no son
19 // directamente accesibles.
20 // Sus métodos serán:
21 Guerrero[] tripulacion = new Guerrero[10];
22 String terricola = "Nave Estelar ";
23 String cyborg = "Cubo Borg ";
24 // Constructor, se le pasa un parámetro de tipo cadena de caracteres que indicará quienes son su nombre. Rellenará la tripulación
    creado 10 objetos con los seres adecuados, a los terricolos les puedes poner el nombre "Terricola 1" y sucesivos. Mostrará un mensaje
    diciendo que se ha creado la nave y y el tipo de nave que se ha creado. Si el nombre no es correcto no rellenará la tripulación y
    mostrará un mensaje de error.
25 // recibeDisparo(int). La nave recibe un disparo, matará al tripulante que se encuentre en la posición que se le pasa como parámetro,
    eso si sigue vivo y si la posición es correcta.
26 // int generaDisparo(int). El método devuelve el disparo generado por el tripulante que se le pasa por parámetro (si está vivo).
27 // boolean cuantosQuedan(). Devuelve cuantos tripulantes vivos quedan en la nave
28
29 public Nave(String nombre) {
30     if (nombre.equals(terricola)) {
31         for (int i = 0; i < tripulacion.length; i++) {
32             tripulacion[i] = new Terricola(i, nombre);
33             System.out.println("Bienvenido terricola" + i);
34         }
35         if (nombre.equals(cyborg)) {
36             for (int i = 0; i < tripulacion.length; i++) {
37                 tripulacion[i] = new Cyborg(i) {
38                 };
39                 System.out.println("Bienvenido cyborg" + i);
40             }
41         }
42     } else {
43         System.out.println("Nombre no aceptado");
44     }
45 }
46
47
48 void recibeDisparo(int identificador) {
```

```
49     for (Guerrero guerrero : tripulacion) {
50         if (generaDisparo(identificador) == guerrero.identificador()) {
51             guerrero.alcanzado();
52         }
53     }
54 }
55
56 int generaDisparo(int identificador) {
57     int dispara = 0;
58     for (Guerrero guerrero : tripulacion) {
59         dispara = guerrero.dispara(identificador);
60     }
61     return dispara;
62 }
63
64 int cuantosQuedan() {
65     int contador = 0;
66     for (Guerrero guerrero : tripulacion) {
67         if (guerrero.estoyVivo() == true)
68             ;
69         contador++;
70     }
71     return contador;
72 }
73 }
74 }
```

src\JAAS270224\SerVivo.java

```
1 package JAAS270224;
2
3 // SerVivo. Clase abstracta con un atributo, vivo, que va a indicar si está vivo o no,
4 // y un atributo identificador de tipo entero. Tiene un método, estoyVivo(),
5 // que devuelve el valor del atributo vivo, y otro para devolver el valor del identificador.
6 // El atributo identificador no se podrá modificar y ambos serán directamente accesibles sólo
7 // desde las clases que hereden de ella. Tendrá un constructor al que se le pasa el valor del
8 // atributo identificador.
9
10 public abstract class SerVivo {
11     protected boolean vivo;
```

```
12     protected final int identificador;
13
14     public SerVivo(int identificador) {
15         this.identificador = identificador;
16     }
17
18     boolean estoyVivo() {
19         return vivo;
20     }
21
22     int saberIdentificador() {
23         return identificador;
24     }
25 }
26
```

src\JAAS270224\Terricola.java

```
1 package JAAS270224;
2
3 import java.util.concurrent.ThreadLocalRandom;
4
5 // Terricola. Clase que hereda de SerVivo. Tendrá un nuevo atributo nombre, una cadena de caracteres que no se podrá modificar y
6 encapsulada, Los métodos serán:
7
8 public class Terricola extends SerVivo implements Guerrero {
9     private final String nombre;
10    // constructor al que se le pasa una cadena de caracteres con el nombre. Automáticamente se contarán el número de objetos Terricola
11    que se han creado para poner un valor de identificador único y consecutivo, empezando por el 1.
12    public Terricola(int identificador, String nombre) {
13        super(identificador);
14        for (int i = 1; i < 10; i++) {
15            identificador++;
16        }
17        this.nombre = nombre;
18    }
19    @Override
20    public int dispara(int maximo) {
21        System.out.println("Soy " + nombre + identificador + " disparo a " + nombre + identificador);
22        int daño = ThreadLocalRandom.current().nextInt(0, maximo);
23    }
24 }
```

```
21         return daño;
22     }
23     @Override
24     public void alcanzado() {
25         System.out.println("Soy " + nombre + identificador+" estoy "+ !vivo);
26     }
27     @Override
28     public boolean estoyVivo(){
29         return vivo;
30     }
31
32     @Override
33     public int identificador() {
34         // TODO Auto-generated method stub
35         throw new UnsupportedOperationException("Unimplemented method 'identificador'");
36     }
37
38
39
40 }
41
```