

ACTIVIDADES DE LA UNIDAD DE TRABAJO 8 – PRO

Autor: Derimán Tejera Fumero. 1ºDAW

Fecha: 08/02/2022

Actividades de comprobación

8.1. Sobre una subclase es correcto afirmar que:

- a) Tiene menos atributos que su superclase.
- b) Tiene menos miembros que su superclase.
- c) Hereda los miembros no privados de su superclase.**
- d) Hereda todos los miembros de su superclase.

8.2. En relación con las clases abstractas es correcto señalar que:

- a) Implementan todos sus métodos.
- b) No implementan ningún método.
- c) No tienen atributos.
- d) Tienen algún método abstracto.**

8.3. ¿En qué consiste la sustitución u overriding?

- a) En sustituir un método heredado por otro implementado en la propia clase.**
- b) En sustituir un atributo por otro del mismo nombre.
- c) En sustituir una clase por una subclase.
- d) En sustituir un valor de una variable por otro.

8.4. Sobre la clase Object es cierto indicar que:

- a) Es abstracta.
- b) Hereda de todas las demás.
- c) Tiene todos sus métodos abstractos.
- d) Es superclase de todas las demás clases.**

8.5. ¿Cuál de las siguientes afirmaciones sobre el método equals () es correcta?

- a) Hay que implementarlo, ya que es abstracto.
- b) Sirve para comparar solo objetos de la clase Object.
- c) Se hereda de Object, pero debemos reimplementarlo al definirlo en una clase.**
- d) No hay que implementarlo, ya que se hereda de Object.

8.6. ¿Cuál de las siguientes afirmaciones sobre el método toString () es correcta?

a) Sirve para mostrar la información que nos interesa de un objeto.

b) Convierte automáticamente un objeto en una cadena.

c) Encadena varios objetos.

d) Es un método abstracto de Object que tenemos que implementar.

8.7. ¿Cuál de las siguientes afirmaciones sobre el método getClass () es correcta?

a) Convierte los objetos en clases.

b) Obtiene la clase a la que pertenece un objeto.

c) Obtiene la superclase de una clase.

d) Obtiene una clase a partir de su nombre.

8.8 Una clase puede heredar

a) De una clase.

b) De dos clases.

c) De todas las clases que queramos.

d) Solo de la clase Object.

8.9. La selección dinámica de métodos:

a) Se produce cuando una variable cambia de valor durante la ejecución de un programa.

b) Es el cambio de tipo de una variable en tiempo de ejecución.

c) Es la asignación de un mismo objeto a más de una variable en tiempo de ejecución.

d) Es la ejecución de distintas implementaciones de un mismo método, asignando objetos de distintas clases a una misma variable, en tiempo de ejecución.

8.10. ¿Cuál de las siguientes afirmaciones sobre el método super() es correcta?

a) Sirve para llamar al constructor de la superclase.

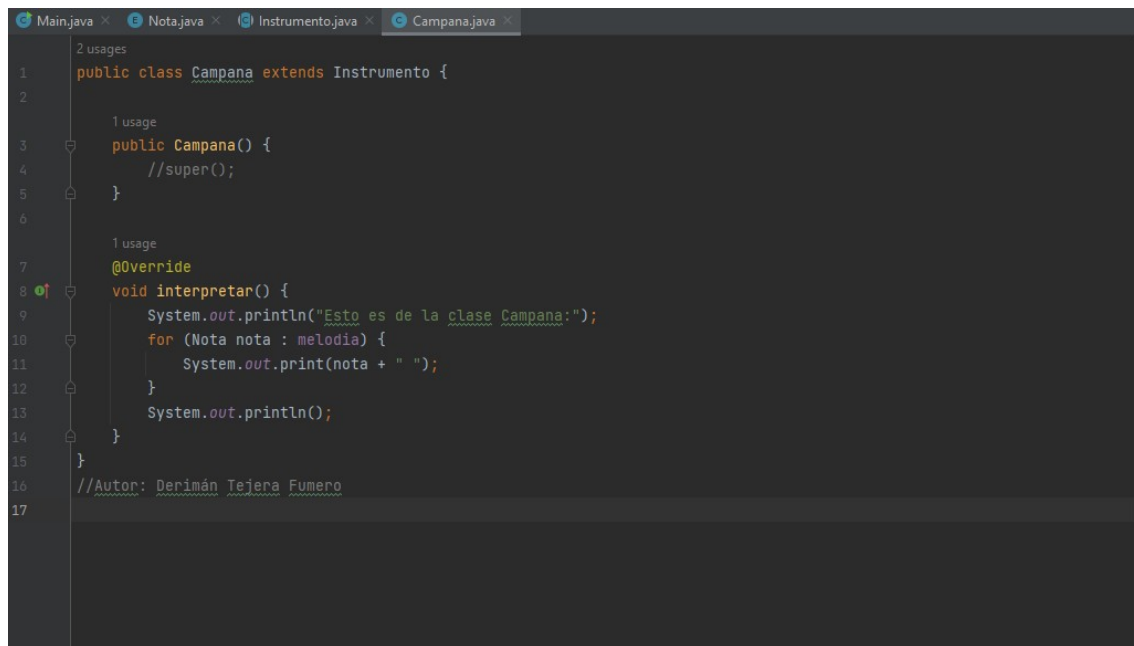
b) Sirve para invocar un método escrito más arriba en el código.

c) Sirve para llamar a cualquier método de la superclase.

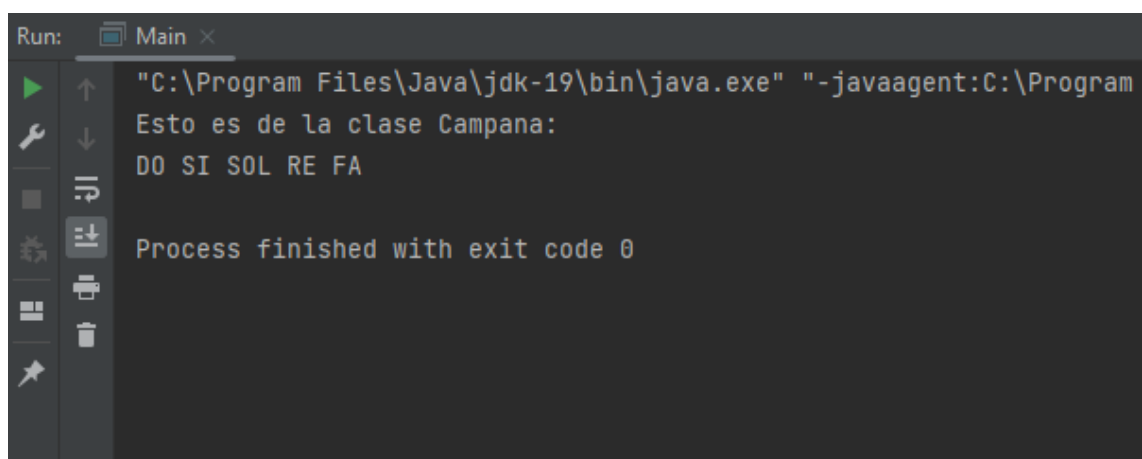
d) Sirve para hacer referencia a un atributo de la superclase.

Actividades de aplicación

8.11. Crea la clase Campana que hereda de Instrumento (definida en la Actividad resuelta 8.4).



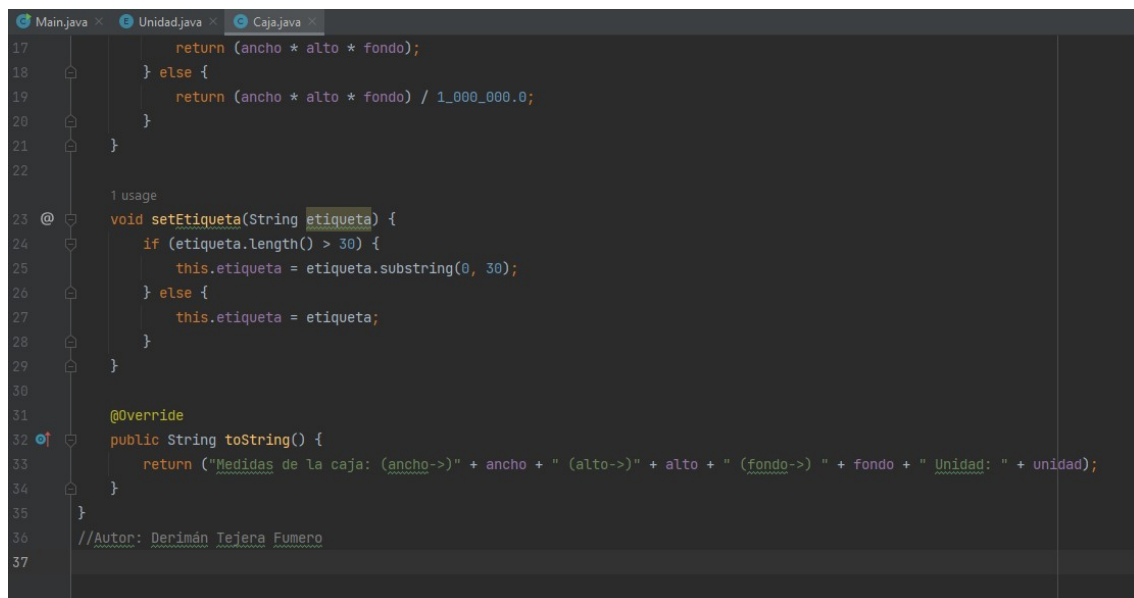
```
1 public class Campana extends Instrumento {
2
3     1 usage
4     public Campana() {
5         //super();
6     }
7
8     1 usage
9     @Override
10    void interpretar() {
11        System.out.println("Esto es de la clase Campana:");
12        for (Nota nota : melodia) {
13            System.out.print(nota + " ");
14        }
15        System.out.println();
16    }
17    //Autor: Derimán Tejera Fumero
```



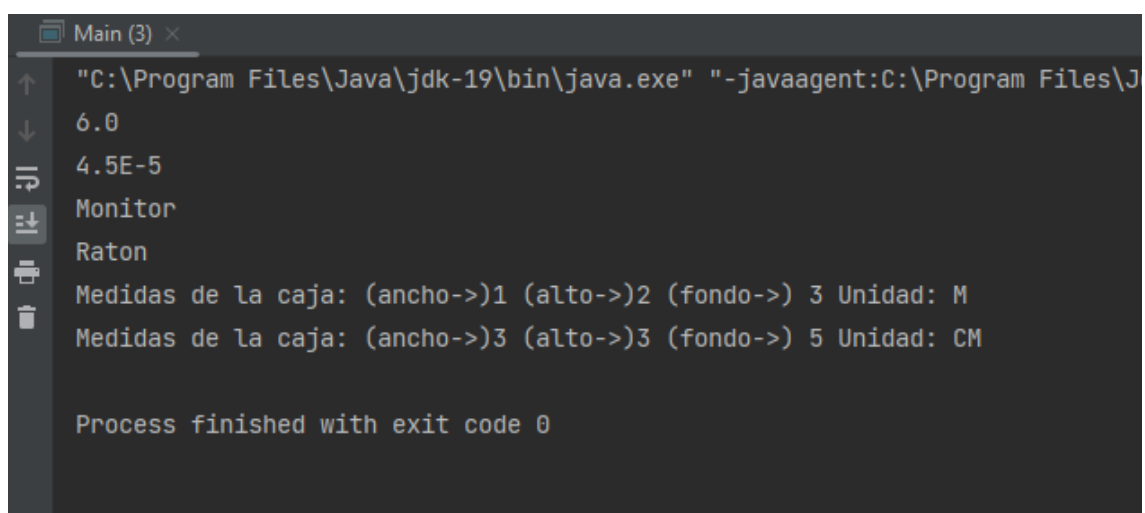
```
Run: Main x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program
Esto es de la clase Campana:
DO SI SOL RE FA
Process finished with exit code 0
```

8.12. Las empresas de transporte, para evitar daños en los paquetes, emban todas sus mercancías en cajas con el tamaño adecuado. Una caja se crea expresamente con un ancho, un alto y un fondo y, una vez creada, se mantiene inmutable. Cada caja lleva pegada una etiqueta, de un máximo de 30 caracteres, con información útil como el nombre del destinatario, dirección, etc. Implemento la clase Caja con los siguientes métodos:

- **Caja (int ancho, int alto, int fondo, Unidad unidad):** que construye una caja con las dimensiones especificadas, que pueden encontrarse en «cm» (centímetros) o «m» (metros).
- **double getVolumen():** que devuelve el volumen de la caja en metros cúbicos.
- **void setEtiqueta (String etiqueta):** que modifica el valor de la etiqueta de la caja.
- **String toString():** que devuelve una cadena con la representación de la caja.



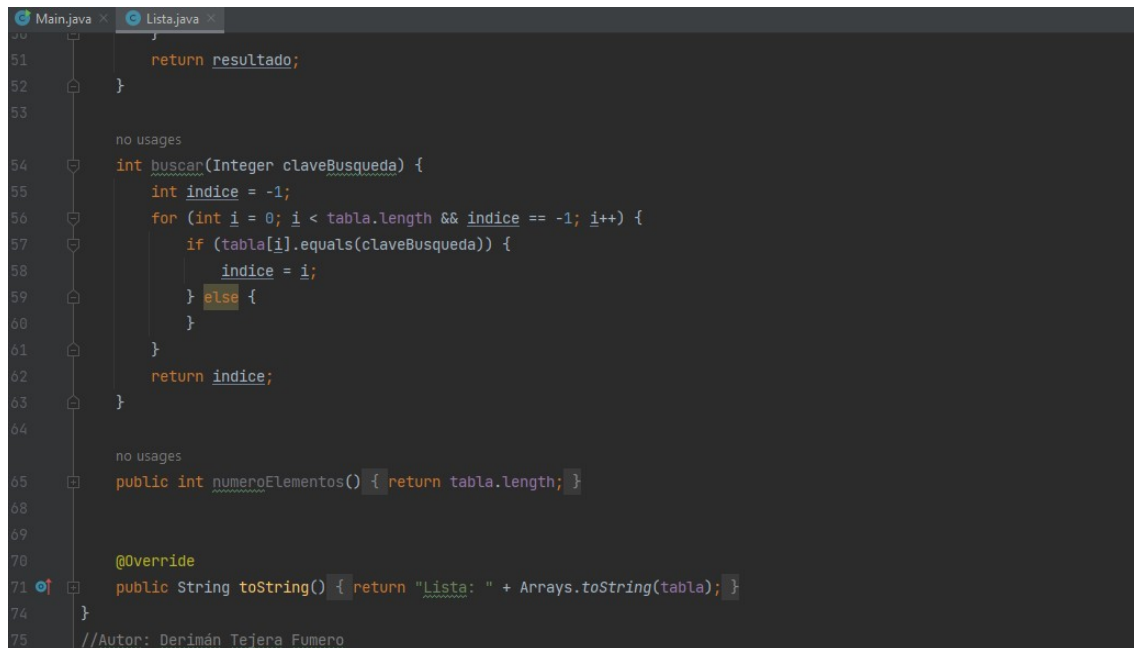
```
17         return (ancho * alto * fondo);
18     } else {
19         return (ancho * alto * fondo) / 1_000_000.0;
20     }
21 }
22
23 1 usage
24 @ void setEtiqueta(String etiqueta) {
25     if (etiqueta.length() > 30) {
26         this.etiqueta = etiqueta.substring(0, 30);
27     } else {
28         this.etiqueta = etiqueta;
29     }
30 }
31
32 @Override
33 public String toString() {
34     return ("Medidas de la caja: (ancho->)" + ancho + " (alto->)" + alto + " (fondo->)" + fondo + " Unidad: " + unidad);
35 }
36 //Autor: Derimán Tejera Fumero
37
```



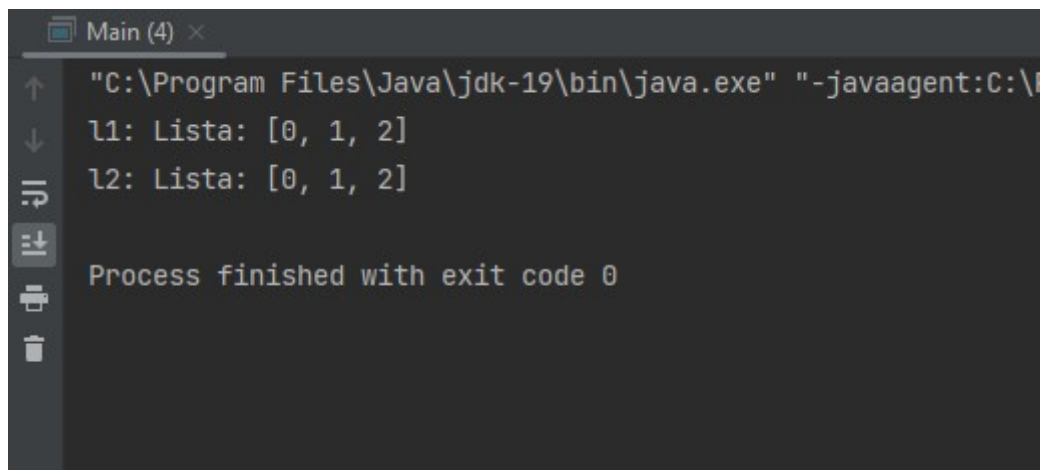
```
Main (3) x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\J
6.0
4.5E-5
Monitor
Raton
Medidas de la caja: (ancho->)1 (alto->)2 (fondo->) 3 Unidad: M
Medidas de la caja: (ancho->)3 (alto->)3 (fondo->) 5 Unidad: CM

Process finished with exit code 0
```

8.14. Reimplementa la clase Lista de la Actividad resuelta 7.11. sustituyendo el método mostrar() por el método toString().



```
51         return resultado;
52     }
53
54     no usages
55     int buscar(Integer claveBusqueda) {
56         int indice = -1;
57         for (int i = 0; i < tabla.length && indice == -1; i++) {
58             if (tabla[i].equals(claveBusqueda)) {
59                 indice = i;
60             } else {
61             }
62         }
63         return indice;
64     }
65
66     no usages
67     public int numeroElementos() { return tabla.length; }
68
69
70     @Override
71     public String toString() { return "Lista: " + Arrays.toString(tabla); }
72 }
73
74 //Autor: Derimán Tejera Fumero
```



```
Main (4) x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\...
l1: Lista: [0, 1, 2]
l2: Lista: [0, 1, 2]
Process finished with exit code 0
```

8.15. Escribe en la clase Lista un método equals () para compararlas. Dos listas se considerarán iguales si tienen los mismos elementos (incluidas las repeticiones) en el mismo orden.

```
Main.java x Lista.java x
70 @Override
71 public String toString() { return "Lista: " + Arrays.toString(tabla); }
74
75 @Override
76 public boolean equals(Object obj) {
77
78     Lista other = (Lista) obj;
79     // this: 1,2
80     // other: 1,2,3
81     if (tabla.length != other.tabla.length) {
82         return false;
83     }
84
85     for (int i = 0; i < tabla.length; i++) {
86         if (this.tabla[i] != other.tabla[i]) {
87             return false;
88         }
89     }
90     return true;
91 }
92
93 }
94 //Autor: Derimán Tejeda Fumero.
```

```
Main (5) x
↑ "C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\
↓ l1: Lista: [0, 1, 2]
↺ l2: Lista: [0, 1, 2]
⇅ true
⇅
🖨 Process finished with exit code 0
🗑
```

8.16 Escribe la clase Pila heredando de Lista (ver Actividad resuelta 7.13).

```
Main.java x Lista.java x Pila.java x
2 usages
1 public class Pila extends Lista {
    1 usage
2 void apilar(Integer nuevo) {
3     insertarPrincipio(nuevo);
4 }
5
2 usages
6 Integer desapilar() {
7     return eliminar(indice: 0);
8 }
9 }
10
11 //Autor: Derimán Tejera Fumero.
12
```

```
Main (1) x
↑ "C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\
↓ Pila@568db2f2
↶ 9
↷ 8
⌵ 7
⌶ 6
⌵ 5
⌶ 4
⌵ 3
⌶ 2
⌵ 1
⌶ 0

Process finished with exit code 0
```

8.17. Escribe la clase Cola heredando de Lista (ver Actividad final 7.18).


```
Main.java x Lista.java x Cola.java x
2 usages
1 public class Cola extends Lista {
    1 usage
2     public void encolar(Integer nuevo) {
3         insertarFinal(nuevo);
4     }
5
    2 usages
6     public Integer desencolar() {
7         return eliminar( indice: 0);
8     }
9 }
10 //Autor: Derimán Tejera Fumero.
```

```
Main (2) x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\
Cola@568db2f2
0
1
2
3
4
5
6
7
8
9

Process finished with exit code 0
```