

## Práctica 2 (15/10/2021) – Herencia y Polimorfismo

Leer detenidamente hasta el final antes de comenzar.

En esta práctica toda la información debe ser mostrada y recogida a través de ventanas (JOptionPane). Se podrá mostrar por consola mensajes del programador que no sean para el usuario.

Haz un programa que presente el siguiente comportamiento:

- Pregunta al usuario qué componente desea elegir (entre los disponibles en un array (Stock), generado “automáticamente” partir del Enum ComponentList – getItems()).
- En función de la opción seleccionada, se creará un objeto de dicha clase y se llamará al doTask de ese componente.

### Pasos a seguir:

- Crear todas las clases, interfaces y enumeraciones del Diagrama de Clases UML adjunto, incluyendo las relaciones de herencia e implementación que correspondan.
- Crear los constructores de las clases
- Incluir los métodos de las interfaces
- Crear el método getItems() : String [] de ComponentList, que devuelve un array de Strings con todos los posibles componentes que se pueden crear. **(Repasar diapositiva 37, tema 2) y se mostrarán las opciones con un JOptionPane de tipo showOptionsDialog<sup>1</sup>.**
- Implementar los métodos abstractos heredados de la superclase o de las interfaces.
- Crear los atributos del main:
  - o selectComp: es un arrayList donde añado el componente creado, no es obligatorio.
  - o Stock : es el String donde guardo los valores retornados al llamar al getItems del ComponentList. Lo necesitaré para el JOptionPane.
- Crear el JOptionPane que muestra las opciones
- Gestionar la opción marcada y llamar al doTask del elemento seleccionado.

### Detalles a tener en cuenta:

- Limpieza de código
- Se deberán incluir comentarios. (@anotaciones y javaDocs no necesarios)

1. Tened en cuenta que `JOptionPane.showOptionDialog`, tiene la siguiente estructura y devuelve un número entero [0-n) con la opción clicada:

```
public static int showOptionDialog(Component parentComponent, Object message, String title, int optionType, int messageType, Icon icon, Object[] options, Object initialValue)
```

Ejemplo de uso: `int x = JOptionPane.showOptionDialog(null, "Message", "WindowTitle", JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null, array_opciones, array_opciones[0]);`

2. Para comprobar si el sensor IR detecta algo o no, se llamará al método `detect()` que retornará un valor random de tipo booleano. Por tanto, `doTask`, mostrará en una ventana si se ha detectado algo o no.
3. Cuando se llame a un actuador, se generará también un valor random previamente y en función del resultado, se llamará a:
  - a. `start()` – si es true (Mostrando por pantalla que el actuador se ha encendido)
  - b. `stop()` – si es false (Mostrando por pantalla que el actuador se ha apagado)

*Intentad descubrir vosotros como se hacen los valores random, de tipo boolean.*

