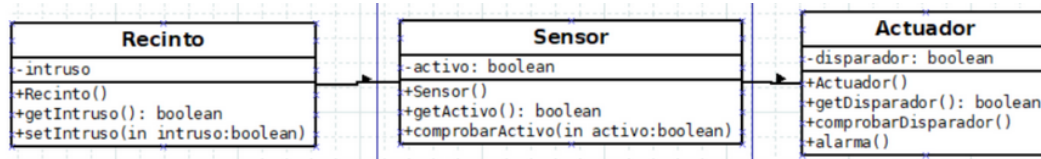
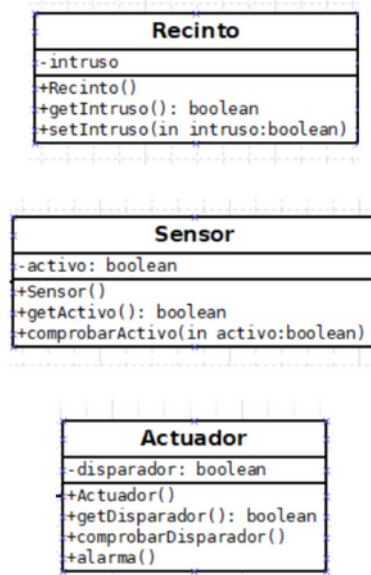


Ejercicio 6 - Asociación

Vamos a implementar la siguiente relación entre clases:



Las clases aparecen detalladas aquí:



Hay que realizar las siguientes tareas:

- Implementar las clases y sus relaciones. Métodos a comentar:
 - `comprobarActivo()`: este método recibe por parámetro el atributo `intruso` de la clase `Recinto`. Si es verdad cambia a verdadero el atributo `activo` de la clase `Sensor`, sino cambia a falso el atributo.
 - `comprobarDisparador()`: dentro de la clase se obtiene el atributo `activo` de la clase `Recinto`. Si es verdad cambia a verdadero el atributo `disparador` de la clase `Actuador`, sino cambia a falso el atributo.

- `alarma()` : muestra en consola el mensaje : `"ALERTA INTRUSO"` si el atributo disparador es verdad.
- Clase `Flow` que contenga el método `main` . Se realiza lo siguiente:
 - Crear los objetos necesarios y la secuencia de acciones correcta para que funcione el sistema de alarma.

```
public class Flow {

    public static void main(String[] args) {

        Recinto casa = new Recinto();
        Sensor sensorPuerta = new Sensor();
        Actuador alarmaSeguridad = new Actuador();

        casa.setIntruso(true);
        sensorPuerta.comprobarActivo(casa.getIntruso());
        alarmaSeguridad.comprobarDisparador(sensorPuerta.getActivo());

        alarmaSeguridad.alarma();

    }

}

class Recinto {

    private boolean intruso;

    public Recinto() {
        this.intruso = false;
    }

    public boolean getIntruso() {
        return this.intruso;
    }

    public void setIntruso(boolean intruso) {
        this.intruso = intruso;
    }

}

class Sensor extends Recinto {

    private boolean activo;
```

```

    public Sensor() {
        super();
        this.activo = false;
    }

    public boolean getActivo() {
        return this.activo;
    }

    public void comprobarActivo(boolean intruso) {
        this.activo = intruso;
    }
}

class Actuador extends Sensor {

    private boolean disparador;

    public Actuador() {
        super();
        this.disparador = false;
    }

    public boolean getDisparador() {
        return this.disparador;
    }

    public void comprobarDisparador(boolean activo) {
        this.disparador = activo;
    }

    public void alarma() {
        System.out.println(getDisparador() ? "ALERTA INTRUSO" : "");
    }
}

```

¿Qué dificultades has encontrado?

No he encontrado ninguna dificultad durante la realización de la tarea.

¿Que conceptos teóricos has manejado en el ejercicio?

He trabajado con el concepto de herencia entre las diferentes clases.

¿Ha sido necesario emplear algún concepto adicional?

He implementado el método `super()` en los constructores de las clases, un concepto que vimos en la práctica anterior. También, he empleado la asignación de valores por defecto para la inicialización de las variables de las clases.