

1 <http://hilite.me/> herramienta para copiar código en html

2 Ejercicios JAVA2

2.11 Ejercicio 11: Crear una clase que sea control remoto que tenga un canal, volumen y si está apagado o encendido.

```
1
2
3
4 public class Control_remoto {
5     private final static int VOL_MAX=5;
6     private final static int NCH_MAX=10;
7     private int canal;
8     private int volumen;
9     private boolean isEncendido;
10
11     Control_remoto() {
12         this.canal=0;
13         this.volumen=0;
14         this.isEncendido=false;
15     }
16
17     Control_remoto(int canal,int volumen,boolean isEncendido){
18         if(canal>NCH_MAX) {canal=NCH_MAX;}
19         if(volumen>VOL_MAX) {volumen=VOL_MAX;}
20         this.canal=canal;
21         this.volumen=volumen;
22         this.isEncendido=isEncendido;
23     }
24
25
```

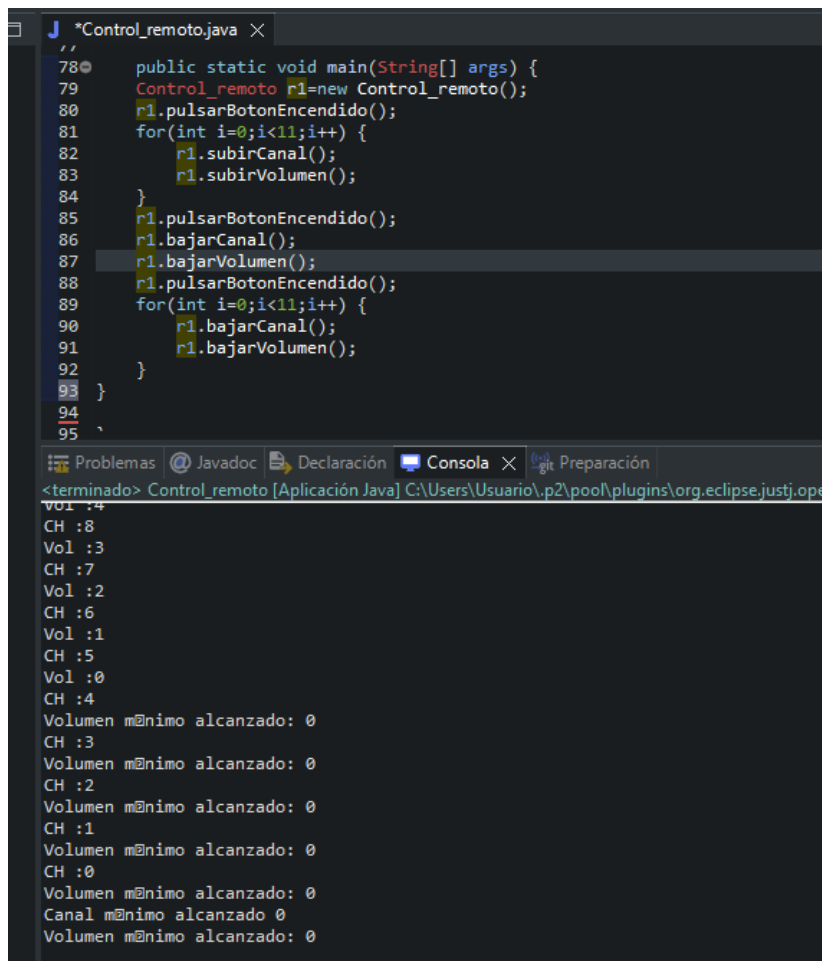
Ilustración 1 clase Control remoto

2.11.1 Crear los métodos de aumentar de canal, disminuir de canal, subir/bajar volumen y encender o apagar.

```
1
2 public void subirVolumen() {
3     if(isEncendido) {
4         if(this.volumen<VOL_MAX) {
5             System.out.println("Vol :"+ ++this.volumen);
6         }
7     }else {
8         System.out.println("Volumen máximo alcanzado: "+this.volumen);
9     }
10 }else {
11     System.out.println("Encienda primero.");
12 }
13 }
14
15 public void bajarVolumen() {
16     if(isEncendido) {
17         if(this.volumen>0) {
18             System.out.println("Vol :"+ --this.volumen);
19         }else {
20             System.out.println("Volumen mínimo alcanzado: "+ this.volumen);
21         }
22     }else {
23         System.out.println("Encienda primero.");
24     }
25 }
26
27 public void subirCanal() {
28     if(isEncendido) {
29         if(this.canal<NCH_MAX) {
30             System.out.println("CH :"+ ++this.canal);
31         }else {
32             System.out.println("Canal máximo alcanzado :"+this.canal);
33         }
34     }else {
35         System.out.println("Encienda primero.");
36     }
37 }
38
39 public void bajarCanal() {
40     if(isEncendido) {
41         if(this.canal>0) {
42             System.out.println("CH :"+ --this.canal);
43         }else {
44             System.out.println("Canal mínimo alcanzado "+this.canal);
45         }
46     }else {
47         System.out.println("Encienda primero.");
48     }
49 }
50 }
51 public void pulsarBotonEncendido() {
52     String txt= (isEncendido?"Apagando...":"Encendiendo..");
53     isEncendido=!isEncendido;
54     System.out.println(txt);
55 }
```

Ilustración 2 metodos mando remoto

2.11.2 Hacer distintas pruebas de subir/bajar volumen, canal o encender/apagar.



The screenshot shows the Eclipse IDE with a Java file named `*Control_remoto.java` open. The code is as follows:

```
//
78 public static void main(String[] args) {
79     Control_remoto r1=new Control_remoto();
80     r1.pulsarBotonEncendido();
81     for(int i=0;i<11;i++) {
82         r1.subirCanal();
83         r1.subirVolumen();
84     }
85     r1.pulsarBotonEncendido();
86     r1.bajarCanal();
87     r1.bajarVolumen();
88     r1.pulsarBotonEncendido();
89     for(int i=0;i<11;i++) {
90         r1.bajarCanal();
91         r1.bajarVolumen();
92     }
93 }
94
95 `
```

The console output shows the execution of the program, displaying channel and volume levels and messages when the minimum volume is reached:

```
<terminado> Control_remoto [Aplicación Java] C:\Users\Usuario\AppData\Local\Temp\org.eclipse.justj.openjdk.hotspot.jre.full\jre-17\bin\java.exe
Vol :4
CH :8
Vol :3
CH :7
Vol :2
CH :6
Vol :1
CH :5
Vol :0
CH :4
Volumen mínimo alcanzado: 0
CH :3
Volumen mínimo alcanzado: 0
CH :2
Volumen mínimo alcanzado: 0
CH :1
Volumen mínimo alcanzado: 0
CH :0
Volumen mínimo alcanzado: 0
Canal mínimo alcanzado 0
Volumen mínimo alcanzado: 0
```

Ilustración 3 pruebas Mando remoto

2.12 Crear una clase coche.

2.12.1 Con los siguientes atributos: o Marca o Modelo o Velocidad máxima o Tipo combustible o Velocímetro o Tacómetro

```
public class Coche {  
  
    private String marca;  
    private String modelo;  
    private int vMax;  
    private int rvMax;  
    private String tipocombustible;  
    private int velocimetro;  
    private int tacometro;  
    private boolean isEncendido;  
    private boolean isMarchaAtras;  
    private int anguloVolante;  
  
    Coche(String marca,String modelo,int vMax,String tipoCombustible)  
    {  
        this.marca=marca;  
        this.modelo=modelo;  
        this.vMax=vMax;  
        this.tipocombustible=tipoCombustible;  
        this.velocimetro=0;  
        this.tacometro=0;  
        this.isEncendido=false;  
        this.isMarchaAtras=false;  
        this.rvMax=40;  
    }  
}
```

2.12.2 Con los siguientes métodos: o Arrancar o Apagar o Acelerar o Frenar o Girar el volante o Dar marcha atrás ,

```
public void pulsarBotonStartStop() {
    if(this.velocimetro==0) {
        this.isEncendido=(this.isEncendido)?false:true;
    }else {
        System.out.        ("Paré el coche primero");
    }
}

public void estado() {
    System.out.        ((this.isEncendido)?"Encendido":"Apagado");
    System.out.        ("V="+this.velocimetro);
    System.out.        ("Sentido public void acelerar() {
    if(isEncendido) {
        if(!isMarchaAtras) {
            if(this.velocimetro<=(this.vMax-10)) {
                this.velocimetro+=10;
                System.out.        ("Acelerando 10 km/h.");
            }else {
                System.out.        ("Imposible acelerar la Velocidad actual es
:"+this.velocimetro+" igual que la máxima del vehiculo("+this.vMax+)");
            }
        }else if(isMarchaAtras){
            if(this.velocimetro<=(this.rvMax-10)) {
                this.velocimetro+=10;
                System.out.        ("Acelerando 10 km/h.");
            }else {
                System.out.        ("Imposible acelerar la Velocidad
actual(reverser) es :"+this.velocimetro);
            }
        }
    }else {
        System.out.        ("Debe encender el vehiuolo primero");
    }
};

"+ ((this.isMarchaAtras)?"Reverse":"Directo"));
}

public void darMarchaAtars() {
    if(this.velocimetro==0) {
        this.isMarchaAtras=true;
    }else {
        System.out.        ("Debe parar el vehiculo primero");
    }
};

public void darMarchaAdelante() {
    if(this.velocimetro==0) {
        this.isMarchaAtras=false;
```

```

        }else {
            System.out.        ("Debe parar el vehiculo primero");
        }

    };

    public void frenar() {
        if(isEncendido) {
            if(!isMarchaAtras) {
                if(this.velocimetro>=(0+10)) {
                    this.velocimetro-=10;
                    System.out.        ("Frenando 10 km/h.");
                }else {
                    System.out.        ("Imposible frenar mas la velocidad es:
"+this.velocimetro);
                }
            }else if(isMarchaAtras){
                if(this.velocimetro>=(0+10)) {
                    this.velocimetro-=10;
                    System.out.        ("Frenando 10 km/h.");
                }else {
                    System.out.        ("Imposible frenar mas la velocidad es:
"+this.velocimetro);
                }
            }
        }else {
            System.out.        ("Debe encender el vehiulo primero");
        }
    };

    public void girarVolante(int angulo) {
        if(angulo<0) {
            if((this.anguloVolante+angulo)>=-540) {
                this.anguloVolante+=angulo;
                System.out.        ("girando para la izquierda....");
            }else {
                this.anguloVolante=-540;
                System.out.        ("No se pude girar mas para la izquierda");
            }
        }
        if(angulo>0) {
            if((this.anguloVolante+angulo)<=+540) {
                this.anguloVolante+=angulo;
                System.out.        ("girando para la derecha....");
            }else {
                this.anguloVolante=540;
                System.out.        ("No se pude girar mas para la derecha");
            }
        }
        System.out.        ((this.anguloVolante<=0)?"Direccion
"+Math.abs(this.anguloVolante)/18+" º a la izquierda":"Direccion
"+Math.abs(this.anguloVolante)/18+"º a la derecha");
    };

```

2.12.3 Hacer pruebas con un coche.

```
<terminado> Coche [Aplicación Java] C:\Users\Usuario\p2\p001\p10
Encendido
V=30
Sentido Reverse
Acelerando 10 km/h.
Imposible acelerar la Velocidad actual(reverser) es
Par el coche primero
girando para la izquierda....
Direccion 28 a la izquierda
No se pude girar mas para la izquierda
Direccion 30 a la izquierda
No se pude girar mas para la derecha
Direccion 30 a la derecha
girando para la izquierda....
Direccion 0 a la izquierda
Encendido
V=40
Sentido Reverse
Frenando 10 km/h.
Frenando 10 km/h.
Frenando 10 km/h.
Frenando 10 km/h.
Imposible frenar mas la velocidad es: 0
Encendido
V=0
Sentido Reverse
Encendido
V=0
Sentido Directo
Acelerando 10 km/h.
Frenando 10 km/h.
Encendido
V=0
Sentido Directo
Apagado
V=0
Sentido Directo
```

Ilustración 4 resultados coche

```
123
124
125 public static void main(String[] args) {
126     Coche c1=new Coche("Nissan","Xtrail",200,"Diesel");
127     c1.acelerar();
128     c1.pulsarBotonStartStop();
129     c1.estado();
130     c1.acelerar();
131     c1.darMarchaAtars();
132     c1.estado();
133     c1.pulsarBotonStartStop();
134     for(int i=0;i<=19;i++) {
135         c1.acelerar();
136         c1.estado();
137     }
138     c1.pulsarBotonStartStop();
139     c1.frenar();
140     c1.estado();
141     for(int i=0;i<=19;i++) {
142         c1.frenar();
143         c1.estado();
144     }
145     c1.estado();
146     c1.darMarchaAtars();
147     c1.estado();
148     c1.acelerar();
149     c1.acelerar();
150     c1.acelerar();
151     c1.estado();
152     c1.acelerar();
153     c1.acelerar();
154     c1.pulsarBotonStartStop();
155     c1.girarVolante(-510);
156     c1.girarVolante(-200);
157     c1.girarVolante(2000);
158     c1.girarVolante(-541);
159     c1.estado();
```

Ilustración 5 Sentencias Pruebas

2.13 Ejercicio 13 : A partir de este código:

```
public class Ejercicio13_1 {  
  
    public int actualFuel = 10;  
  
    public void showDetails() {  
        System.out.println("La capacidad actual es de " +  
this.actualFuel + " litros.");  
    }  
  
    public static void main(String[] args) {  
        Ejercicio13_1 c0 = new Ejercicio13_1();  
        c0.showDetails();  
        System.out.println("Actualización capacidad");  
        c0.actualFuel = -8;  
        c0.showDetails();  
    }  
}
```

No permitir introducir valores negativos a actualFuel.

```
public class Fuel {  
    private int actualFuel;  
    Fuel(int actualFuel){  
        if(actualFuel>=0) {  
            this.actualFuel=actualFuel;  
        }else {  
            this.actualFuel=0;}  
    }  
    public void showDetails() {  
        System.out.        ("La capacidad actual es de " +this.actualFuel + " litros.");  
    }  
    public void rellenar_actual_Fuel(int cantidad) {  
        if((this.actualFuel+cantidad)>=0) {  
            this.actualFuel=cantidad;  
        }else {  
            this.actualFuel=0;  
        }  
    }  
    public static void main(String[] args) {  
  
        Fuel deposito = new      (7);  
        deposito.        ();  
        Fuel deposito2=new      (-9);  
        deposito2.        ();  
        //Sacar el nombre de la instancia??? sin poner un atributo nombre dentro de la clase.  
        deposito.        (-8);  
        deposito.        ();  
        deposito.actualFuel=-100;  
        deposito.        ();  
    }  
}
```

SACAR NOMBRE DE UNA INSTANCIA ¿??????? Sin tenerlo como atributo

2.14 Ejercicio 14: crear clase personas, doctor, policía y profesor y una clase principal donde se llame a los detalles de cada uno de ellos.

```
import java.time.LocalDate;

public class Persona {
    private int dni;
    private String nombre;
    private String apellido1;
    private String apellido2;
    private LocalDate fechaNac;

    Persona(int dni,String nombre,String apellido1,String apellido2,LocalDate fNac){
        this.dni=dni;
        this.nombre=nombre;
        this.apellido1=apellido1;
        this.apellido2=apellido2;
        this.fechaNac=fNac;
    }
    public int getDni() {
        return dni;
    }
    public void setDni(int dni) {
        this.dni = dni;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getApellido1() {
        return apellido1;
    }
    public void setApellido1(String apellido1) {
        this.apellido1 = apellido1;
    }
    public String getApellido2() {
        return apellido2;
    }
    public void setApellido2(String apellido2) {
        this.apellido2 = apellido2;
    }
    public String getFechaNac() {
        return "Fecha nacimeinto "+fechaNac.toString() +" era el día de la semana "+fechaNac.getDayOfWeek().toString();
    }
    public void setFechaNac(LocalDate fechaNac) {
        this.fechaNac = fechaNac;
    }
    public String toString() {
        String txt="+++++++\n";
        txt+="Dni:"+this.dni+"\n";
        txt+="Nombre :"+this.nombre+"\n";
        txt+="Apellidos :"+this.apellido1+" "+this.apellido2+"\n";
        txt+=(this.getFechaNac());
        return txt;
    }
}
```

```

import java.time.LocalDate;

public class Policia extends Persona {
    private String comisaria;
    private int distrito;
    private final static int NOMINA=2200; Todos lo mismo y cambio en todos ese es el uso?

    Policia(int dni, String nombre, String apellido1, String apellido2, LocalDate fNac,String
comisaria,int distrito) {
        super(dni, nombre, apellido1, apellido2, fNac);
        this.comisaria=comisaria;
        this.distrito=distrito;
    }
    public String getComisaria() {
        return comisaria;
    }
    public void setComisaria(String comisaria) {
        this.comisaria = comisaria;
    }
    public int getDistrito() {
        return distrito;
    }
    public void setDistrito(int distrito) {
        this.distrito = distrito;
    }
    public static int getNomina() {
        return NOMINA;
    }
    public String toString() {
        String txt="*****\n";
        txt+="Dni:"+this.getDni()+"\n";
        txt+="Nombre :"+this.getNombre()+"\n";
        txt+="Apellidos :")+this.getApellido1()+" "+this.getApellido2()+"\n";
        txt+=(this.getFechaNac()+"\n");
        txt+="Trabaja en la comisaria :"+this.                ()+"\n");
        txt+="En el distrito "+this.                ()+"\n");
        txt+="Cobra un sueldo de "+Policia.getNomina());
        return txt;
    }
}

```

```
1 import java.time.LocalDate;
2
3 public class Principal {
4
5     public static void main(String[] args) {
6
7         Policia madero1=new Policia(34896415,"Cesar","Bouzas","Soto",LocalDate.of(1977, 7, 4),"A coru
8         System.out.println(madero1.toString());
9         Persona persona0=new Persona(34896414,"Manuel","Bouzas","Nuñez",LocalDate.of(2011, 12, 10));
10        System.out.println(persona0.toString());
11        Persona personal=(Persona)madero1;//No se pierden datos??????
12        System.out.println(personal.toString());
13    }
14 }
15 }
16 }
```

<terminado> Principal [Aplicación Java] C:\Users\Usuario\.p2\pool\plugins\org.eclipse.jdt.launcher\org.eclipse.jdt.launcher.win32.x86_64_18.0.1.v20220517-1800.jar

```
*****
Dni:34896415
Nombre :Cesar
Apellidos :Bouzas Soto
Fecha nacimeinto 1977-07-04 era el día de la semana MONDAY
Trabaja en la comisaria :A coruña
En el distrito 15179
Cobra un sueldo de 2200
+++++
Dni:34896414
Nombre :Manuel
Apellidos :Bouzas Nuñez
Fecha nacimeinto 2011-12-10 era el día de la semana SATURDAY
*****
Dni:34896415
Nombre :Cesar
Apellidos :Bouzas Soto
Fecha nacimeinto 1977-07-04 era el día de la semana MONDAY
Trabaja en la comisaria :A coruña
En el distrito 15179
Cobra un sueldo de 2200
```

Dudas:

- Paso Policia a Persona pero el método sobrescrito me sigue utilizando el de policía ,no se perderían datos??
- Se pude llamar al constructor del padre sin indicar parámetros y que se encomiende él pedirlos o siempre se tienen que crear con paramteros(si los tiene claro).Consructores muy largos.
- Esta bien aplicado el final static para tener todos los polis la misma nomina y poder cambiársela a todos a la vez.

2.15 Ejercicio 15: Crear la clase mercancía con los siguientes atributos : o Id o Nombre o Responsable o Zona o Área o Estantería o Cantidad Y crear la clase mercancía perecedera que extienda de mercancía con el atributo fecha caducidad. Mostrar por pantalla sus características principales. **Ejercicio 16.** Crear interfaz máquina (con los métodos encender, apagar y mantenimiento) y clases avión y tractor que la implementen