

OCP

OPEN/CLOSED PRINCIPLE

**¿Esta nuestro diseño preparado para
ser extendido?**

The Open/Closed Principle (OCP)

“

Software entities (classes, modules, functions, etc.) should be open for extension but closed for modification

— *Bertrand Meyer*

”

mal

```
class Figura {}  
class Cuadrado extends Figura {}  
class Circulo extends Figura {}
```

```
class Pintor {  
    void pinta (Collection<Figura> figuras) {  
        for (Figura figura: figuras) {  
            if (figura instanceof Cuadrado) {  
                pinta((Cuadrado) figura);  
            } else if (figura instanceof Circulo) {  
                pinta((Circulo) figura);  
            }  
        }  
    }  
  
    void pinta (Cuadrado cuadrado) {  
        // ...  
    }  
  
    void pinta (Circulo circulo) {  
        // ...  
    }  
}
```



```
interface Figura {  
    void pinta();  
}
```

```
class Cuadrado implements Figura {  
  
    @Override  
    public void pinta() {  
        // ...  
    }  
}
```

```
class Circulo implements Figura {  
  
    @Override  
    public void pinta() {  
        // ...  
    }  
}
```

bien

```
class Pintor {  
  
    void pinta (Collection<Figura> figuras) {  
        for (Figura figura: figuras) {  
            figura.pinta();  
        }  
    }  
}
```