INFORME EJERCICIO 1

Principios de diseño usados

Principio de sustitución de Liskov:

Un ejemplo de este principio en el código es el método setModo: este recibe como parámetro un objeto tipo ModoFun, pero gracias a que se trata de una interfaz sus métodos son abstractos y se implementan en cada subclase, no se sobrescriben. De este modo, el funcionamiento de ModoFun no es modificado por las subclases sino que se implementa en estas, permitiendo que la función setModo reciba parámetros de tipo Off, Manual etc.

Principio de inversión de la dependencia:

Otra vez ModoFun es un ejemplo, ya que al ser una interfaz común a todos los modos de funcionamiento, el atributo modo presente en el termostato no depende de métodos concretos, sino de los métodos abstractos de la interfaz, que se implementan por parte de las subclases.

Principio abierto-cerrado:

Gracias al mecanismo de herencia, se pueden realizar modificaciones en el termostato sin tener que realizar cambios drásticos en el código. Por ejemplo, si quisiésemos añadir un estado más al termostato, bastaría con crear una nueva clase que herede de ModoFun, implementar el nuevo método que representa el modo de funcionamiento en la subclase, y añadirlo también en la interfaz como método abstracto.

Patrones de diseño usados

En esta práctica se han usado dos patrones: el patrón estado y el patrón instancia única.

El primero es el patrón en el que se basa fundamentalmente la práctica, ya que según este patrón un objeto puede cambiar su comportamiento al cambiar su estado interno, y esto es exactamente lo que hacen los objetos de la clase Termostato: al cambiar su atributo modo, cambian su comportamiento. Para hacer esto de forma más simple, sin tener que crear un objeto de cada modo de funcionamiento para poder cambiar de estado, se utiliza el patrón instancia única, que nos permite tener una sola instancia de cada clase, y acceder a ella mediante un método de la propia clase. De este modo, no necesitamos crear varios objetos de cada modo de funcionamiento, sino que ya están creados al inicio de la ejecución

Los diagramas que explican el patrón estado van por separado, para que sean más fáciles de ver.