Modelado Y Programación
Práctica 2

Alan Ignacio López Carrillo 420014760
Jonathan Bautista Parra 419003227
Javier Axel Pérez Juárez

State

El patrón State sugiere que crees nuevas clases para todos los estados posibles de un objeto y extraigas todos los comportamientos específicos del estado para colocarlos dentro de esas clases.
En lugar de implementar todos los comportamientos por su cuenta, el objeto original, llamado contexto, almacena una referencia a uno de los objetos de estado que representa su estado actual y delega todo el trabajo relacionado con el estado a ese objeto.

Desventajas:
- Esta implementación no es deseable cuando hay muchos estados de tipo diferente.
- Esta implementación implica una gestión bastante especializada de los estados (el contexto debe conocer los diferentes tipos de estados). Esto puede hacer que sea más difícil la modificación del diagrama de estado.

Template

Es un patrón de diseño que define una estructura algorítmica en la súper clase, delegando la implementación a las subclases. Es decir, define una serie de pasos, en donde los pasos serán redefinidos en las subclases. Permite que las subclases redefinan ciertos pasos del algoritmo sin cambiar su estructura.

Desventajas:
- Se puede producir ambigüedad si no se escribe bien.
- Si el método plantilla llama demasiados métodos abstractos, se cansará pronto de utilizar AbstractClass como superclase.

Iterator

El patrón de diseño Iterator proporciona una forma de acceder secuencialmente a los elementos de un objeto compuesto por agregación sin necesidad de desvelar su representación interna.

Desventajas:
- Un Iterator no es thread-safe. Las implementaciones típicas de Iterator (es decir, el patrón en su forma más pura) tienen un comportamiento no definido si la colección se modifica mientras se está iterando.