

Practica 2

Luis Geovanni Méndez Ávila

24 de Octubre 2020

1.-

Patrón state

Principios:

- Separa el comportamiento de un objeto en distintas clases, en las cuales las acciones que pueda hacer el objeto pueden variar dependiendo del estado actual del objeto principal.
- Permite tener una mejor perspectiva de los posibles estados en los que puede estar un objeto.

Desventajas:

- Se tienen que crear muchas variables de clase al objeto principal las cuales representan cada uno de los posibles estados que puede tener.

Patrón template

Principios:

- Crea una interfaz general de algun paso de un algoritmo, de la cual implementarán clases que tienen relacion con el paso pero que tienen algunas implementaciones más específicas.
- Hay una clase principal que ejecuta el algoritmo deseado en la cual se le pasa una referencia a alguna de las clases que implementan la plantilla del paso específico con su respectiva variación.

Desventajas:

- A simple vista no noto alguna.

2.-

Compilar y ejecutar

Descomprimir el zip y con una consola dirigirte hasta la carpeta src. Despues compila con el comando `javac *.java` , una vez complilado ejecutas con `java Main`.

Nota.

Creo que la idea de ambos patrones fue bien implementada, sin embargo, mi complicación fue al pasar la casa a construir, tuve que andar cambiando firmas a algunos métodos y agregando algunos de más para que pudiera funcionar de alguna manera. Por lo cual los diagramas UML quedan distintos de la implementación final . Además por alguna razón no actualiza el estado actual , por lo que al robot después de darle la indicacion del tipo de casa a construir no cambia su estado inactivo.

Me faltó documentar varias partes del código , por lo que no quedó comentado fueron métodos que fui agregando sobre la marcha.