

Práctica 2

Concha Zamora Marco Antonio - 320028421

Decuir Fuentes Rodrigo André - 320022711

Demetrio Torres Yanahi - 319169922

Equipo: Tukú Wana Min Nakú

-
1. Menciona los principios de diseño esenciales del patrón State, Template e Iterator.

State permite que un objeto modifique su comportamiento cada vez que hay un cambio su estado interno. Es muy común que en cualquier aplicación existan estructuras con una colección de estados entre los cuales un objeto realiza distintas transiciones.

Template define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos. Esto permite que las subclases redefinan ciertos pasos de un algoritmo sin cambiar su estructura.

Iterator se usa para ofrecer una interfaz de acceso secuencial a una determinada estructura ocultando la representación interna y la forma en que realmente se accede, o lo que es lo mismo, permite realizar recorridos sobre objetos compuestos independientemente de la implementación de éstos.

2. Menciona una desventaja de cada patrón.

State

- ✎ Aplicar el patrón puede resultar excesivo si sólo tienen unos pocos estados o el estado rara vez cambia.

Template

- ✎ Los métodos de plantilla tienden a ser más difíciles de mantener cuantos más pasos tengan.

Iterator

- ✎ Aplicar el patrón puede resultar excesivo si la aplicación funciona únicamente con colecciones sencillas.

3. Anotaciones acerca de la práctica.

a) Patrón de diseño State

Identificamos que el robot contaba con estados determinados y condiciones que lo llevaban de un estado a otro. Los estados están definidos en la interfaz de EstadosRobot, la cual es implementada por cada una de las clases de cada estado, donde fijamos una conducta determinada para cada acción que se pretenda llevar a cabo.

b) Patrón de diseño Template

Las hamburguesas están construidas bajo un diseño estándar que es algo similar a la "base" de la hamburguesa, una vez definida dicha "base", las variaciones son

implementadas a través de la sobrecarga de distintos métodos en la clase abstracta "Hamburguesa", con ello logramos tener distintas hamburguesas que implementan la misma clase pero con distintas características.

c) Patrón de diseño Iterator

Dados los menús que se plantean en el problema, primero declaramos la interfaz IteradorMenú con las operaciones necesarias para iterar sobre los menús que tienen como elementos a las hamburguesas. Luego para cada tipo de menú se crea una clase que va a implementar la interfaz iterador y que va a iterar sobre cada hamburguesa del menú; para el menú especial IteradorMenuEspecial va a usar una hashtable de hamburguesas para recorrer el menú, para el menú del día IteradorMenuDelDia va a usar un ArrayList para recorrer el menú y para el menú general se va a usar un arreglo para recorrer el menú.

4. Instrucciones de ejecución.

Entra a la carpeta "src" y compila el código con:

```
javac *.java
```

Desde la misma carpeta ejecuta el programa con el siguiente comando:

```
java McBurguesas
```

Para generar la documentación, desde la carpeta src ejecuta el siguiente comando:

```
javadoc -d documentacion *.java
```