

Trabajo Práctico – State, Strategy y Abstract Factory

Objetivo: Comprender los patrones de diseño State, Strategy y Abstract Factory. Discutir alternativas de diseño e implementación.

Ejercicio 1 – Responder y justificar la respuesta

Para cada uno de los casos indique el patrón que utilizaría para resolverlo y una pequeña justificación.

Caso a: Una máquina de video juegos funciona ingresando fichas. Además de los controles para jugar, la maquina posee un único botón de inicio y una ranura para ingresar fichas. Cuando la maquina se enciende presionar el botón de inicio genera un cartel en la pantalla que indica que ingresen fichas, si se introduce una ficha y se presiona el botón de inicio se comienza el juego para un único jugador; si en cambio antes de presionar el inicio se ingresaron dos fichas pueden jugar dos jugadores. Cuando termina el juego, vuelve al momento inicial.

Caso b: Un auto todo terreno posee un motor que se adapta a los diferentes terrenos por los que transita para poder mejorar la performance y el consumo de combustible. En este motor se pueden utilizar diferentes modos, los cuales son seleccionados por el conductor antes de la travesía:

- Terreno fangoso: mucha potencia y poca velocidad.
- Súper Autopista: Mucha velocidad y aminora la potencia de tracción.
- Ciudad: equilibrio entre potencia y tracción.

Ejercicio 2 - Calculadora

Se desea modelar una calculadora simple. Las operaciones que permite realizar son suma (+), resta (-), multiplicación (*) división (:) y resultado (=).

Dicha calculadora recibe los operandos u operadores uno a uno e ir almacenando resultados parciales hasta recibir el operador “=”.

Ejercicio 3 - Logueo de Información

El controlador de vuelo de un avión registra valores de velocidad, temperatura y altitud de la nave. Estos valores pueden serle seteados de manera directa y, además, entiende los mensajes incrementarVelocidad(float aumento) y decrementarVelocidad(float decremento) que modifican el valor actual de la velocidad sumando o restando ese valor.

Por razones de seguridad este controlador debe loguear toda modificación que registre en sus medidas, indicando la fecha, hora, medida modificada (velocidad, temperatura o altitud), valor anterior y nuevo valor.

El controlador puede configurarse con uno de varios posibles modos de realizar el logueo, e incluso el usuario puede cambiar este modo aun cuando el controlador se encuentra operando, sin detener su funcionamiento. Estos modos son los siguientes:

- Logueo Base de Datos: la información a loguear se almacena en una base de datos, cuyo nombre, usuario y password pueden ser configurados.
- Logueo Archivo Texto Plano: la información a loguear se almacena en un archivo de texto plano, dedicando una línea a cada log, con sus valores

separados por comas. El nombre del archivo y su ubicación pueden ser configurados.

- Logueo Archivo XML: la información a loguear se almacena en un archivo en formato XML. Cada log tiene un tag, dentro del cual se tienen otros tags anidados, uno para cada dato del log. El nombre del archivo y su ubicación pueden ser configurados.
 - Logueo Red: la información a loguear se envía por la red hacia otro sitio, para asegurar la preservación de la misma. El puerto por el que se envía la misma puede ser configurado.
1. Realice un diagrama de clases UML de la solución completa.
 2. Implemente en Java la solución contemplando sólo los casos Logueo Archivo Texto Plano y Logueo Archivo XML.
 3. Realice tests que le permitan chequear el funcionamiento de los dos modos del punto 2, incluyendo el cambio de un modo a otro sin detener el controlador de vuelo. En este último caso, el logueo de medidas deberá quedar en parte en un archivo y en parte en otro.

Ejercicio 4 - Reproductor MP3

Se desea desarrollar un sistema que permita controlar la reproducción de música en un reproductor MP3. El sistema permite realizar las siguientes acciones:

- Play: Reproduce la canción seleccionada, si es que el reproductor está en modo de selección de canciones. Caso contrario produce un error.
- Pause: Pausa la canción que se está reproduciendo. Si la canción está en pausada retoma la reproducción, caso contrario genera un error.
- Stop: Para la reproducción o la pausa, y pasa a selección de canciones. Caso contrario no hace nada.

Se asume que existe una clase "Song" que entiende los mensajes play(); pause(); stop(); con la cual usted debe interactuar.

1. Realizar un diagrama de clases UML.
2. Implementar la solución en Java.
3. Realizar test que permitan chequear que los cambios de estados funcionan correctamente ante diferentes combinaciones de operaciones.

Ejercicio 5 - Sistemas de Alarma

Una Empresa de Seguridad se dedica a la instalación de Sistemas de Alarmas en inmuebles. Cada Sistema de Alarma instalado en un inmueble está monitoreado por la Empresa de Seguridad, y está constituido por los siguientes componentes:

- Una Central Local: es quien coordina a todos los componentes del Sistema de Alarma, por lo que todos estos componentes se conectan a esta central.
- Tablero de Control: es la interfaz mediante la cual el usuario opera el Sistema de Alarma. Mediante este componente puede activar, desactivar y monitorear el sistema.

- Sensores de Movimiento: son pequeños dispositivos que detectan movimiento en un área determinada, avisando a la central cuando lo detectan.
- Sensores de Apertura: son pequeños dispositivos que detectan la apertura de puertas y ventanas, avisando a la central cuando la detectan.
- Alarma: es el dispositivo sonoro (tipo sirena) que se activa por orden de la central.
- Dispositivo de Comunicación: es el dispositivo que comunica al Sistema de Alarma con la Empresa de Seguridad, lo cual permite el monitoreo a distancia del inmueble. A través de él, la central local comunica a la empresa eventos tales como activación/desactivación del sistema y disparo de la alarma.

Cada nuevo Sistema de Alarma se instala mediante los siguientes pasos:

1. Instalación de una Central Local.
2. Instalación de un Tablero de Control.
3. Instalación de Sensores de Movimiento (cantidad especificada por el usuario).
4. Instalación de Sensores de Apertura (cantidad especificada por el usuario).
5. Instalación de una Alarma.
6. Instalación de un Dispositivo de Comunicación.

Si bien el proceso de instalación para todos los Sistemas de Alarmas es el anteriormente descripto, cada uno puede ser instalado de acuerdo a dos tipos distintos de configuración: *Standard* o de *Alta Seguridad*.

La diferencia entre una y otra configuración es el tipo de componentes por los que está compuesto el sistema, de forma tal que los componentes de la segunda configuración son de mayor confiabilidad (vulnerabilidad, errores y falsos positivos) que los de la primera configuración. En este sentido, cada configuración tiene las siguientes particularidades.

En la Configuración *Standard* se instalan sensores de movimiento infra-rojos, sensores de apertura magnéticos, un dispositivo de comunicación telefónico y una alarma que corresponde a un modelo básico.

En la Configuración de *Alta Seguridad* se instalan sensores de movimiento por ultrasonido, sensores de apertura ópticos, el dispositivo de comunicación es de radiofrecuencia y la alarma corresponde a un modelo antirrobo, que se dispara ante el primer intento de remoción de su carcasa.

En cuanto a la Central Local y Tablero de Control, ambas configuraciones contemplan el uso del mismo tipo.

1. Especifique qué Patrón de Diseño cree apropiado aplicar en este contexto y por qué.
2. Diseñe un diagrama de clases que modele el problema.
3. Especifique la relación entre las clases de su modelo y los roles de las clases participantes del patrón de diseño seleccionado (según el libro de Gamma).
4. No es necesario que implemente el patrón Observer, pero explique cómo ese patrón se articularía con el seleccionado aquí.