



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°5

1. ¿Que es un patrón de diseño?
2. Investigue e identifique las diferentes categorías de patrones de diseño que existen.
3. Explique cómo funciona el patrón Singleton para establecer la conexión con una base de datos de Postgresql o MySql.
4. Investigue y aplique el patrón DAO (data access object) a un sistema que usted elija para conectarlo con la base de datos.
5. Aplique el patrón Builder para la construcción de tres tipos de tortas, una torta deberá tener como atributo la masa y el relleno. Las tortas deben ser de Vainilla, coco y una a su elección para demostrar el objetivo del patrón solicitado.
6. Aplique el patrón Factory Method para la creación de juegos físicos y digitales. Los juegos comparten un id y un importe. También el método abstracto getPrecio, que se encarga de calcular el precio de uno u otro. Para ello los juegos físicos tienen un atributo que es el precio de traslado (a caso de ejemplo elija usted). Y los juegos digitales el precio depende de la plataforma en la cual se compra teniendo como atributo el precio de la plataforma. Estos valores deben ser float y multiplicarlos al importe.
7. Patrón Facade.

Cree una clase Helper con un método que permita traducir una palabra de español a inglés. Establezca traducciones para al menos 5 palabras.

Cree una clase Helper dos métodos: uno que convierta una cadena a mayúsculas y otro que convierta a minúsculas.

Luego, implemente un patrón Facade para permitir el acceso a estos tres métodos desde una nueva clase.

Para probar la solución, elija tres palabras e imprima para cada una

- Su traducción a inglés
- Su versión con solo mayúsculas
- Su versión con solo minúsculas

8. Como parte de un sistema de administración de archivos debe implementar, usando el patrón Composite, las clases necesarias para organizar carpetas y archivos.



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°5

Todo elemento posee un nombre y un método que indica si es carpeta o no. En el caso de las carpetas cuentan con un listado de elementos que contienen.

El objetivo final será crear una serie de elementos y su contenido para luego imprimir el nombre de cada uno.

Ejemplo de salida:

- Carpeta 1
 - Carpeta 2
 - Archivo 1
 - Archivo 2
 - Archivo 3
- Carpeta 3
 - Archivo 4

9. Patrón decorator.

Dada una clase Producto con un método public String getPrecio() que el precio del mismo como String y con dos decimales.

Eje: 90000,50

Es necesario crear dos decorators, uno para cuando el sistema usa moneda estadounidense(\$USD) y otro para cuando el sistema usa moneda argentina (\$ARS).

Cada decorator debe implementar getLineDescription de forma de que se devuelva el símbolo de la moneda antes del valor.

Eje: \$ARS 90000,50 o \$USD 90000,50

Pruebe la implementación con un producto, imprimiendo:

- Lo que retorna el método getPrecio()
- Lo que retorna el método getPrecio() para el primer decorator.
- Lo que retorna el método getPrecio() para el segundo decorator.

10. Aplique el patrón de diseño Observer para un sistema de reporte del clima, defina la clase Sistema Meteorológico que extienda de la clase Observable y tenga como atributo el estado del clima en forma de String. Cada vez que se modifica el clima esté notifica a



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°5

todos los observadores. El observador va a ser la clase Reportero que implementa la interfaz Observer, él será el encargado de imprimir por consola el clima cuando cambia en el sistema meteorológico.

Pruebe el sistema creando instancias de cada clase y cambiando el clima.

11. Aplique el patrón State para simular el funcionamiento de atención de una caja en un banco. La clase Banco tiene una caja y este puede atender una persona, suspender, cerrar y abrir la caja. La caja tiene el nombre del cajero y el estado actual, los estados posibles son:

- Abierta: imprime por pantalla al cliente próximo en la fila.
- Suspendida: en este estado solo atiende a personas mayores a 60 años de edad, en otro caso imprime un mensaje de espera.
- Cerrada: no atiende personas y muestra mensajes indicando su estado.

Pruebe el correcto funcionamiento del sistema creando personas con diferentes edades y cambiando los estados de la caja.

12. Patrón Strategy.

En una plataforma de streaming de películas se desea consultar el catálogo. Sin embargo hay diferentes situaciones que podrían llevar a que el sea filtrado de diferentes formas, un ejemplo de ello es cuando la cuenta que está usando el sistema pertenece a un niño. En este caso el catálogo solo debe mostrar películas que no sean para mayores de 13 años.

Utilizando el patrón strategy defina una estrategia general que retorne el catálogo completo y luego una estrategia para niños que filtra las películas por edad.

Escriba en consola el listado implementando la estrategia para todo el catálogo. Luego imprima el listado usando la estrategia para menores de 13.

Finalmente, implemente una tercera estrategia que retorne el catálogo de películas para menores de 18 años. Compruebe el resultado en consola.

13. Aplique el patrón MVC, Singleton y observer en el siguiente programa. En el centro de control climatológico se recaba información de 3 formas diferentes. En el mar se tiene una boya marina junto a la costa de la ciudad que informa la temperatura y salinidad del agua. En tierra se tiene una estación meteorológica que brinda al centro de control valores de presión atmosférica, temperatura y velocidad del viento. Por último, se tiene acceso a un satélite que informa el nivel de radiación UV.



PROGRAMACIÓN ORIENTADA A OBJETOS

TRABAJO PRÁCTICO N°5

Aplique el patrón observer en la relación entre los dispositivos que recopilan los valores y el centro de control, toda esta información tendrá que ser almacenada en una base datos, en esta conexión debe de implementar el patrón Singleton. Por último desarrolló una interfaz gráfica aplicando MVC que permite ingresar el periodo de fechas y este devuelve los datos en base.