

## **PRÁCTICO N° 8**

**TEMA: Patrones de Diseño.**

### **Objetivos**

Con este práctico se espera que el estudiante pueda:

- Analizar el problema e identificar el patrón de diseño más conveniente.
- Identificar las ventajas provistas por cada patrón en cada situación.
- Instanciar la plantilla del patrón de diseño al problema real.
- Aplicar el patrón de diseño para elaborar una solución conveniente en cada caso.

Para cada uno de los ejercicios planteados a continuación:

- a) Identificar el o los patrones de diseño que modelan la solución.
- b) Justificar la elección.
- c) Modelar con un diagrama de clases de diseño.

En algunos casos se solicitará la implementación la solución en Java.

### **1 - Procesador de textos**

Se está desarrollando una aplicación de edición de textos que será utilizada tanto en Estados Unidos como en Argentina. Como se sabe la construcción de las fechas es diferente según donde nos encontremos, en Estados Unidos es mes/día/año, mientras que en Argentina es día/mes/año. Desarrollar utilizando patrones de diseño los generadores de fecha para que puedan confeccionarse documentos para uno y otro destino.

## 2 - Evaluador de funciones matemáticas

Se desea desarrollar un evaluador de funciones matemáticas de una variable en java. Una función matemática posee un nombre, tipo de función (Lineal, Cuadrática o Trigonométrica), una variable independiente y está formada por una expresión matemática de la siguiente forma:

- una constante es una expresión válida.
- una variable es una expresión válida.
- si  $p$  y  $q$  son expresiones válidas, entonces  $p + q$ ,  $p - q$ ,  $p * q$ ,  $p : q$  son expresiones válidas.

Un ejemplo de una función válida podría ser:

$$f(x) = x + 3 * 4$$

El evaluador deberá tener implementado el método *evaluar( valor: Integer): Double*. Por ej.

$$f.evaluar(10) = 22$$

Implemente la solución en Java.

## 3 - Imágenes Satelitales

Un estudio agropecuario dedicado al cultivo de cereales y crianza de ganado realiza el análisis de imágenes satelitales para determinar factibilidad de realizar alguna actividad. Se ha encomendado a un grupo de licenciados en computación el desarrollo de un sistema informático que permita descargar las imágenes de un servidor externo. El problema a resolver es que las imágenes satelitales son muy extensas (y de gran tamaño) por lo que la descarga de la imagen con la resolución máxima insumiría un tiempo inaceptable (en algunos casos). En ocasiones una resolución de calidad media puede ser útil y en otros no. Se requiere que el sistema descargue la resolución de la imagen a medida que sea necesaria y demandada por un usuario.

## 4 - LogDisk

Se desea implementar una clase que monitoree los accesos a disco de las diferentes aplicaciones de un sistema. La clase LogDisk deberá poseer un atributo de tipo Date donde se registrará la fecha de creación del LogDisk, dos contadores que van registrar la cantidad de accesos de lectura y de escritura respectivamente, y los métodos addRead() y addWrite que incrementarán los contadores cuando corresponda. Tenga en cuenta que la clase LogDisk deberá poseer una única instancia ya que de lo contrario se perderían todos los datos cada vez que se genere una nueva instancia.

Implemente en Java la clase LogDisk.

## **5 – Lista Polimórfica**

Modelar el tipo de datos ListaPolimorfica que pueda contener elementos de tipo Integer, String o elementos sean de tipo ListaPolimorfica también.

## **6 - Juegos de Mesa**

Se desea implementar una plataforma informática de juegos de mesa que provea la funcionalidad de producir 3 juegos diferentes. El juego “A”, “B”, y “C”. Cada juego utiliza fichas, dados y cartas de diferentes tipos.

## **7 - Software fotográfico**

Un analista en computación desarrolla un software para edición de fotografías. Ha decidido utilizar una librería de Corel Draw para la definición de nuevos filtros. La librería cuenta con las funciones de: iluminar(); pixelar(); resaltarBlancos(); resaltarColoresPrimarios(); enfocar(); desenfocar(); quitarOjosRojos(); mosaico() entre otros. El analista desea crear dos filtros nuevos: ResaltarColores() (el cual debería invocar a resaltarBlancos() y resaltarColoresPrimarios()). Y el segundo filtro: Artista() (el cual debería invocar a las acciones de iluminar(); enfocar(); quitarOjosRojos() y mosaico()).

Implemente en Java una solución aplicando el patrón elegido.

Nota: Las funciones solo deberán mostrar un mensaje por pantalla.

## **8 - Movimientos Bancarios**

Diferentes aplicaciones Clientes se conectan a un único Servidor para enviarle los valores bancarios de cada día. Dependiendo de si estos valores son positivos o negativos, el servidor crea el movimiento de Banco correspondiente (Depósito para los positivos, Débito para los negativos).

## **9 - Seguridad S.A.**

Una empresa de la ciudad posee un sistema de control basado en sensores. Cuando el sistema de seguridad está activado cada sensor detecta una determinada situación y ejecuta una acción particular, como por ejemplo, realizar una llamada telefónica, activar los sistemas contra incendios, activar una sirena, etc. En la actualidad el sistema cuenta con 3 tipos de sensores (sensores para incendios, de movimiento, de sonido) y se prevé en el futuro incorporar otros sensores como el de gas.

## **10 - Lista**

Modelar la implementación de una clase Lista y dos recorridos sobre ella:  
Recorrido hacia adelante y recorrido hacia atrás.

## **11 - Árbol Binario**

Modelar recorridos en pre-orden de un árbol binario.

Implemente en Java

## **12 - Inmobiliaria “Buenos Muchachos”**

Una inmobiliaria vende propiedades en la ciudad y zona. La empresa ha solicitado el desarrollo de un sistema para la gestión y venta de propiedades. Los empleados de la inmobiliaria presentan las propiedades a los clientes y al momento de realizar una operación generan una solicitud de venta. A uno de los equipos afectados al proyecto se le encomendó el diseño e implementación de las historias de usuario relacionadas a la creación de las solicitudes de ventas. Luego de un análisis por parte del equipo de analistas, se tomaron las siguientes decisiones con respecto al diseño:

- Definir la clase GestiónSolicitud cuya responsabilidad será la de crear las solicitudes de compra dependiendo de la naturaleza de la solicitud.
- Las solicitudes de compra podrán ser de dos tipos diferentes. Solicitud de compra de contado o bien a través de un crédito. Cada tipo de solicitud cuenta con el método calculoMontoFinal() el cual calculara de maneras diferentes el monto final de la propiedad defendiendo si la operación se realiza de contado o con un crédito.

- a) Determine el patrón de diseño más apropiado para modelar este fragmento del sistema. ¿A qué categoría pertenece el patrón seleccionado?
- b) Observando la plantilla del patrón de diseño e instancie la misma al problema e implemente en Java.

### 13– Chat

El siguiente código Java implementa un chat aplicando un patrón de diseño.

Construya el diagrama de clases UML subyacente y determine de qué patrón se trata.

```
public interface IUserioChat {

    public void recibe(String de, String msg);
    public void envia(String a, String msg);

}

public class Usuario implements IUserioChat {
    private String nombre;
    private SalonDeChat salon;

    public Usuario(SalonDeChat salonDeChat) {
        salon = salonDeChat;
    }

    public void recibe(String de, String msg) {
        String s = "el usuario " + de + " te dice: " + msg;
        System.out.println(nombre + ": " + s);
    }

    public void envia(String a, String msg) {
        salon.envia(nombre, a, msg);
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

}

public interface ISalonDeChat {

    public abstract void registra(Usuario participante);
    public abstract void envia(String de, String a, String msg);
}
|
}
```

```

public class SalonDeChat implements ISalonDeChat {

    private HashMap<String, Usuario> participantes = new HashMap<String, Usuario>();

    public void registra(Usuario user) {
        participantes.put(user.getNombre(), user);
    }

    public void envia(String de, String a, String msg) {
        if (participantes.containsKey(de) && participantes.containsKey(a)) {
            Usuario u = participantes.get(a);
            u.recibe(de, msg);
        } else {
            System.out.println("Usuario inexistente");
        }
    }
}

```

```

public static void main(String[] args) {
    SalonDeChat s = new SalonDeChat();

    Usuario u = new Usuario(s);
    u.setNombre("Juan");
    s.registra(u);

    Usuario u1 = new Usuario(s);
    u1.setNombre("Pepe");
    s.registra(u1);

    Usuario u2 = new Usuario(s);
    u2.setNombre("Pedro");
    s.registra(u2);

    u.envia("Pepe", "Hola como andas?");
    u1.envia("Juan", "Todo ok, vos?");
    u2.envia("Martin", "Martin estas?");
}

```

Problems @ Javadoc Declaration Console Search

<terminated> Main (11) [Java Application] C:\Program Files\Java\jre6\bin

Pepe: el usuario Juan te dice: Hola como andas?

Juan: el usuario Pepe te dice: Todo ok, vos?

Usuario inexistente