

TRABAJO PRÁCTICO - PATRÓN COMPOSITE

Trabajo Práctico - Patrón Composite

Objetivos

Comprender el patrón de diseño Composite. Discutir alternativas de diseño e implementación.

Cultivos

Diseñe una solución para el siguiente problema. Se debe modelar un sistema que controla el uso que se le da a porciones productivas de tierra. Las porciones pueden ser puras de un cultivo o mixtas, estas ultimas se dividen en cuatro regiones iguales. En cada región se pueden plantar solamente soja, trigo o subdividir la porción en cuatro siguiendo las mismas reglas anteriores. La figura muestra un ejemplo de esto. A partir del cultivo, el sistema debe determinar las ganancias anuales de cada tipo de cultivo para una región. La región cultivada con soja da una ganancia anual de \$500, la de trigo \$300. En caso que la región este subdividida, la ganancia será proporcional a la cantidad de sub parcelas que posean ese cultivo. El sistema debe responder a la ganancia anual que dará la parcela tanto para soja como para trigo.



1. Implemente en Java
2. Describa las alternativas que puede utilizar para calcular la ganancia anual. Relacione las alternativas.
3. Explique la discusión sobre quiénes deben implementar las operaciones de agregado y borrado de hijos.
4. Sobre su implementación, indique los roles que desempeñan cada una de las clases diseñadas en relación al patrón Composite.

ShapeShifter

ShapeShifter es un nuevo tipo objetos que posee una estructura del tipo arbórea. Estos objetos deben poder componerse, detectar la mayor profundidad y "achatarsse". En esta primera etapa es necesario que simplemente trabajen con Integer. Como la idea es un tanto difusa, se provee una interfaz java y una serie de descripciones con ejemplos de cada una de las funcionalidades necesarias.

```

public interface IShapeShifter{
    public IShapeShifter compose(IShapeShifter);
    public int deepest();
    public IShapeShifter flat();
    public List<Integer> values();
}

```

Se espera que la clase que implemente la interfaz cumpla con:

- public IShapeShifter **compose**(IShapeShifter);
 - recibe un IShapeShifter y retorna un nuevo IShapeShifter que es el resultado de componer al receptor con el que es enviado como parámetro.
 - La composición se realiza al mismo nivel de profundidad.
 - En la **Figura 1** se muestran esquemas de IShapeShifter, cada letra simbolizaria un nombre de variable. Entonces las siguientes expresiones deberían ser verdaderas:
 - a.compose(b) es igual a c
 - d.compose(c) es igual a d
 - d.compose(e) es igual a f
- public int **deepest**(IShapeShifter)
 - Retorna un numero que representa la profundidad máxima alcanzada en composiciones que contiene.
 - Continuando con las figuras de ejemplo, las siguientes expresiones deben ser validas:
 - a.deepest() es igual a 0.
 - c.deepest() es igual a 1.
 - f.deepest() es igual a 3.
- public IShapeShifter **flat**() Achata un IShapeShifter.
 - Si el IShapeShifter posee una profundidad maxima ≥ 1 , entonces retorna un IShapeShifter de profundidad maxima 1 con todos los IShapeShifter de profundidad 0 contenidos. En cualquier otro caso, retorna el mismo IShapeShifter.
 - Siguiendo los siguientes ejemplos:
 - a.flat() es igual a a.
 - f.flat() es igual a g.
 - Importante: No es necesario establecer un orden particular de los IShapeShifter en los resultantes.
- public List<Integer> **values**()
 - Retorna una lista de enteros con los valores incluidos en el IShapeShifter.
 - Siguiendo el ejemplo:
 - a.values retorna una lista con el entero 1.
 - d.values retorna una lista con los enteros 3,1 y 2.
 - Importante: No es necesario establecer un orden particular de los enteros en la lista.

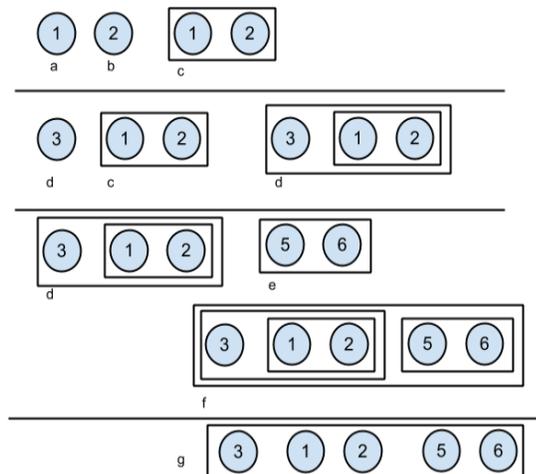


Figura 1. Ejemplos

Ejercicios:

1. Realizar un diagrama de clases UML donde muestre un diseño orientado a objetos que resuelva el problema aplicando el patrón Composite.
2. Indicar en el diseño cuales son los roles del patrón.

3. Implementar en java todo lo necesario para resolver la funcionalidad de la interface con clases que la implementen
4. Escribir un método en el cual muestre como se instancia el ejemplo a y c.

Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by/4.0/)
