

Patrón Composite

Considerar una aplicación en la que se desea tener los datos de una Uea, incluyendo las Ueas que forman parte de su rama de seriación.

En este caso se trata de una jerarquía del mismo tipo de elementos (Uea), por lo que se aplicará el patrón *Composite*

Se creará un nuevo proyecto llamado **CompositePattern** con los siguiente paquetes:

- uam.patrones.composite.clases
- uam.patrones.composite.principal

Se creará la clase Uea en el paquete de clases

Uea.java

```
package uam.patrones.composite.clases;

public abstract class Uea {

    private String nombre;
    private String clave;
    private int credits;

    abstract public void agregar(Uea c);
    abstract public void eliminar(Uea c);
    abstract public void mostrar(int profundidad);

    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getClave() {
        return clave;
    }
    public void setClave(String clave) {
        this.clave = clave;
    }
    public int getCredits() {
        return credits;
    }
    public void setCredits(int credits) {
        this.credits = credits;
    }

    public String toString() {
        return getNombre() + " " + getClave() + " "
            + getCredits();
    }
}
```

El patrón *Composite* indica que debe contener lo que cualquier elemento puede hacer, sus características y la forma en que se agregarán, eliminarán o imprimirán sus elementos.

Dos clases implementarán estas funcionalidades, una tratándose como si fuera un elemento sencillo (una Hoja) o como una colección (un Compuesto).

Hoja.java

```
package uam.patrones.composite.clases;

public class Hoja extends Uea{

    @Override
    public void agregar(Uea c) {

    }

    @Override
    public void eliminar(Uea c) {

    }

    @Override
    public void mostrar(int profundidad) {
        System.out.println(toString());
    }

}
```

En este caso, cuando se trata de un elemento sencillo, solo tiene sentido el que imprima su información. Cuando se trata de un elemento compuesto, si tiene sentido darle funcionalidad al resto de las operaciones.

ComopositePattern.java

```
package uam.patrones.composite.clases;

import java.util.LinkedList;

public class CompositePattern extends Uea{

    private LinkedList<Uea> dependientes =
        new LinkedList<Uea>();

    @Override
    public void agregar(Uea c) {

        dependientes.add(c);

    }

}
```

```

@Override
public void eliminar(Uea c) {

    dependientes.remove(c);

}

@Override
public void mostrar(int profundidad) {
    String nivel = "-";
    System.out.println(toString());
    for (int i = 0; i < dependientes.size(); i++) {
        for(int j=0;j<profundidad;j++)
            nivel+="-";
        System.out.print(nivel);
        dependientes.get(i).mostrar(profundidad+1);
    }

}

}

```

Finalmente se crea una clase Principal para darle funcionalidad al proyecto.

Principal.java

```

package uam.patrones.composite.principal;

import uam.patrones.composite.clases.CompositePattern;
import uam.patrones.composite.clases.Hoja;

public class Principal {

    public static void main(String[] args) {

        patronComposite();

    }

    private static void patronComposite() {

        CompositePattern tallerMatematicas = new CompositePattern();
        CompositePattern introduccionAlCalculo = new CompositePattern();

        CompositePattern calculoDiferencial = new CompositePattern();
        Hoja complementosMatematicas = new Hoja();
        Hoja calculoIntegral = new Hoja();

        tallerMatematicas.setClave("1112026");
        tallerMatematicas.setNombre("Taller de Matematicas");
        tallerMatematicas.setCreditos(7);
    }

}

```

```
introduccionAlCalculo.setClave("1112027");
introduccionAlCalculo.setNombre("Introduccion al Calculo");
introduccionAlCalculo.setCreditos(6);

complementosMatematicas.setClave("1112013");
complementosMatematicas.setNombre("Complementos de Matematicas");
complementosMatematicas.setCreditos(9);

calculuDiferencial.setClave("1112028");
calculuDiferencial.setNombre("Calculo Diferencial");
calculuDiferencial.setCreditos(9);

calculoIntegral.setClave("1112029");
calculoIntegral.setNombre("Calculo Integral");
calculoIntegral.setCreditos(9);

tallerMatematicas.agregar(introduccionAlCalculo);
tallerMatematicas.agregar(complementosMatematicas);
introduccionAlCalculo.agregar(calculuDiferencial);

calculuDiferencial.agregar(calculoIntegral);

//tallerMatematicas.mostrar(0);
}
}
```