**Abstract Factory**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Adapter**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Builder**

lassification :

Intention en théorie :

Separe le constructeur de ses objets complexes.

Si un objet Maison possède en attribut un String nom, un int id et un objet pièce alors le builder va s’occuper de construire tous les éléments objets de la Maison

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Bridge**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Command**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Composite**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Decorator**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Facade**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Factory Method**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Flyweight**

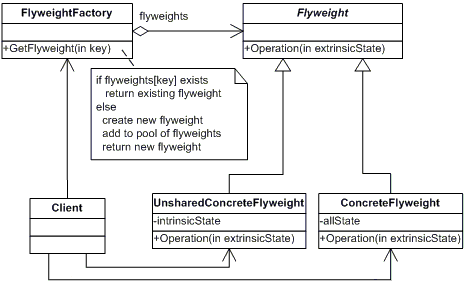
Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Iterator**

Classification :

Intention en théorie :

On veut parcourir des listes sujettes à modifications (pendant le parcours de cette même liste)

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Mediator**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



**Memento**

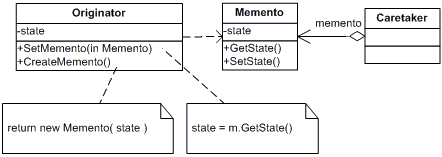
Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Observer**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Prototype**

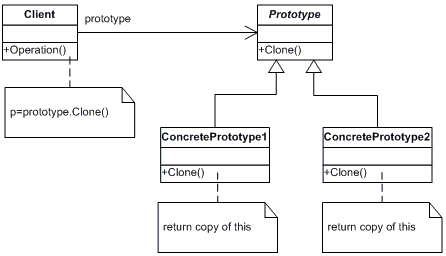
Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Proxy**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Strategy**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Singleton**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Template Method**

Classification :

Intention en théorie :

Intention dans le programme :

Interactions entre les composants :

Autre exemple :



Différence avec Forme canonique :

**Anti Patterns**