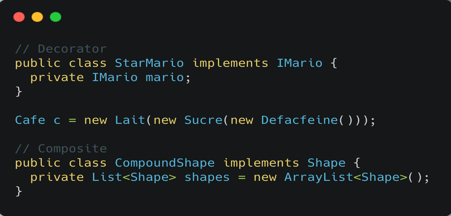
**Decorator** :

* Si une classe implémente une interface et un attribut de cette même interface.
* Ou si une instanciation se fait via un chainage de new (mais ne peut pas se terminer par null).

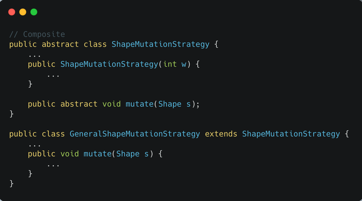
**Facade :** Si le projet contient plusieurs classes et que le main ne fait appel qu’à une seule class.

**Iterator :** Mot clés hasNext() et next() dans la classe.

A screen shot of a computer code

Description automatically generated**Composite :**

* Classe abstraite ou interface avec des méthodes communes utilisées par toutes les classes dérivées.
* Les classes dérivées effectuent des opérations simples, sans contenir d’autres objets.
* La classe principale de toutes les composites contient une liste des composites sur lesquelles sont utilisés les méthodes communes.

**Observer :**

* Une classe principale qui contient une liste d’instances d’une interface ou d’une classe abstraite commune. Et une méthode pour ajouter ou supprimer à cette liste. Et une méthode pour notifier tous les observateurs en appelant une méthode commune.
* Les observateurs dans la listes doivent implémenter ou hériter de la classe abstraite qui contient une méthode type update().

A computer screen shot of a code

Description automatically generated**Strategy :**

* Si une classe a un attribut dont le type est une interface ou abstract
* Si cette interface définit une méthode utilisée dans la classe
* Si la méthode de l’interface est dans une autre classe
* A screen shot of a computer program

  Description automatically generatedEt que l’interface ne possède qu’une seule méthode

A screenshot of a computer code

AI-generated content may be incorrect.**Prototype :**

* Si interface/abstract qui contient une méthode de clonage
* Et si une classe extends/implements cette interface et la méthode

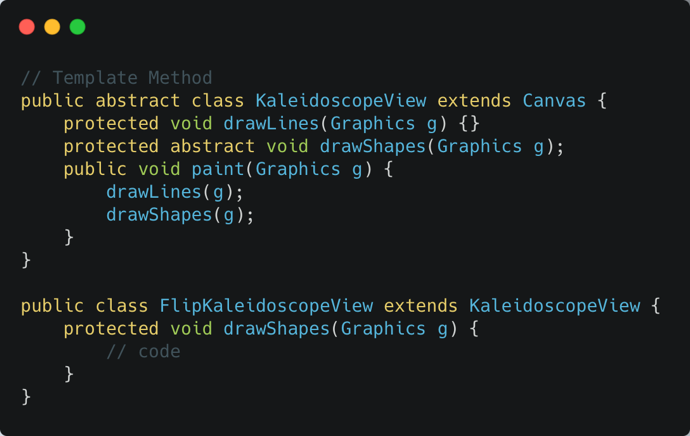
**AbstractFactory :**

* Si interface/abstract donne méthode pour créer des objets AF
* Si classe extends/implements l’interface et la méthode
* Si les objets créer partage une interface/abstract
* Si autre classe utilise l’interface AF pour créer des objets

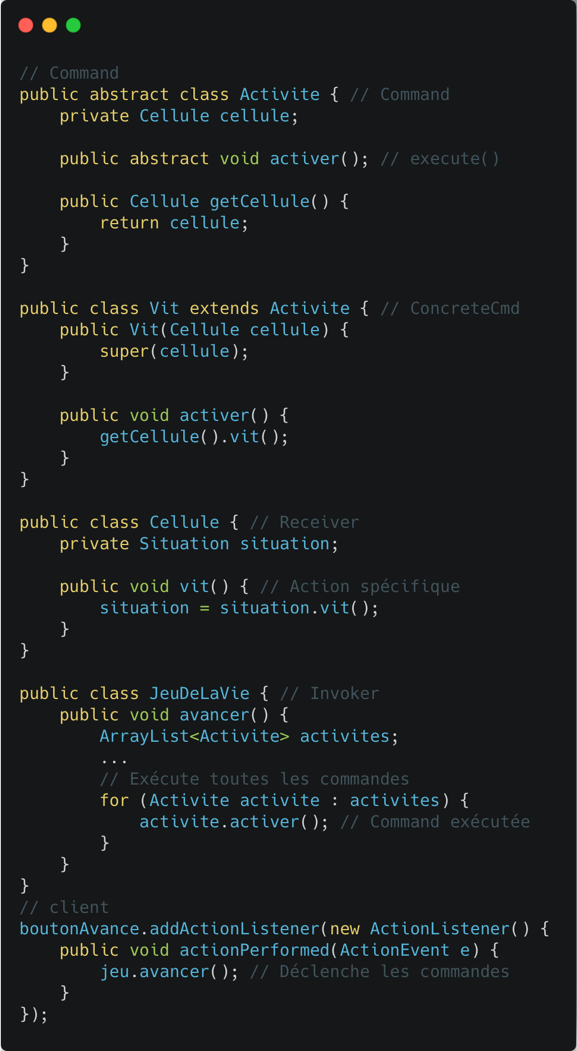
A screen shot of a computer program

AI-generated content may be incorrect.**A screenshot of a computer program

AI-generated content may be incorrect.**T**emplate Method :**

* Si abstract a méthode concrète
* Si la méthode appelle des méthode concrète et abstraite
* Si sous-classes impl. méthodes abstraite

**A screen shot of a computer program

AI-generated content may be incorrect.Builder :**

* Si une classe interne ou distincte contient de setter pour un objet
* Si setter return l’instance du Builder
* Si Builder à une méthode de build()
* Si produit final est une instance complexe configurée vie le Builder

**Command :**

* Si abstraite/interface possède execute() ou similaire
* Si classe impl. cette méthode
* Si autre classe effectue l’action en réponse
* Si autre classe invoke execute()
* Si Client passe par Invoker pour execute()

**Visitor :**

* Si abs/interface possède accept()/demande() ou similaire
* Si classes concrètes appellent des méthode d’un Visitor dans la méthode impl. de l’inter/abs
* A screen shot of a computer program

  AI-generated content may be incorrect.A screen shot of a computer program

  AI-generated content may be incorrect.Si interface Visitor définit une méthode pour chaque type d’elt à visiter
* Si classes concrète de Visitor impl méthodes de IVisitor
* Si Client parcours un Visitor

**State :**

* Si une classe (Context) contient un état
* Si les modifications d’état sont gérées par des methodes d’une autre classe
* Si inter/abs. (State) Défini des comportements communs aux états
* Si classes concrètes (ConcreteState) représentes les états possibles