

# Conception – Interface *(vaisseau et inventaire)*

---

## Affichage général : Jpanel

---

### Objectif

- Affichage général ;
- Mise en forme.

### Documentation :

- OpenClassroom : <https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/23108-creez-votre-premiere-fenetre>
- Tuto Oracle : <https://docs.oracle.com/javase/tutorial/uiswing/components/panel.html>
- Javadoc : <https://docs.oracle.com/javase/7/docs/api/javax/swing/JPanel.html>

### Implémentation :

Ex : Générer la fenêtre de l'application

```
import javax.swing.JFrame;

public class Test {
    public static void main(String[] args){

        JFrame fenetre = new JFrame();

        //Définit un titre pour notre fenêtre
        fenetre.setTitle("Ma première fenêtre Java");

        // Définit sa taille : 400 pixels de large et 100 pixels
        de haut
        fenetre.setSize(400, 100);

        // Nous demandons maintenant à notre objet de se
        positionner au centre
        fenetre.setLocationRelativeTo(null);

        // Termine le processus lorsqu'on clique sur la croix
        rouge
        fenetre.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Et enfin, la rendre visible
        fenetre.setVisible(true);
    }
}
```

## Organisation du projet / conception générale : M.V.C.

---

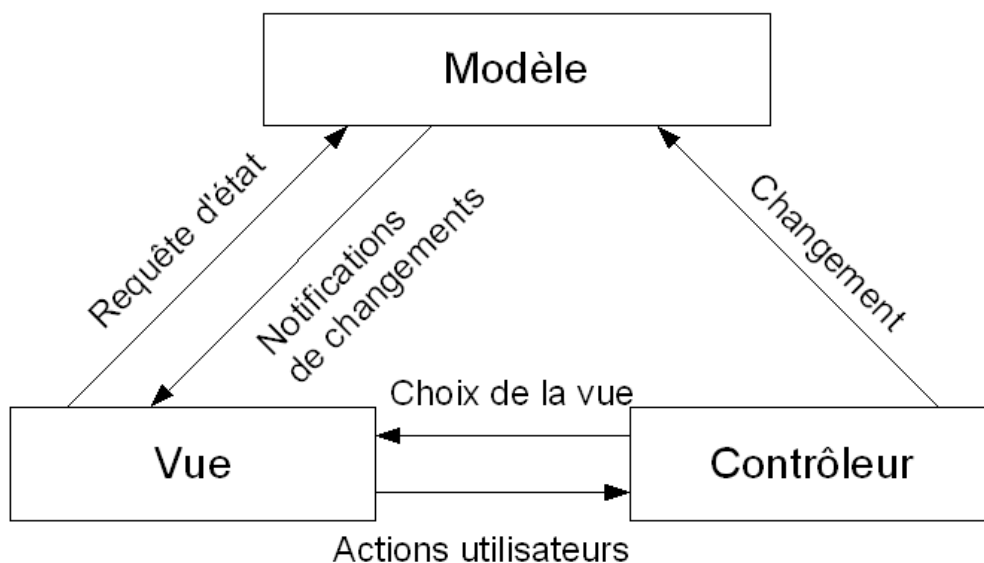
### Objectifs :

- Structurer le projet ;
- Faciliter l'implémentation et les échanges modèles-interface ;
- Faciliter le partage du travail : modèle, vue, contrôleur.

### Documentation :

- OpenClassroom : <https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/25552-structurez-mieux-votre-code-avec-le-pattern-mvc>
- <https://baptiste-wicht.developpez.com/tutoriels/conception/mvc/>

### Implémentation :



Le pattern MVC est un pattern composé du pattern observer et du pattern strategy.

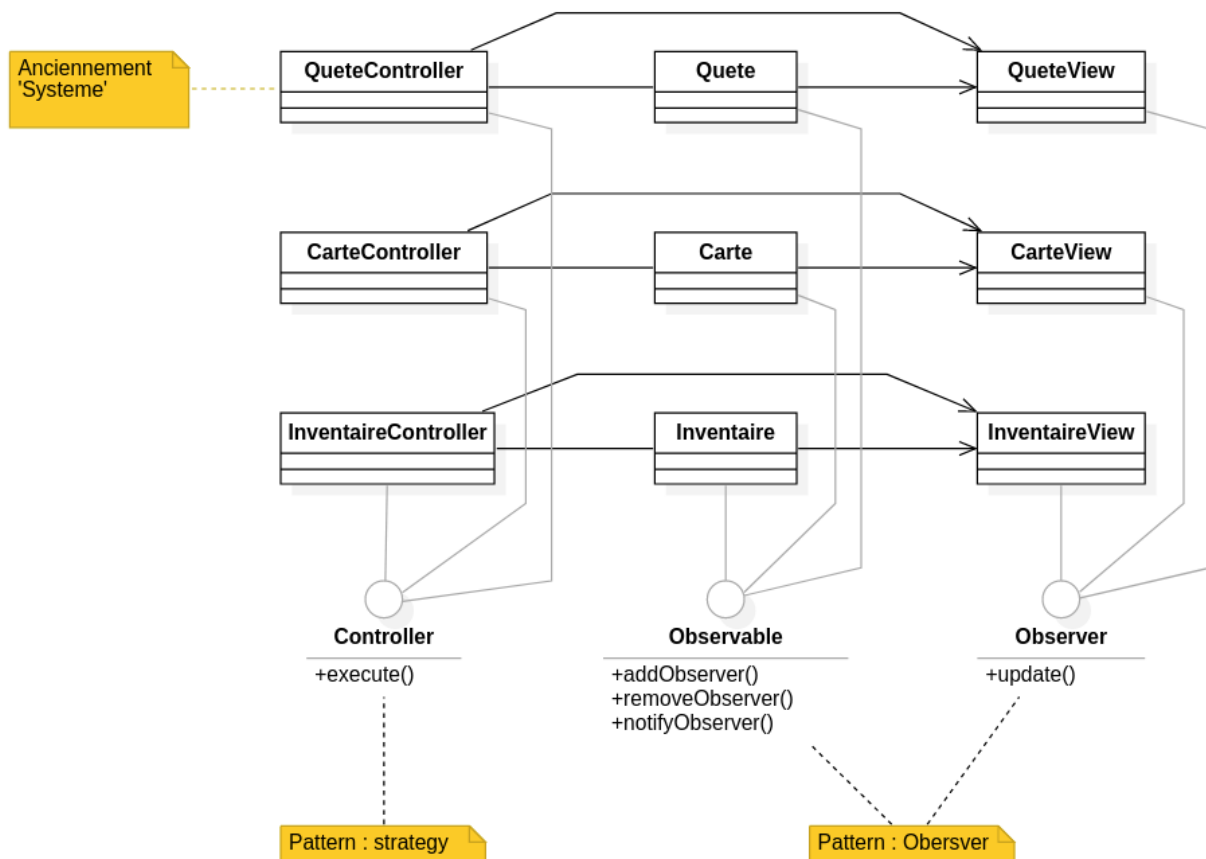
Avec ce pattern, le code est découpé en trois parties logiques qui communiquent entre elles :

- Le modèle (données)
- La vue (fenêtre)
- Le contrôleur qui lie les deux.

L'implémentation du pattern observer permet au modèle de tenir informés ses observateurs.

L'implémentation du pattern strategy permet à la vue d'avoir des contrôles différents.

Utiliser ce pattern permet de découpler trois acteurs d'une application, ce qui permet plus de souplesse et une maintenance plus aisée du code.



\*

## Mise en forme : Jpanel et JavaFX

### Objectifs :

- Structurer l'affichage et paramétrer l'interface ;
- Styliser l'interface.

### Documentation :

- OpenClassroom : <https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/5014486-decouvrez-java-fx>
- SceneBuilder (logiciel d'édition d'interface) : <https://gluonhq.com/products/scene-builder/>

**Implémentation :** Via fichiers css.