

## DÉPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

# Test Blanc COO et Design Patterns

**Filière :**

« Génie du Logiciel et des Systèmes Informatiques Distribués »

**GLSID**

## Compte rendu Design Patterns

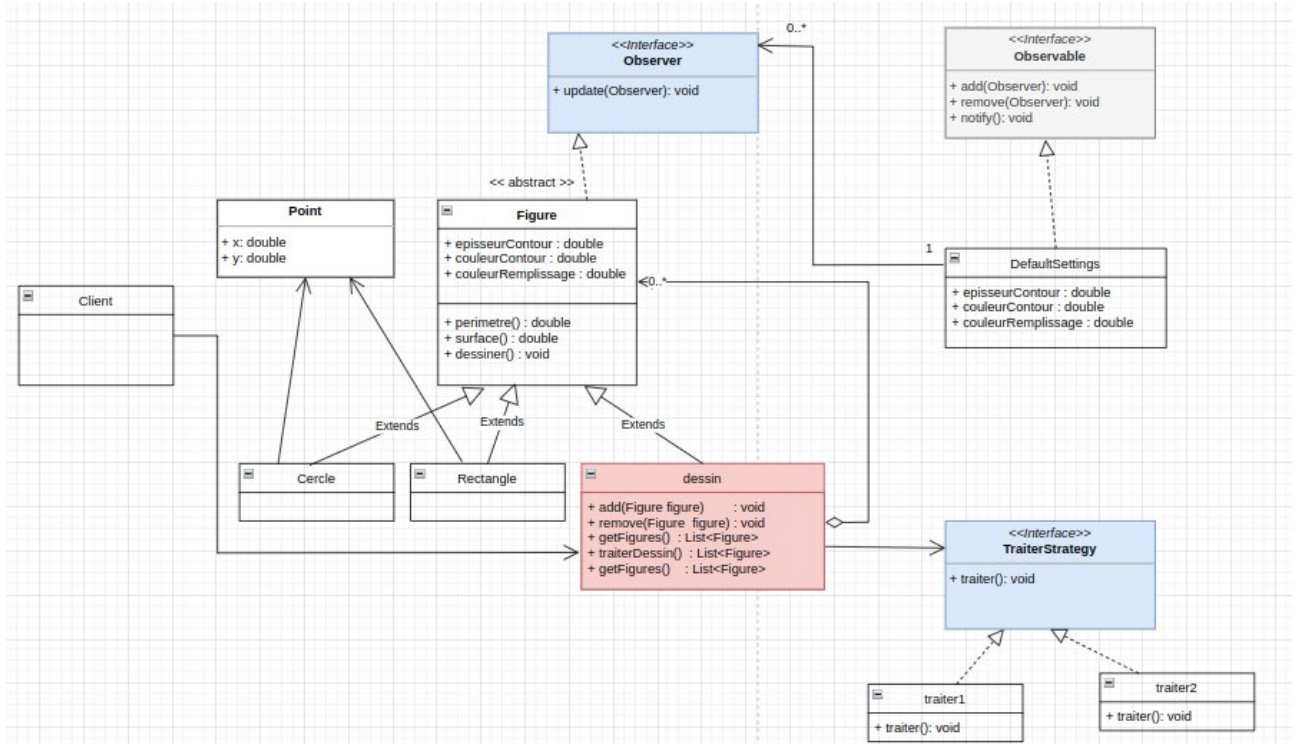
Réalisé par :

RAMLI Abdelmajid

**Année Universitaire : 2022-2023**

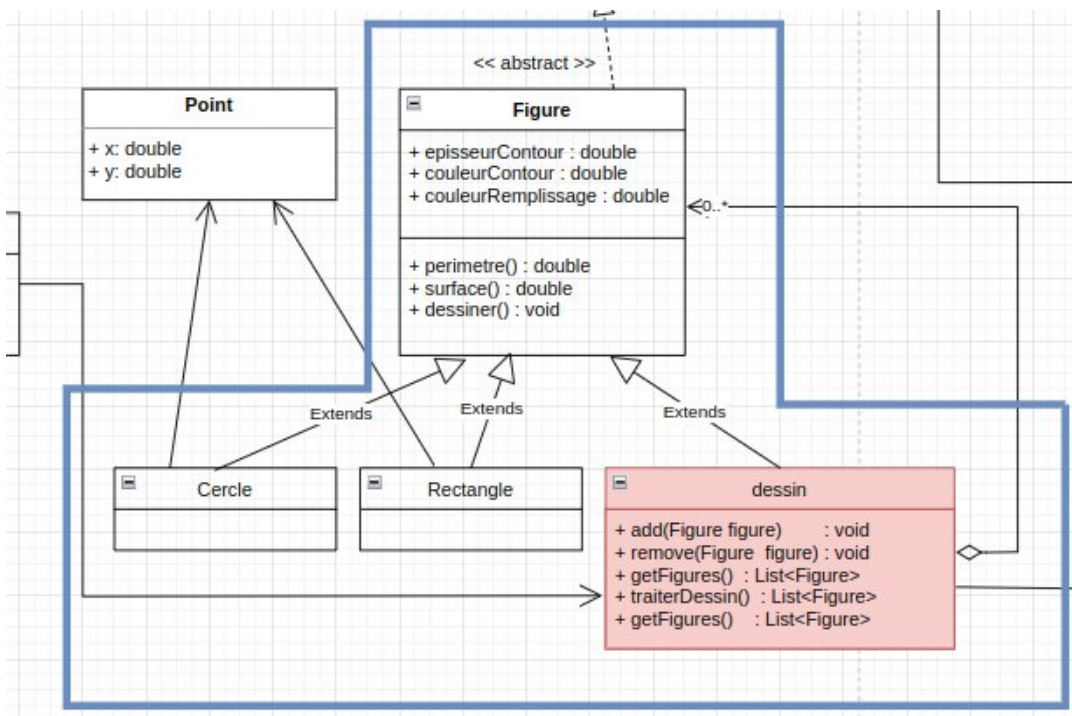
1. Etablir un Diagramme de classe du modèle en appliquant les design patterns appropriés en justifiant les designs patterns appliqués :

Diagramme :



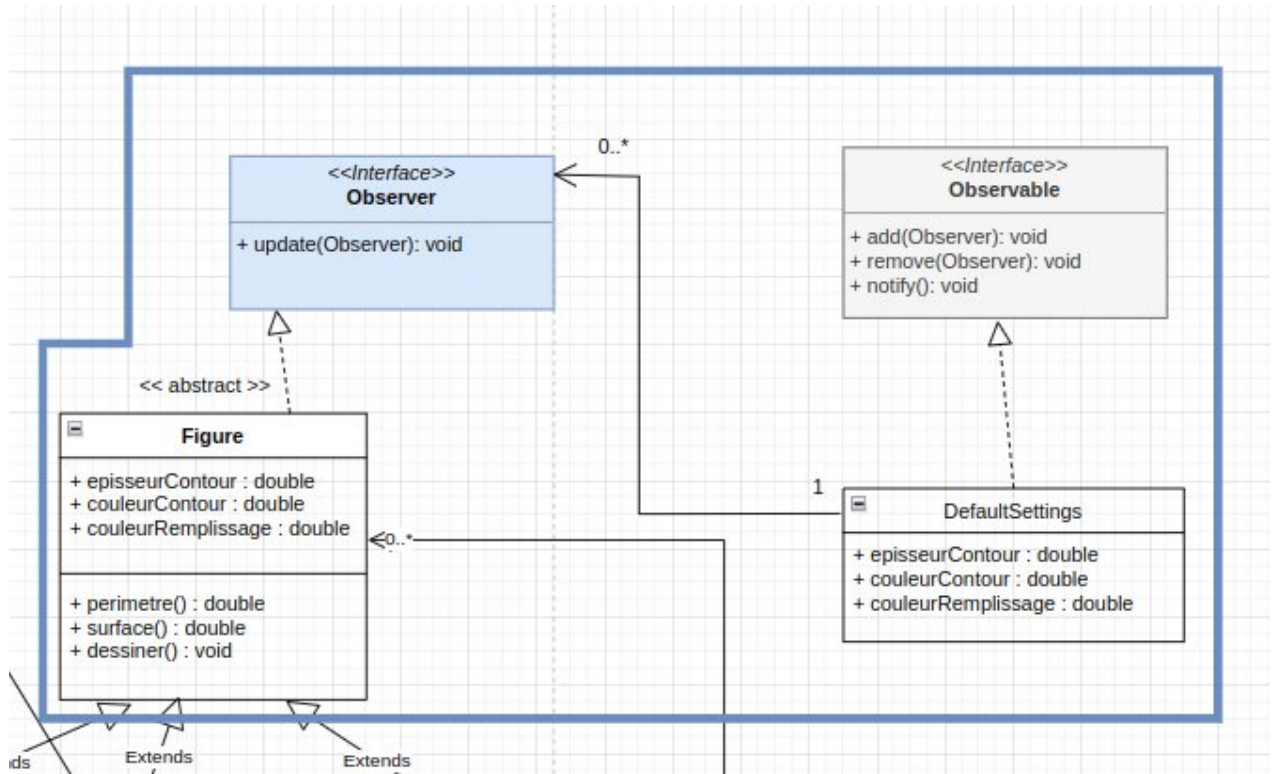
Pattern composite :

Un dessin peut avoir plusieurs figures ou dessins . Un figure est un cercle ou un rectangle :



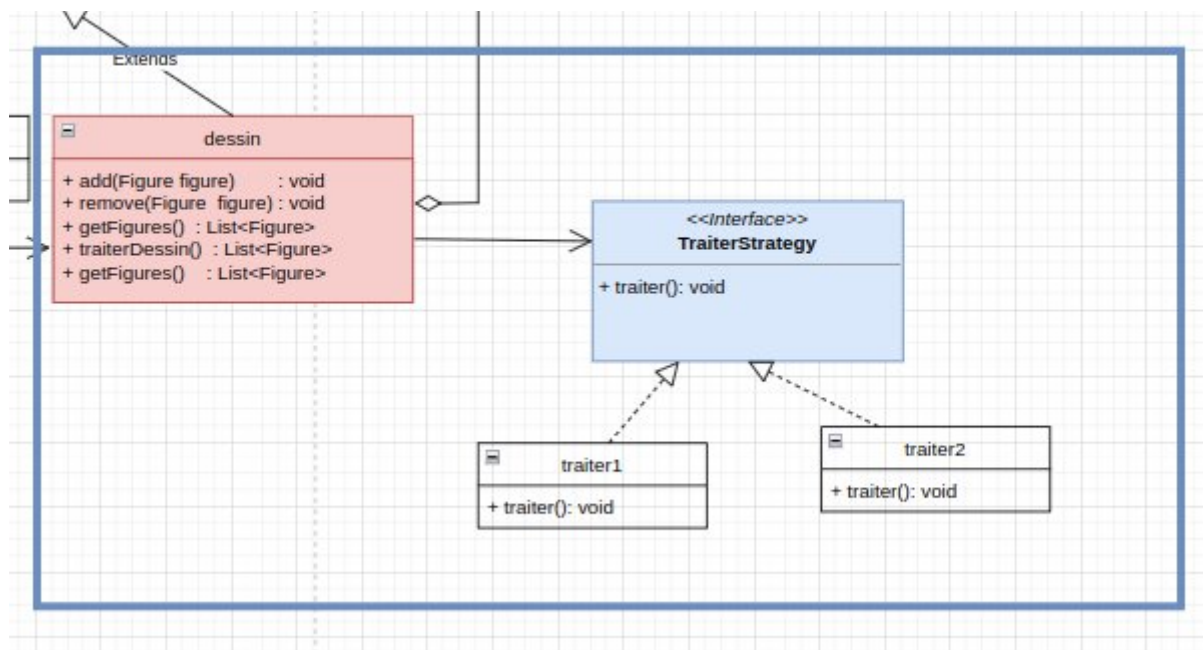
## Pattern Observer :

Un figure a une configuration par défaut. Lorsque cette configuration change tous les figures doivent être notifiées :

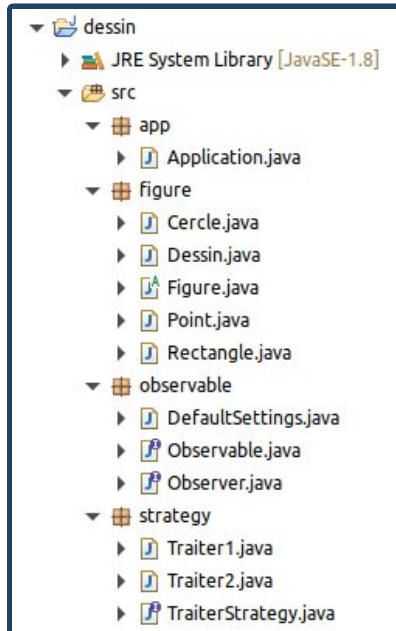


## Pattern strategy :

La méthode traiter peut évoluer. Et peut avoir plusieurs implémentations :



2. Faire une implémentation du modèle en utilisant un projet Maven sans prendre en considération des aspects techniques :



```
public static void main(String[] args) {
    Observable settings=new DefaultSettings();
    Dessin dessin=new Dessin();
    Figure cercle=new Cercle(new Point(0,0),2);
    Figure rectangle=new Rectangle(new Point(1,1),5,2);
    settings.add(cercle);
    settings.add(rectangle);

    dessin.add(cercle);
    dessin.add(rectangle);

    cercle.dessiner();
    rectangle.dessiner();

    dessin.setTraiterStrategy(new Traiter1());
    dessin.traiterDessin();
    dessin.setTraiterStrategy(new Traiter2());
    dessin.traiterDessin();

    System.out.println("contour : "+((DefaultSettings)settings).getCouleurContour());
    System.out.println("remplissage : "+((DefaultSettings)settings).getCouleurRemplissage());
    ((DefaultSettings)settings).setCouleurContour("grey");
    ((DefaultSettings)settings).setCouleurRemplissage("violet");
    System.out.println("contour : "+((DefaultSettings)settings).getCouleurContour());
    System.out.println("remplissage : "+((DefaultSettings)settings).getCouleurRemplissage());

    try {
        dessin.export();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Résultat :

```
cercle : centre : x : 0.0 y : 0.0 rayon : 2.0
rectangle : coin : x : 1.0 y : 1.0 l : 5.0 h : 2.0
traitement 1 effectue
traitement 2 effectue
contour : black
remplissage : bleu
contour : grey
remplissage : violet
dessin enregistrer
```