

# Compte-rendu du TP : Logiciel de dessin

*fait par Claire LARNAUDIE*

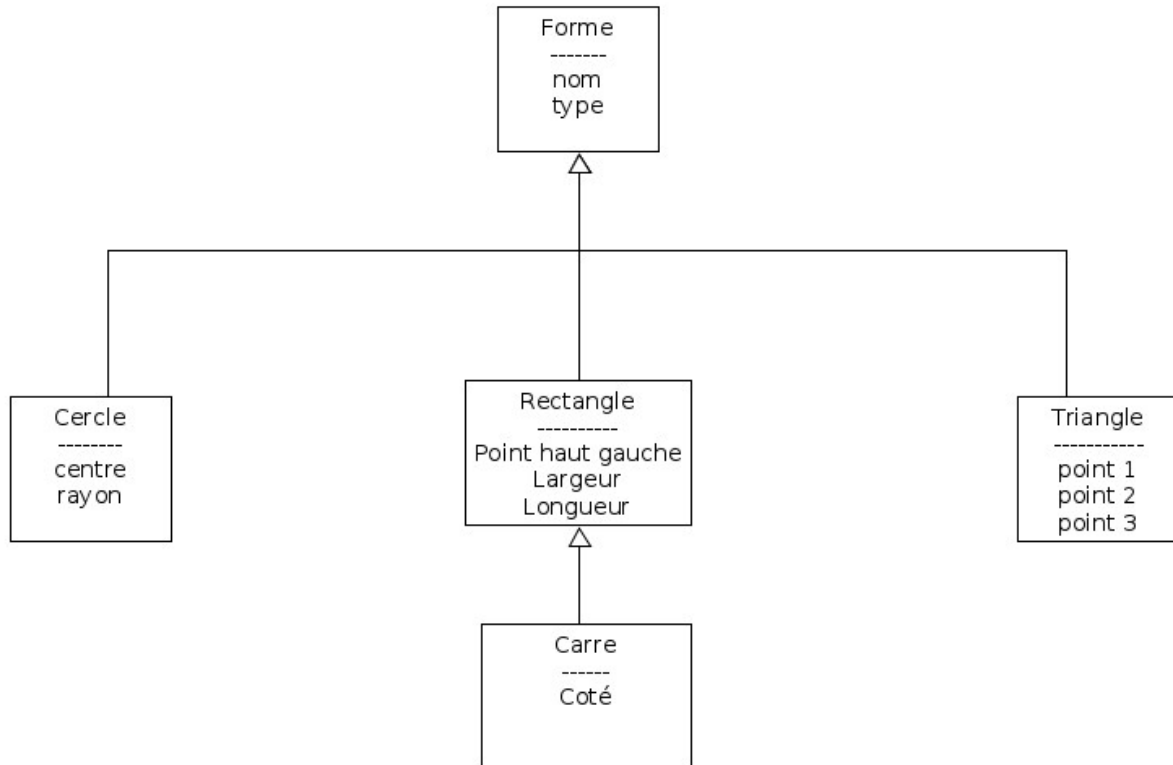
Les schémas UML ont été réalisés avec l'extension UMLet d'Eclipse.

Le projet a été réalisé avec Eclipse et Maven.

Les classes Java ont été vérifiées avec SonarLint.

## 1/ Formes graphiques

Le schéma UML des formes est le suivant.

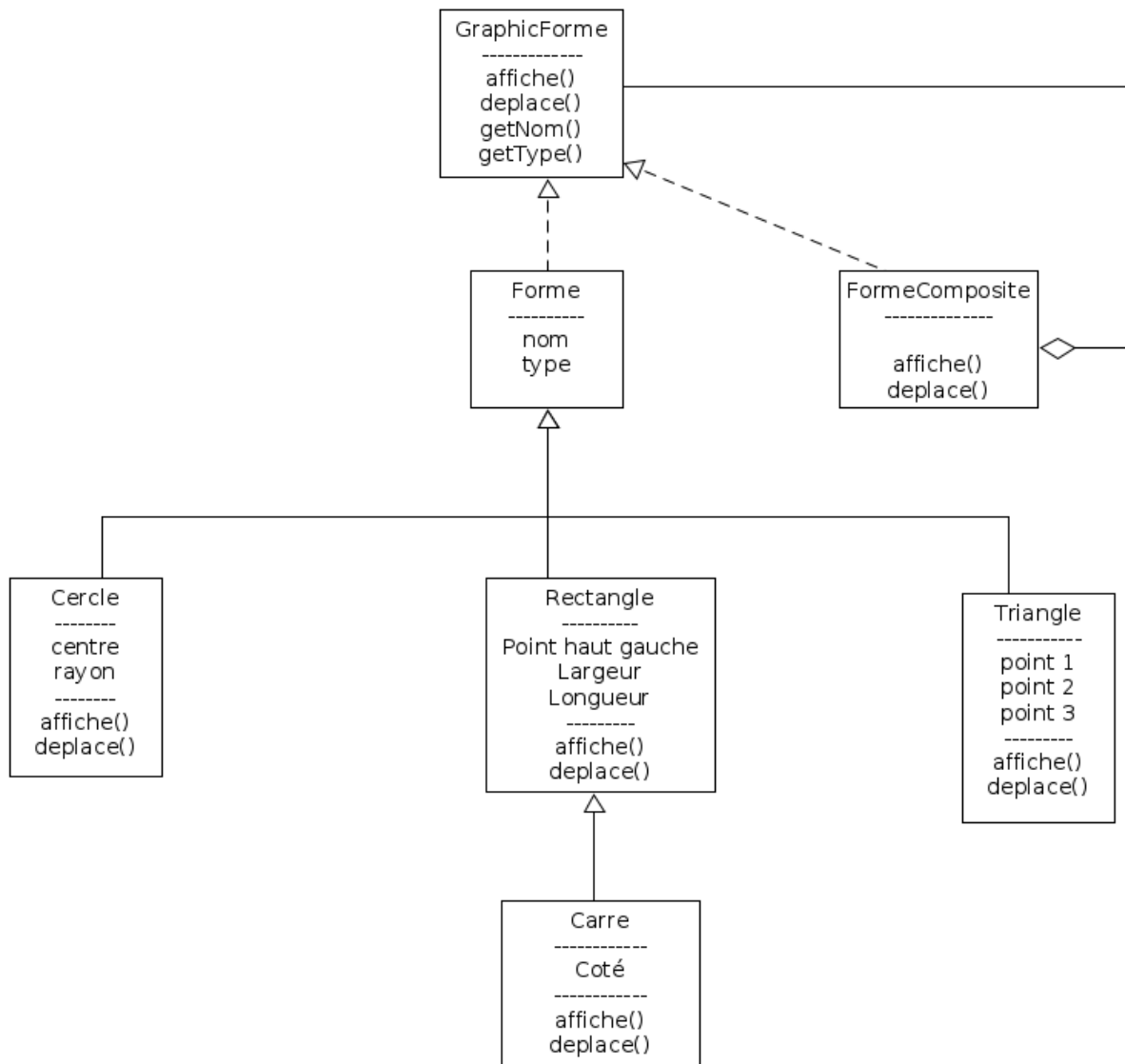


Les classes Java : **Forme**, **Carre**, **Cercle**, **Rectangle** et **Triangle** sont définies dans le package `main.java.forme`.

Les classes Java **Forme**, **Carre**, **Cercle**, **Rectangle** et **Triangle** sont définies dans le package `main.java.forme`

## 2/ Groupe de formes avec le pattern Composite

En appliquant le pattern Composite, le schéma UML du groupe de forme est le suivant.



Les classes Java : `GraphicForme` et `FormeComposite` sont définies dans le package `main.java.forme`.

`GraphicForme` est une interface contenant les méthodes `affiche()`, `deplace()`.

La classe `FormeComposite` (contenant les formes) :

- implémente l'interface `GraphicForme`.
- contient une liste de `GraphicForme`.

La classe `Forme` est déclarée abstraite et implémente l'interface `GraphicForme`.

Les méthodes de l'interface `GraphicForme` sont définies au niveau des classes `Cercle`, `Carre`, `Rectangle` et `Triangle`.

### 3/ Formes et groupes persistant avec le pattern DAO et JDBC

La base de données est Derby.

Les tables sont créées en utilisant l'utilitaire ij de Derby.

Pour passer du modèle UML à la définition des tables, on utilise les principes vus en cours de Conception et Base de données.

5 tables sont créées :

CARRE contient les formes carrées

CERCLE contient les formes carrées

RECTANGLE contient les formes carrées

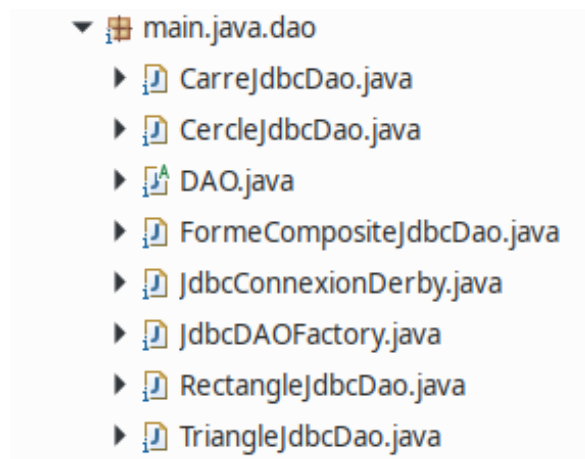
TRIANGLE contient les formes carrées

DESSIN pour regrouper les formes

La classe DAO abstraite contient les méthodes CRUD.

Pour chaque table, une classe héritant de DAO implémente les méthodes CRUD.

Les classes suivantes sont définies dans le package main.java.dao.



### 4/ Classe DrawingTUI

Cette classe contient les méthodes :

nextCommand() pour analyser la commande de l'utilisateur

afficheDessin() pour afficher les caractéristiques du dessin

On utilise le patron de conception SINGLETON afin qu'il y ait une instance unique de classe

### 5/ Modèle de conception Command

L'interface Command contient la méthode execute.

Les classes implémentant l'interface Command sont :

AfficheDessinCommand : Commande pour afficher le dessin

CreateFormeCommand : Commande pour ajouter une forme au dessin

LoadDessinCommand : Commande pour charger le dessin depuis la base de données

MoveFormeCommand : Commande pour déplacer une forme

SaveDessinCommand : Commande pour sauvegarder le dessin

Ces classes sont dans le répertoire main.java.command.

## 6/ Classe DrawingApp

C'est la classe principale. Elle interagit avec DrawingTUI pour récupérer la prochaine commande et l'exécuter.

## 7/ Tests

Les tests des classes ont été faits avec JunitTest.



### Résultats du test avec TestFormeCompositeJdbcDao

```
Tests de creation en base
Creation et recherche OK
FormeComposite 1
CERCLE : Cercle 1 Centre = (50,60) rayon = 10
CARRE : Carre 1 Point Haut Gauche = (30 40) cote = 10
RECTANGLE : Rectangle 1 Point Haut Gauche = ( 70 80) Largeur = 5 Longueur = 40
TRIANGLE: Triangle 1 Point 1 = (20 30) Point 2 = (40 60) Point 3 = (30 70)

Tests de suppression
Suppression et recherche OK

Tests de mises a jour
Mise à jour d'un élément et recherche OK
FormeComposite 1
CERCLE : Cercle 1 Centre = (60,75) rayon = 10
CARRE : Carre 1 Point Haut Gauche = (40 55) cote = 10
RECTANGLE : Rectangle 1 Point Haut Gauche = ( 80 95) Largeur = 5 Longueur = 40
TRIANGLE: Triangle 1 Point 1 = (30 45) Point 2 = (50 75) Point 3 = (40 85)
```

## 8/ Exécution du programme

Le scénario est le suivant :

### Étape 1

- Lancement de l'application
- Le nom du dessin est dessinCl
- Ajout d'un triangle, d'un carré, d'un cercle, d'un rectangle
- Affichage du dessin
- Sauvegarde du dessin
- Arrêt de l'application

### Étape 2

- Lancement de l'application
- Chargement du dessin : dessinCl (il doit correspondre à celui sauvegardé)
- Affichage du dessin
- Déplacement du cercle

Ci-dessous le résultat du scénario avec en vert les commandes de l'utilisateur.

```
Saisissez le nom du dessin
dessinCl
Pour creer une forme : nom de la forme, Type de forme, attributs de la forme
Attributs : Cercle : centreX centreY rayon Triangle : xPoint1 yPoint1 xPoint2 yPoint2 xPoint3 yPoint3
           Carre : xPointHautGauche yPointHautGauche cote Rectangle : xPointHautGauche yPointHautGauche largeur longueur
Pour deplacer une forme : move, nom de la forme, deplacement en x, deplacement en y
Pour charger un dessin : load nomDuDessin
Pour sauvegarder le dessin : save
tr TRIANGLE 10 20 15 25 30 45
TRIANGLE: tr Point 1 = (10 20) Point 2 = (15 25) Point 3 = (30 45)
cr CARRE 45 65 10
CARRE : cr Point Haut Gauche = (45 65) cote = 10
ce CERCLE 12 25 30
CERCLE : ce Centre = (12,25) rayon = 30
re RECTANGLE 5 10 20 30
RECTANGLE : re Point Haut Gauche = ( 5 10) Largeur = 20 Longueur = 30
affiche
dessinCl
TRIANGLE: tr Point 1 = (10 20) Point 2 = (15 25) Point 3 = (30 45)
CARRE : cr Point Haut Gauche = (45 65) cote = 10
CERCLE : ce Centre = (12,25) rayon = 30
RECTANGLE : re Point Haut Gauche = ( 5 10) Largeur = 20 Longueur = 30
save
Sauvegarde reussie
fin
Fin du programme
```

```
Saisissez le nom du dessin
autreDessin
Pour creer une forme : nom de la forme, Type de forme, attributs de la forme
Attributs : Cercle : centreX centreY rayon Triangle : xPoint1 yPoint1 xPoint2 yPoint2 xPoint3 yPoint3
           Carre : xPointHautGauche yPointHautGauche cote Rectangle : xPointHautGauche yPointHautGauche largeur longueur
Pour deplacer une forme : move, nom de la forme, deplacement en x, deplacement en y
Pour charger un dessin : load nomDuDessin
Pour sauvegarder le dessin : save
load dessinCl
dessinCl
CERCLE : ce Centre = (12,25) rayon = 30
CARRE : cr Point Haut Gauche = (45 65) cote = 10
RECTANGLE : re Point Haut Gauche = ( 5 10) Largeur = 20 Longueur = 30
TRIANGLE: tr Point 1 = (10 20) Point 2 = (15 25) Point 3 = (30 45)
move ce 4 3
CERCLE : ce Centre = (16,28) rayon = 30
affiche
dessinCl
CERCLE : ce Centre = (16,28) rayon = 30
CARRE : cr Point Haut Gauche = (45 65) cote = 10
RECTANGLE : re Point Haut Gauche = ( 5 10) Largeur = 20 Longueur = 30
TRIANGLE: tr Point 1 = (10 20) Point 2 = (15 25) Point 3 = (30 45)
fin
Fin du programme
```