# 

***Noms :***

***Bouzidi asma***

***Belkadi hayet***

***TP architecture des logiciel***

# Partie01 :

# 03- le rôle de chacune de ses classes :

Cercle : créer et afficher un cercle avec un rayon et un centre fixe.

Cercle Début : crée un début pour le train et le faire déplacer.

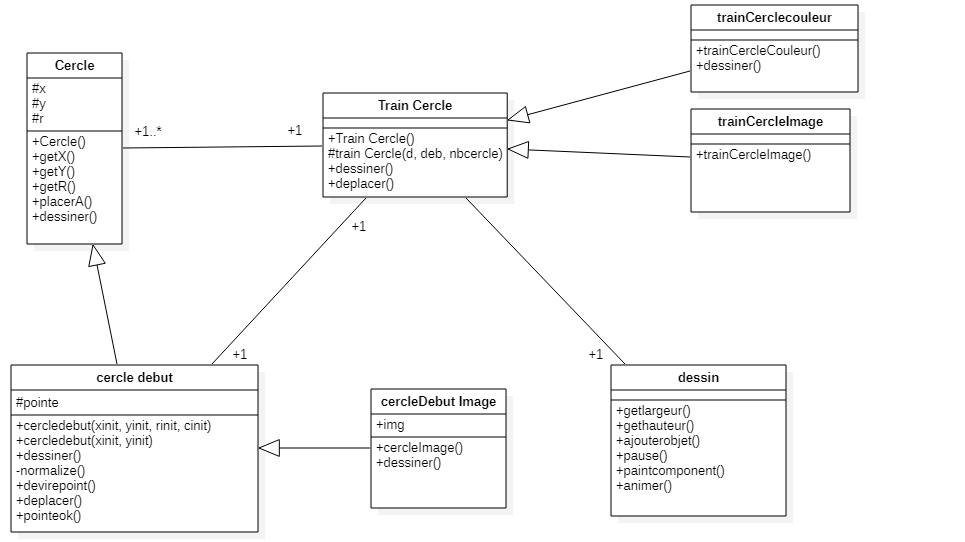
CrcleDebutImage : afficher l’image du début de cercle

Dessin : responsable de l’affichage et l’animation des trains cercle, effectuer le dessin et définir la zone de dessin

Train Cercle : crée et afficher un train de cercle, composé de plusieurs cercle, en précisant l’endroit où il va être déplacé et le nombre de cercle qui le compose

TrainCercleCouleur : affecter une couleur aux trains

# 04- la structure du code

****

# Partie02 :

# 5- la solution proposé est celle-ci :

Solution 1 : L’interface IObjetDessinable contient la méthode public void dessiner() , et elle doit être implémenté par la classe visage ,on ajoute une liste de visage dans la classe Dessin ,et on doit ajouter aussi une classe ajouterObjetVisage() a la classe dessin , qu’elle va permettre d’ajouter un objet Visage .

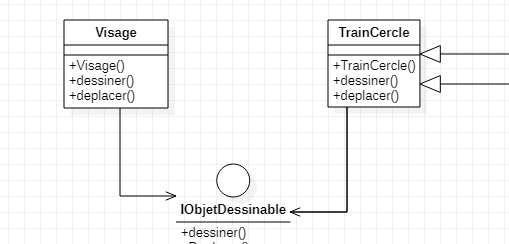
Solution 2 : on mettent la classe TrainCercle implémente l’interface IObjetDessinable pour pouvoir utiliser la la méthode ajouterObjetVisage() pour les deux types Visage et TrainCercle. Et pour cela on doit changer le type de variable ListeTrain de TrainCercle a un objet IObjetDessinable.

**private** **final** List<IObjetDessinable> listeTrain = **new** CopyOnWriteArrayList();

Et pour les deux solutions on doit ajouter à la classe main un constructeur de la classe Visage ,et appeler la méthode ajouterObjetVisage () pour ajouter un objet.

R ! : on a opté pour la 2eme solution

# 6-modelisation de la solution :



7 - Problème : après avoir apporter toutes les modifications aux classes application java, Dessin on trouve un problème dans l’animation du l’objet visage , a cause du la méthode déplacer() appelé par la méthode animer() , dans le quelle elle et défini dans les deux class TrainCercle et Visage

8- Solution : On ajoute une méthode abstract deplacer() dans l’interface IObjetDessinable et changer la méthode animer() de la classe dessin comme suit :

**public** **void** animer() {

// changer le type de variable c

**for** (IObjetDessinable c : listeTrain) {

c.deplacer();

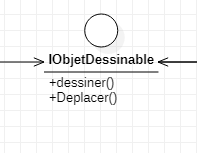
}

}

Et faire une appelle a la méthode deplacer() dans la boucles while de main

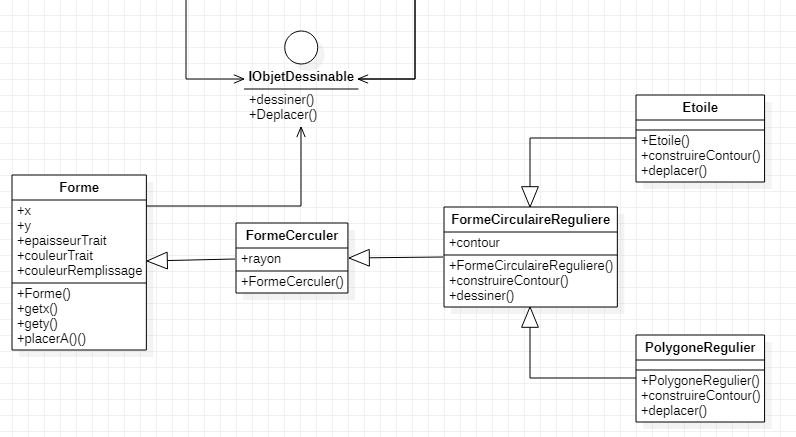
v.deplacer();

# 9- implémentation de solution :



# Partie 03 :

# 12-la structure des nouvelles classes :



# Partie 04 :

# 14-le problème :

# Partie 05 :

# 15- le rôle des classes :

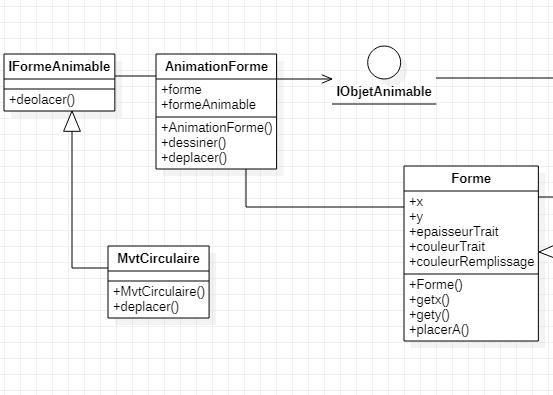
Forme : responsable de la créatine des formes on définissent la position (x,y), l’épaisseur de Trait, la couleur de conteur et la couleur de rempliage de la forme

AnimationForme : responsable de déplacement des formes

IFormeAnimable : généraliser l’animation (déplacement) des formes

MvtCirculaire : définir un type de mouvement circulaire

# 16- diagramme de class :



# 17- nouveau mouvement :

Vu que la classe AnimationForme a besoin de deux methode deplacer() et dessiner() pour ce compiler correctement et elle est liée a une interface IObjetAnimable et donc on liée l’interface IObjetAnimable avec l’interface IObjetDessinable

Et On faisant une appelle au constructeur de classe IObjetAnimable on définissent le premier paramètre comme une forma étoile et le deuxième un mouvement circulaire (MvtCirculaire) dans la classe main comme suit

d.ajouterObjet(**new** AnimationForme(

**new** Etoile(350, 100, 50, 8.f, Color.***yellow***, Color.***YELLOW***),

**new** MvtCirculaire(250, 250, 180, 0, 5)

));

On oura une étoile animable avec des mouvements circulaires.