

TP 3 Java Transformations Géométriques

La classe **Graphics2D** permet d'effectuer les transformations géométriques nécessaires (**Graphics2D.scale**, **translate**, **rotate (angle en radians)** ou **Graphics2D.shear (cx, cy))** qui peuvent aussi être réalisées avec la classe **AffineTransform** (on peut faire une série d'opérations géométriques avec la même **classe AffineTransform**).

Le code suivant qui **combine** une ellipse avec un rectangle introduit la **classe AffineTransform** pour réaliser une transformation géométrique:

```
Graphics2D g2 = (Graphics2D) g;  
Ellipse2D oval = new Ellipse2D.Float(100, 200, 80, 60);  
Rectangle2D rect = new Rectangle2D.Float(120, 230, 90, 80);  
Area forme = new Area(oval);  
forme.add(new Area(rect)); // ou forme.intersect(new Area(rect));  
g2.draw(forme);  
AffineTransform at = new AffineTransform();  
at.translate(100, 100); // Translation de (x=100, y=100)  
at.rotate (0.5); // Rotation d'un angle de 0,5 radians  
forme.transform(at); //appliquer la translation suivie de la rotation  
g2.draw(forme); // Dessiner la forme, ou utiliser g2.fill(forme)
```

Exercice 1 :

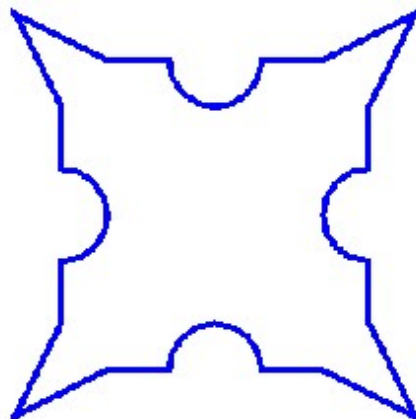
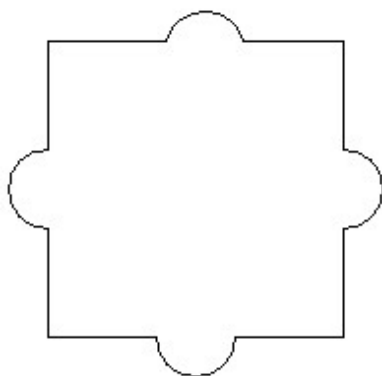
- 1) Faire un cisaillement sur le rectangle A qui commence au point (50, 70) avec largeur 50 et hauteur 80, de la valeur 2 dans la direction y, ensuite la valeur 2 dans la direction x.
- 2) Refaire un cisaillement sur le rectangle A avec les mêmes coordonnées de la valeur 2 dans la direction X, ensuite la valeur 2 dans la direction Y.
- 3) Refaire un cisaillement sur le rectangle A avec les mêmes coordonnées de la valeur 2 dans les 2 directions X et Y en même temps.

Utiliser `Rectangle2D r= new Rectangle2D.Double (x, y, width, height).`

En conclure si les résultats des 3 opérations sont les mêmes ?

Exercice 2 :

- 1) Appliquer une rotation du rectangle A qui commence au point (30, 40) avec largeur 50 et hauteur 80 d'un angle de 30 degrés (la rotation se fait autour de l'origine par défaut).
- 2) Appliquer une rotation sur le même rectangle A d'un angle de 30 degrés autour de son centre (point situé en plein milieu du rectangle).
- 3) Utiliser une boucle pour afficher un message 'Bienvenue au mode Graphique de Java' au milieu de la fenêtre de l'Applet avec une série de rotations horaires en changeant de couleur.
- 4) Utiliser une boucle pour effectuer séparément une série de rotations horaires des 2 formes ci-dessous d'un angle de 30 degrés tout en effectuant un changement de couleur.



Exercice 3 :

1) Appliquer le changement d'échelle du rectangle (30, 40, largeur=50, hauteur=80) avec les valeurs (1.5, 1.5) et (3, 3) séparément de sorte que :

- (a) La position de début de la forme agrandie coïncide avec le début de la forme initiale
- (b) La forme initiale (petite) soit située au milieu de la forme agrandie.

Pour les 2 cas (a) et (b), faire une translation avec des valeurs à calculer pour arriver à la position désirée.

2) Appliquer aussi séparément un changement d'échelle des 2 formes précédentes de l'exercice 2 avec les valeurs (1.5, 2) et (2.5, 2).

Exercice 4 :

Charger une image et l'afficher, ensuite:

1) faire une rotation de l'image autour de son centre d'un angle de 45 degrés. Se servir de l'exercice 2 pour la rotation.

(Utiliser les instructions suivantes: avec **`import java.awt.Image, java.awt.Toolkit`**
`image= Toolkit.getDefaultToolkit().getImage (fichier_image);` et **`Graphics2D.drawImage (image, x, y, largeur, hauteur, this);`**).

2) Faire un changement d'échelle sur l'image (valeur de 2 sur x, et 1.5 sur y)