

## TP 2 Java

Se servir des fonctions supplémentaires (de **Graphics g**):

**g.drawString ("chaine", x, y);** // affiche une chaine de caractères à la position (x , y)

**g.setFont (new Font ( "Arial", Font.BOLD , 16);** // "TimesRoman" . . ,  
Font.ITALIC,

*Utiliser aussi la fonction **getBounds (Rectangle r=getBounds());** ) pour connaître la largeur et la hauteur du JFrame en cours d'exécution.*

1) Afficher les nombres de 1 à 30 ainsi que leurs carrés et racines carrées, chaque nombre (avec le carré et la racine carrée) dans une ligne tout en utilisant un changement de couleur (dégradé de couleur) d'une ligne à la suivante, de façon à avoir l'affichage ajusté suivant:

<b>Nombre:= 1</b>	<b>Carré = 1</b>	<b>Racine carrée = 1</b>
<b>Nombre= 2</b>	<b>Carré = 4</b>	<b>Racine carrée = 1.414213562 ....</b>

2) **Afficher un texte animé "Bienvenue ou Welcome..."** qui change aléatoirement de position, de taille et de couleur toutes les fractions de secondes.

---

La librairie awt contient plusieurs classes qui permettent de faire le dessin graphique. La classe **Graphics2D** est une extension de la classe **Graphics** et permet d'effectuer des opérations et transformations géométriques supplémentaires (**Graphics2D.scale**, **translate**, **rotate (angle en radians)** ou **Graphics2D.shear ( cx, cy )** ).

Java.awt.geom.\* contient plusieurs classes relatives aux primitives géométriques ( comme: **Point2D**, **Rectangle2D**, **Ellipse2D**, **Line2D**, **Point2D**, **Area** et même **Path2D** ou **GeneralPath** –ces derniers permettent de tracer une forme graphique généralisée-).

La classe **Area** permet de combiner des formes géométriques simples pour avoir des formes plus complexes (**Area.add**, **intersect**, **subtract** et **exclusiveOr**).

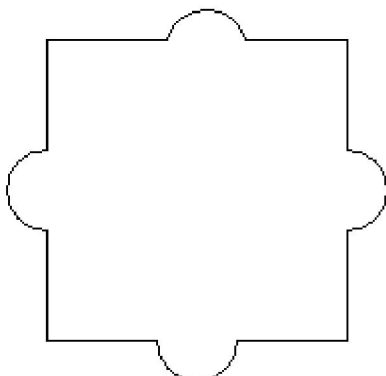
Le code suivant permet de combiner une ellipse avec un rectangle pour obtenir une forme plus complexe (dans la fonction **paint** ou une fonction générée **dessin**):

```
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);  
Écrire soit: forme.add(new Area(rect)); ou forme.intersect(new Area(rect));  
g2.draw(forme); ou g2.fill(forme);
```

Pour spécifier l'épaisseur de trait du dessin (ex: 3): **g2.setStroke(new BasicStroke(3.0f));**

3) **Ecrire le code java (avec Area ...) qui permet d'afficher les formes suivantes (carrés de longueur 150, avec des cercles ou ellipses sur les côtés):**

1)



2)

