

TP-TD de Programmation II n° 6

Vendredi 10 novembre - 13h30

Exercice 1 :

Nous vous fournissons pour ce TP des classes permettant de faire des dessins simples **Dessin** et **TestDessin**. Compilez et exécutez cette dernière, vous devriez observer un triangle noir.

1. Regardez le code de **TestDessin**. Observez comment sont définis les 3 points du triangle, et comment est appelé la méthode de dessin. Si vous souhaitez utiliser d'autres couleurs prédéfinies, vous pouvez regarder l'api de la classe **Color** où vous trouverez leurs noms. Cela vous sera utile pour distinguer des figures qui se chevauchent.
2. La classe **Figure** sera la classe mère dont hériteront nos cas particuliers. Celle ci contiendra les coordonnées du centre de la figure, des accesseurs pour repérer ce centre, et une méthode abstraite utilisée plus tard pour traduire une figure en un polygone.

Voici la trame de votre classe :

```
import java.awt.Polygon;
public abstract class Figure{
    // coordonnées du centre approximatif de la figure
    private int posX;
    private int posY;

    public Figure(int x, int y){
        posX = x;
        posY = y;
    }
    .....
}
```

Ajoutez les accesseurs concrets **public int getPosX()** et **public int getPosY()** qui respectivement donneront la position horizontale du centre de la figure (abscisse), et la position verticale du centre de la figure (ordonnée). Ajoutez également la déclaration de méthode abstraite **public abstract Polygon creePolygon();** qu'on ne peut définir davantage sur des figures pour le moment.

3. On définit une classe **Rectangle** comme cas particulier de **Figure**. Une figure possédant déjà une position centrale, il nous suffit de préciser quelle est la largeur et la hauteur. Pour rester simple, on choisit de toujours aligner la largeur avec la ligne horizontale. Ecrivez cette classe ainsi que la méthode **creePolygon()** qui lui est associée. Testez la.
4. Introduisez une classe **Carre** dans la hiérarchie de classe. Quel est d'entre les carrés, et les rectangles ceux qui ont le plus de caractéristique ? Ecrivez et testez la.
5. On considère une famille de triangle isocèles pour lesquels la base est toujours horizontale, et la hauteur verticale. Définissez un constructeur pour cette classe, utilisant les paramètres base et hauteur. Ecrivez la méthode **creePolygon()** correspondant à ces triangles.
6. On introduit une méthode **déforme(double dx, double dy)** qui déforme une figure selon l'axe des *x* et selon l'axe des *y*, en conservant le placement de son centre. Dans le cas d'un rectangle la nouvelle hauteur sera *hauteur * dy* et idem pour la largeur. Quel est le type retour de la méthode ?

7. On envisage d'intégrer d'autres types de figures : le cercle, et l'ovale. Ces figures peuvent elles avoir un lien avec les carrés et les rectangles ? Proposez encore d'autres figures.
8. Pour chacune des méthodes décrites ci-dessous demandez-vous si elle est abstraite ou non, et dans quelles classes il convient de la définir ou de la redéfinir. Ecrivez les et testez les.
 - `double estDistantDe(Figure fig)` qui calcule la distance entre le centre de deux figures.
 - `déplacement(int x, int y)` qui déplace une figure.
 - `double surface()` qui calcule la surface de la figure.
 - `boolean estPlusGrand(...)` qui compare les surfaces de deux figures.

Exercice 2 :

Présentation du projet