

*RAPPORT DE CONCEPTION  
PROJET FINAL- POO*

*ATALLA Salim & NISO Aram*  
*L2 Info - groupe 3870*

---

Nous avons à programmer un ensemble de classes et interfaces en Java qui dessinent des figures géométriques en rotation sur elles-mêmes tout en se déplaçant sur des chemins dans le plan cartésien et cela en les réalisant dans une fenêtre graphique.

Nous avons commencé le travail et choisi nos chemins et nos figures et voici l'avancement de notre travail. Vous allez y trouver les classes et interfaces principales de notre programme et ajoutant un diagramme à la fin de notre rapport.

## Classe «View»

- Rôle: Créer un 'JFrame' qui sera l'espace du travail.
- Super-classe: JFrame

- Attributs:

GUI gui	"Graphical User Interface", qui contient toute la partie graphique du programme.
---------	----------------------------------------------------------------------------------

- Méthodes:

View ( string title )	Constructeur qui sert à construire un view.
void ContentFrame()	Les composants du "View".

-----

# Classe «GUI»

- Rôle: C'est la partie graphique du programme.
- Super-classe: JPanel
- Attributs:

JLabel formLabel	C'est un string qui définit la formPanel.
JLabel pathLabel	C'est un string qui définit la pathPanel.
JRadioButton squareSpiralButton	Bouton pour choisir un chemin du type Square Spiral.
JRadioButton SpiralButton	Bouton pour choisir un chemin du type Spiral.
JRadioButton SquareButton	Bouton pour choisir un forme du type Square.
JRadioButton CircleButton	Bouton pour choisir un forme du type circle.
formGroup	Regroupe les radioButtons des formes.
ButtonGroup pathGroup	Regroupe les radioButtons des chemins.
JButton RunButton	Pour démarrer le programme.
Timer chrono	Sert de faire une pause entre chaque dessin.

- Méthodes:

GUI ()	Constructeur qui construit la partie graphique.
ContentPane()	Complément de constructeur.
menuPane()	Contient formPane et pathPane.
formPane()	Contient les composants qui concernent les formes.
pathPane()	Contient les composants qui concernent les chemins.
runPane()	Contient les composants qui concernent le bouton 'Run'.
void actionPerformed(ActionEvent e)	'Action Listener' pour les boutons du GUI.
void paintComponent(Graphics g)	C'est la partie du dessin.

---

## Interface «Form»

- Rôle: Responsable de définir les méthodes nécessaires de chaque forme.
- Méthode:

void translation(int dx, int dy)	Changer les coordonnées d'un objet à une valeur des x et des y donnés.
void setPosition(int x, int y)	Initialiser l'objet aux nouvelles coordonnées.
void drawForm(Graphics g)	dessiner un form après l'avoir créé.

---

## Interface «Path»

- Rôle: Responsable de définir les méthodes nécessaires de chaque chemin/trajectoire.
- Méthodes:

drawPathForCircle(Circle circle, Graphics g)	Méthode pour dessiner un cercle pour un chemin /trajectoire particulier.
void drawPathForSquare(Square square, Graphics g)	Méthode pour dessiner un carré pour un chemin /trajectoire particulier.

---

# Classe «Circle»

- Rôle: Pour définir et ajuster un cercle.
- Super-classe: Implements du l'interface 'Form'.
- Attributs:

Point center	Représente le centre d'un cercle.
int radius	Représente le rayon d'un cercle.

- Méthodes:

Circle(Point center, int radius)	Constructeur pour construire un cercle.
void translation(int dx, int dy)	ajouter x et y aux coordonnées (Valeurs negatives sont acceptées).
void setAtPosition(int x, int y)	Initialiser le cercle aux nouvelles coordonnées.
void drawForm(Graphics g)	Dessir la forme.
Point getCircleCenter()	Getter pour avoir le Centre.
int getCircleRadius()	Getter pour avoir le Radius.

-----

## Classe «Square»

- Rôle: Pour définir et ajuster un Carré.
- Super-classe: Implements du l'interface 'Form'.
- Attributs:

Point point	Représente l'angle haut_gauche d'un carré.
int side	Représente la côté d'un carré.

- Méthodes:

Square(Point point, int side)	Constructeur pour construire un carée.
void translation(int dx, int dy)	ajouter x et y aux cordonnées (Valeurs negatives sont acceptées).
void setAtPosition(int x, int y)	Initialiser le point en bas à gauche du carée aux nouvelles coordonnées.
void drawForm(Graphics g)	Dessir la forme.
getSquareSide()	Getter pour avoir le Side.
Point getSquarePoint()	Getter pour avoir Point.

-----

# Classe «squareSpiral»

- Rôle: Définir le chemin de squareSpiral.
- Super-classe: Implements du l'interface 'Path'.
- Attributs:

int x, y	Représentes le point du départ.
Int i	Un compteur.
int d	La distance entre les lignes du chemin.

- Méthodes:

squareSpiral()	Constructeur pour construire le chemin.
void drawPathForCircle(Circle circle, Graphics g)	Ajuster le chemin pour un cercle.
drawPathForSquare(Square square, Graphics g)	Ajuster le chemin pour un carée.

-----

# Classe Spiral

- Rôle: Une classe de type path, elle est responsable de faire un dessin de forme spirale.
- Super-classe: Implements du l'interface 'Path'.

- Attributs:

double r	C'est un double qui fait la différence du rayon, du circle de la spirale.
double Theta	Le double qui fait la différence du rayon à chaque tour.
int i	un compteur nécessaire pour changer Theta pour chaque tour, (si 'i' est négatif alors le chemin va se dessiner au côté opposé).
int x, y	ils sont des coordonnées temporaire, on les utilise pour mieux organiser le code.

- Méthodes:

Spiral()	Constructeur pour construire le chemin.
void drawPathForCircle(Circle circle, Graphics g)	Ajuster le chemin pour un cercle.
void drawPathForSquare(Square square, Graphics g)	Ajuster le chemin pour un carée.

-----

Après avoir montré toutes les classes et interfcas, voici le diagramme concernant aussi:



