Architecture logicielle

Projet “Dessin vectoriel”

Oleksandr AVLESI

**Rendu final**

Le but du projet consistait en réalisation dans un programme en Java d’un script qui permettrait de définir un dessin logique et dont interprétation donnerait la possibilité de représenter ce dessin à l’aide des différents outils graphiques. Le but de ce rapport c’est de clarifier une architecture du projet réalisé et expliquer quels étaient les raisons qui ont amené à l’implémentation qui a été faite.

Une classe principale d’une application s’appelle *Dessin*. Toutes les requêtes de la création des objets graphiques vont être adressées à cette classe. On peut la considérer comme une classe manager.

Les fonctions principales que cette classe fournit ce sont :

* draw(Chemin chemin) – permet de dessiner une figure
* label() – permet d’étiqueter une figure
* fill(Chemin chemin, Color color) – permet de remplir une figure d’un couleur
* insert(Chemin chemin, int x, int y, int width, int height) – permet de clipper une figure dans une zone carrée indiquée

*Chemin* c’est une classe des modèles des figures qui peuvent être dessinées. Pour créer des figures le client peut se servir des fonctions de Dessin suivantes :

* Chemin createPoint(int x, int y) – créer un point
* Chemin createLine(int x1, int y1, int x2, int y2) créer une ligne
* Chemin createArc(int center\_x, int center\_y, int rx, int ry, int startAngle, int arcAngle) créer une arc
* Chemin createEllipse(int center\_x, int center\_y, int rx, int ry) – créer un ellipse
* Chemin createCircle(int center\_x, int center\_y, int r) – créer une circle
* Chemin createRectangle(int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4) – créer un rectangle

Chaque figure peut avoir son propre couleur et ses paramètres de l’épaisseur. Ce paramétrage est fait à l’aide de la classe *Crayon*. A chaque figure on peut associer une instance de Crayon.

Pour créer un crayon, il faut se servir d’une fonction de *Dessin* correspondante :

* Crayon createCrayon(Color color, int thickness)

Pour associer un crayon avec une figure, il faut utiliser une fonction de Chemin

* setCrayon(Crayon crayon)

Si un crayon n’est pas associé explicitement, une figure va utiliser un crayon par défaut dont les paramètres c’est un couleur noir et une épaisseur d’une valeur 1.

Si un client veut obtenir une figure plus complexe, il peut se servir de fonctions de Chemin :

* Chemin connectWithLine(Chemin chemin) – fonction permet d’obtenir une nouvelle figure qui se compose des deux figures jointes par une ligne
* Chemin connectWithBezier(Chemin chemin, int x1, int y1, int x2, int y2) – fonction permet d’obtenir une nouvelle figure qui se compose des deux figures jointes par une courbe de Bézier. Ici x1, y1, x2, y2 ce sont des points de contrôle d’une courbe de Bézier cubique.

Initialement dans le projet il y a deux implémentations de la représentation des dessins vectoriels – celle qui est basée sur la librairie awt de java et celle basée sur svg. Pour choisir une représentation que vous voulez, il faut fournir un paramètre correspondant à un constructeur de la classe Dessin.

Voici un exemple d’utilisation des fonctionnalités décrites ci-dessus:

// créer un dessin et choisir une représentation svg

Dessin dessin = **new** Dessin(DrawType.***SVG***);

// pour choisir une représentation awt il faut faire DrawType.AWT

//créer un crayon

Crayon pen = d.createCrayon(Color.***BLUE***, 1);

//créer un cercle en precisant le center (800, 800) et un radius (100)

Chemin cercle = d.createCircle(800, 800, 100);

//associer un crayon à un cercle

cercle.setCrayon(pen);

// remplir un cercle d'un couleur jaune

d.fill(cercle, Color.***YELLOW***);

//dessiner un cercle

d.draw(cercle);

// ajouter un texte

d.label("Cool text", 800, 750);