Haute École Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

ATL – Ateliers Logiciels

Exercice Ascii Paint

Héritage polymorphisme - Mise en pratique

1 Mini Projet : AsciiPaint - remise 1

Le but de cet exercice est de créer une application permettant de créer et afficher différentes formes géométriques dans la console.

Voici à quoi pourrait ressembler une illustration contenant 4 cercles dans la console :

Votre application permettra

1. d'ajouter une nouvelle forme : cercle, rectangle ou carré ;
2. d'afficher l'illustration ;
3. d'afficher la liste des formes présentes dans le dessin ;
4. de bouger une forme ;
5. de changer sa couleur.

Ces manipulations se feront dans la console via des commandes textuelles.

Par exemple :

Version du document
19 septembre 2023

```
add circle 5 3 1 c
add rectangle 10 10 5 20 r
show
list
move 1 10 20
color 0 C
```

permettent respectivement :

- ▷ d'ajouter un cercle centré en (5,3), de rayon 1 et de couleur `'c'` ;
- ▷ d'ajouter un rectangle dont le coin supérieur gauche est en (10,10), la largeur est de 5, la hauteur de 20 et la couleur `'r'` ;
- ▷ d'afficher le dessin ;
- ▷ d'afficher la liste numérotée des formes présentes ;
- ▷ de bouger la forme numéro 1 de 10 points horizontalement et de 20 points verticalement ;
- ▷ de changer la couleur de la forme 0 (le cercle) en un C.

2 Diagramme de classes

Ci-joint, un diagramme de classes (incomplet) dont vous pouvez vous inspirer pour votre implémentation.

Nous présentons quelques classee utiles pour structurer votre application.

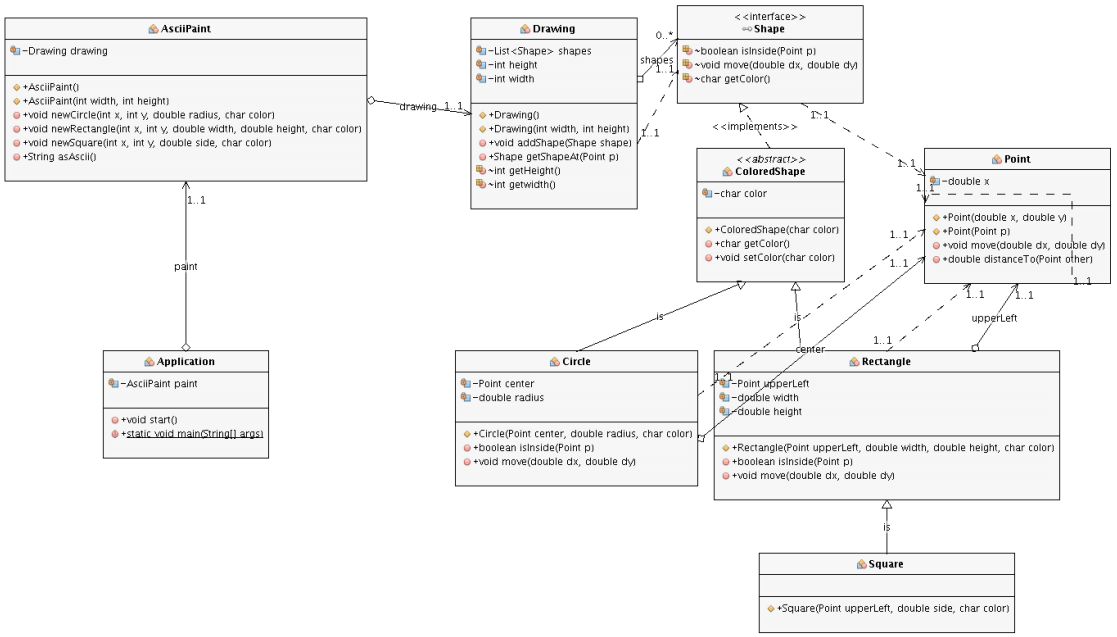
L'interface `Shape` représente une forme et définit les comportements attendus par toute forme. Elle déclare les méthodes :

- ▷ `move(double dx, double dy)` permettant de déplacer une forme ;
- ▷ `isInside(Point p)` retournant vrai si le point donné se trouve à l'intérieur de la forme, et faux sinon ;
- ▷ `getColor()` retournant un caractère d'affichage, par exemple le caractère 'c' ;
- ▷ `setColor(char color)` modifiant la couleur de la forme ;

Les classes `Circle`, qui représente un cercle, et `Rectangle`, qui représente un rectangle, implémentent l'interface `Shape`. La classe `Square`, représentant un carré, sera une sous-classe de `Rectangle`.

La classe `Drawing` représente une illustration sous la forme d'une collection de formes. Elle a une longueur et une largeur (50x50, 100x30, etc) et propose une méthode qui permet de l'afficher dans la console.

L'illustration contient donc une liste de formes et, pour se dessiner, elle parcourt chaque point de chaque ligne en demandant à chaque forme si le point est intérieur ou non de façon à afficher un blanc (si aucune forme n'occupe cette case) ou le caractère d'affichage de la forme (sa couleur) sinon.



3 Structure du code

Vous devez structurer votre code en suivant l'architecture Model-View-Controller (MVC).

Vous aurez au minimum 3 *packages* :

- pour le modèle : `g12345.atl.ascii.model`
- pour la vue : `g12345.atl.ascii.view`
- pour le contrôleur : `g12345.atl.ascii.controller`

3.1 Le modèle

Le modèle contient les classes et une interface : `Point`, `Shape`, `Circle`, `Rectangle`, `Square`, `Drawing` et `AsciiPaint`.

`AsciiPaint` est la façade du modèle et contient les méthodes permettant de modifier le modèle : ajouter un forme, bouger une forme, changer la couleur, etc. La façade contient aussi les méthodes permettant de récupérer les informations nécessaires à l'affichage.

3.2 La vue

La vue n'est pas présente dans le diagramme fourni. Elle propose les méthodes permettant l'affichage : afficher le dessin, afficher la liste des formes, etc.

3.3 Le contrôleur

La classe `Application` est le contrôleur.

Le contrôleur

- gère la boucle applicative (tant qu'on a pas fini, l'application demande une commande et l'exécute)
- traduit les commandes de l'utilisateur en action sur le modèle et/ou sur la vue.

Pour interpéter les commandes de l'utilisateur nous vous recommandons d'utiliser les REGEXP de java : https://www.w3schools.com/java/java_regex.asp, et en particulier la notion de groupe pour récupérer les éléments d'une commande.