

Programmation objet : C++, JAVA

TP N° 5

Objectifs :

- Relations entre les objets

L'objectif de ce TP est de mettre en œuvre les principes des relations. Le support de cours *Relations* est fortement conseillé. Il est à votre disposition sur Moodle → S5A → Langages Orientés Objets.

Partie 1 – C++ (durée 1 UC)

Contexte : Création d'une fenêtre contenant des objets graphiques

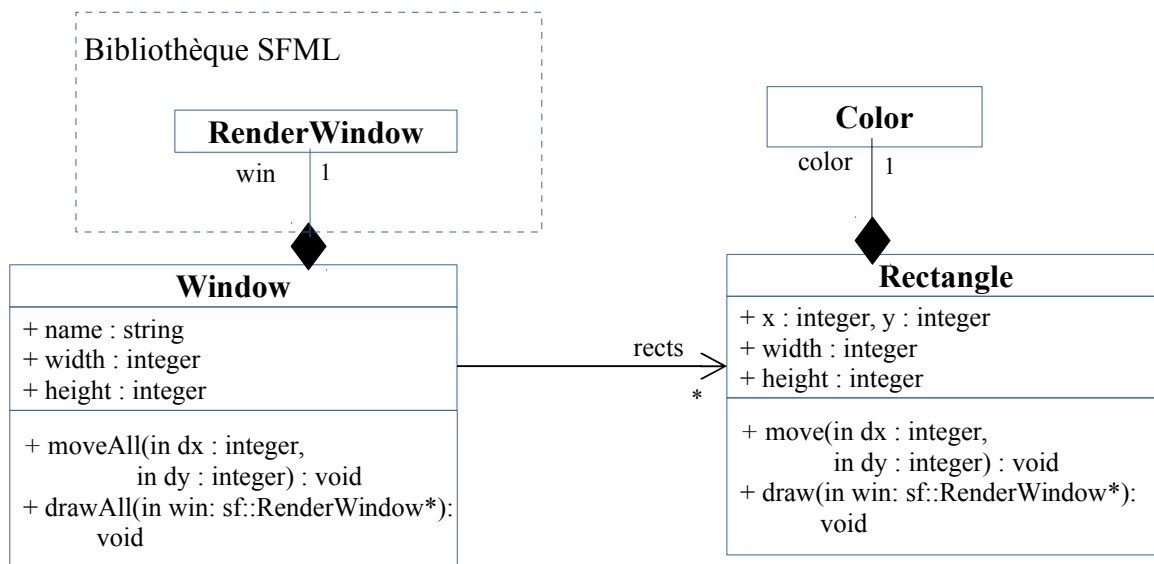
Question 1 : Création de la relation entre les classes *Window* et *Rectangle*.

Téléchargez le code CPP fourni sur Moodle pour le TP4.

Le dossier CPP contient :

- une classe *Window* qui se sert de la bibliothèque **SFML 2.1** pour dessiner une fenêtre (la relation entre cette classe et la *RenderWindow* est déjà implémentée),
- le squelette de la classe *Rectangle* (presque entièrement vide, c'est à vous de l'implémenter),
- une classe *Color* (entièrement implémentée),
- un fichier *Appli* contenant le *main*,
- un *makefile* pour compiler le code.

Le code implémente le diagramme de classes suivant sauf les méthodes et les relations. C'est à vous de bien les implémenter et de modifier le code fourni comme cela vous semble nécessaire.



Fournir une implémentation pour les méthodes suivantes :

Dans la classe *Rectangle*:

- constructeur et destructeur,
- *move()* qui applique au *Rectangle* un déplacement défini par les paramètres *dx*, *dy*,
- *draw()* qui dessine le *Rectangle* (un exemple d'implémentation est déjà fourni mais attention ! Ce code n'est qu'un exemple, vous devez l'adapter à votre problème. Il est là juste pour vous éviter la lourde tâche

de chercher dans la documentation de la bibliothèque SFML comment dessiner les formes géométriques, ce qui sortirait des objectifs de ce TP).

Dans la classe *Window* :

- *drawAll()* qui dessine tous les objets graphiques de la *Window*,
- *moveAll()* qui applique aux objets graphiques un déplacement défini par les paramètres *dx*, *dy*.

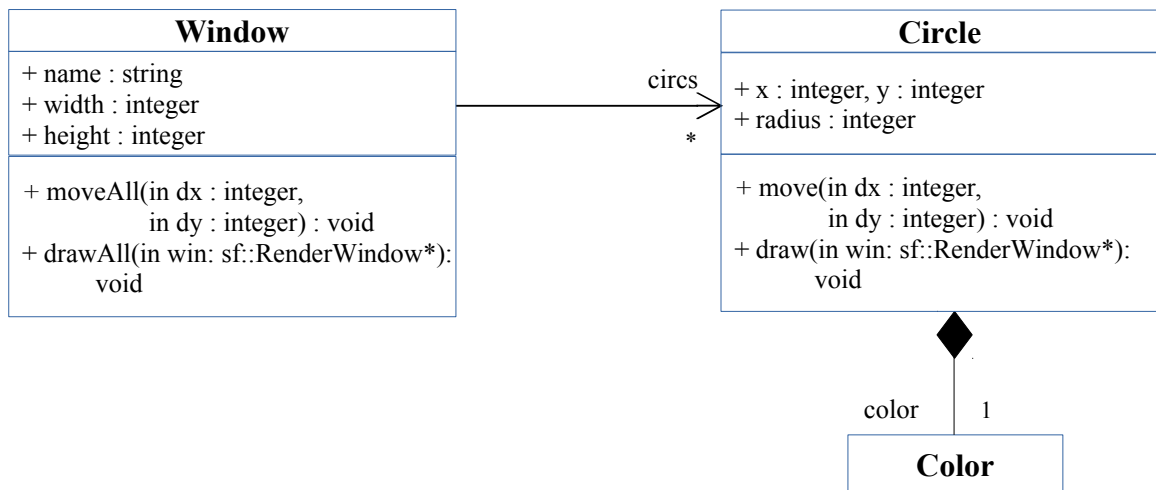
Dans le fichier *Appli.cpp* modifiez le *main* :

Créez 3 objets de type *Rectangle* en vous servant de la fonction *rand()* pour leur donner une position, une dimension et une couleur au hasard.

Si vous avez bien implémenté les relations et les fonctions demandées vous verrez les objets rectangle apparaître dans la fenêtre et vous pourrez le bouger avec les flèches sur votre clavier.

Question 2 : Création de la classe *Circle*.

Ajoutez une classe *Circle* en suivant le diagramme de classes suivant :



Dans le fichier *Appli.cpp* modifiez le *main*.

Créez 3 objets de type *Circle* en vous servant de la fonction *rand()* pour leur donner une position, un rayon et une couleur au hasard.

Exemple de fonction *draw()* pour la classe *Circle* (code à adapter à votre problème) :

```
void Circle::draw(sf::RenderWindow *win) const {
    int radius = 8;
    sf::CircleShape shape(radius);
    shape.setFillColor(sf::Color(255, 0, 0));
    shape.setPosition(50,50);
    win->draw(shape);
}
```

Partie 2 – Java (durée 1 UC)

Question 1 : Création de la relation entre les classes *Window/Rectangle* et *Window/Circle*.

Résolvez le problème proposé dans la première partie du TP en langage Java.

À nouveau vous pouvez vous servir du code fourni sur Moodle. Pour l'affichage graphique les packages *awt* et *swing* sont utilisés mais à nouveau vous n'auriez qu'à implémenter les relations entre les classes.

Dans la version Java la méthode *drawAll()* doit s'appeler *paint*, mais elle est déjà définie pour vous, il ne manque qu'appeler la fonction *draw* sur chaque élément graphique de la fenêtre.

Attention : il faut passer le contexte graphique *g* à la *draw*. Ce contexte fournit les fonctionnalités nécessaires pour dessiner plusieurs formes (rectangles, cercles,...).

Des commentaires **//TODO:** vont vous aider pour savoir où implémenter le code demandé.