

Contexte :

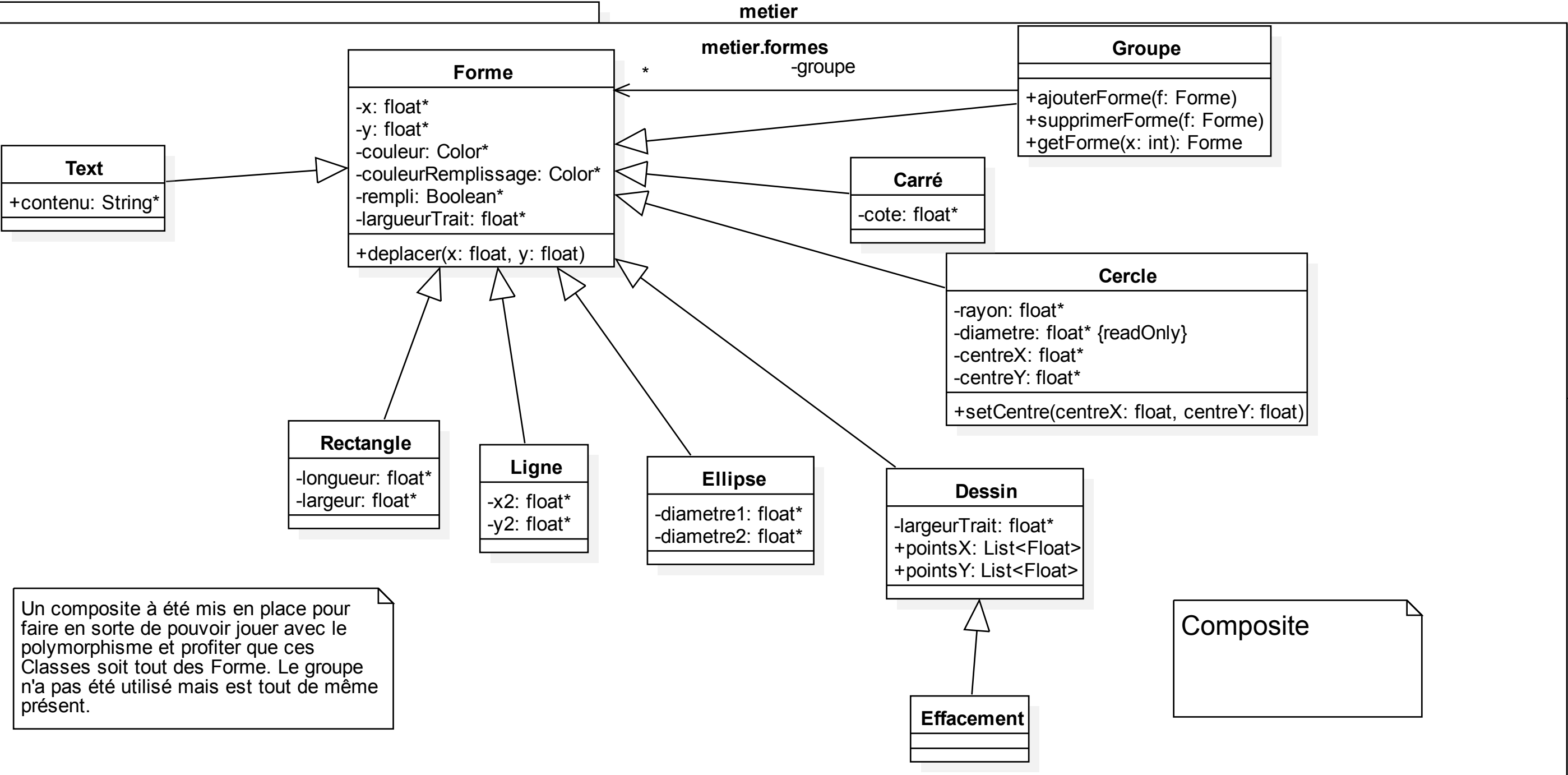
Ce projet réside en la réalisation d'un logiciel de dessin type "Paint" qui permet de dessiner avec son curseur ou bien de sélectionner des figures afin de les tracer. J'ai donc réalisé un logiciel nommé ColorMaster permettant d'effectuer cela. On peut par exemple tracer un rectangle en appuyant sur le bouton correspondant et en cliquant sur le canvas puis en déplaçant la souris pour tracer sa forme. De plus, j'ai fait en sorte que l'on puisse faire un mécanisme de retour en arrière et en avant afin de simplifier l'utilisation de celui-ci. Ces undo et redo forment un historique d'action qui vont ensuite pouvoir être sauvegardé lors de la sauvegarde du dessin. Lorsque l'on finit de traiter son fichier et qu'on l'ouvre à nouveau on a donc encore accès au retour en arrière et avant. De plus j'ai voulu implémenter une mécanique permettant de suivre les fichiers qui ont été ouvert récemment et ainsi les ouvrir depuis une liste.

La difficulté de ce projet a résidé dans l'équipe puisque le camarade qui était censé fournir un travail équivalent au miens n'a même pas écrit une seule ligne de code.

Cela m'a donc pénalisé dans la réalisation du projet et je n'ai pas totalement pu faire ce que je voulais.

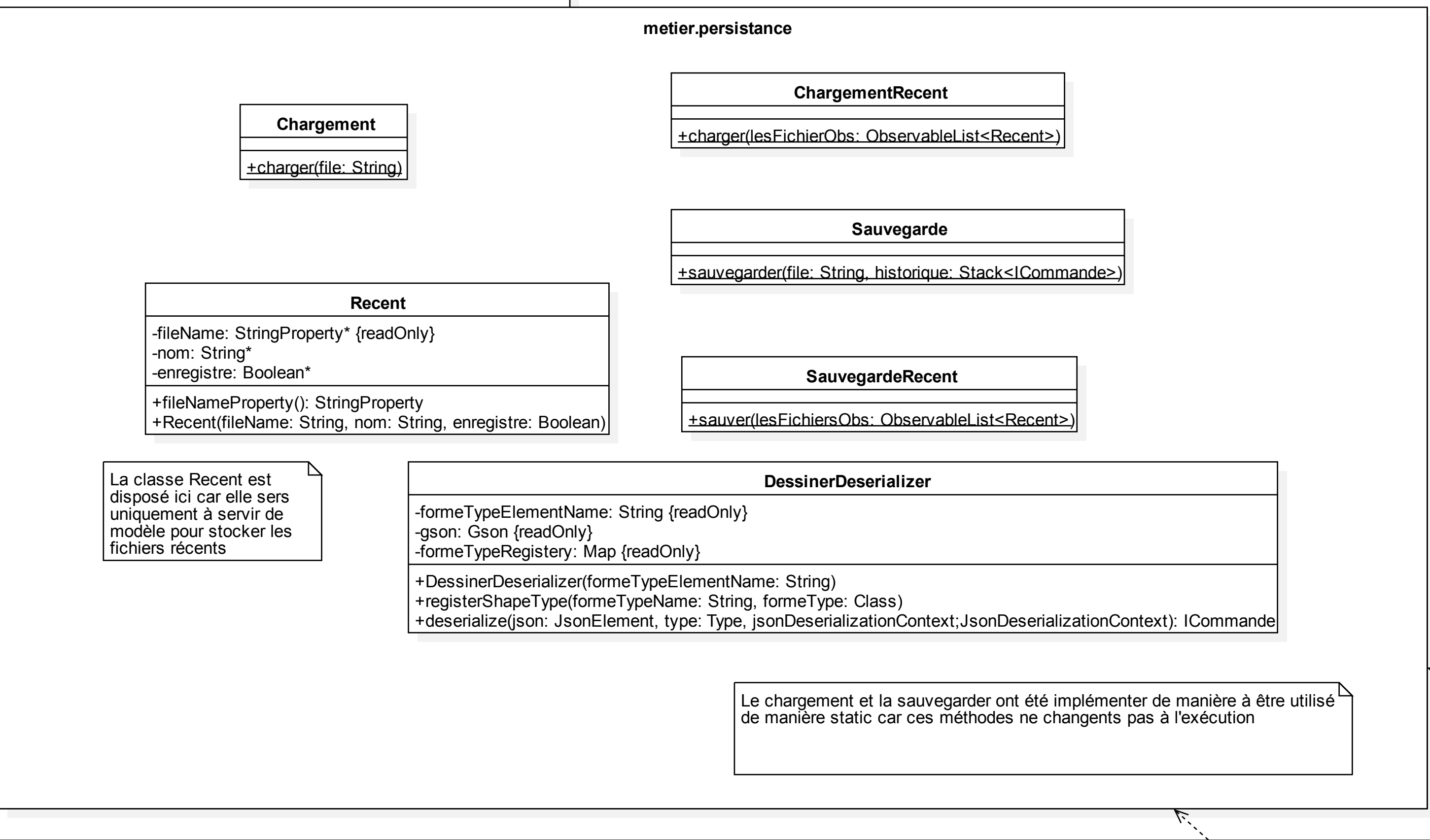
En effet il manque énormément de refactorisation de code qui me permettrait par exemple de rendre la responsabilité des classes unique à certains endroits ou bien rendre l'interface plus responsive, ect...

Les propriétés qui terminent pas des * possèdent des getters et setters dans la classe correspondante.
Les constructeurs ne sont pas représenter quand l'objet est instancier de manière vide.
La plupart des méthodes hérités d'un héritage ne sont pas à nouveau mentionné pour gagner de la place et de la lisibilité.
On remarque par ailleurs la présence de 2 patrons de conceptions différents que sont la Commande et le Composite.



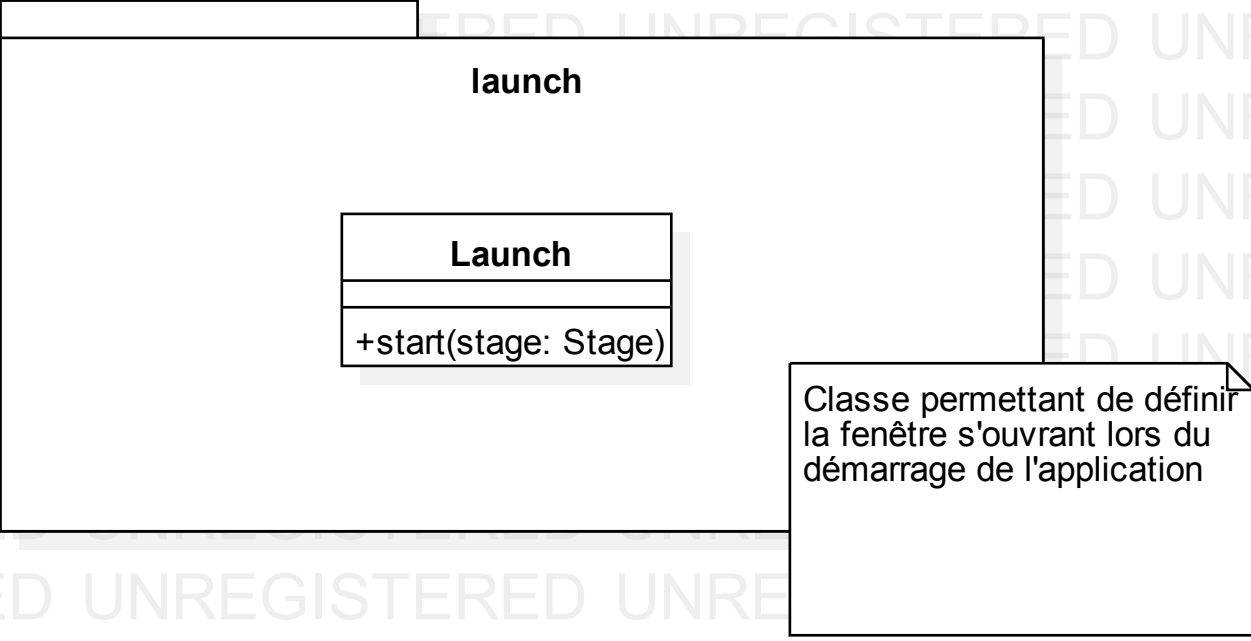
Un composite à été mis en place pour faire en sorte de pouvoir jouer avec le polymorphisme et profiter que ces Classes soit tout des Forme. Le groupe n'a pas été utilisé mais est tout de même présent.

Composite

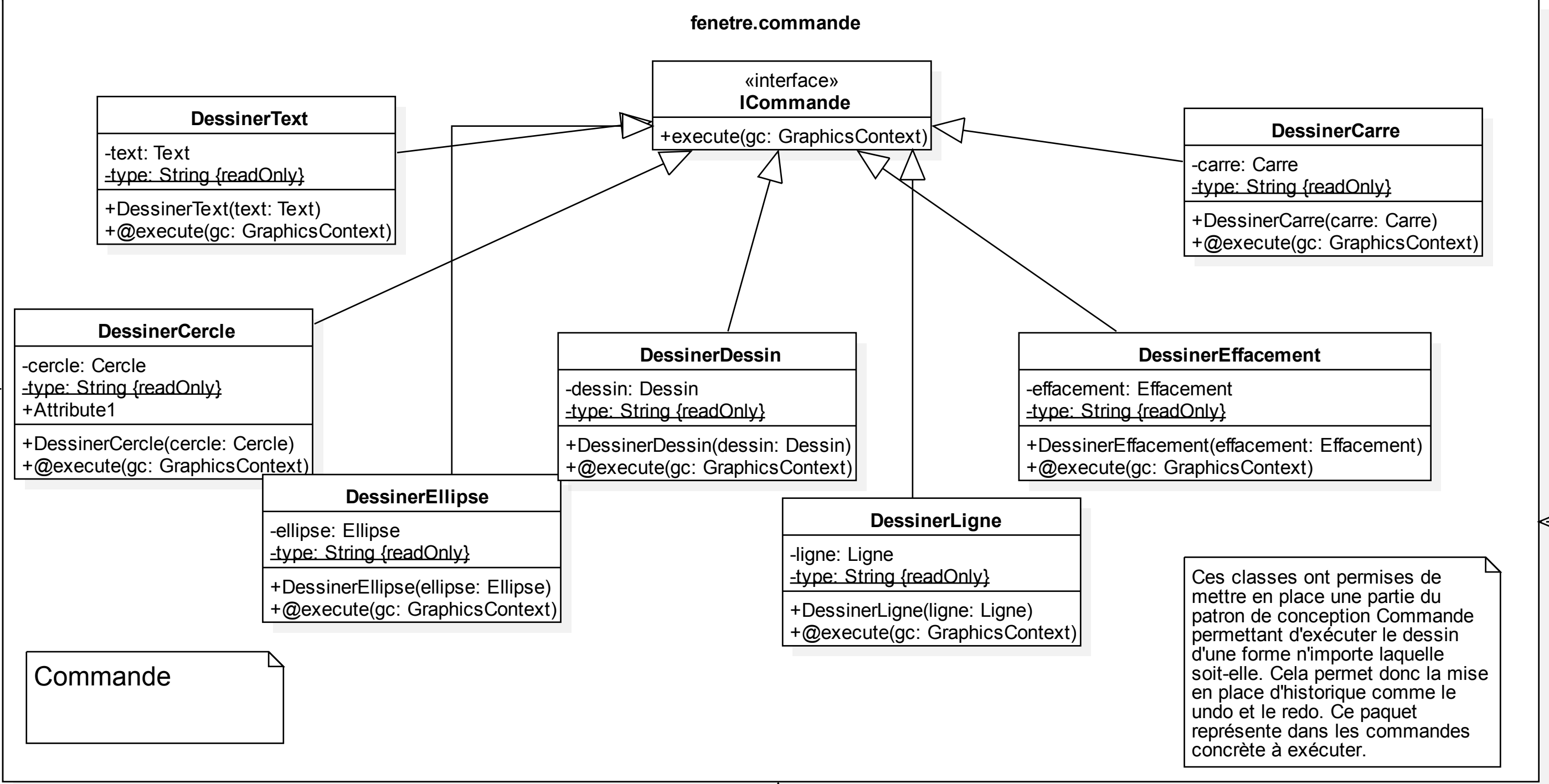


La classe Recent est disposé ici car elle sers uniquement à servir de modèle pour stocker les fichiers récents

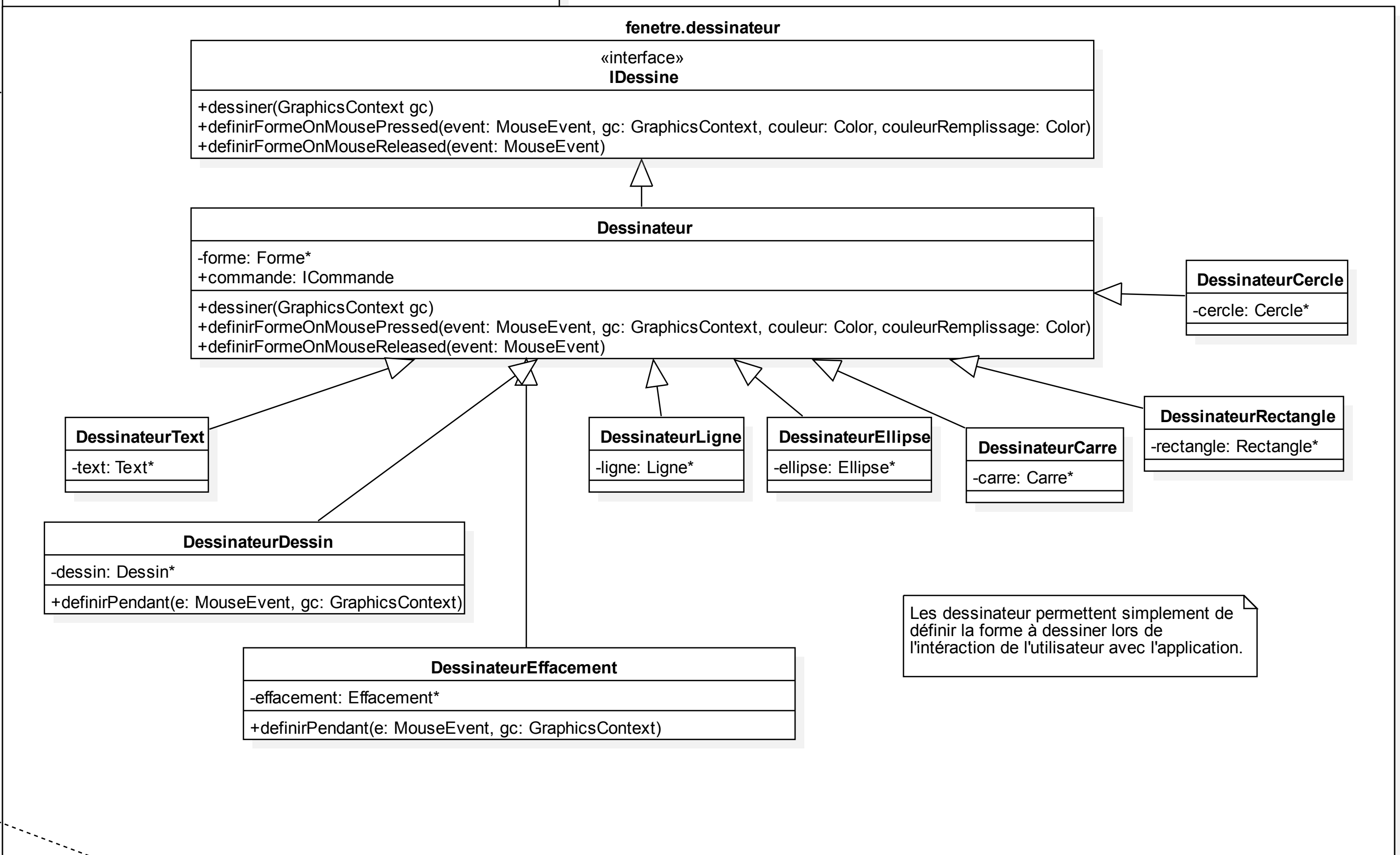
Le chargement et la sauvegarder ont été implémenter de manière à être utilisé de manière static car ces méthodes ne changents pas à l'exécution



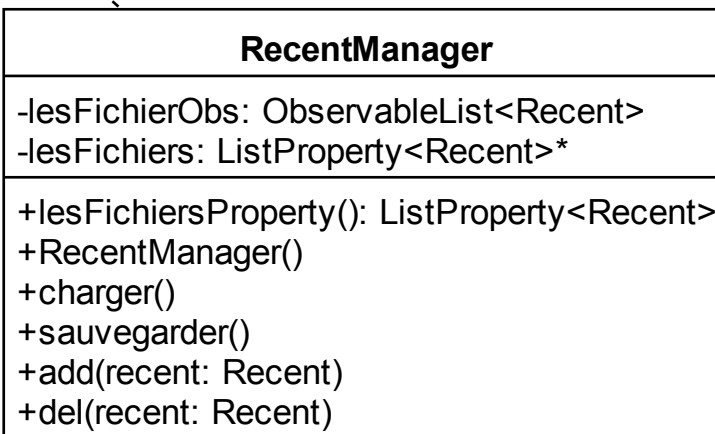
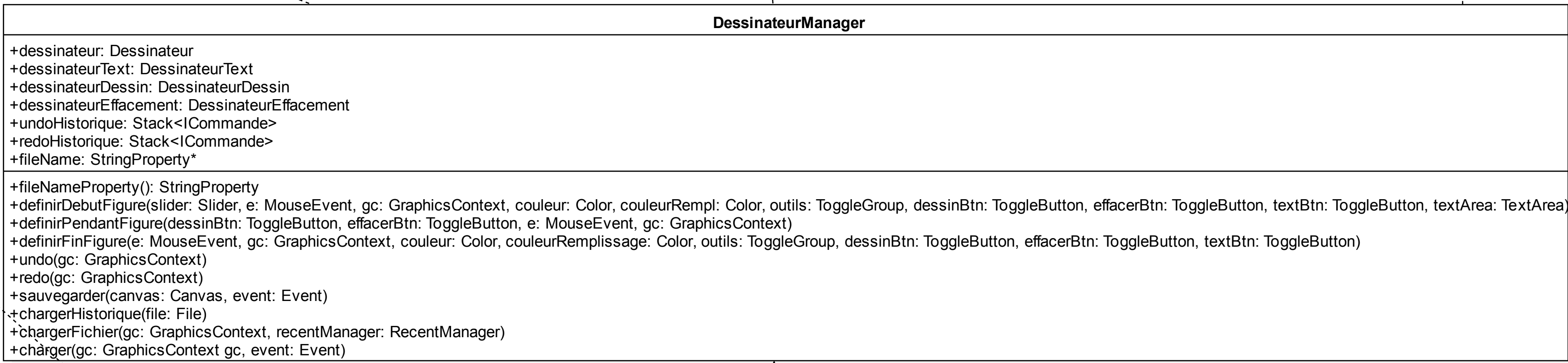
Classe permettant de définir la fenêtre s'ouvrant lors du démarrage de l'application



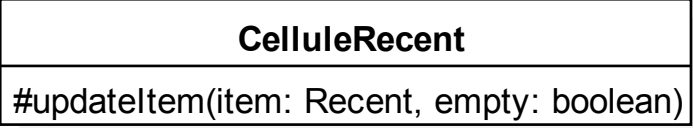
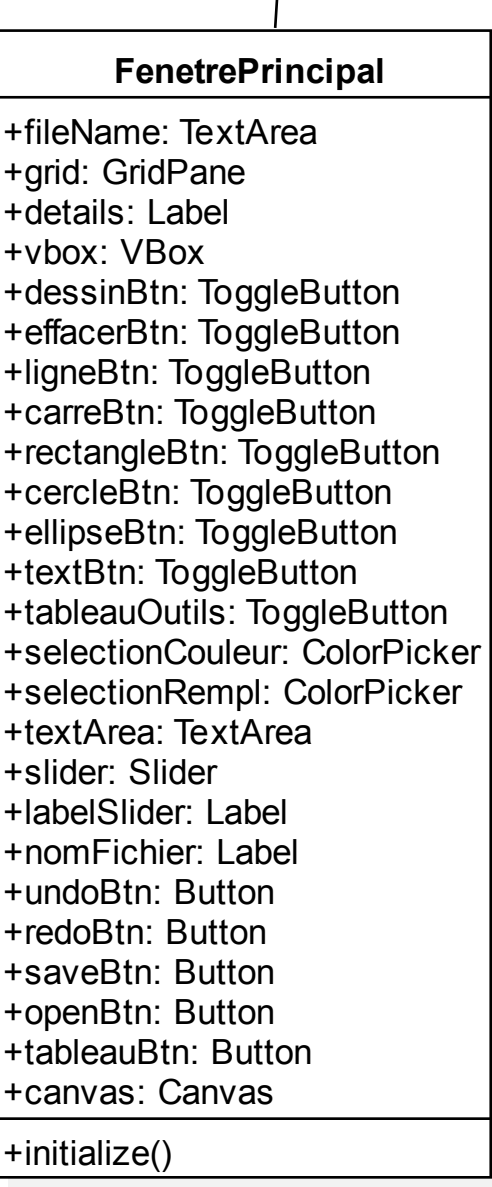
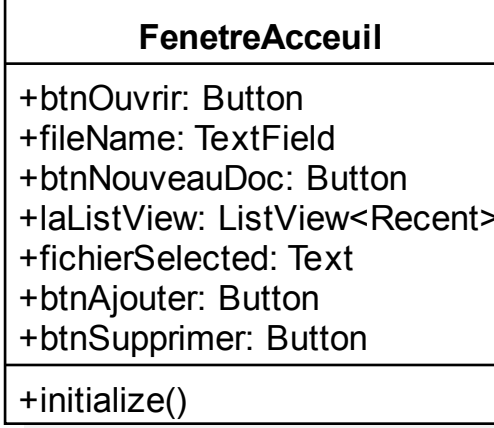
Ces classes ont permises de mettre en place une partie du patron de conception Commande permettant d'exécuter le dessin d'une forme n'importe laquelle soit-elle. Cela permet donc la mise en place d'historique comme le undo et le redo. Ce paquet représente dans les commandes concrète à exécuter.



Les dessinateur permettent simplement de définir la forme à dessiner lors de l'interaction de l'utilisateur avec l'application.



Code-behind des fenêtres comportant le même nom mais en extension FXML. C'est la méthode initialize qu'il est indiqué tout les interactions avec les éléments de la vue. Ces deux utilise le manager afin d'utiliser les différentes méthode du modèle



Classe nécessaire à la mise en place d'une CellFactory et permettant de définir comment un fichier récent s'affiche

ColorMaster



Utilisateur

Ouvrir nouveau document

Ouvrir document récent

Dessiner

Enregistrer son dessin et ses historiques

Charger son dessin et ses historiques