

RAPPORT PROJET ACO

Université Rennes 1

Istic-Miage

Groupe 1B | Master 1 Miage

Membres du binôme

Djedjemel Arnould Cyriaque

Padonou DieuDonné

Encadrant

Adrien Le Roch

Contenu

I. Présentation du projet

1. Contexte
2. Diagramme de classe
 - a. Version 1
 - b. Version 2
 - c. Version 3

II. Mise en Œuvre du projet

1. Choix personnels de conception
2. Synthèse des résultats de tests
3. Fonctionnement du projet
 - a. Version 1
 - b. Version 2
 - c. Version 3

Conclusion

I. Présentation du projet

1. Contexte

Ce projet consistait à développer un éditeur de texte avec les fonctionnalités telles que :

- Insérer du texte
- Sélectionner une partie du contenu de l'éditeur
- Modifier la sélection
- Copier un texte sélectionné
- Coller un texte sélectionné
- Couper un texte sélectionné
- Rejouer toutes les actions jouées auparavant
- Annuler/Rétablir une action jouée

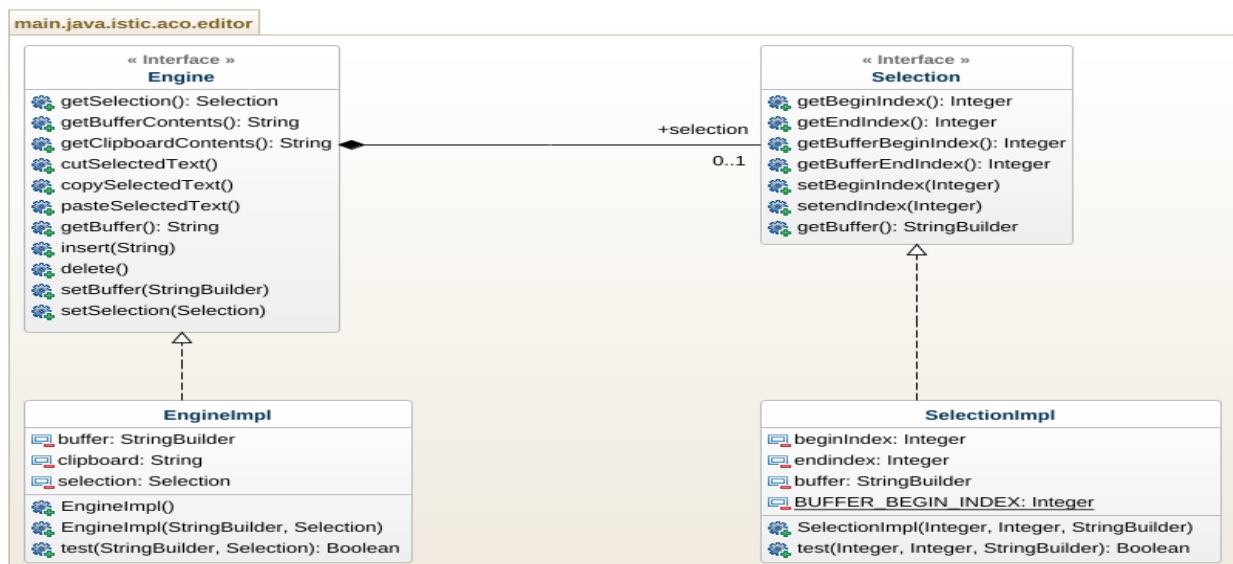
Il fallait pour cela, travailler avec le langage Java, respecter au maximum les bonnes pratiques et standards du développement logiciel vues en cours, mais aussi mettre en œuvre les patrons de conception *Command* et *Memento*.

Ainsi, ce projet a eu différentes versions allant de la version 1 à la 3 permettant d'apercevoir l'impact de l'utilisation des différents patrons de conception sur le projet. Il y a eu entre autres :

- Version 1 : Éditeur classique sans application de pattern
- Version 2 : Version 1 avec implémentation du pattern Command, début d'implémentation du pattern Memento avec l'action replay.
- Version 3 : Version 2 avec une meilleure implémentation du pattern Memento, action Annuler/Rétablir (Undo/Redo).

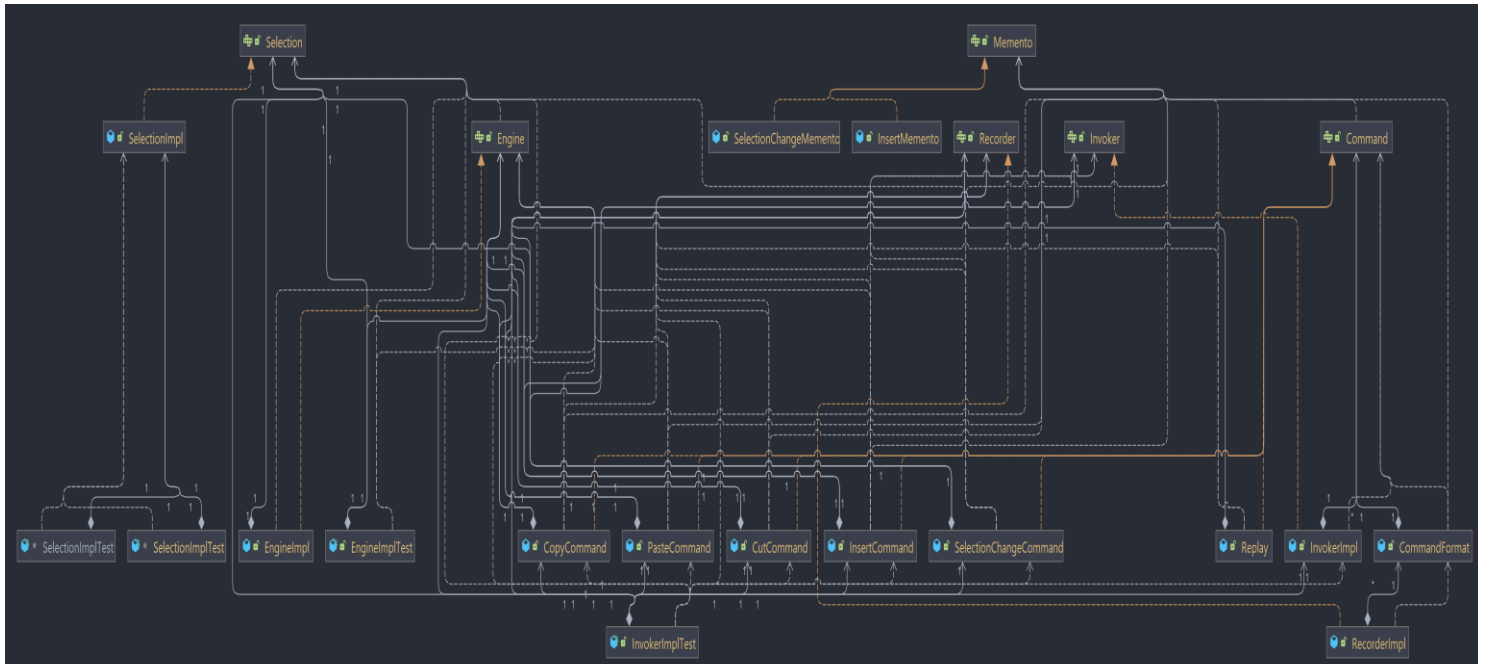
2. Diagramme de classes

a. Version 1



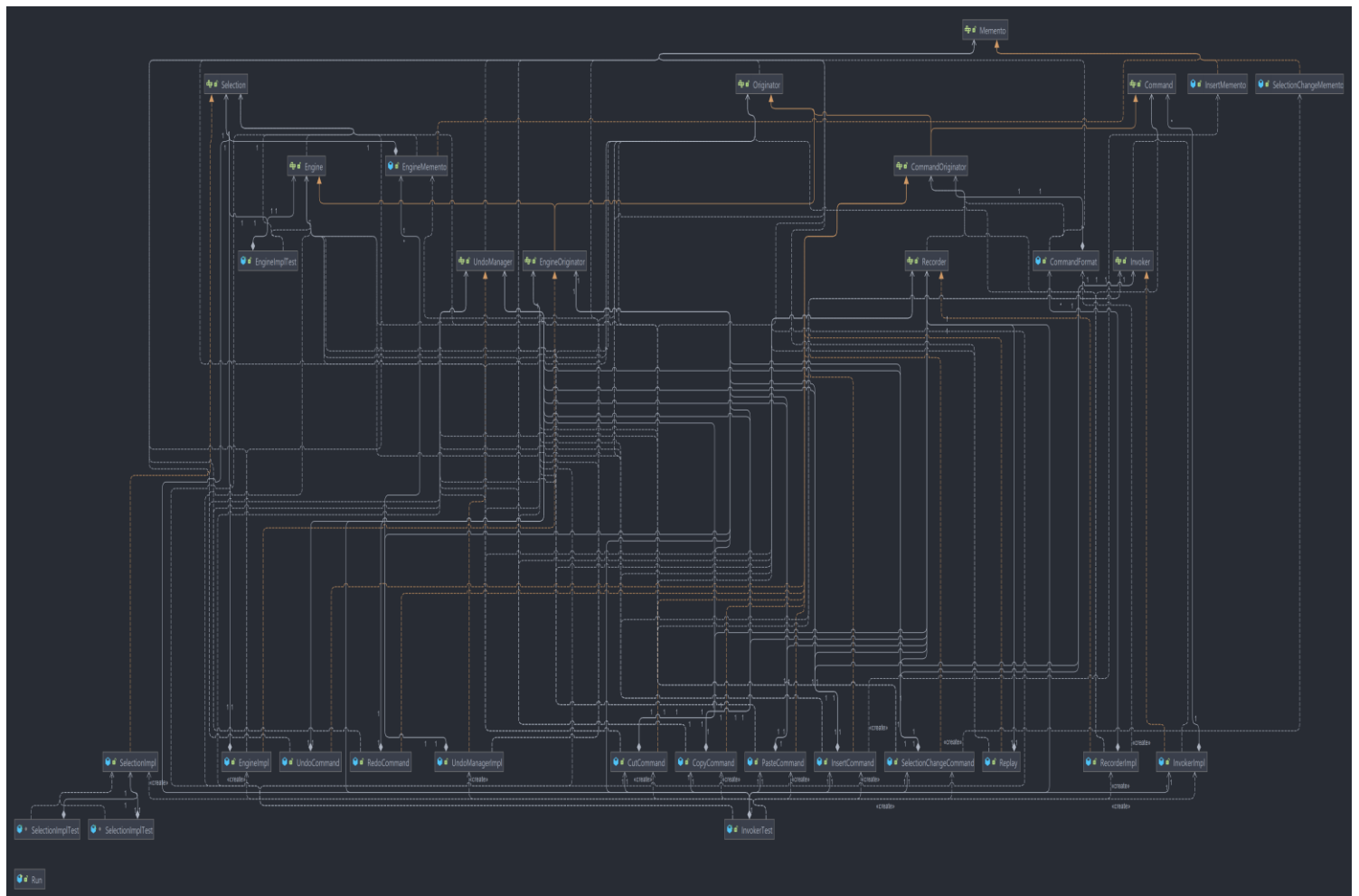
Voir plus sur le dépôt, (diagramme_Version1.png).

b. Version 2



Voir plus sur le dépôt, (diagramme_Version2.png).

c. Version 3



Voir plus sur le dépôt, (*diagramme_Version3.png*).

II. Mise en Œuvre du projet

1. Choix personnels de conception

Pour la conception tout d'abord, nous avons fait le choix de ne pas travailler avec une interface graphique, ni de classe Main. Seules les classes Tests permettent de vérifier si le projet fonctionne correctement.

Ensuite pour les actions Annuler/Rétablir, nous avons décidé de sauvegarder les états de la classe Engine et de faire des mouvements sur les actions précédentes et antérieur pour représenter les actions Annuler/Rétablir.

Ainsi, nous avons une classe *EngineMemento* représentant ses états, aussi nous avons une interface *UndoManager* pour enregistrer ses états et itérer entre elles selon la situation.

Ce choix s'inspire dans la logique d'implémentation du pattern Memento, la classe Engine devient ainsi *Originator*, la nouvelle interface UndoManager le *Caretaker* et la classe EngineMemento, le *concreteMemento*. Il existe déjà une interface Memento dont hérite la classe EngineMemento.

2. Synthèse des résultats de tests

- Placez vous sur la branche master
- Allez dans le dossier *src*, ensuite le dossier *tests*, et enfin le dossier *reports*
- Vous y trouverez une image du résultat du coverage et les fichiers de rapports des tests

3. Fonctionnement du projet

a. Version 1

- Placez vous sur la branche « *merge-branch* » du dépôt.
- Lancez la classe *EngineTest* et observer le résultat, vous pouvez modifier la valeur des variables pour avoir des résultats différents.

b. Version 2

- Placez vous sur la branche « *merge-branch-2* » du dépôt.
- Lancez la classe *InvokerTest* et observer les résultats, vous pouvez modifier la valeur des variables pour avoir des résultats différents.

c. Version 3

- Placez vous sur la branche « *master* » du dépôt.
- Lancez la classe *Tests* et observer les résultats, vous pouvez modifier la valeur des variables pour avoir d'autres résultats.

Conclusion

Grace à ce projet, nous avons pu apprendre à coder en respectant les bonnes pratiques, mais aussi qu'en utilisant les patrons de conception, ceux-ci facilitent la conception et permettent de produire un code plus robuste, réutilisable et facilement maintenable.