

**OMD**

## **Rapport TP 2**

Nous attestons que ce travail est original, qu'il indique de façon appropriée tous les emprunts, et qu'il fait référence de façon appropriée à chaque source utilisée.

# Mini-éditeur

## Introduction

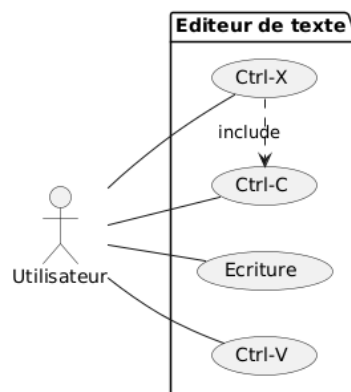
Le but de ce TP est de modéliser et réaliser un éditeur de texte basique. Nous avons plusieurs prérequis à respecter dans la réalisation de cet éditeur. Les objectifs de ce TP sont de présenter une technique de conception et de programmation à objets simple, de montrer les différents moyens de coordination des objets, de montrer la constructions parallèle des diagrammes statiques et dynamiques, de montrer la mise en œuvre des patrons de conceptions, et de montrer le passage des modèles de conception au programme. Pour réaliser ce TP, nous allons utiliser PlantUML afin de réaliser les diagrammes, et Java afin de programmer notre éditeur de texte.

## Version n°1

Les attentes par rapport à notre éditeur de texte sont les suivantes :

- Le texte est contenu dans un buffer (zone de travail),
- Il existe une notion de sélection de texte, avec des commandes utilisateur permettant de déplacer le début et la fin de la sélection,
- Copie de la sélection dans le presse-papier,
- Copie de la sélection dans le presse-papier puis effacement de la sélection,
- Remplacement (« collage ») de la sélection par le contenu du presse-papier,
- L'interface homme-machine est d'un type quelconque (textuelle ou graphique)

Nous avons donc pour commencer réalisé le diagramme de cas d'utilisation.



Et nous avons ensuite décrits chaque cas d'utilisations :

**Nom du cas d'utilisation :** Ecriture.

**Brève description :** L'utilisateur écrit dans l'éditeur.

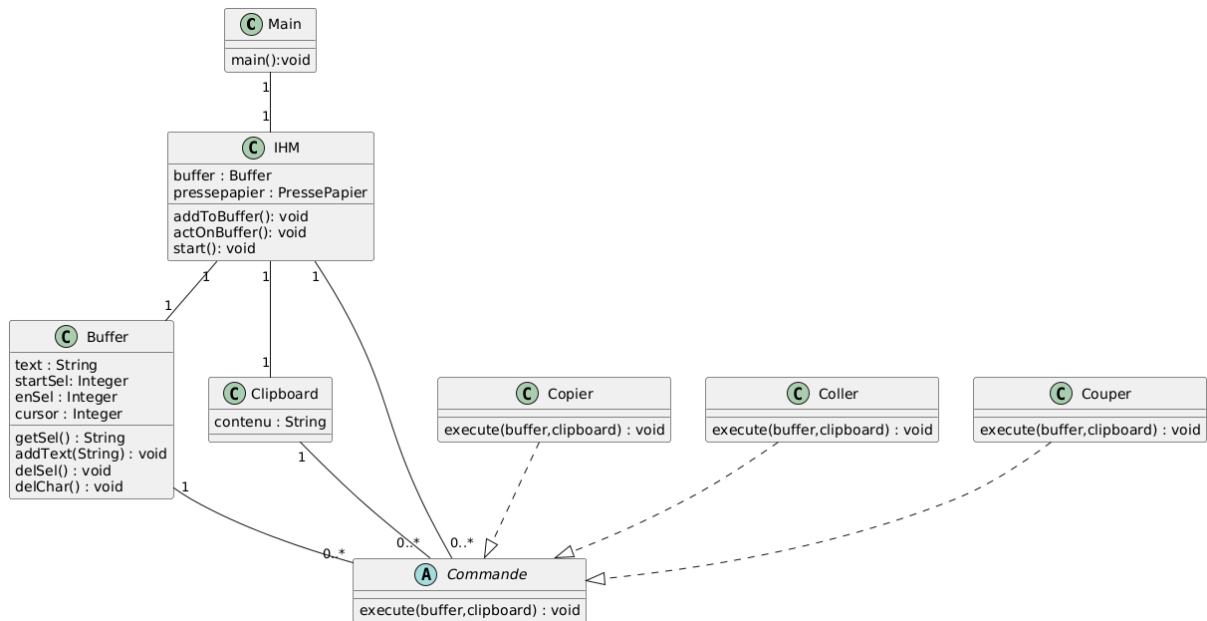
**Acteurs :** Utilisateur.

**Contexte :** L'utilisateur veut écrire dans l'éditeur de texte.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Données en entrée et pré-conditions :</b> L'éditeur est ouvert.</p> <p><b>Scénario principal :</b> L'éditeur saisit au clavier le texte voulu, qui s'affiche dans l'éditeur.</p>                                                                                                                                                                                                                                                                                                              |
| <p><b>Nom du cas d'utilisation :</b> Copier.</p> <p><b>Brève description :</b> L'utilisateur copie une sélection de texte.</p> <p><b>Acteurs :</b> Utilisateur.</p> <p><b>Contexte :</b> L'utilisateur veut copier une sélection.</p> <p><b>Données en entrée et pré-conditions :</b> L'éditeur est ouvert.</p> <p><b>Scénario principal :</b> L'utilisateur sélectionne du texte et choisit de copier la sélection, le texte sélectionné sera donc copié dans le presse-papier.</p>                |
| <p><b>Nom du cas d'utilisation :</b> Coller.</p> <p><b>Brève description :</b> L'utilisateur colle une sélection de texte.</p> <p><b>Acteurs :</b> Utilisateur.</p> <p><b>Contexte :</b> L'utilisateur veut coller une sélection.</p> <p><b>Données en entrée et pré-conditions :</b> L'éditeur est ouvert.</p> <p><b>Scénario principal :</b> L'utilisateur met le curseur sur l'emplacement où il veut que le texte soit collé et colle le contenu du presse-papier.</p>                          |
| <p><b>Nom du cas d'utilisation :</b> Couper.</p> <p><b>Brève description :</b> L'utilisateur coupe une sélection de texte.</p> <p><b>Acteurs :</b> Utilisateur.</p> <p><b>Contexte :</b> L'utilisateur veut couper une sélection.</p> <p><b>Données en entrée et pré-conditions :</b> L'éditeur est ouvert.</p> <p><b>Scénario principal :</b> L'utilisateur sélectionne du texte et choisit de couper la sélection, la sélection va être effacée et le texte sera copié dans le presse-papier.</p> |

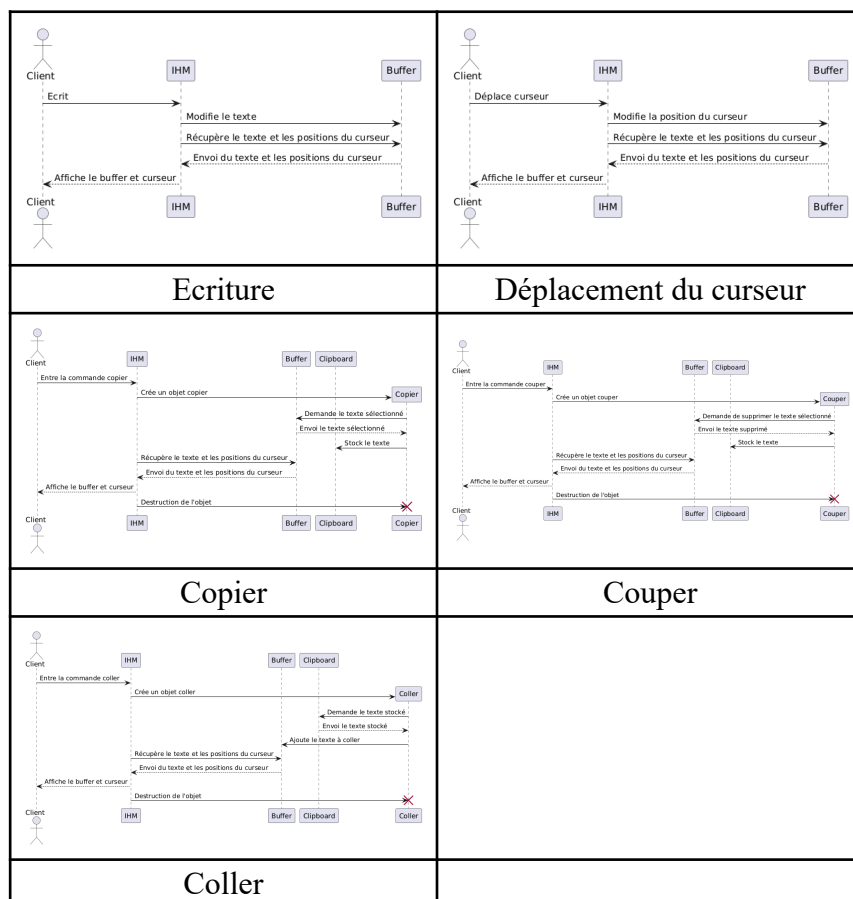
À l'aide du diagramme de cas d'utilisation et des scénarios d'utilisation, nous réalisons ensuite le diagramme de classe et les diagrammes de séquences.

Ce diagramme de classe montre l'architecture de notre éditeur de texte. La classe main contient la méthode principale qui crée les objets nécessaires : un buffer pour stocker le texte, un clipboard pour stocker le contenu temporaire, et une interface utilisateur graphique IHM. La classe IHM utilise ces objets pour interagir avec l'utilisateur et gérer le texte, le buffer stocke le texte actuel, ainsi que les indices de début et de fin pour la sélection, et fournit des méthodes pour obtenir et modifier cette sélection, le clipboard stocke le texte copié ou coupé et permet de récupérer ou de définir ce contenu. L'interface commande impose une méthode execute(Buffer, Clipboard) que les classes Copier, Coller, et Couper implémentent pour réaliser leurs actions spécifiques. Ces trois classes sont des implémentations concrètes du modèle de commande, chacune définissant comment elles interagissent avec le Buffer et le Clipboard. Pour les liens, la classe main crée et utilise IHM, qui elle-même crée et manipule un buffer et un clipboard pour gérer le texte. Les classes Copier, Couper, et Coller implémentent l'interface Commande et dépendent du buffer et du clipboard pour exécuter leurs actions.



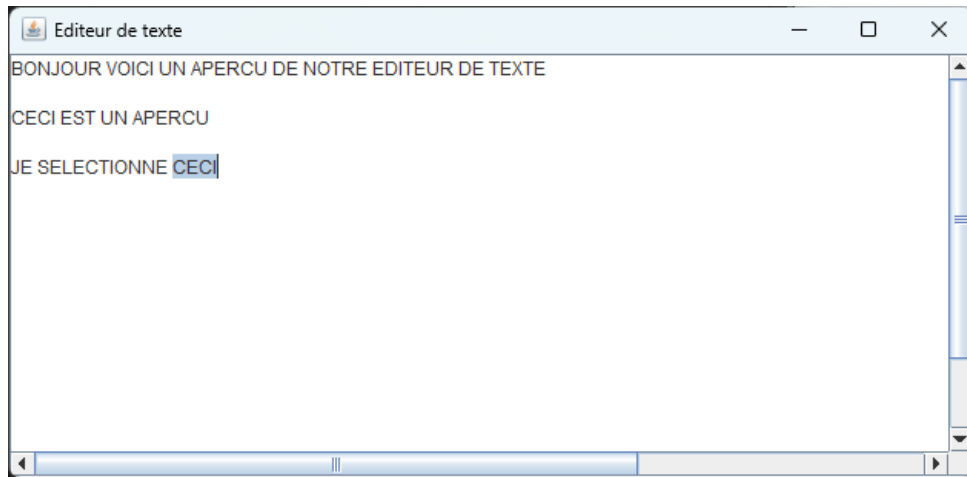
Notons que les getters et setters pour chacun des attributs des classes ne sont pas représentés pour des questions de lisibilité.

Les diagrammes de séquence représentent les étapes réalisées pour chaque cas d'utilisations.



Nous sommes ensuite passer à la partie développement de notre éditeur. Nous avons donc fais des classes pour chaque instance du diagramme de classe. Nous avons, à l'aide des librairie java.awt et javax.swing, réalisé une interface graphique, qui nous a permis d'afficher le texte, et de récupérer les interactions avec la souris pour bouger le curseur. Cependant, nous avons bloqué ou ignoré toutes autres fonctionnalités, car nous devions implémenter nous même le reste de l'éditeur de texte.

Voici un aperçu de la fenêtre de notre éditeur de texte.



Comme vous l'avez certainement remarqué, nous avons fait le choix de n'implémenter que le texte majuscule. De plus, il apparait qu'un bruit venant de windows intervient dès que l'on veut supprimer un caractère, nous n'avons malheureusement pas trouvé comment régler ce problème. Nous avons pû utiliser des barres de défilement verticale et horizontale. Grâce au librairie utilisées, nous pouvons afficher le curseur et la zone de texte sélectionnée. La zone sélectionnée peut être définie grâce à la souris, ou alors en maintenant shift et en apuyant sur les flèches vers la gauche ou le bas.

Cette version permet bien toute les fonctionnalités de base désirée.

## ***Version n°2***

Nous voulons maintenant implémenter de nouvelles fonctionnalités. Celles-ci sont listées ci-dessous.

- Enregistrer/rejouer les actions de l'utilisateur (e.g., script),
- Réaliser le défaire/refaire, avec une capacité quelconque dans le défaire (autrement dit on peut revenir au début).

## ***Conclusion***