***Rapport TP noté n°1***

Ce compte rendu a pour but de présenter le travail que j’ai effectué pour ce TP, qui avait pour but de créer les classes « Fraction » et « Liste chainée ».

Ce document me permet de détailler et expliquer les choix que j’ai fait vis-à-vis de l’implémentation de ces classes.

Cette dernière était dans un premier temps destiné à stocker des valeurs de type entier, puis je l’ai rendu générique afin qu’elle puisse stocker d’autres types de valeur, comme des objets de type « Fraction », des réels etc.

J’ai réalisé l’ensemble de ce code grâce au logiciel Visual Studio 2022.

De plus, ce TP est contenu dans une Solution nommée « TP\_note », contenant 3 projets :

« fraction », « liste\_chainee » et « liste\_chainee\_gen ».

Ces 3 projets correspondent respectivement à l’implémentation de : la classe « Fraction », la classe « Liste chainée » et la classe « Liste chainée » rendue générique.

Chacun de ces projets contient un « menu » présent dans le programme principal, permettant de tester les différentes méthodes implémentées.

1. **La classe « Fraction »**

Permet de créer des objets de type fraction, qui sont attribués d’un numérateur et d’un dénominateur.

Méthode inverse : Cette méthode modifie directement l’objet qui a permis de l’appelé, donc ne renvoie pas un objet correspondant à l’inverse de l’objet ayant appelé cette méthode.

Méthode pgcd : Renvoie le pgcd de deux fractions en utilisant l’algorithme d’Euclide.

Méthode simplifier : Divise le numérateur et le dénominateur par le pgcd de ces derniers. De plus, si le dénominateur est négatif, je multiplie ces deux éléments par -1 pour que ce soit le numérateur qui soit négatif (plus cohérent).

Gestion des exceptions : Pour cette classe, il y’a eu 2 exceptions qui ont nécessité d’être traitées : le cas ou le dénominateur est nul et le cas ou on souhaite diviser une fraction par une fraction nulle.

1. **La classe « Liste chainée »**

Cette classe permet de stocker des nœuds en chaine.

Ces nœuds contiennent chacun deux éléments : un pointeur vers un nœud et un entier.

Chaque objet de cette classe est attribué d’une « tête de liste » : m\_debut, qui est un pointeur vers un nœud.