***TP 2 : Les tableaux et les chaines de caractères***

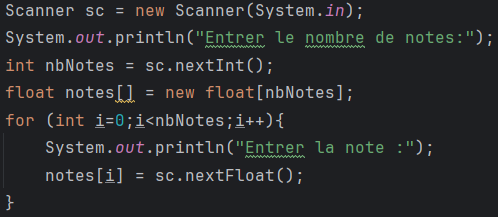
***Réalisé par : Ilyas El Mabrouki (GLSID)***

***Exercice 1 :***

*1/ Effectuer la saisie des notes :*

Pour effectué cette fonctionnalité, il est essentiel de savoir le nombre de notes pour créer un tableau sans perdre beaucoup d’espace mémoire.

Ensuite, à chaque itération de la boucle qui va se répéter selon le nombre de notes, on demande une note et on la stocke dans le tableau.



*2/ Trier et afficher la liste des notes :*

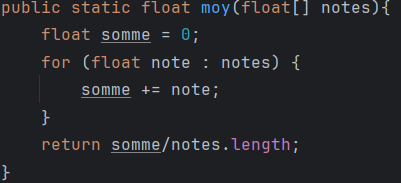
Afin d’afficher les notes triées, on a utilisé la fonction *sort* de la classe *Arrays* qui prend en paramétre un tableau.

Une boucle a été éffectué ensuite pour l’affichage des notes.



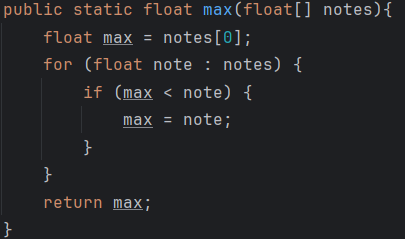
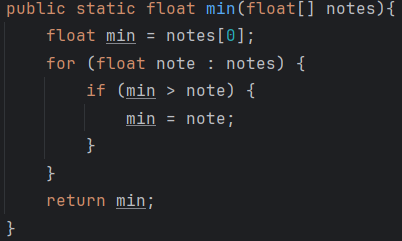
*3/ Afficher la note moyenne :*

La fonction moyenne prend comme paramétre le tableau des notes et retourne la somme des notes divisés sur le nombre des notes.



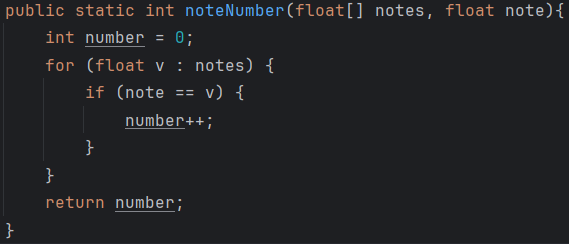
*4/ Afficher la note maximale et minimale :*

La fonction pour touver la note maximale et minimale sont très similaire, il suffit de faire une boucle sur les notes, et faire les comparaisons et les affectations nécessaires si les conditions sont vérifiées.

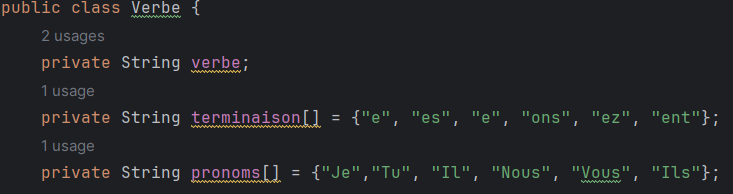
*5/ Affichez le nombre d’étudiants ayant une note saisie par l’utilisateur :*

Pour trouver le nombre des notes qui ont la méme note saisie par l’utilisateur, on a effuctué une boucle et à chaque fois on trouve une égalité entre la note saisie et la note actuel, on incrémente une variable qui joue le role d’un compteur et à la fin on la retourne.



***Exercice 2 :***

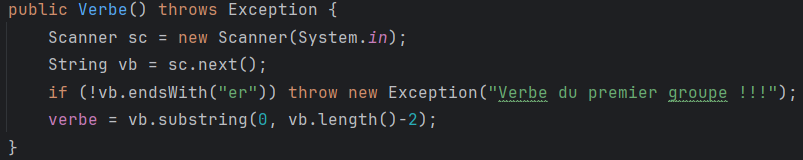
1/ La classe Verbe :



*2/ Effectuer la saisie du verbe :*

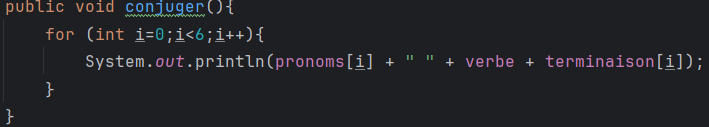
Le constructeur sera le responsable du saisie et verification du validité du verbe, vue que le role du constructeur est d’initialiser les attributs.

On a commencé par lire le texte saisie par l’utilisateur, ensuite, on a vérifié si le contenu saisie termine par l’expression ‘er’, si cette condition n’est pas vérifié on génére une exception qui va arrété l’éxecution du code, par contre si la condition est vérifié, le texte sera stocké mais avec une suppression des deux derniers lettres pour simplifier la fonctionnalité du conjugaison après.

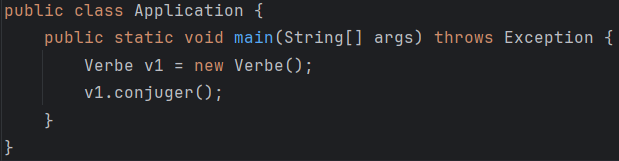


*2/ Conjuger le verbe :*

La fonction conjuger est très simple, il consiste seulement à afficher les 6 cas de conjugaison, en utilisant les tableaux pronoms et terminaisons.

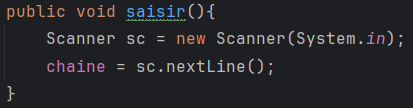


4/ la fonction main :

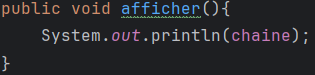


***Exercice 3 :***

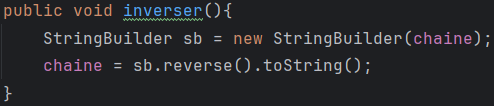
1/ Effectuer la saisie :



2/ Afficher la chaine du caractére :

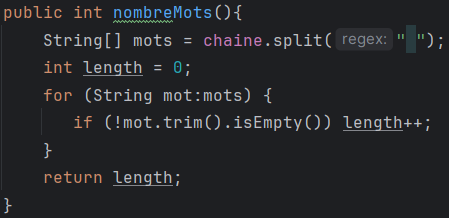


3/ Inverser la chaine du caractére :

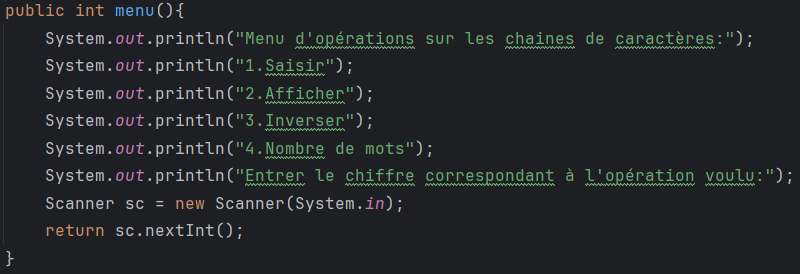


4/ Compter le nombre de mots :

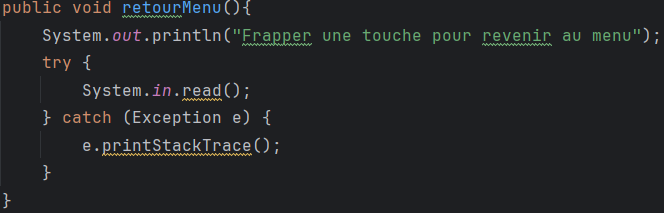
Pour compter le nombre de mots qui composent la chaine de texte, je n’ai pas effectué seulement des divisions, mais aussi une vérification des mots résultant de cette division qui peuvent etre des chaines vides. La verification valide a été suivi par une incrémentation d’un compteur.



5/ Afficher menu + Effectuer choix d’opération :

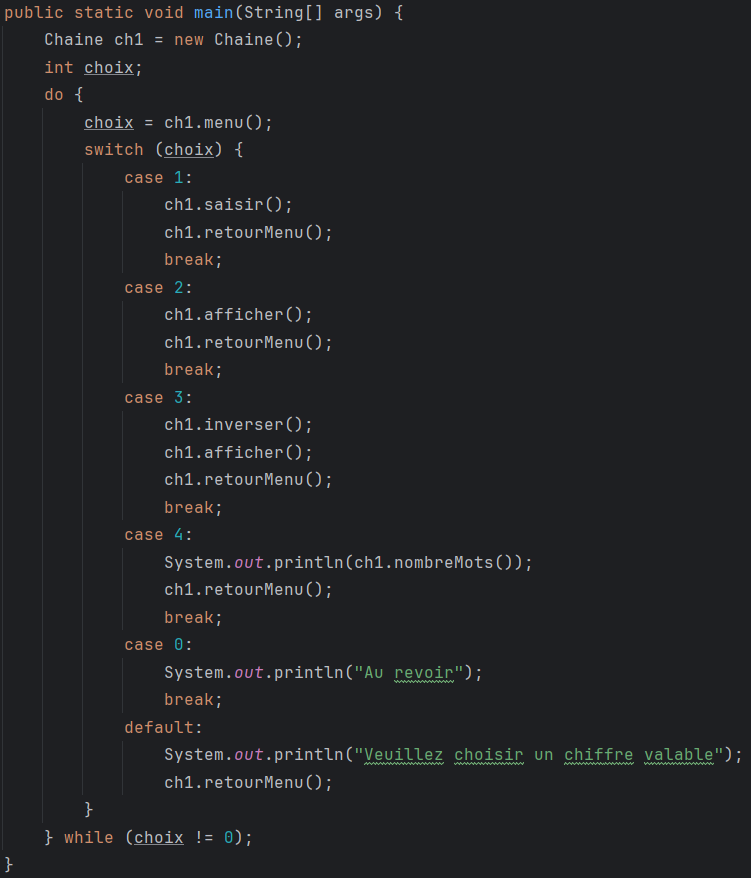


6/ Retour au menu :



7/ Fonction main :

La fonction main est sous forme d’une boucle qui commence à chaque fois par l’affichage d’un menu et le stockage du chiffre d’opération choisi par l’utilisateur, ensuite la fonction qui correspond au chiffre sera exéuter en plus de la fonction *retourMenu(),* l’itération va terminé par vérifier la valeur de choix. Si ce dernier est différent de 0, une autre itération sera exécuté sinon la boucle va etre terminé.



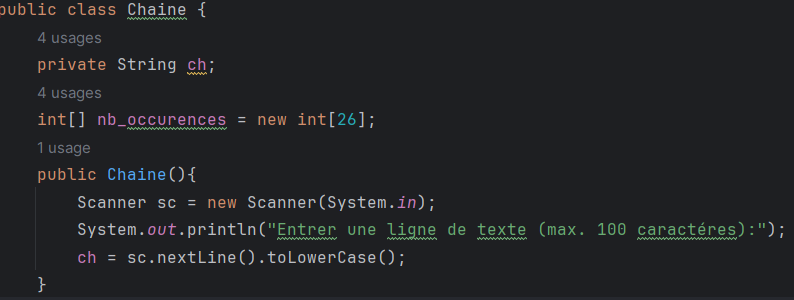
***Exercice 4 :***

1/ Classe Chaine :

La classe chaine est composé de 2 attributs :

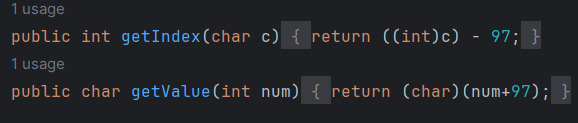
* ch pour stocker la chaine de caractére
* nb\_occurences pour stocker la valeur d’occurrence pour chaque lettre

En plus du constructeur, qui a éffectué l’initialisation de la chaine de caractére après avoir lit le texte saisie par l’utilisateur.



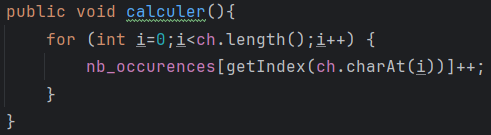
2/ Effectuer la convertion caractére / entier :

Ces convertions sont essentiels pour savoir l’index qui correspond à chaque lettre, et aussi pour savoir la valeur qui correspond à chaque index.



3/ Calculer le nombre d’occurences :

Pour se faire, il suffit d’incrémenter la valeur de l’élément qui correspond à la lettre actuel, après avoir effectuer une boucle sur la chaine.



4/ Affichage de la liste des lettres :

La fonction d’affichage contient une boucle sur le tableau des occurrences et effectue une vérification sur l’existence de la lettre actuel dans la chaine de caractére, et à chaque fois la condition est vérifié, on affiche la lettre avec le nombre de fois cette derniére a été répété dans la chaine.

