

RAPPORT TP3 PRALG ALEXANDRE DÉCAVÉ

Classe Eleve :

- J'ai choisi de coder des méthodes SetNote, GetNote, SetName et GetName, pour pouvoir récupérer les notes et les noms, et les modifier si jamais on a besoin (pour après tester les tris en changeant une note par exemple).
- J'ai codé un constructeur et un destructeur mais je ne suis pas sûr de l'utilité. Je ne savais pas s'il fallait faire des pointeurs pour les attributs de la classe. Je ne savais donc pas s'il fallait initialiser note et name vides et après mettre leur valeur avec les fonctions Set, ou s'il fallait directement initialiser les deux valeurs.

Remplissage aléatoire du vecteur d'élèves :

- J'ai testé des nombres différents d'élèves pour voir le nombre de comparaisons
- Pour la génération aléatoire de nombres, je me suis inspiré de ce que j'avais fait dans le TP1 avec les matrices pour générer les coefficients aléatoirement. J'ai donc utilisé le `srand(time(0))` et les `rand()%N` pour avoir des entiers aléatoires entre 0 et N-1.
- Pour la génération aléatoire de caractères, j'ai cherché sur des forums différentes méthodes. J'ai retenu celle dans mon `main.cpp` : écrire 'A' + `rand()%26` génère une majuscule aléatoire entre A et Z, idem avec les minuscules en mettant 'a'.

Foncteurs :

- Je n'étais pas sûr de savoir s'il fallait tout écrire dans le .h ou s'il fallait déclarer les foncteurs dans le .h et les coder dans le .cpp (j'avais en tête le fait que les templates doivent être codés uniquement dans le .h).
- Pour la fonction des foncteurs, je me suis inspiré des slides.

Affichage des élèves :

- J'avais un doute sur la phrase « l'affichage du vecteur trié se fait dans une fonction à part ». Je ne savais pas s'il fallait faire une fonction qui faisait le tri du vecteur et son affichage, ou s'il fallait juste faire une fonction qui affichait le vecteur, et l'utiliser dans le main après avoir fait le tri.
- J'ai retenu plutôt l'idée de faire juste une fonction qui donne un affichage du vecteur, que l'on utilisera après avoir fait les tris.

Ajout des compteurs dans les foncteurs :

- J'ai eu du mal à comprendre comment implémenter le compteur ptr, s'il fallait le mettre à 0 dans la classe ou pas. Je ne l'ai pas fait car j'ai compris qu'on le faisait pointer sur compteur dans le main.
- De même, j'ai eu du mal avec l'idée du `cmp.ptr = &compteur`. Si j'ai bien compris, on initialise une variable compteur (int) à 0. Ensuite, on initialise une variable de la classe CompareNote (par exemple) s'appelant cmp. Ensuite, on indique que l'attribut ptr qui est un pointeur doit pointer sur la variable compteur. Ainsi quand on appelle le foncteur on incrémente la variable pointée par `cmp.ptr`.

Affichage du nombre de comparaisons :

- Il y a légèrement plus de comparaisons pour les noms, ce qui paraît intuitif car il y a plus d'éléments à comparer (les caractères). Mais on peut se dire que si on prend une classe de 20 élèves, la répartition des majuscules sera plutôt uniforme donc on ne comparera que les majuscules en moyenne.
- Sinon, en changeant la note d'un des élèves à 20 pour une classe de 20 élèves, je suis passé de 103 comparaisons à 125 pour les notes. Pour une classe de 30 élèves, je suis passé de 124 à 154 comparaisons.
- Les élèves sont bien triés à l'affichage, tout va bien.