

Note : pour chaque classe écrite, il faudra créer un fichier Classname.h (où se trouvera la déclaration de la classe) et un fichier Classname.cpp qui contiendra l'implémentation. Les tests se feront dans un fichier main.cpp.

La compilation se fait unité de compilation par unité de compilation, plus une phase finale d'édition des liens.

Pour automatiser cela, voir le cours de C++ sur le Makefile.

1 Classe Array

Nous allons faire une classe de tableau de double appelée Array.

1. Implémenter une classe Array qui contient la taille d'un tableau et son pointeur vers les données.
2. Implémenter plusieurs constructeurs :
 - (a) celui par défaut (taille nulle et pointeur nul),
 - (b) qui prend un entier, alloue la mémoire du tableau, et l'initialise à zéro,
 - (c) qui prend un entier et un double. Le tableau est créé et toutes ses valeurs sont égales au double envoyé en argument.
3. Implémenter le destructeur.
4. Implémenter une méthode qui renvoie la taille, puis coder les accesseurs et les mutateurs adéquats.
5. On peut surcharger l'opérateur crochet []. S'en servir pour renvoyer un élément à un indice donné, ou modifier un élément à un indice donné.
6. Surcharger l'opérateur somme + pour concaténer deux Arrays.
7. Est-il nécessaire de faire un opérateur copie ?

2 Nombres aléatoires

1. Que font les fonctions rand() et srand() ?
2. Implémenter une classe RandomArray héritant de Array.
3. Implémenter un constructeur pour générer un tableau de N doubles aléatoires compris entre A et B.
4. Implémenter les méthodes pour calculer la moyenne et l'écart-type du tableau.
5. On cherche à faire une marche aléatoire dans le plan. On part de $(X, Y) = (0, 0)$ et on se déplace de $(+1, 0)$, $(0, -1)$, $(-1, 0)$, $(0, 1)$ avec la même probabilité.
 - (a) Implémenter une classe RandomWalk, reprenant le modèle de classe RandomArray mais en remplaçant le type doubles par le conteneur `std::pair<double, double>`. Le conteneur représentera un vecteur 2D.
 - (b) Implémenter une méthode pour le déplacement 2D, et l'imprimer dans un fichier (afficher la trajectoire avec R ou gnuplot).
 - (c) Calculer la distance quadratique (le carré de la distance au point de départ pour tous les temps).

3 Nombres complexes

1. Implémenter une classe Complex, et coder les constructeurs, destructeur, et autres accesseurs et mutateurs adéquats.
2. Surcharger les opérateurs permettant les opérations arithmétiques élémentaires sur les complexes =, +, -, *, / (note pour les matheux en herbe $z_1/z_2 = z_1\bar{z}_2/(x_2^2 + y_2^2)$ où $\bar{z} = a - ib$ si $z = a + ib$)

4 Fractales

On va calculer plusieurs types de fractales grâce à la classe Complex :

Mandelbrot : Soit la suite dans \mathbb{C} , $U_{n+1} = U_n^2 + c$ pour $U_0 = 0$ et c parcourant le carré $[-2; 2]^2$.

Imprimez les points pour lesquels la suite reste bornée < 4 (pour 200 itérations).

Julia : Même suite mais c est fixe ($c = 0.1$), cette fois c'est U_0 qui parcourt le carré. Même question.