

Examen

Exercice 1 Quels sont les paradigmes supportés par C++?

Exercice 2 Créer une classe `vect` simulant un vecteur de l'espace euclidien de dimension variable. On mettra en attribut privé `int dim` la dimension du vecteur et `double * adr` l'adresse du vecteur (dont les composantes seront de type `double`). Faire les méthodes suivantes pour gérer la partie dynamique et libérer l'espace mémoire : un constructeur prenant la dimension en paramètre, un destructeur, un constructeur de recopie et l'opérateur `=` surdéfini.

Exercice 3 Faire une fonction `compte` renvoyant un pointeur de pointeur sur `int` représentant les n premiers entiers par ligne de taille croissante, par exemple l'appel `compte(4)` affiche :

```
1
1 2
1 2 3
1 2 3 4
```

Exercice 4 On précisera où interviennent les changements dans la classe `vect` de l'exercice 2.

1. Faire les opérateurs binaires `==` et `+` prenant en paramètre des `vect` mais n'étant pas des fonctions membres de `vect`.
2. Ecrire une classe abstraite `boule` avec: deux attributs privés `vect centre` et `double rayon`, et une fonction membre virtuelle `bool contient(point p)`.
3. Faire une classe `boule2` dérivée de `boule` redéfinissant la méthode `contient` testant si le point en paramètre est ou non dans la boule. on rappelle que la distance entre $x = (x_1, \dots, x_n)$ et $y = (y_1, \dots, y_n)$ est $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$. (On pourra supposer que `sqrt()` est disponible).
4. Ecrire une fonction `bool intersecte(boule b1, boule b2)` renvoyant `false` si l'intersection des deux boules est vide.