Travaux Dirigés de C++ n°5

Cours de programmation

—*Licence d'informatique*—

La surcharge d'opérateurs. Les exceptions.

Nous sommes maintenant prêts pour bien comprendre la surchage d'opérateurs. Nous verrons aussi dans ce TD un mécanisme puissant pour signaler et gérer les erreurs: les exceptions.

Dans tout le TD nous retravaillons sur la classe représentant les matrices d'entiers déjà écrite pour le dernier exercice du TD précédent.

▸ **Exercice 1.** *(**set_new_handler**) Ecrire une fonction qui doit être appellée en cas d'erreur d'allocation de mémoire.*

▸ **Exercice 2.** *(**Exceptions**) Modifier la méthode* `get` *pour qu'elle lève une exception si on essaie d'acceder à une position en dehors de la matrice.*

Dans les exercices suivants des exceptions seront levées en cas de besoin.

▸ **Exercice 3.** *(**Exceptions**) Ecrire une méthode* `plusEgal` *sur les matrices dont le rôle est identique à celui qu'on donnerait à l'opérateur* `+=`*. En cas d'erreur, la méthode lèvera une exception. Ecrire une fonction* `main` *dans laquelle on déclare deux matrices* `a` *et* `b` *de tailles différentes et qui contient l'expression* `a.plusEgal(b)`*. Au cas où l'évaluation de l'expression fournit une exception, on affichera un message d'erreur puis on propagera l'exception reçue.*

▸ **Exercice 4.** *(**Surcharge d'opérateurs**) Surchargez les opérateurs* `+=`*,* `<<` *pour qu'ils agissent sur les matrices. Surchargez l'opérateur* `+` *de deux manières différentes (en temps que méthode de la classe des matrices et en temps que fonction). Surchargez le* `-` *unaire. Surchargez l'opérateur* `()` *pour que l'affectation* `m(3,4)=5` *ait le sens qu'on lui devine.*

▸ **Exercice 5.** *(**L'opérateur d'affectation**) Surchargez l'opérateur d'affectation* `=` *pour la classe* `Matrice`*.*

Surchargez le comparateur `==` *pour qu'il teste l'égalité de deux objets de classe* `Matrice`*. L'implantation de cet opérateur devra être symétrique.*

Surchargez les opérateurs `++` *et* `--` *(qui ajoutent et enlèvent* `1` *à chaque élément de la matrice). Comment dissocier* `m++` *de* `++m` *?*