S8-PIG « Programmation des Interfaces graphiques en C++ »

TD #2

1 – Classe Damier statique

Créer une classe *DamierStat* qui permet de manipuler un tableau d'entiers de dimensions (4, 5). Programmez le constructeur et les méthodes nécessaires pour l'usage suivant :

```
int main()
{
    DamierStat D1;            // Création du Damier et init des cases à 0
    D1.Init(7) ;              // Initialisation de toutes les cases à 7
    D1.Set(2, 4, -2) ;        // Modif. de la case (2, 4) avec valeur -2
    D1.Print();               // Affichage du tableau
}
```

Complétez le programme principal, en créant un second objet par allocation dynamique (n'oubliez pas de libérer la mémoire qui lui a été allouée avant de quitter la fin du programme).

Remarque : Cette classe n'est pas très pratique puisque les dimensions du tableau sont codées en dur dans la classe. Pour s'affranchir de cette contrainte, nous devons utiliser une allocation dynamique de mémoire.

2 – Classe Damier dynamique

Il s'agit maintenant de modifier la classe précédente de manière à ce que les dimensions du tableau soient transmises comme des arguments au constructeur (classe *DamierDyn*). Le tableau sera nécessairement alloué de manière dynamique (référez-vous à l'exemple vu en cours).

Vous veillerez à ce que le constructeur initialise le tableau avec la valeur transmise en troisième argument (on fixera une valeur par défaut à cet argument).

Dans ce cas de figure où votre classe manipule de la mémoire allouée dynamiquement, vous DEVEZ reprogrammer un destructeur, un constructeur de recopie et l'opérateur = : ces 3 méthodes existent pas défaut mais elles ne travaillent pas correctement.

Proposez également une méthode privée *DamierDyn::Redim(…)* qui permet de redimensionner dynamiquement le damier (factoriser le code utilisé à plusieurs reprises dans des méthodes privées).

Exemple de programme principal :

```
int main()
{
    DamierDyn D1(4,5);        // Création d'un tableau initialisé à la valeur 0
    D1.Init(7) ;
    D1.Set(2,4, -2);
    DamierDyn D2(D1);         // Constructeur de recopie
    D1.Redim(6,2);            // Redimensionnement de D1
    D2 = D1;                  // Redimensionnement de D2 (à l'image de D1)
}
```

Créez un dernier objet, en utilisant cette fois-ci une allocation dynamique (tout objet, qu'il gère ou non de la mémoire allouée dynamiquement, peut être créé soit de manière statique, soit de manière dynamique).