

Programmation Orientée Objet

TD n°2

Exercice 1

1. Quels seront les résultats fournis par l'exécution du programme suivant (la déclaration de la classe A, sa définition et le programme d'utilisation ont été regroupés en un seul fichier) :

```
#include <iostream>
using namespace std ;

class A
{
private :
    int x, y;
public :
    A();                // constructeur I
    A(int abs);         // constructeur II
    A(int abs, int ord); // constructeur III
    A(const A &) ;      // constructeur IV
    ~A() ;              // destructeur
    void affiche();
} ;

A::A()
{
    x = 1 ; y = 0 ;
    cout << "constructeur I" << " " << x << " " << y << endl ;
}

A::A(int abs)
{
    x = abs ; y = 0 ;
    cout << "constructeur II" << " " << x << " " << y << endl ;
}

A::A(int abs, int ord)
{
    x = abs ; y = ord ;
    cout << "constructeur III" << " " << x << " " << y << endl ;
}

A::A (const A & a)
{
    cout << "constructeur IV (de copie) : " << a.x << " " << a.y << endl ;
    x = a.x ; y = a.y ;
}
```

```

A::~~A ()
{
    cout << "destruction          : " << x << " " << y << endl ;
}

void A::affiche()
{
    cout << "point de coordonnees : " << x << " " << y << endl;
}

void fct (A, A*) ;          // prototype fonction indépendante fct

main ()
{
    cout << "début main" << endl ;
    A a;
    A b(2);
    A c(10,20);
    A d = a ;
    A * adr = new A (3,3) ;
    fct (a, adr) ;
    A e(30,40);
    A tab[6]={4,c,0,e};
    for(int i=0;i<6;i++)
        tab[i].affiche();
    cout << "fin main" << endl ;
}

void fct (A d, A * add)
{
    cout << "entrée fct" << endl ;
    delete add ;
    cout << "sortie fct" << endl ;
}

```

2. Remplacer les constructeurs I, II et III par un seul et même constructeur qui aura les mêmes fonctionnalités.

Exercice 2

On se propose de réaliser une classe nommée **set_int** permettant de manipuler des ensembles de nombres entiers. On devra pouvoir réaliser sur un tel ensemble les opérations classiques suivantes :

- savoir si un entier donné lui appartient,
- lui ajouter un nouvel élément (uniquement s'il n'appartient pas à l'ensemble),
- connaître son cardinal (nombre d'éléments),
- afficher l'ensemble des entiers.

Ici on conservera les différents éléments de l'ensemble dans un tableau alloué dynamiquement par le constructeur. Un argument (auquel on pourra prévoir une valeur par défaut) lui précisera le nombre maximal d'éléments de l'ensemble.

1. Ecrire le code C++ permettant de coder une telle classe.
2. Ecrire un programme utilisant la classe `set_int` pour déterminer le nombre d'entiers différents contenus dans 20 entiers saisis au clavier.
3. Que faudrait-il faire pour qu'un objet du type `set_int` puisse être transmis par valeur, soit comme argument d'appel, soit comme valeur de retour d'une fonction ?
4. On considère à présent deux ensembles d'entiers. On désire ajouter membre à membre les valeurs contenues dans le deuxième ensemble à celles contenues dans le premier. Proposer deux fonctions différentes permettant cette opération :
 - une méthode de la classe `set_int` prenant un paramètre de type `set_int`,
 - une fonction extérieure à la classe `set_int` qui prenant deux paramètres de type `set_int`.

Pour cette dernière, les attributs de la classe `set_int` étant privés, on devra ajouter à la classe `set_int` des méthodes `Set` et `Get` pour chacun des attributs permettant d'accéder aux attributs pour les lire ou les modifier. Dans le cas d'un tableau, on accèdera à un seul élément du tableau dont l'indice sera passé en paramètre.
5. Comment pourrait-on adapter la classe `set_int` pour qu'elle dispose d'une fonction membre `Number_set` fournissant le nombre d'ensembles existant à un instant donné ?

Exercice 3

Créer une classe *point* ne contenant qu'un constructeur sans arguments, un destructeur et un membre donné privé présentant un numéro de point (le premier crée portera le numéro 1, le suivant le numéro 2..). Le constructeur affichera le numéro du point créé et le destructeur affichera le numéro du point détruit. Écrire un petit programme d'utilisation créant dynamiquement un tableau de 4 points et le détruisant.