

Exercice 74

Énoncé

Comment concevoir le type classe `chose` de façon que ce petit programme :

```
main()
{
    chose x ;
    cout << "bonjour\n" ;
}
```

fournisse les résultats suivants :

```
création objet de type chose
bonjour
destruction objet de type chose
```

Que fournira alors l'exécution de ce programme (utilisant le même type `chose`) :

```
main()
{
    chose * adc = new chose
}
```

Exercice 75

Énoncé

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

```
#include <iostream>
using namespace std ;
class demo
{
    int x, y ;
public :
    demo (int abs=1, int ord=0)    // constructeur I (0, 1 ou 2 arguments)
    {
        x = abs ; y = ord ;
        cout << "constructeur I      : " << x << " " << y << "\n" ;
    }
    demo (demo &) ;                // constructeur II (par recopie)
    ~demo () ;                     // destructeur
} ;

demo::demo (demo & d)              // ou demo::demo (const demo & d)
{
    cout << "constructeur II (recopie) : " << d.x << " " << d.y << "\n" ;
    x = d.x ; y = d.y ;
}
demo::~~demo ()
{
    cout << "destruction              : " << x << " " << y << "\n" ;
}

main ()
{
    void fct (demo, demo *) ;      // proto fonction indépendante fct
    cout << "début main\n" ;
    demo a ;
    demo b = 2 ;
    demo c = a ;
    demo * adr = new demo (3,3) ;
```

```

    fct (a, adr) ;
    demo d = demo (4,4) ;
    c = demo (5,5) ;
    cout << "fin main\n" ;
}
void fct (demo d, demo * add)
{
    cout << "entrée fct\n" ;
    delete add ;
    cout << "sortie fct\n" ;
}

```

Exercice 76

Énoncé

Créer une classe `point` ne contenant qu'un constructeur sans arguments, un destructeur et un membre donnée privé représentant un numéro de point (le premier créé 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.

Exercice 77

Énoncé

1. 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 : lui ajouter un nouvel élément, connaître son cardinal (nombre d'éléments), savoir si un entier donné lui appartient.

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.

2. Écrire, en outre, un programme (`main`) utilisant la classe `set_int` pour déterminer le nombre d'entiers différents contenus dans 20 entiers lus en données.
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 ?

N.B. Le chapitre 17 vous montrera comment résoudre cet exercice à l'aide des composants standard introduits par la norme, qu'il ne faut pas chercher à utiliser ici.