

Déclaration, définition et utilisation des variables en C++

Rappel :

Déclaration et utilisation des variables :

```
int i ;      =>    i=0 ;
int j = 5 ;
int fctplus ( int a , int b ) ;
           =>    int res  = fctplus ( i , j ) ;
```

Déclaration et utilisation des pointeurs :

```
int i ;      =>    i=0 ;
int j = 5;
int pti = NULL;
int ptj = &j;
int *pt = new int(9);
i = *ptj ;
pti = &i;
int fctplus ( int *a , int *b ) ;
           =>    int res  = fctplus ( &i , &j ) ;
           =>    int res  = fctplus ( pti , ptj ) ;
```

Les références :

Les *références* sont un outil permettant de générer un *alias* (étiquette) sur une variable ou un objet. Il s'agit d'une abstraction plus sécurisée et de plus haut niveau que les pointeurs. En effet, contrairement à ces derniers, une référence :

- Doit être obligatoirement initialisée dès sa création ;
- ne peut référencer un autre objet que celui auquel elle a été attachée lors de sa création.

Déclaration d'une référence : utilisation du 'et commercial' &
Type & nom = val ;

```
int i= 8 ;
int &refi = i ;
nbCmpx cmp1 ( 4 , 5 );
nbCmpx &refcmp = cmp1 ;
```

Passage de paramètre à une fonction ou méthode:

```
int fctplus ( int &a , int &b ) ;
int i=5 , j=6 ;
int & refi = i , refj = j ;
           =>    int res  = fctplus ( i , j ) ;
           =>    int res  = fctplus ( refi , refj ) ;
```

Mise en œuvre :

Analyser les codes ci-dessous et indiquer les résultats d'exécution dans le dessin de la console.

```
class nbcmpx
{
public :
    int x,y;
    nbcmpx(int a , int b){x=a ; y=b;}
    affiche(){cout << "\n  x = " << x << "\t\t y = " << y << endl;}
    change(int a , int b){x=a;y=b;}
    change(int *a , int * b){x=*a ; y=*b;}
    change(int &a , int &b){x=a ; y=b;}
};

void echange(int & refa , int &refb)
{
    int tmp;
    tmp = refa ;
    refa = refb;
    refb = tmp;
}

ftest1()
{
    int i=5;
    int j=45;
    int &refi = i;
    refi++;
    cout << "1) i = " << i << "\t j = " << j << "\t\t refi = " << refi << endl<< endl;
    refi = j;
    refi++;
    cout << "2) i = " << i << "\t j = " << j << "\t\t refi = " << refi << endl<< endl;
    int & refj = j;
    j = 3;
    cout << "3) i = " << i << "\t j = " << j << "\t\t refi = " << refi << "\t\t refj=" <<
    refj << endl<< endl;
    echange(i , j);
    cout << "4) i = " << i << "\t j = " << j << "\t\t refi = " << refi << "\t\t refj=" <<
    refj << endl<< endl;
}

void ftest2()
{
    nbcmpx n2(10 , 12);
    nbcmpx & ref1 = *new nbcmpx(14,39);
    nbcmpx &ref2 = n2;
    ref1.affiche();
    ref2.affiche();
    ref2 = ref1;
    ref2.affiche();
    n2.change(50,51);
    ref2.affiche();
}
```

Compléter ci-dessous le résultat de l'exécution de la fonction ftest1 :

```
C:\profil\moureaux\cpp2007-2008\Preference.exe
1> i =          j =          ref i =
2> i =          j =          ref i =
3> i =          j =          ref i =          ref j=
4> i =          j =          ref i =          ref j=
```

Compléter ci-dessous le résultat de l'exécution de la fonction ftest2 :

```
C:\profil\moureaux\cpp2007-2008\Preference.exe
x =          y =
x =          y =
x =          y =
x =          y =
```

Résultat :

```
C:\profil\moureaux\cpp2007-2008\Preference.exe
1> i = 6      j = 45      ref i = 6
2> i = 46     j = 45      ref i = 46
3> i = 46     j = 3       ref i = 46      ref j=3
4> i = 3      j = 46      ref i = 3      ref j=46
```

```
C:\profil\moureaux\cpp2007-2008\Preference.exe
x = 14      y = 39
x = 10      y = 12
x = 14      y = 39
x = 50      y = 51
```