

## Laboratoire n°7 en c++ : héritage simple

Réaliser une classe ligne. Pour la créer, utiliser 3 fichiers :

- un fichier d'en-tête contenant la déclaration de la classe
- un fichier d'implémentation contenant les définitions de la classe
- un fichier d'utilisation contenant le programme principal.

La classe ligne contient :

- 4 coordonnées réelles : x1, y1 (coordonnées de l'origine de la ligne) et x2,y2 (coordonnées de l'extrémité de la ligne)
- un constructeur initialisant les données membres. Par défaut le point d'origine aura pour coordonnées (0,0) et le point d'extrémité aura pour coordonnées (10,10)
- une fonction affiche permettant d'afficher les coordonnées de la ligne
- une surcharge de l'opérateur == permettant de comparer 2 lignes et retournant true ou false
- une surcharge de l'opérateur > permettant de comparer la longueur de 2 lignes et retournant true (si la longueur de la première ligne est plus grande que la deuxième), sinon false
- une fonction permettant de connaître à tout moment le nombre total d'objets créés de la classe ligne.

Créer une classe lignestyle dérivée de la classe ligne comportant :

- 2 données membres : epaisseur (épaisseur de la ligne, un entier) et style (style de la ligne, une chaîne de caractères)
- un constructeur initialisant les données membres. Par défaut, l'épaisseur sera 2 et le style sera continu.
- une méthode affiche qui affiche les coordonnées, l'épaisseur et le style de la ligne.

Créer un programme qui :

- crée des objets lignes et teste les fonctions de la classe ligne
- affiche le nombre d'objets créés
- crée des objets lignestyle et teste les fonctions de la classe lignestyle.

```

#include<iostream.h>
#include<math.h>
#include<string.h>

class ligne{
protected:    float x1,y1;
               float x2,y2;
public:       ligne(float xx1=0.0F,float yy1=0.0F,float xx2=10.0F,float yy2=10.0F);
               void afficher();
               friend bool operator==(ligne l1,ligne l2);
               friend bool operator>(ligne l1,ligne l2);
               static int j;
               static int get_value();
               ~ligne();
};
class lignestyle:public ligne{
    int epaisseur;
    char style[20];
public:    lignestyle(int ep=2,char* st="continu");
           void affiche();
};

//définitions

#include"labo7.h"

int ligne::j=0; //init. de la variable static

ligne::ligne(float xx1,float yy1,float xx2,float yy2){
    x1=xx1;
    y1=yy1;
    x2=xx2;
    y2=yy2;
    ++j;
}
void ligne::afficher(){
    cout<<"voici les coordonnees de la ligne:"<<endl;
    cout<<"origine de la ligne:("<<x1<<","<<y1<<")"<<endl;
    cout<<"extremite de la ligne:("<<x2<<","<<y2<<")"<<endl;
}
bool operator==(ligne l1,ligne l2){
    if(l1.x1==l2.x1&&l1.y1==l2.y1&&l1.x2==l2.x2&&l2.y2==l1.y2) return true;
    else return false;
}
bool operator>(ligne l1,ligne l2){
    if(sqrt(pow((l1.x2-l1.x1),2)+pow((l1.y2-l1.y1),2))>sqrt(pow((l2.x2-
l2.x1),2)+pow((l2.y2-l2.y1),2))) return true;
    else return false;
}
}

```

```

int ligne::get_value(){
    return j;
}
ligne::~ligne(){
    --j;
}
lignestyle::lignestyle(int ep,char st[20]){
    epaisseur=ep;
    strcpy(style,st);
}
void lignestyle::affiche(){
    cout<<"("<<x1<<","<<y1<<)"<<" et ("<<x2<<","<<y2<<)"<<endl;
    cout<<"epaisseur:"<<epaisseur<<endl;
    cout<<"style:"<<style<<endl;
}

```

```
//programme principal
```

```
#include"labo7.h"
```

```

void main()
{
    ligne l1;
    cout<<"nombres d'objet(s) cree(s):"<<l1.get_value()<<endl;
    ligne l2;
    cout<<"nombres d'objet(s) cree(s):"<<l1.get_value()<<endl;

```

```

l1.affiche();
l2.affiche();

```

```

if((l1==l2)==true)
{
    cout<<"les lignes sont confondues"<<endl;
}
else cout<<"les lignes sont differentes"<<endl;

```

```
cout<<"comparaison des longueurs:"<<endl;
```

```

if((l1>l2)==true){
    cout<<"la ligne 1 > ligne 2"<<endl;
}
else cout<<"la ligne 2 > ligne 1"<<endl;
cout<<endl;

```

```

lignestyle lt1(4,"discontinu");
cout<<"style de la ligne:\n";
lt1.affiche();
}

```