

Héritage

- Point et Point3D

Soit la classe Point

```
class Point
{
public:
    Point(int,int);
    void afficher();
    int getX();
    int getY();
    void setX(int);
    void setY(int);
    void faireTranslation(int, int);
private:
    int x_, y_;
};
```

Soit la classe Point3D

```
class Point3D:public Point
{
public:
    Point3D(int, int,int);
    void afficher();
    void faireTranslation(int, int,int);
    void faireMiseAEchelle(int scaling);
    int getZ();
    void setZ(int);
private:
    int z_;
};
```

Soit le programme principal

```
int main()
{
    Point P1(2,3);
    P1.afficher ();
    P1.faireTranslation(2,5);
    cout <<"X " << P1.getX()<<endl;
    cout <<"Y " << P1.getY()<< endl;

    Point3D P2(1,2,3);

    P2.afficher();
    P2.faireTranslation(1,1,1);
    cout << "X " << P2.getX()<<endl;
    cout << "Y " << P2.getY()<< endl;
    cout << "Z " << P2.getZ()<< endl;

    P2.faireMiseAEchelle (4);
    P2.afficher();

    // exécuter la méthode afficher de Point
    P2.Point::afficher();
```

Sortie du programme principal

```
X 2
Y 3
X 4
Y 8
X 1
Y 2
Z 3
X 2
Y 3
Z 4
X 8
Y 12
Z 16
X 8
Y 12
Press any key to continue . . .
```

Question 1

Écrire l'implémentation de la classe Point

Question 2

Écrire l'implémentation de la classe Point 3D sachant que

- le constructeur de Point3D doit appeler celui de Point.
- la fonction membre afficher() de Point3D doit appeler celle de Point et ajouter des instructions pour afficher la valeur de l'attribut z_
- la fonction membre faireTranslation(*) doit appeler celle de Point et ajouter les instructions pour déplacer la valeur de l'attribut z_;
- la fonction membre faireMiseAEchelle() appartient seulement à la classe Point3D et multiplie la valeurs des attributs par le paramètre scaling.
- les fonctions membres getZ() et setZ();

Faire exécuter vos implémentations des classes avec le programme principal donné. Vous devez obtenir le même résultat.