

Introduction à la POO : « is a », notion d'héritage

I. Préambule :

Soit le code suivant :

```
class Point
{
    private float x, y;

    public Point(float ax, float ay)
    {
        x = ax;
        y = ay;
    }
    public Point( Point aP)
    {
        x = aP.x;
        y = aP.y;
    }

    public float donneX()
    {
        return x;
    }
    public float donneY()
    {
        return y;
    }

    public void Dessine(Gdk.Window wnd, Gdk.GC gc)
    {
        wnd.DrawLine( gc, this.x, this.y - 10, this.x, this.y + 10);
        wnd.DrawLine( gc, this.x - 10, this.y, this.x + 10, this.y);
    }
}
```

```

class Cercle
{
    protected Point c; // centre du cercle
    protected float r; // rayon

    public Cercle(Point aP, float aR)
    {
        c = new Point( aP);
        r = aR;
    }
    public void modifCentre(float ax, float ay)
    {
        c.modif(ax, ay);
    }
    public virtual void Dessine(Gdk.Window wnd, Gdk.GC gc)
    {
        wnd.DrawArc( gc, c.donneX() - r, c.donneY() - r, 2 * r, 2 * r, 0, 360);
    }
}

```

Un Cercle de centre c et de rayon r est l'ensemble des point qui sont à la distance r du point c.

Travail à faire :

- Tracer le diagramme de classe de ce code.
- Créer une nouvelle classe appelée Disque.
 - i. Solution 1 : reprendre le code du cercle en remplaçant DrawArc par FillEllipse
 - ii. Utiliser le mécanisme d'héritage. :
un disque est un cercle que l'on remplit avec un pinceau.

NB : si je construis mon code Disque en utilisant le copier/Coller (i) alors quand je vais modifier le code de Cercle, il faudra aussi penser à modifier le code de Disque.

Un disque est un cercle qui est plein. Tous ce qui est dans le disque et qu'on retrouve dans le cercle est enlevé de la déclaration de la classe disque. Et on indique que la classe disque hérite de la classe Cercle. Dans la classe cercle les attributs privés utilisé dans le Disque passent de privé à protégé.

L'application Windows construite autour de cette classe Segment se trouve [ici](#).

Pour la lire, installer MonoDevelop.

II. Définitions

II.1 Héritage du point de vue du programme

Une classe est **dérivée** ou **hérite** d'une classe de **base** si elle a :

- au moins les attributs de la classe de base, même si elle n'y a pas accès.
- Au moins les méthodes de la classe de base, même si elle n'y a pas accès.

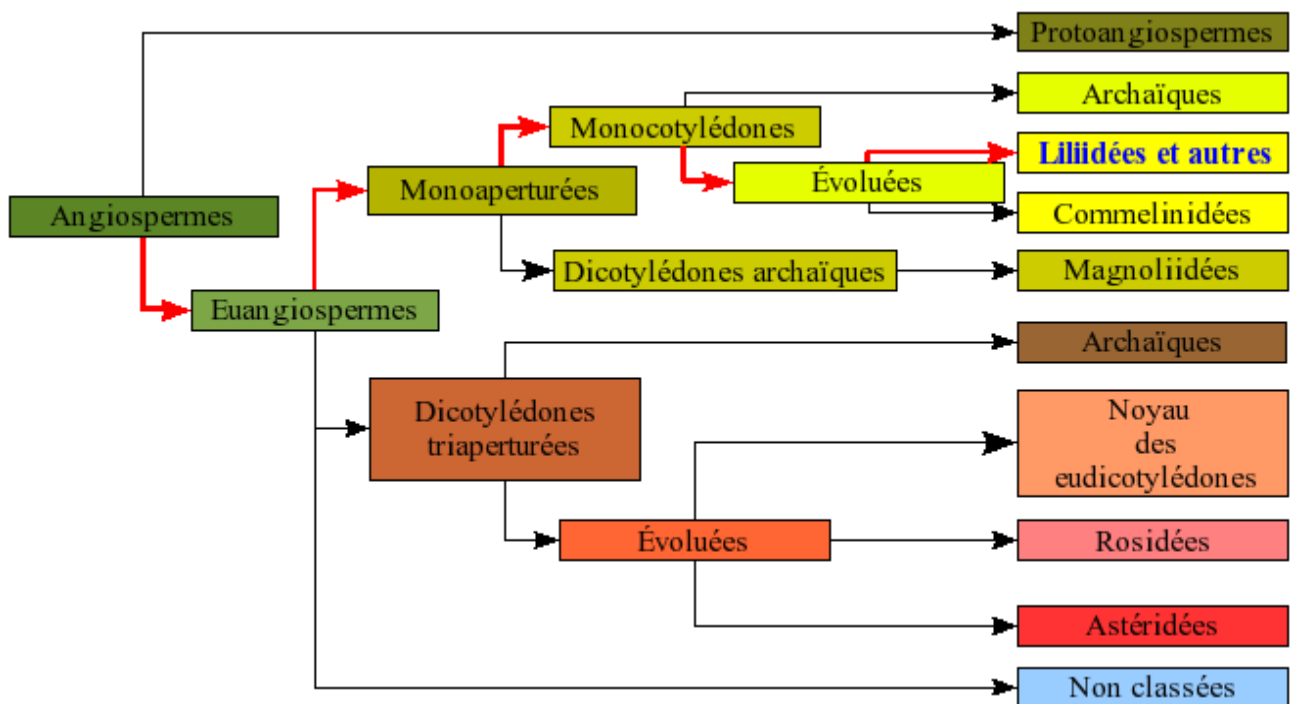
II.2 Héritage du point de vue de l'analyse

Permet la manipulation de concepts abstraits.

Une classe est **est dérivée** ou **hérite** d'une classe de **base** si elle constitue une spécialisation de la classe de base.

Ce concept permet de classifier les objets.

Exemple des classification des êtres vivants :



(source Wikipédia)

cf http://fr.wikipedia.org/wiki/Classification_phylogénétique

III. Implémentation en C#

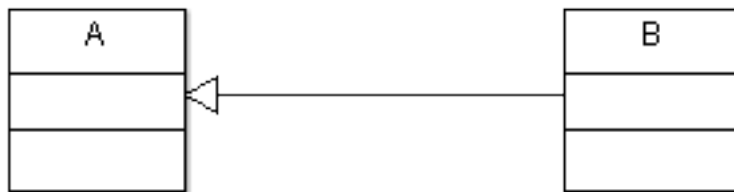
En C# ou java, il y a une seule façon de faire.

Pour C# : dans la déclaration de classe utilisation des « : »

Pour Java : utiliser le mot clef « extends ».

Avec UML, on utilise un lien avec une flèche en direction de la classe de base.

Ici, B hérite de A :



IV. Exemples

Diagramme de classe en utilisant le logiciel de votre choix (Visio, ArgoUML).

1. Un élève est une personne qui est inscrit dans une promotion. Un professeur est une personne qui assure des cours pour plusieurs promotions.
2. Les moyens de transports :
 - Un vélo, une moto sont des deux roues.
 - Un deux roues, une voiture, un camion sont des véhicules terrestre.
 - Un voilier, un scooter de mer, une péniche sont des bateaux.
 - Un avion, un bateau, un véhicule terrestre sont des moyens de transports.
3. Les volumes : essayer de construire une modélisation des différents volumes que vous connaissez.
4. ...