**Notes de cours sur les interfaces**

* Ce ne sont pas des classes mais des interfaces ( public interface Modele )
* Les interfaces sont uniquement constituées de :
  + méthodes abstraites publiques ( pas nécessaire d'écrire abstract )
  + constantes
* Les interfaces permettent l'héritage multiple

**Héritage**

* **Jusqu'à présent, héritage simple : une sous-classe ne peut hériter d'une seule superclasse :**
* Dans la réalité, on peut cependant concevoir un objet selon différents aspects :
  + Une Personne peut être perçue comme un Client, un Patient, une Proie facile
  + Une Voiture peut être perçue comme un Bien taxable ou comme un moyen de transport.
* Pour représenter un "semblant" d'héritage multiple … une classe peut mettre en œuvre 0,1 ou plusieurs interfaces

**EX Modélisation de l'inventaire d'une chaîne de magasins**

*1public abstract class Article*

*private String nom;*

*private double prix;*

*private String codeBarres;*

*public abstract int calculerTempssurTablettes {} ;*

*6public interface Remboursable*

*public boolean estRemboursable();*

*4public interface Taxable*

*Public static final POURCENTAGE\_TAXE = 15.2%*

*public double calculerTaxe();*

*5public class CoupeCheveux implements Taxable*

*private String nom;*

*private double prix;*

*public double calculerTaxe(){…}*

*3public class PinteLait extends Article*

*private GregorianCalendar dateExpiration;*

*public int calculerTempssurTablettes ()*

*2public class Jouet extends Article implements Taxable, Remboursable*

*private int ageMinimum;*

*public int calculerTempssurTablettes ()*

*public double calculerTaxe(){*

*return getPrix() \* Taxable.POURCENTAGE\_TAX;*

*}*

*public boolean estRemboursable();*

**Interfaces : but**

* **Spécifier un ensemble de fonctionnalités à couvrir**
* **Description d'une capacité ( Comparable, Assurable… )**