
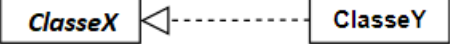
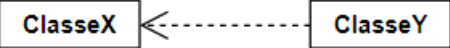






Héritage 	<pre>class ClasseY : public ClasseX { // ClasseY hérite des attributs et méthodes de ClasseX ... // avec les mêmes propriétés d'accès public private protected }</pre>
Réalisation / Implémentation 	<p>La <i>ClasseX</i> est une classe abstraite, donc avec au moins une méthode <i>virtual pure</i>, c.a.d. définie par =0 ; La <i>ClasseX</i> définit une interface, pas d'instanciation possible. La <i>ClasseY</i> hérite de la <i>ClasseX</i> et implémente les méthodes <i>virtual pures</i> définies dans la <i>ClasseX</i>. Instanciation possible de <i>ClasseY</i>.</p>
Dépendance 	<pre>class ClasseY { // ClasseY utilise et dépend de ClasseX lors de ses calculs public: /// METHODES void methodeDeY () { ClasseX* jUtiliseUnXlocalement; ... jUtiliseUnXlocalement -> methodeDeX(); } }</pre>
Association unilatérale 	<pre>class ClasseY { // Association unilatérale: ClasseY connaît ClasseX ClasseX* monPoteX; // implémente l'association unilatérale } // attention aux éventuelles multiplicités</pre>
Association bilatérale 	<pre>class ClasseY { // Association unilatérale : ClasseY connaît ClasseX ClasseX* monPoteX; // implémente l'association <u>dans un sens</u> } // attention aux éventuelles multiplicités class ClasseX { // <u>plus</u> association unilatérale : ClasseX connaît ClasseY ClasseY* monPoteY; // implémente l'association <u>dans l'autre sens</u> } // attention aux éventuelles multiplicités</pre>
Agrégation 	<pre>class ClasseY { // Agrégation (dit assemblage <u>faible</u>). Les composants list<ClasseX*> lesComposants; // existent <u>hors</u> de l'objet. Leurs adresses sont enregistrées. } //cf. ajouterComposant(ClasseX*), supprimerComposant(ClasseX*), existeComposant(ClasseX*), ~ClasseY()</pre>
Composition 	<pre>class ClasseY { // Composition (dite assemblage <u>forte</u>), <u>ses</u> composants list<ClasseX> mesComposants; // peuvent exister <u>dans</u> l'objet. Les objets sont dits <u>embarqués</u>. } //cf. ajouterComposant(ClasseX), supprimerComposant(ClasseX), existeComposant(ClasseX), ~ClasseY()</pre>

