

Héritage

L'héritage est un principe permettant de créer une nouvelle classe à partir d'une classe existante. Le nom d'héritage provient du fait que la classe enfant contient les attributs et les méthodes de sa classe parent. L'intérêt de l'héritage est de pouvoir définir de nouveaux attributs et de nouvelles méthodes pour la classe enfant, qui viennent s'ajouter à ceux de la classe parent.

L'Héritage permet donc de prendre une classe globale et de la spécialisée de plus en plus. Comme par exemple prendre la classe Humain (parent) lui faire hérités les différents membres du corps sur la classe enfant, ainsi si l'ont créer la classe homme et femme (enfants), les deux classes auront les parties du corps du parent.

Composition

La composition est un cas particulier de l'agrégation, qui implique une inclusion plus forte de l'objet enfant dans l'objet parent. Alors qu'un même objet peut être agrégé par plusieurs objets différents, un même objet ne peut pas être "composé" (ou partagé) par plusieurs objets. Il y a de plus une notion de durée de vie commune entre un composant et ses composés : les composés d'un composant sont en général détruits en même temps que le composant.

Argumentaire

L'héritage est en effet une manière simple de partager les fonctionnalités d'une classe parent à une classe enfant sans trop de soucis. Certes, mais le problème c'est que si l'ont fait que des héritage le code deviendra de moins en moins flexible car si à un moment le code doit être changé dans une classe parent cela veux dire que soit la classe enfant va hérité de toute la classe parent et nous auront des données qui ne nous serviront pas, soit nous seront obligés de refaire de nouvelles classes et de réorganiser le code pour que le principe de fonctionnement ne change pas.

En occurrence nous utilisons justement la composition, cela va rendre le code plus flexible et ce ne sera pas lier directement à la classe contrairement à l'héritage. Je pense que la flexibilité est un très grand atout du coup on privilégieras cette méthode qui permettra d'obtenir un code propre