Les design patterns

Aide à conception :

- les principes SOLID, s'applique sur les objets en général (s'applique sur api et implémentation)
- sémantique : distingue 2 types de classes (s'applique sur l'api)
- design pattern : briques élémentaires, généralement constituée de plusieurs classes

DP forme en quelques sorte des briques élémentaires pour résoudre une problématique particulière.

Il faut être capable de 2 choses :

- face à une problématique, voir les DP qui peuvent s'appliquer (ie faciliter la conception du code) ;
- face à un code, reconnaître s'il y a un/des DP utilisés et lesquels (et surtout éviter de “casser” le DP, ce qui réduirait la qualité du code).

Le second point permet de faciliter l'échange entre les devs, en leur donnant un cadre de codage commun. Quand on reconnait un DP dans un code, on comprend tout de suite la problématique qui est abordé et donc ce que le code fait

Classification

Historiquement, on distinguer les premiers DP, issu de GoF, des autres DP plus récent.
software DP

- création : créer des objets, factory et builder
- structurel : proxy, wrapper, bridge, composite, décorateur, Flyweight facade
- comportement : visiteur (cf http://accu.org/index.php/journals/2021), stratégie, state, chaine responsability

Architecture DP : http://en.wikipedia.org/wiki/Architectural_pattern

- MVC, SOA, ECS

Chapitre précédent	Sommaire principal	Chapitre suivant

Cours, C++