

SOLID

DEPENDENCY INVERTING PRINCIPLE

MAGNUM2-QUALITÉ D'SI-THOMAS HAESLÉ & QUENTIN BURG

Viser toujours à atteindre un couplage faible dans les logiciels.

Les abstractions de « haut niveau » ne doivent pas dépendre des abstractions « bas niveau »

Les abstractions ne doivent pas dépendre de détails, mais les détails d'implémentation doivent dépendre des abstractions.

Indissociable du I, cela se traduit par :

- **La mise en oeuvre de l'injection de dépendance**

INDÉPENDANTES

SOLID

DEPENDENCY INVERSION PRINCIPLE

Vise toujours à atteindre un couplage faible dans les logiciels.

Les abstractions de « haut niveau » ne doivent pas dépendre des abstractions « bas niveau »

Les abstractions ne doivent pas dépendre de détails, mais les détails d'implémentation doivent dépendre des abstractions.

Indissociable du I, cela se traduit par :

- La mise en oeuvre de l'injection de dépendance

INDÉPENDANT DES IDIOMES

CLEAN / ONION

UNCLE BOB ARCHITECTURE / JEFFREY PALERMO ARCHITECTURE

Objectifs est d'avoir un logiciel :

- Indépendant des frameworks
- Testable
- Indépendant de l'UI
- Indépendant de la DB
- Indépendant des API externes

