

Pt12 SOLID

Al fitxer adjunt es proposen unes implementacions inicials que conté alguna violació dels principis SOLID, a més d'altres males pràctiques. Es tracta d'identificar quin o quins principis s'estan violant i proposar una implementació alternativa que segueixi aquests principis, sempre des de la perspectiva de que el codi és dinàmic, que canvia amb el temps amb la incorporació de noves funcionalitats.

Cada exercici es troba un package i conté un test anomenat Client.java amb una sortida. Després de fer el refactoring, aquesta sortida hauria de ser la mateixa.

Poseu amb comentaris dins del codi quins principis s'estan violant. Podeu crear les classes i interfícies que creieu convenient.

Creeu un projecte anomenat solid, i per cada Exemple creeu un package anomenat ex1, ex2, etc, amb la refactorització i amb un test per provar la nova implementació.

A l'hora d'avaluar els Exemples es tindrà en compte les bones pràctiques que hem vist fins ara a M03 i M05.

SOLID no és cap regla, com per exemple java naming conventions, són unes recomanacions perquè el nostre codi sigui més mantenible (fàcil d'utilitzar i estendre, etc), però dependrà molt del context en què ens trobem. Pensem també que quantes més classes tingui el nostre sistema més complex es torna. Són aspectes que s'han de valorar segons el cas específic. Farem alguns exemples a classe i ho parlarem.

Per simplificar els exercicis, cada exercici ha de resoldre un únic principi:

- ex1: Single Responsibility Principle
- ex2: Open/Closed Principle
- ex3: Interface Segregation Principle
- ex4: Liskov Substitution Principle
- ex5: Dependency Inversion Principle