TP2 – UQAM

Principes SOLID :

* Single Responsibility Principle (SRP) – Principe de responsabilité unique :
  + **Une classe ne doit avoir qu'une seule raison de changer**.
  + Autrement dit, une classe doit avoir une seule responsabilité.
* **Open/Closed Principle (OCP) – Principe ouvert/fermé**
  + **Les entités logicielles (classes, modules, fonctions) doivent être ouvertes à l'extension mais fermées à la modification**.
  + Cela signifie que nous devons pouvoir ajouter de nouvelles fonctionnalités **sans modifier le code existant**.
* Liskov Substitution Principle (LSP) – Principe de substitution de Liskov
  + **Une classe dérivée doit pouvoir être utilisée comme sa classe de base sans affecter le bon fonctionnement du programme**.
* Interface Segregation Principle (ISP) – Principe de ségrégation des interfaces
  + **Une classe ne doit pas être forcée d’implémenter des méthodes qu’elle n’utilise pas**:
    - Cela signifie qu’il est préférable d’avoir **plusieurs interfaces spécifiques** plutôt qu’une seule grosse interface.
* Dependency Inversion Principle (DIP) – Principe d'inversion des dépendances
  + Les modules de haut niveau ne doivent pas dépendre des modules de bas niveau. Les deux doivent dépendre d'abstractions :
    - Autrement dit, il vaut mieux **programmer contre une interface** plutôt qu'une implémentation concrète.