Open Closed Principle

Overview

- Bertrand Meyer proposed the open-closed principle (OCP)
  - classes and methods should be Open for extension (new functionality) and Closed for modification
  - a class should be easily extendable without modifying the class itself
- a module is said to be open if it is still available for extension
  - it should be possible to add fields to the data structures it contains, or new elements to the set of functions it performs
- a module is said to be closed if it is available for use by other modules
  - assumes that the module has been given a well-defined, stable description
  - the interface in the sense of information hiding (not a java interface)
- general idea of this principle is that it tells you to write your code so that you will be able to add new functionality without changing the existing code
  - prevents situations in which a change to one of your classes also requires you to adapt all depending classes
  - reduces tight coupling
- Robert C. Martin considered this principle as the “the most important principle of object-oriented design”

Overview (cont’d)

- unfortunately, Bertrand Mayer proposed the use of inheritance to achieve the open/closed principle
- however, inheritance introduces tight coupling if the subclasses depend on implementation details of their parent class
- others redefined the Open/Closed Principle to the Polymorphic Open/Closed Principle
  - uses interfaces instead of super classes to allow different implementations
  - interfaces can be reused through inheritance but implementation need not be
  - can easily substitute without changing the code that uses them
    - multiple implementations can be created and polymorphically substituted for each other
- interfaces are closed for modifications
  - you can provide new implementations to extend the functionality of your software
  - new implementations must implement the interface
- interfaces introduce an additional level of abstraction which enables loose coupling
  - interfaces are independent of each other and don’t need to share any code (usually)