DESIGN PATTERNS

Clement Jim Jeffrey

Design Pattern:

A design pattern is a general **repeatable** solution to a **commonly-occurring problem** in software design.

Why design pattern

- So you think you can write good OO programs?
- To reuse ancient's wisdom on software design
  - More flexible code
  - Avoid the pitfalls
- To communicate more effectively

Gof and design pattern

- Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, the so called “Gang of four”

- The book is currently the 36th print since 1994

Object Orientation Concepts

- encapsulation
  - information hiding
  - separation of interface/implementation
  - public/protected/private attributes/methods
  - control of what level of access to the object
    - read/write access to the property
    - visible to self/derived classes/every class/friends/package

Object Orientation Concepts

- polymorphism
  - use the same interface to handle different types
  - C++ virtual functions or Java non-static methods
  - the method invoked depends on the object type being referenced at runtime
  - Late binding/dynamic binding

Object Orientation Concepts

- Inheritance
  - the “is-a” relationship
  - generalization/specialization
  - code sharing in base class
  - function extension in derived class

Design Principles

- depend on interface, not implementation
  - loose coupling between objects
- prefer delegation to inheritance
  - inherited class has more responsibility than user/client class
- polymorphism instead of control structures
  - nested and/or scattered control structures are inflexible

What are Design Patterns

- Proven solutions to recurring design problems
  - Proven: they are really applied in the field and work
  - Recurring design problems: the problems will occur again and again
- With design patterns, you don't have to reinvent the wheel
- Design patterns provide good solutions, not functionally correct solutions

What to Expect from Design Patterns

- A common design vocabulary
  - just like Linked Lists in data structures or Quick Sort in algorithms
- A documentation and learning aid
  - learning design patterns help you understand designs in real systems and make better design
  - documentation using design patterns are easier to write and understand

What to Expect from Design Patterns

- An adjunct to existing methods
  - design patterns show how to use OO constructs effectively
  - provide a smooth transition from analysis to design and then to implementation
- A target for refactoring
  - refactor to patterns