

Design Pattern Study

daybreaker@SPARCS, KAIST

2007. 4. 9

Prerequisites

- Java OOP에 대한 이해 (class, interface, ...)
 - Not java language itself!
- CS230 System Programming 이상의 **삽질 경험**
- C/C++ 기초 약간
- UML notation : textbook 부록 참조

Contents

- What is Design Pattern?
- Study Plan

What is Design Pattern?

- 경험 많은 프로그래머들의 비법(?)
 - 이렇게 하니까 좋더라~
 - 자주 사용되는 문제들을 해결하는 핵심
- 왜 사용하나?
 - Class Design의 어려움
 - 매번 다시 고민(삽질)하기 싫어서

What is Design Pattern?

- Pattern의 구성요소
 - Pattern Name
 - 일종의 vocabulary 같은 것
 - Problem
 - Pattern을 적용할 대상
 - Solution
 - Design, relationships, consequences, responsibilities of elements
 - Template과 같아서 구체적인(specific) 해답을 제시하지는 않음
 - Consequences
 - Pattern 적용 결과
 - Trade-offs

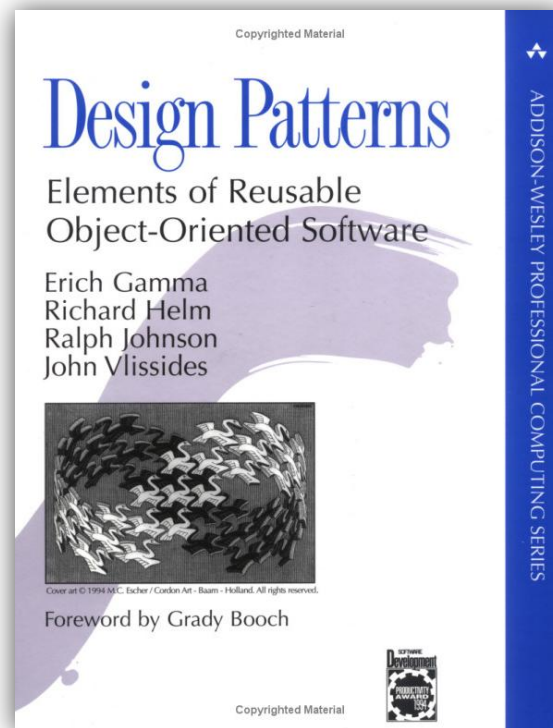
Descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context

What is Design Pattern?

- Categories
 - Creational
 - Abstraction of instantiation process
 - creation
 - Structural
 - How classes/objects are composed for larger structure
 - composition
 - Behavioral
 - Algorithms/assignments of responsibilities between objects
 - communication, interconnection

Study Plan

- Textbook
 - Design Patterns, Elements of Reusable Object Oriented Software
 - 구입 : 강컴(kangcom.com), 약 4만원
 - 번역판도 있음 (번역의 질은 잘...)
 - 예제 코드는 C++ 및 Smalltalk로 제공



Study Plan

- Design Pattern, itself
 - Creational Patterns
 - Structural Patterns
 - Behavioral Patterns
- Discussions & Applications
 - Linux Shell
 - Web programming

Study Plan

- The Catalog of Design Patterns (9~11 sessions)
 - Abstract Factory, Builder, Factory Method
 - Prototype, Singleton
 - Adapter, Bridge, Composite
 - Decorator, Façade
 - Flyweight, Proxy
 - Chain of Responsibility, Command
 - Interpreter, Iterator, Mediator
 - Memento, Observer, State
 - Strategy, Template Method, Visitor
- Application (2~3 sessions)
- 2 sessions/week after the midterm exam

귀귀귀~

이제 다시 시작이다...