개발 실무 디자인 패턴

참가자

제한재, 홍광필, 문혜영, 김문수, 강은지, 박상도, Jhang JhinDhal, 신용우, 서경진, 남규진, 박용훈, 고은비, 이충일

> 참고 링크
> https://www.inflearn.com/course/%EC%9E%90%EB%B0%94-%EB%94%94%EC%9E%90%EC%9D%B8-%ED%8C%A8%ED%84%B4/
>
> http://doohyun.tistory.com/56?category=714942
>
> https://www.tutorialspoint.com/design_pattern/index.htm
>
> https://java-design-patterns.com/
>
> https://refactoring.guru/design-patterns
>
> https://sourcemaking.com/design_patterns/
>
> git repository
>
> https://github.com/zebmason/GoFRefactored/blob/master/README.md
>
> https://github.com/iluwatar/java-design-patterns
>
> https://github.com/mariofusco/from-gof-to-lambda

목표

OOP 개발 패턴을 실습을 통해 확실히 이해하여 실무에 즉각 활용할 수 있도록 한다.
- 개발자간의 의사소통 원할
- 재사용 및 유지보수성 증가

커리큘럼

1~3 GoF 디자인패턴 이론

각 패턴의 목적과 활용방법에 대한 지식을 습득하고 실무 서비스로직에 적용 중인 패턴을 공유한다.

1주차 - OOP, UML 그리고 생성 패턴

패턴명	in JDK	in Practice	발표자
OOP 기초, 패턴 구조	객체 지향 프로그래밍의 원칙과 패턴을 읽는 구조에 대해 발표합니다.		Jhang JhinDhal
UML 읽기	책에 있는 UML 을 읽을 수 있도록 간단한 내용 정리		KwangPil, Hong
Abstract Factory	javax.xml.xpath.XPathFactory		박용훈
Builder	java.lang.StringBuilder		강은지
Factory Method	java.util.Calendar		남규진
Prototype	java.lang.Object#clone()		이충일
Singleton	java.lang.Runtime#getRuntime()	내부캐시	김문수

2주차 - 구조 패턴

패턴명	in JDK	in Practice	발표자
Adapter	java.util.Arrays#asList() java.io.InputStreamReader		서경진
Bridge	java.util.Collections#newSetFromMap()		고은비

Composite	java.awt.Component		제한재
Decorator	java.util.Collections#synchronizedXXX()		박상도
Facade	javax.faces.context.ExternalContext	slf4j, 주문service	황영주
Flyweight	java.lang.Integer#valueOf(int)		문혜영
Proxy	java.lang.reflect.Proxy		신용우

3주차 - 행위 패턴

패턴명	in JDK	in Practice	발표자
Chain of Responsibility	java.util.logging.Logger#log()		박용훈
Command	java.lang.Runnable		강은지
Interpreter	java.util.Pattern		남규진
Iterator	java.util.Iterator		이충일
Mediator	java.util.concurrent.Executor#execute()		김문수
Memento	java.util.Date		서경진
Observer	java.util.EventListener		고은비
State	javax.faces.lifecycle.Lifecycle#execute()	방송상태에따른 처리 - 시작 - 종료	제한재
Strategy	java.util.Comparator#compare()	목록 정렬 전략 - 최신순 정렬 - 특정 아이템 최상위 정렬	박상도
Template Method	java.util.AbstractList	할인쿠폰 - 금액할인쿠폰 - %할인쿠폰	황영주
Visitor	java.nio.file.FileVisitor		문혜영

4~6 기본 & 자주쓰는 디자인 패턴 특화 실습

https://github.com/iluwatar/java-design-patterns

난이도 Beginner 인 패턴들을 위주로 각 패턴에 특화된 요구사항을 정의하고 패턴을 적용하여 구현해보는 실습을 진행합니다.

- Factory Method
- Prototype
- Singleton
- Adapter
- Decorator
- Facade
- Proxy
- Template Method
- Iterator
- Observer
- Strategy
- State
- Chain of Responsibility

7~9 디자인 패턴 실습

이론으로 학습한 패턴을 실제로 어플리케이션에 구현하는 작업을 합니다. 구현할 대상은 간단한 주가 데이터 크롤러 및 거래 알고리즘 및 시뮬레이터 입니다. 스터디 리더의 언어와 같은 언어로 같이 프로젝트에 참여할 수도 있고, 새로 객체 지향을 배워본 언어에 개인적으로 적용하고 코드리뷰를 함께 할 수 있습니다. (파이썬, 자바 예정)

7주차 - 요구사항 정의 및 데이터 크롤링

- 만들고자 하는 어플리케이션의 요구사항과 변경사항을 나열합니다.
- finance.naver.com 에서 KOSPI 데이터를 크롤링 하는 간단한 프로그램입니다.
- 크롤러에 대한 failover 기능을 구현하기 위해 naver 에서 실패시 finance.daum.net 에서 가져올 수 있어야 합니다.
- 기타 스터디에서 새로운 요구사항이 구현될 수 있습니다.

8주차 - 알고리즘에 기반하여 ETF 트레이딩을 구현하고 계좌 수익률 구현

- 크롤링한 간단한 데이터를 기반으로 트레이딩 알고리즘을 작성하고 그에 따른 수익률을 추적할 수 있어야 합니다. (트레이딩 알고리즘은 요구사항으로 간단하게 정의되어있습니다)
- 알고리즘은 다양하게 구현될 수 있으며 알고리즘은 언제든지 교체될 수 있어야 합니다.
- 프로그램은 실시간으로 동작하는 모듈이라고 가정하고 작성합니다.

9주차 - 알고리즘 시뮬레이터 구현

- 실시간으로 동작하는 모듈로 가정하고 작성된 프로그램에서 과거의 데이터를 기반으로 시뮬레이터를 구현합니다.