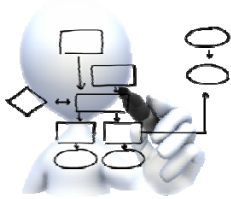


교과목 소개



한국기술교육대학교 컴퓨터공학부 김상진

sangjin@koreatech.ac.kr
www.facebook.com/sangjin.kim.koreatech

강의목표

- 강의목표
 - 설계 패턴을 통해 기존 개발자의 경험 공유
 - 소프트웨어 개발에 설계 패턴 활용 능력 배양
 - 객체지향 설계 및 구현 능력 배양
 - 리팩토링 개념 이해
 - 라이브러리 및 코드 이해 및 분석 능력 배양
- Coverage
 - UML
 - Object oriented design and analysis
 - Design Patterns
 - Refactoring



설계패턴

- 설계패턴은 C. Alexander가 건축학 분야에서 처음 제시함
- 컴퓨터 분야에서는 GoF(Gang of Four)라고 불리는 E. Gamma, R. Helm, R. Johnson, J. Vissides가 가장 큰 공헌을 함
- GoF가 그들의 저서를 통해 소개한 패턴들은 객체지향 기법을 사용하여 소프트웨어를 설계 및 구현할 때 사용할 수 있는 패턴들임
- 소프트웨어 개발과정에서 해결해야 하는 많은 문제들은 다른 사람들이 이미 슬기롭게 해결하였을 것임
- 설계패턴은 단순히 코드 재사용을 가능하게 해주는 것이 아니라 다른 개발자의 경험을 사용할 수 있도록 해줌
 - Instead of code reuse, with patterns you get *experience reuse*
- 설계 패턴은 자료구조, 알고리즘보다도 상위 개념

왜 자바?

- 설계 패턴은 객체지향 설계 및 구현할 때 사용되는 것으로 객체지향 프로그래밍 언어인 C++, C#, 자바 등을 이용하여 개발할 때 사용할 수 있음
 - 각 언어의 특징 때문에 각 패턴을 적용하고 구현하는 방법은 차이가 있음
- 객체지향 프로그래밍 언어 중에서 자바는 다양한 예제(GUI 포함)를 보다 쉽게 프로그래밍할 수 있게 해주며, 언어적 특성보다 패턴 자체에 보다 집중할 수 있도록 해줌
- 따라서 이 수업에서는 자바를 중심으로 설계패턴을 설명하고 실습함
 - C++, Javascript, Python 언어를 이용한 예도 설명함

강의정보

- 교수 정보
 - 연구실: 제4공학관 B301호
 - 전화번호: 교내 1490 (041-560-1490)
 - 전자우편: sangjin@koreatech.ac.kr
 - 강의홈페이지: el.koreatech.ac.kr
 - 강의페이스북페이지: www.facebook.com/KoreaTech.OOAD
- 교재
 - Eric Freeman and Elisabeth Freeman, Head First Design Patterns, O'Reilly, 2004.
- 수업에서 소프트웨어 개발 및 강건한 코딩 관련 졸업생 초청 특강이 있을 예정
- 조교정보

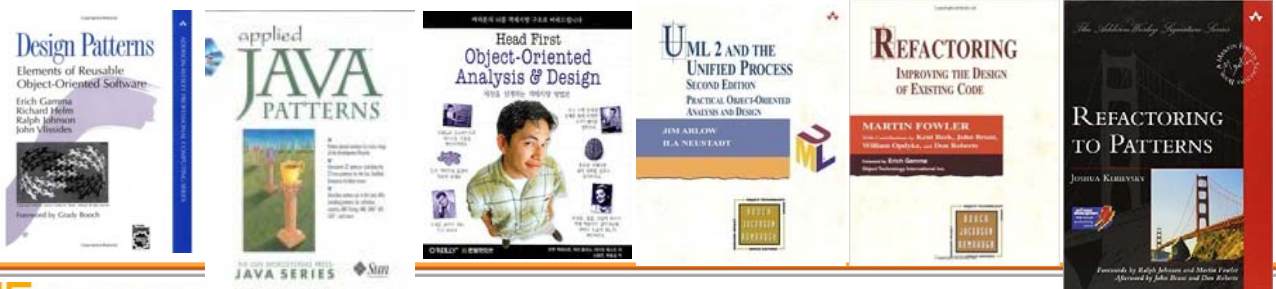


강의정보 - 평가방법

- 강의 평가 방법
 - 출석: 5%, 숙제/실습: 10%, 학기발표: 15%, 중간: 35%, 실습시험/기말: 35%
 - 참고. 바뀔 수도 있음
 - 프로그래밍 숙제는 수업 날짜 하루 전 밤 12시까지 el.kut.ac.kr을 통해 제출되어야 인정됨
 - 1시간 이상 지각할 경우 지각 처리함
 - 합당한 사유를 사전에 통보하면 결석 처리하지 않음
 - 결석할 경우에도 실습 내용을 숙제로 전부 제출해야 함
- Pre-requisite
 - 자바프로그래밍
- 강의 준비물
 - 강의 노트와 실습 노트를 강의 홈페이지에서 다운받아 준비해야 함

강의정보 - 부교재

- GoF, Design Patterns: Elements of Reusable Object-Oriented Software, AW, 1994.
- S. Stelting and O. Massen, Applied Java Patterns, PH, 2002.
- B.D. McLaughlin, G. Pollice, D. West, Head First Object-oriented Analysis & Design, O'Reilly, 2007.
- J. Arlow, I. Neustadt, UML 2 and the Unified Process, 2nd Ed., AW, 2008.
- M. Fowler, Refactoring: Improving the Design of Existing Code, AW, 1999
- J. Kerievsky, Refactoring to Patterns, AW, 2014



강의계획

1a	강의소개	9a	Refactoring 개념 소개
1b	객체지향 개념 재검토 SE와 UML	9b	Adapter/Façade
2a		10a	
2b	패턴소개, Strategy	10b	Template Method
3a		11a	
3b	Observer	11b	Iterator/Composite
4a		12a	
4b	Decorator	12b	State
5a		13a	
5b	Factory, Singleton	13b	Compound
6a		14a	
6b	Command	14b	학기발표
7a		15a	
7b	중간시험	15b	실습/기말시험
8a		16a	
8b		16b	

학기 발표 과제



- 개인 발표
- 졸업작품 또는 기존 개발 프로젝트(타 수업, 외부 대회, 다른 활동)를 설계 패턴을 이용하여 리팩토링함
 - 수업시간에 배운 또는 배우지 않은 패턴 중 자신의 프로젝트에 적용하면 소프트웨어 구조가 향상될 것으로 기대하는 패턴을 선택하여 리팩토링함
 - 구현이 가장 큰 목적이 아님
 - 골격 정도는 구현해야 함
- 보고서 양식 제공함

사용할 소프트웨어 (1/2)

- 자바(java.sun.com)
 - J2SE 버전(J2EE, J2ME): 최신 버전 (Java SE 8u121)
 - www.oracle.com/technetwork/java/javase/downloads/
 - JDK(Java 2 Development Kit)
 - Java documentation
 - 설치방법: 특이사항 없음
- 개발환경 Eclipse(www.eclipse.org/downloads)
 - Eclipse 설치하기 전에 JDK가 설치되어 있어야 함
 - Neon 설치 (eclipse-java-neon-?.zip)
 - Eclipse 설치방법
 - 설치하고자 위치에 압축 해제
 - eclipse.exe 파일 실행



사용할 소프트웨어 (2/2)

- C++11 이상
- Python 2.7.13
- Javascript, Node.js
- creately (<https://creately.com/>)
 - 온라인 UML 작성