Spring 2023

Teamwork #1: Module View

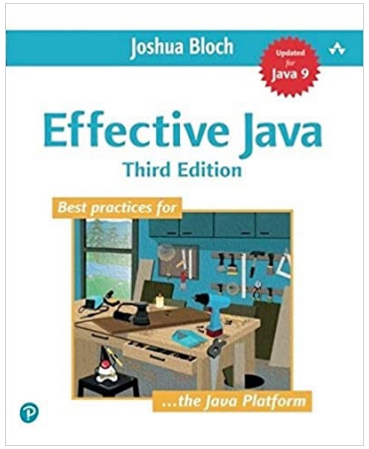
Seonah Lee

Gyeongsang National University

목 차

아마존 도서 추천 시스템

- 각 책 페이지에는 다음의 책 추천이 있음

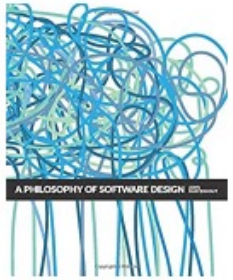


What other items do customers buy after viewing this item?

Page 1 of 8

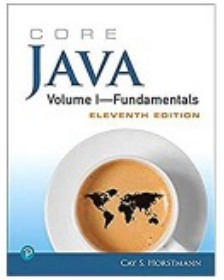
Clean Code: A Handbook of Agile Software Craftsmanship
› Robert C. Martin
1,395
#1 Best Seller in Software Testing
Paperback
$35.18

Clean Architecture: A Craftsman's Guide to Software Structure...
› Robert C. Martin
444
#1 Best Seller in Computer Hardware Design...
Paperback
16 offers from $21.64



A Philosophy of Software Design
John Ousterhout
160
Paperback
$18.95

Head First Java, 2nd Edition
› Kathy Sierra
1,216
Paperback
$34.19



Core Java Volume I--Fundamentals (11th Edition) (Core Series)
› Cay S. Horstmann
37
Paperback
$15.47

Java: A Beginner's Guide, Eighth Edition
› Herbert Schildt
48
Paperback
$20.99

Cracking the Coding Interview: 189 Programming...
› Gayle Laakmann...
1,916
#1 Best Seller in Data Structure and Algorithms
Paperback
$26.99

아마존 도서 추천 시스템

- 제약 사항
  - 아마존 시스템은 수백만명의 고객 정보와 책 정보를 관리함
- 요구사항
  - 이러한 도서 추천은 고객 각각의 인터랙션(책 조회 및 책 구매)에 가치를 둠; 고객 히스토리를 기반하여 도서 추천을 수행함
    - 새로운 고객은 극히 제한된 정보를 가짐, 몇 개의 클릭이 모두일 수 있음
    - 기존 고객은 너무 많은 정보를 가짐. 구매와 선호도 평가 모두 너무 많을 수 있음
  - 도서 추천은 **0.5**초 이내의 실시간에 이루어져야 하며, 동시에 높은 수준의 추천을 수행해야 함

아마존 도서 추천 시스템

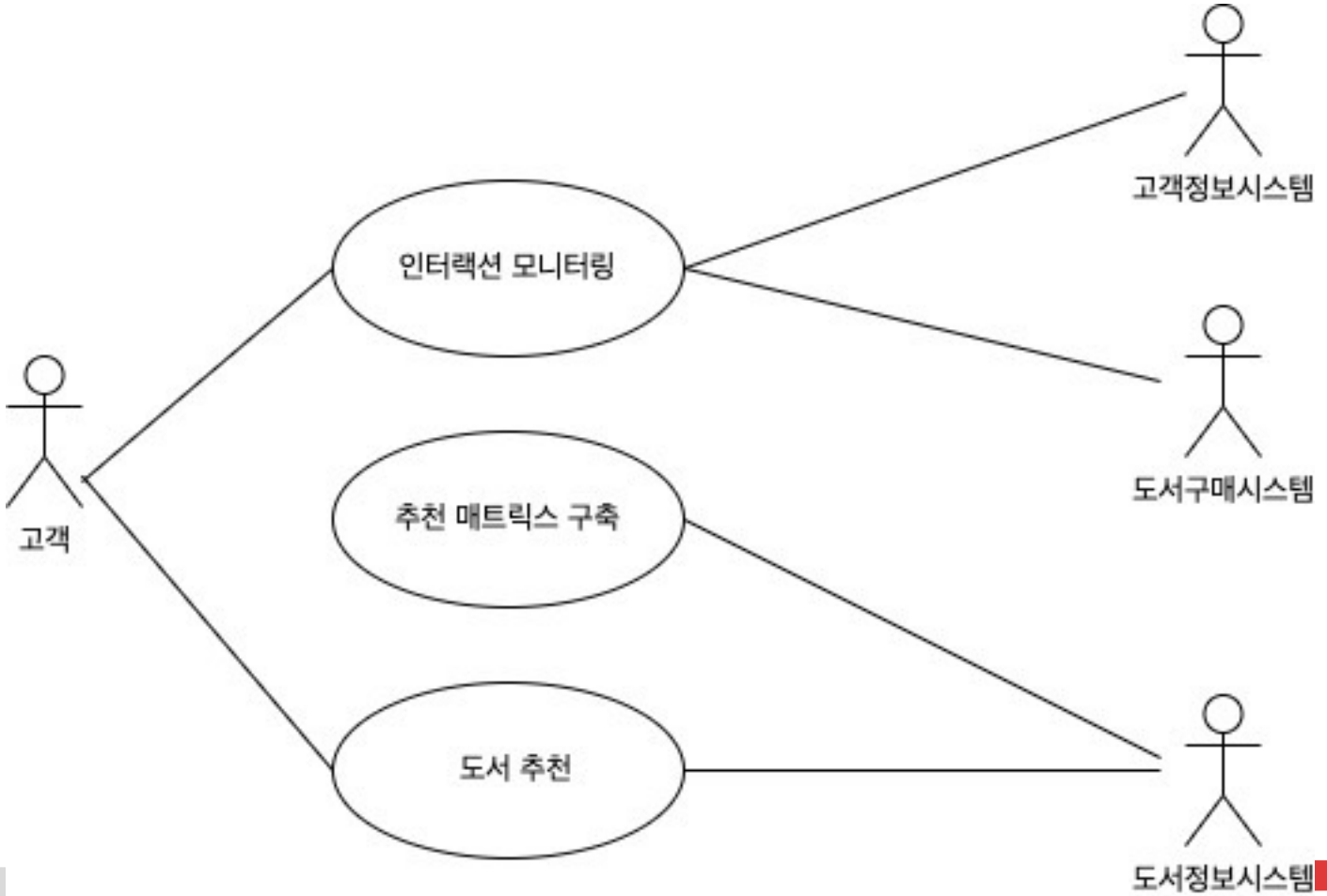
- 도서 추천 기능을 위한 설계 결정 사항
  - 각 사용자의 구매 혹은 선호 항목을 유사한 항목으로 매칭하기 위하여 **Item-to-Item Collaborative Filtering** 방식을 수행함
    - 시스템은 **user:item** 매트릭스가 아닌 **item:item** 매트릭스 구축
    - **Item:item** 매트릭스를 기반으로 도서 추천

```
For each item in product catalog, i1
        for each customer C who purchase i1
                for each item i2 purchased by customer C
                        record that a customer purchased i1 and i2
        for each item i2
                compute the similarity between i1 and i2
```

  - 성능 향상을 위해 오프라인으로 매트릭스 구축, 온라인으로 추천 수행

실습 I: 모듈 분할

- 유스케이스 명세서를 작성하여 모듈 분할 진행; **MVC**로 시스템 분할



유스케이스: 인터랙션 모니터링

유스케이스명	인터랙션 모니터링
개요	고객의 인터랙션 활동을 모니터링하여 기록한다.
관련 액터	고객, 고객정보시스템, 도서구매 시스템
선행조건	사용자가 로그인하지 않은 상태
이벤트 흐름	정상흐름 1. 고객은 아마존 시스템에서 도서를 클릭하여 도서 상세 내용을 조회한다. 2. 시스템은 고객이 클릭하여 상세 내용을 조회한 결과로 (조회, 조회한 도서, 조회 시간)을 기록한다. 3. 고객이 다른 도서를 클릭하여 도서 상세 내용을 조회한다. 4. 시스템은 고객이 클릭하여 상세 내용을 조회한 결과로 (조회, 조회한 도서, 조회 시간)을 기록한다. 5. 고객이 도서를 선택하고 구매 버튼을 누른다. 6. 시스템은 선택 도서를 장바구니에 담아, 구매 준비를 한다. 7. 고객은 결재를 수행하기 위해 체크 아웃 버튼을 누른다. 8. 시스템은 고객이 로그인할 수 있도록 로그인 창을 보인다. 9. 고객은 로그인을 수행한다. 10. 고객은 도서를 주문하는 절차를 수행하여 완료한다. 11. 시스템은 (구매, 구매한 도서, 구매 시간)을 기록한다. 12. 시스템은 구매자의 활동(조회, 구매) 기록을 데이터베이스에 저장한다.
후행 조건	사용자가 로그인한 상태

유스케이스: 추천 매트릭스 구축

유스케이스명	추천 매트릭스 구축
개요	고객의 인터랙션 활동을 기반으로 도서 추천 매트릭스를 구축한다.
관련 액터	도서정보 시스템
선행조건	고객의 인터랙션 데이터베이스에 누적된 조회 및 구매 이력 데이터가 존재
이벤트 흐름	정상흐름 1. 시스템의 타이머가 하루에 한번 데이터 분석을 위한 트리거링을 시작한다. 2. 시스템은 타이머가 트리거링한 액션으로 고객의 인터랙션 데이터베이스에 누적된 조회 및 구매 이력 데이터를 가져온다. 3. 시스템은 조회 및 구매 이력 데이터베이스를 대상으로 5페이지에 있는 도서 추천 매트릭스 구축 알고리즘에 따라 데이터를 스캔하며 반복적으로 매트릭스를 구축한다. 4. 시스템이 현재 누적된 모든 조회 및 구매 이력 데이터를 대상으로 매트릭스 구축을 완료하면 분석 및 구축 행위를 완료한다.
후행 조건	도서 추천 매트릭스 데이터베이스가 존재함

유스케이스: 도서 추천

유스케이스명	도서 추천
개요	고객의 책 조회 활동을 기반으로 구매할 도서를 추천한다.
관련 액터	고객, 도서정보 시스템
선행조건	도서 추천 매트릭스 데이터베이스가 존재함
이벤트 흐름	정상흐름 1. 고객은 아마존 시스템에서 도서를 클릭하여 도서 상세 내용을 조회한다. 2. 시스템은 해당 도서를 쿼리로 만들어 도서 추천 매트릭스 데이터베이스에 접근하여, 해당 도서를 조회한 경우 구매한 도서 목록을 구매 확률과 함께 받는다. 3. 시스템은 추천 섹션에 추천받은 도서 목록을 구매 확률이 높은 순에서 낮은 순으로 왼쪽에서 오른쪽까지 7개를 배치하여 나열한다.
후행 조건	도서 추천 목록이 도서 상세 페이지에 표시

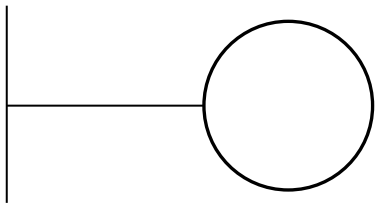
모듈 분할 MVC 기법

- 객체 지향 방법에서의 **MVC** 기법
  - 유스 케이스의 행위에서 **MVC** 역할 분리를 적용하여 클래스를 도출
  - **Model**
    - 데이타베이스 등 저장 매체 입출력을 담당하는 클래스
  - **View**
    - 사용자 유저 인터페이스를 담당하는 클래스
  - **Controller**
    - **Model**과 **View** 사이의 매개 및 중재, 조절 역할을 수행하는 클래스

모듈 분할 MVC 기법

- **View:** 경계 클래스**(Boundary Class)** 도출
  - **MVC** 패턴의 **View**에 해당하는 클래스로서 외부 액터/엔터티와 인터페이스하는 클래스를 포함
  - 사용자 인터페이스 클래스
    - 액터-인터페이스 관계에서 하나 이상의 사용자 인터페이스 클래스를 찾을 수 있음
  - 시스템 인터페이스 클래스
    - 외부 시스템과 커뮤니케이션하는 경계 클래스로서, 기존의 시스템의 인터페이스를 활용하기 위한 어댑터 클래스가 필요할 수 있음

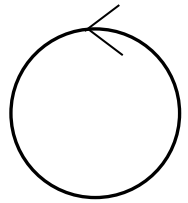
<< boundary>>
클래스명



모듈 분할 MVC 기법

- **MVC** 패턴의 **Control**에 해당하는 클래스로서 시스템을 제어하는 행위를 제공
  - 기본적으로 각 유스케이스에 대하여 하나의 제어 클래스를 생성
  - 유스케이스 내부의 제어 로직을 정의

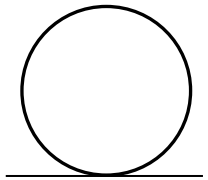
<< control >>
클래스명



모듈 분할 MVC 기법

- **MVC** 패턴의 **Model**에 해당하는 클래스로서 시스템에 저장되는 정보를 표현
  - 보통 각 유스케이스에 대하여 하나의 실체 클래스가 필요
  - 시스템에 정보를 저장하고 관리하는 역할을 수행

<< entity >>
클래스명



실습 | 제출: 모듈 분할 결과

- 시스템의 가장 큰 기능을 서브 시스템들로 나누어 본다.
  - 왜 그렇게 나누는 것이 합당한지 논의하여 합의한다.
- 유스케이스 명세서(혹은 시퀀스 다이어그램)을 작성한다.
- 구현할 시스템에 있어야 할 객체 혹은 모듈을 식별한다.
  - **MVC** 기법을 활용한 객체 도출을 수행한다.
- 각 모듈명과 역할을 목록으로 기술하여 정리한다.

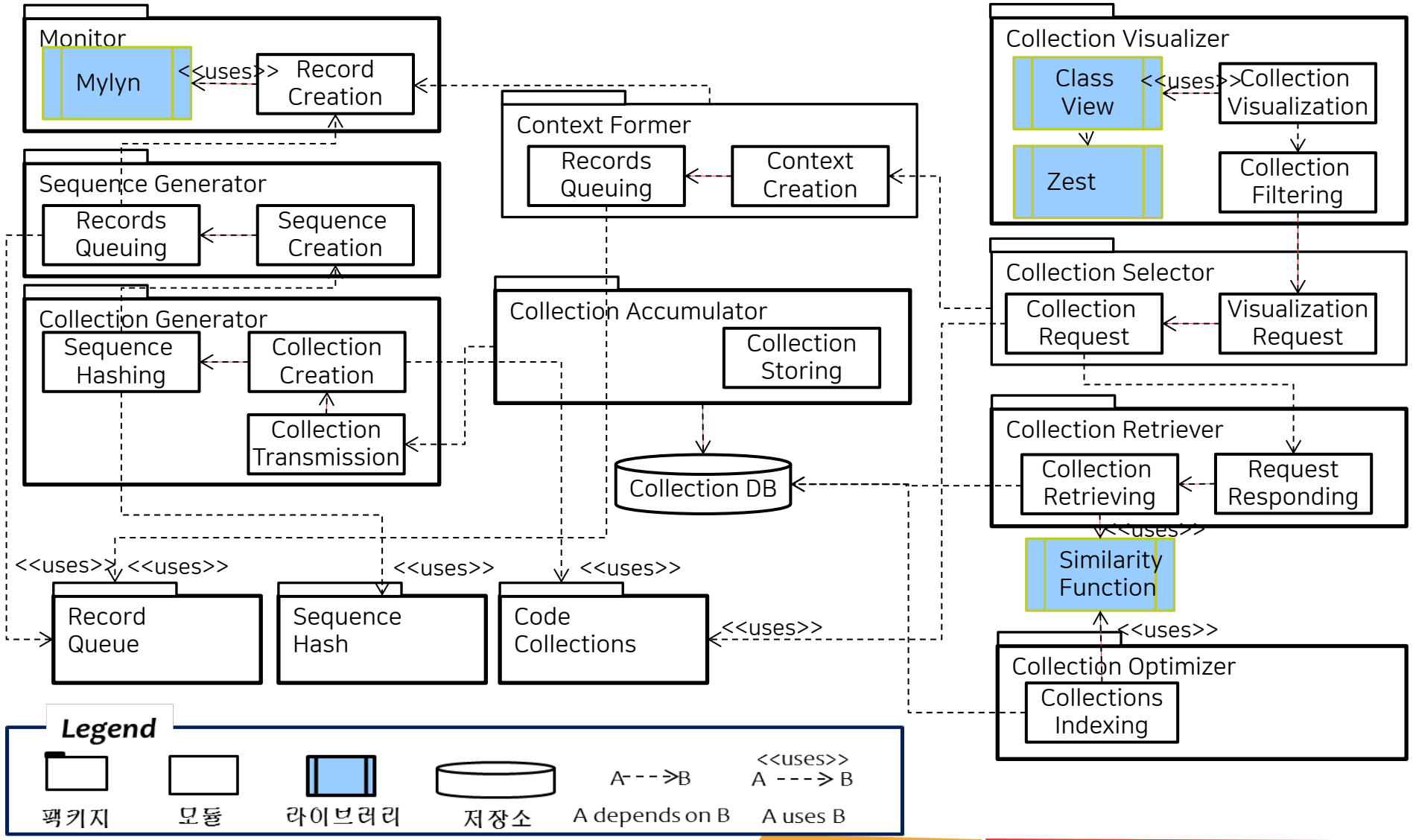
모듈명	역할

실습 II: 모듈 뷰 작성

- 모듈 분할을 완료 후, 각 관심(**Concern**)에 따라 그룹화한다.
  - 관심 별 수직적인 관계가 있다면 **Layered Style**을 고려한다.
- 각 모듈 사이의 사용 혹은 사용가능 관계를 파악한다.
- 이러한 모듈과 관계를 표현할 스타일을 정한다.
  - 스타일: 표현양식(노테이션)과 각 모듈 및 관계의 특성을 정의
- 정해진 스타일에 따라 모듈 뷰를 작성한다.

실습 II 제출: 모듈 뷰 작성 예제

- 예제



실습 결과 제출

- 토론 방식
  - 구글 드라이브에 프리젠테이션 만들어 논의
- 제출 방식
  - 팀별 만든 프리젠테이션 링크를 네이버 까페에 제출
  - 타이틀 **[SW** 아키텍처: 모듈 뷰 작성**]** 로 팀명을 기입하여 제출
    - 가급적 링크는 수업 중 제출 요망
- 강사 평가 (저녁 **7**시 이후 평가 진행)
  - 평가 후 각 팀별 피드백 까페에 답글로 기술

Question?

Seonah Lee

saleese@gmail.com