e-learning

2020 학년도 1 학기

컴퓨터정보과

Use Case 모델의 구조화1

담당교수 : 김계숙

제 4 주차 / 제 1차시

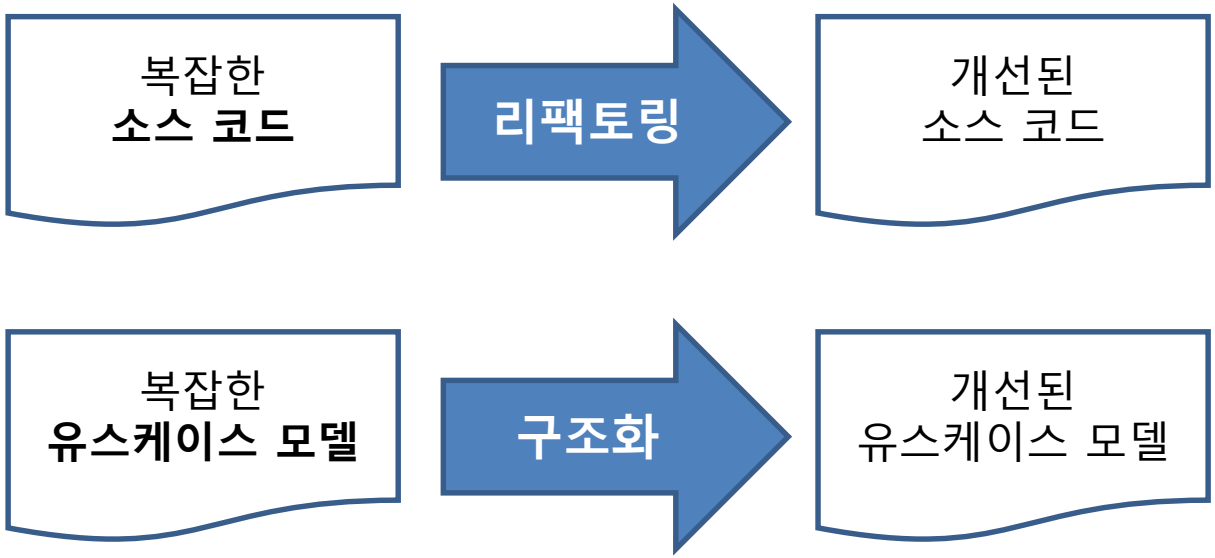
본 자료
절대
유출 불가

DIT 동의과학대학교
DONG-EUI INSTITUTE OF TECHNOLOGY

유스케이스 모델의 구조화

개요

유스케이스 모델의 구조화 목적



유스케이스 모델의 구조화 목적

1. refactoring 이란?

* 리팩토링은 외부 동작을 바꾸지 않으면서 내부 구조를 개선하는 방법으로 소프트웨어 시스템을 변경하는 프로세스다.

* 디자인을 먼저 한 후 코드를 만드는 것이 아니라 일단 돌아가는 코드를 작성하고 그 후에 그 코드가 더 좋은 구성을 갖도록 바꾼다는 것.

* 여러 곳에 산재된 중복 코드 정리

2. refactoring 하는 이유?

* 코드 간결성 및 가독성 증대

* 무엇인가 소스의 변경이 필요할 때(기능 추가나 삭제,수정 작업이 일어날 때) 소스의 리팩토링은 포장이사처럼 편하게 작업하도록 도와준다.

* 리팩토링은 소스의 중복된 부분을 모듈화 시켜준다. 모듈화는 입출력이 명확하기 때문에 이식성을 높여준다.

* 리팩토링을 통해 시스템 복잡도를 줄일 수 있다.

3. 활용

여러 메소드 내에 반복되는 소스를 리팩토링 하여 하나의 메소드로 생성(convertCoffee(resultMap))

구조화 방법

소스 코드 리팩토링 방법

- Extract method
- Pull up field
- …

그룹으로 함께 묶을 수 있는 코드 조각이 있으면, 코드의 목적이 잘 드러나도록 이름을 별도로 지어 뽑아냄(중복된 메소드 하나로)

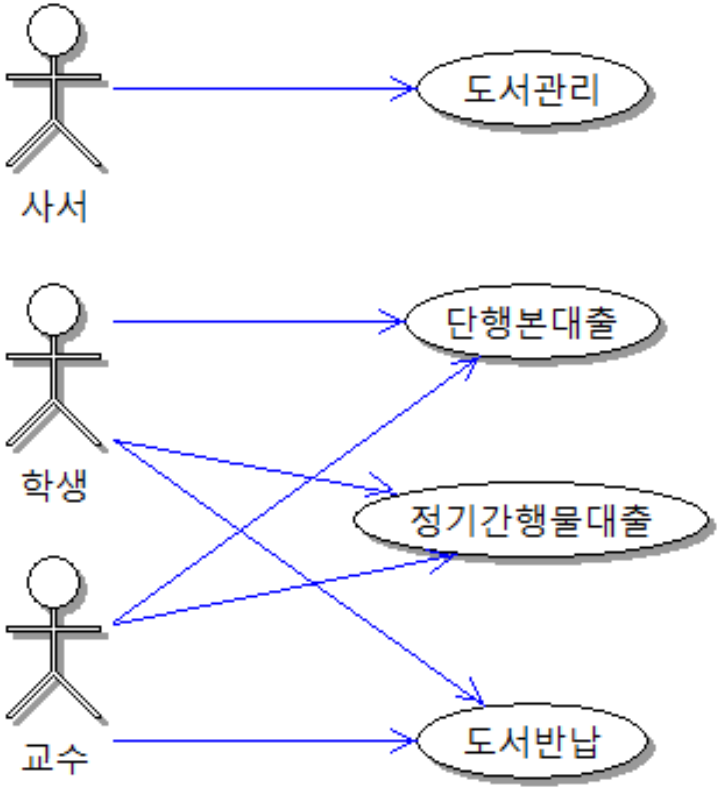
두 서브클래스가 동일한 필드를 가지고 있다면 그 필드를 수퍼클래스로 옮김

유스케이스 모델 구조화 방법

- 액터 일반화
- 유스케이스 일반화
- 유스케이스 포함
- 유스케이스 확장

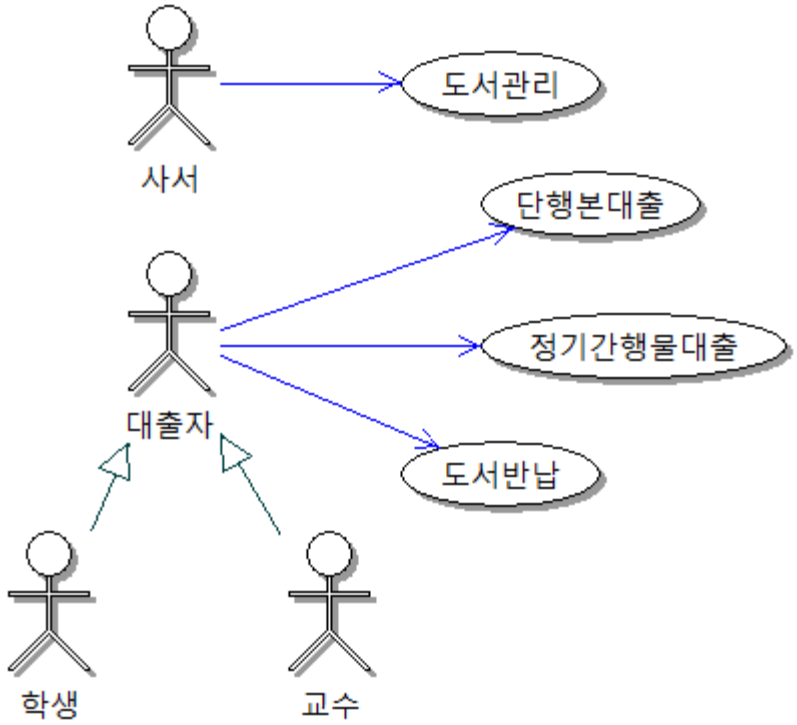
유스케이스 모델의 구조화

- 구조화 이전의 유스케이스 모델 예



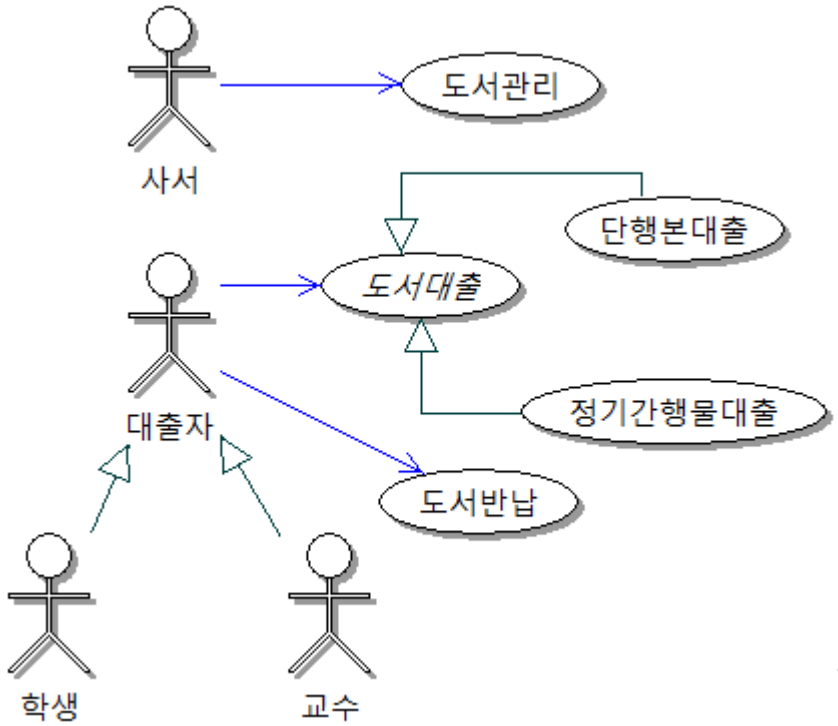
액터의 일반화

- 시스템과 비슷한 방식으로 상호작용을 하는 유사한 액터를 일반화하여 부모 액터를 정의한다



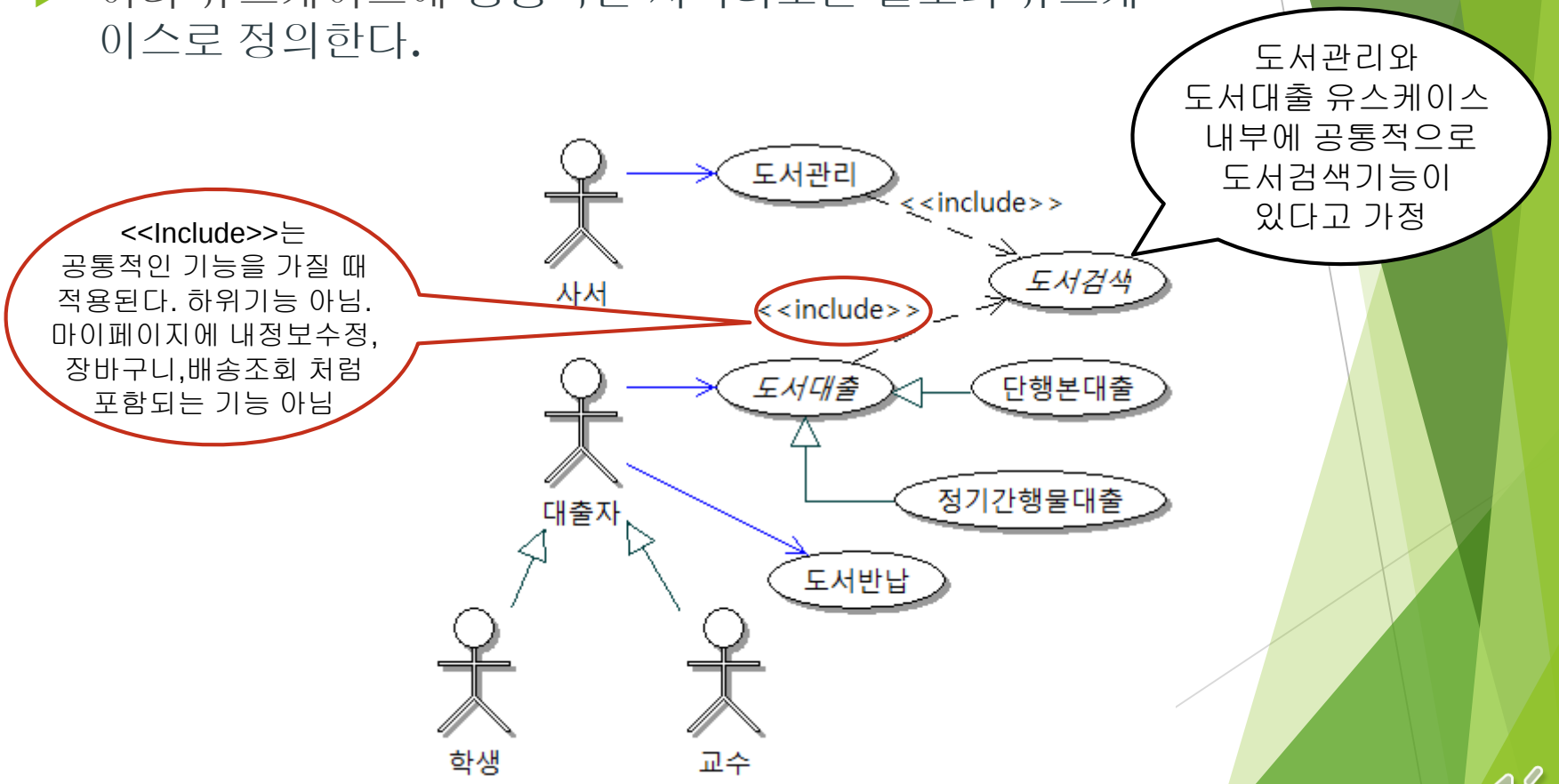
유스케이스의 일반화

- 유사한 기능을 제공하는 유스케이스들을 일반화하여 부모 유스케이스를 정의한다



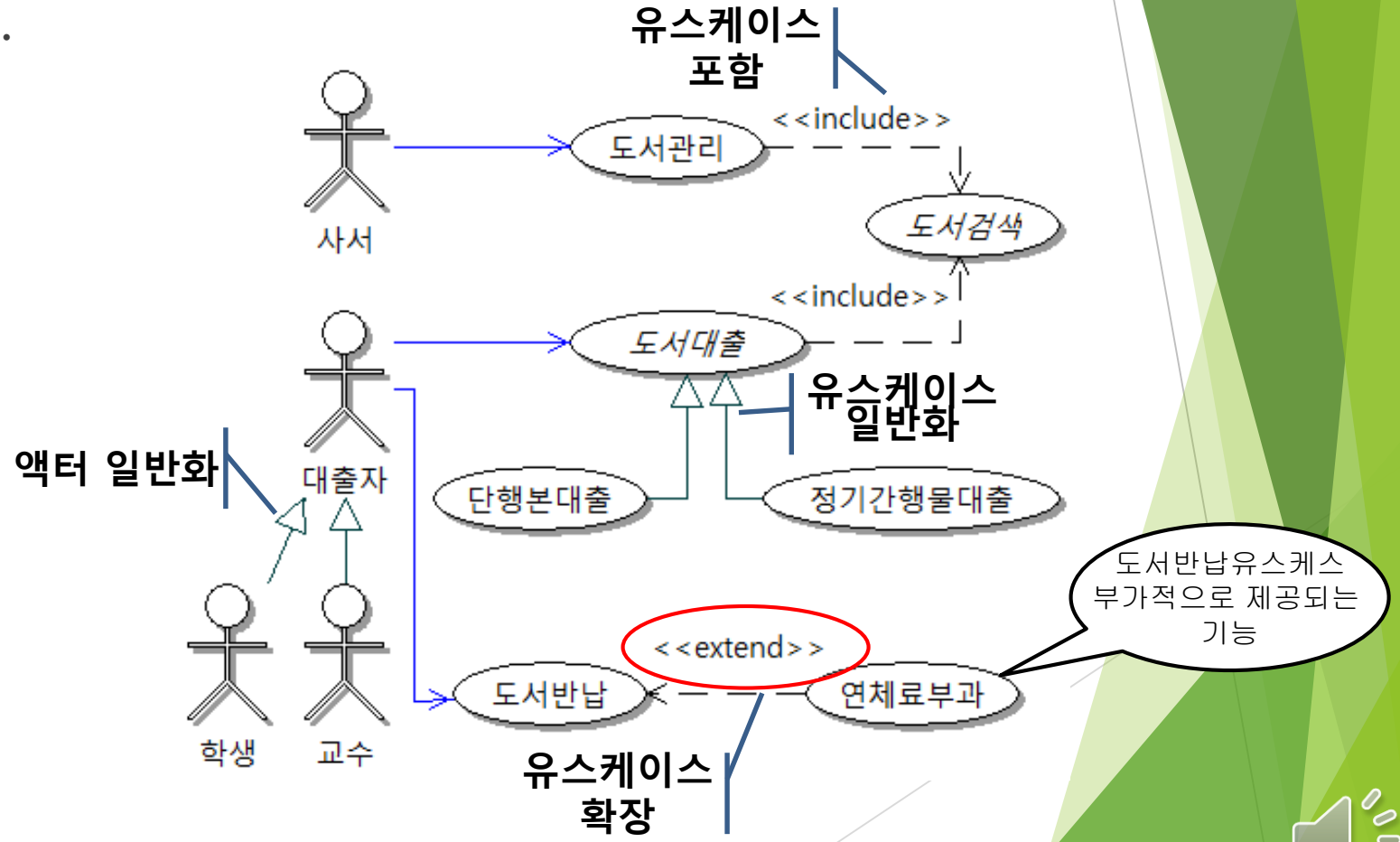
유스케이스의 포함

- 여러 유스케이스에 공통적인 시나리오를 별도의 유스케이스로 정의한다.



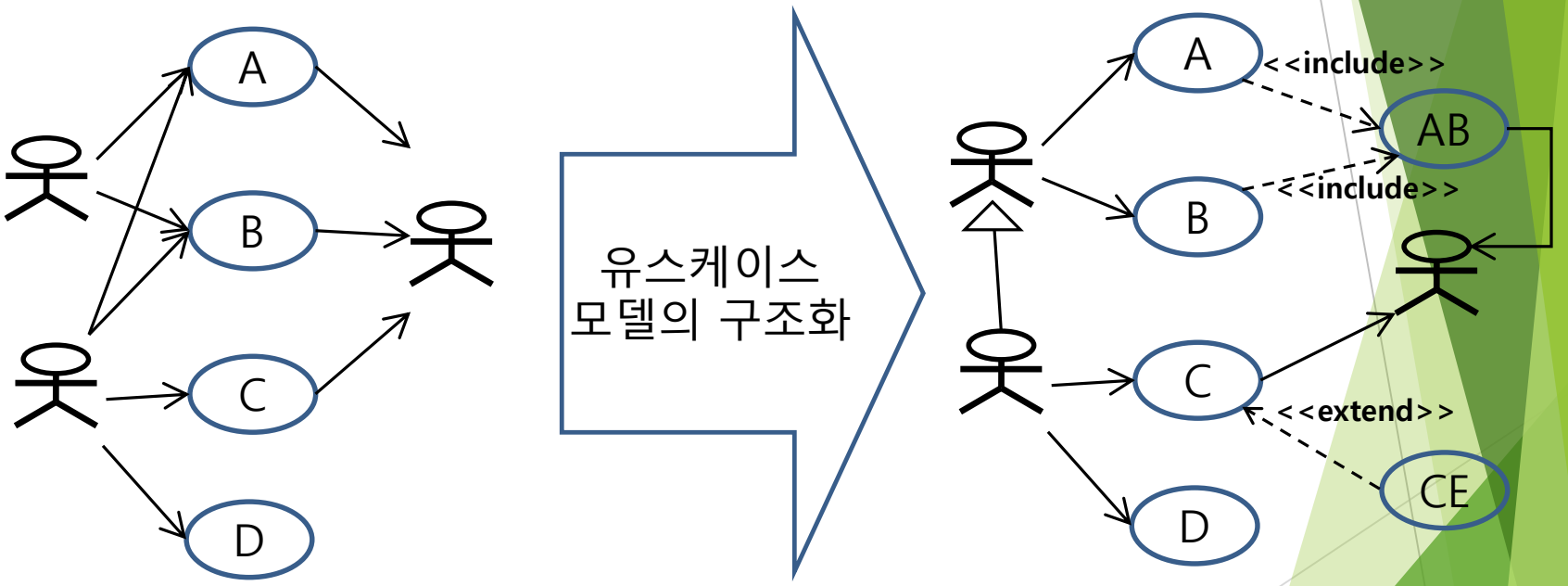
유스케이스의 확장

- 기본 기능에 부가적인 기능을 별도의 유스케이스로 정의한다.



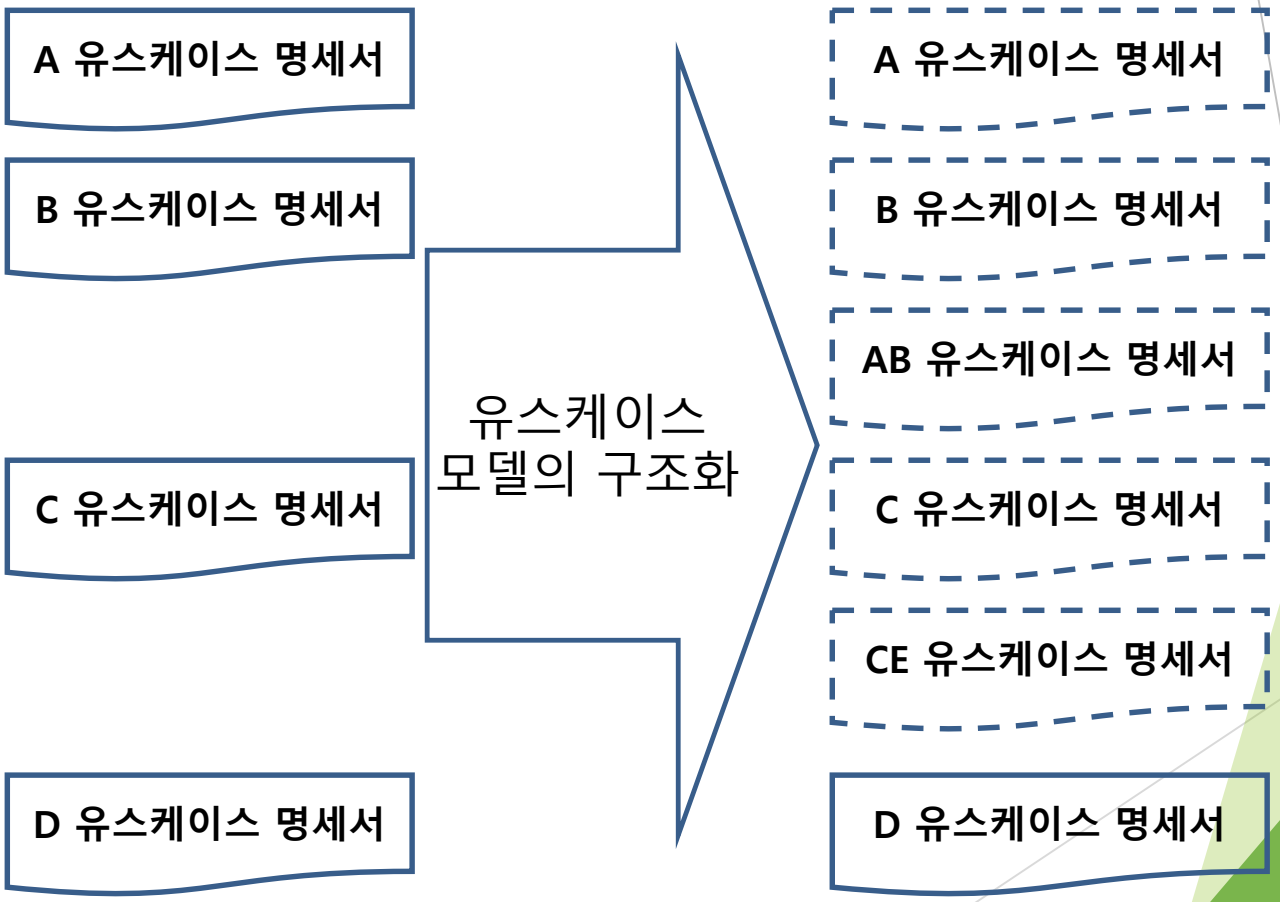
산출물

- 구조화된 유스케이스 모델



산출물

- 수정된 유스케이스 명세서



감사합니다

동의과학대학교
DONG-EUI INSTITUTE OF TECHNOLOGY

e-learning

2020 학년도 1 학기

컴퓨터정보과

Use Case 모델의 구조화2

담당교수 : 김계숙

제 4 주차 / 제 2차시

본 자료
절대
유출 불가

DIT 동의과학대학교
DONG-EUI INSTITUTE OF TECHNOLOGY