



## 공통 모듈 설계



한국기술교육대학교  
온라인평생교육원

## 학습내용

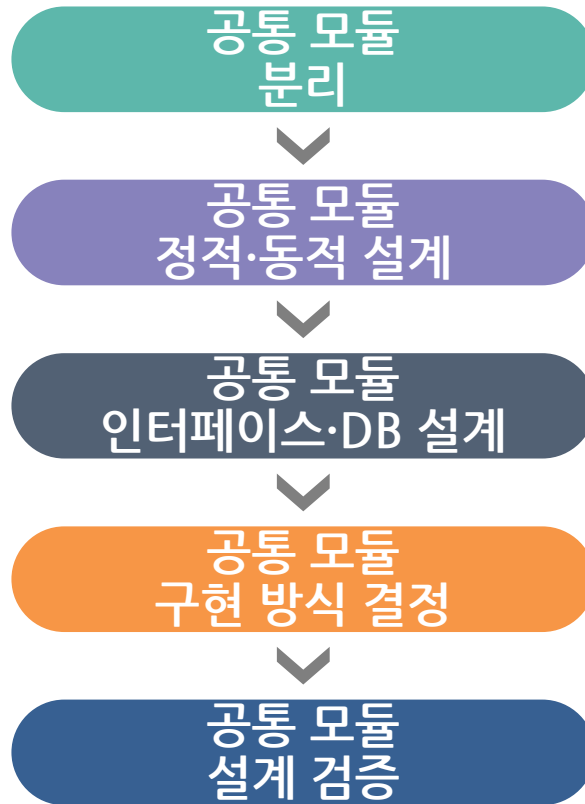
- 공통 모듈 설계 순서
- 공통 모듈 설계 단계별 내용

## 학습목표

- 공통 모듈 설계 순서를 설명할 수 있다.
- 개발할 응용소프트웨어의 전반적인 기능과 구조를 쉽게 이해할 수 있는 공통 모듈을 설계할 수 있다.

# 공통 모듈 설계 순서

## ◎ 공통 모듈 설계 순서 흐름



# 공통 모듈 설계 순서

## ◎ 공통 모듈 설계 순서 흐름

공통 모듈 분리



공통 모듈  
정적·동적 설계



공통 모듈  
인터페이스·DB 설계



공통 모듈  
구현 방식 결정



공통 모듈 설계 검증

- Use Case, Class Diagram에서 공통 모듈 분리
- 공통 모듈을 위한 별도의 Use Case, Class Diagram 구성
- 공통 모듈 명세

# 공통 모듈 설계 순서

## ◎ 공통 모듈 설계 순서 흐름

공통 모듈 분리



공통 모듈  
정적·동적 설계



공통 모듈  
인터페이스·DB 설계



공통 모듈  
구현 방식 결정



공통 모듈 설계 검증

- 공통 모듈에 대한 정적 설계

Class, Component, Package  
Diagram 등

- 공통 모듈에 대한 동적 설계

Sequence, Activity,  
Communication Diagram 등

# 공통 모듈 설계 순서

## ◎ 공통 모듈 설계 순서 흐름

공통 모듈 분리



공통 모듈  
정적·동적 설계



공통 모듈  
인터페이스·DB 설계



공통 모듈  
구현 방식 결정



공통 모듈 설계 검증

- 외부에서 공통 모듈 사용을 위한 인터페이스 설계
- 데이터 구조 및 프로시저 설계

# 공통 모듈 설계 순서

## ◎ 공통 모듈 설계 순서 흐름

공통 모듈 분리



공통 모듈  
정적·동적 설계



공통 모듈  
인터페이스·DB 설계



공통 모듈  
구현 방식 결정



공통 모듈 설계 검증

- Class, Library, Component, Framework, Script 등 공통 모듈 사용을 위한 구현 형태 결정

# 공통 모듈 설계 순서

## ◎ 공통 모듈 설계 순서 흐름

공통 모듈 분리



공통 모듈  
정적·동적 설계



공통 모듈  
인터페이스·DB 설계



공통 모듈  
구현 방식 결정



공통 모듈 설계 검증

- 응집도, 결합도 측정
- 인터페이스 검증



# 공통 모듈 설계 단계별 내용

## ◎ 공통 모듈 분리

### ❖ 공통 모듈이란?

- 다른 모듈들에 의해서 공통적으로 사용되는 모듈

예

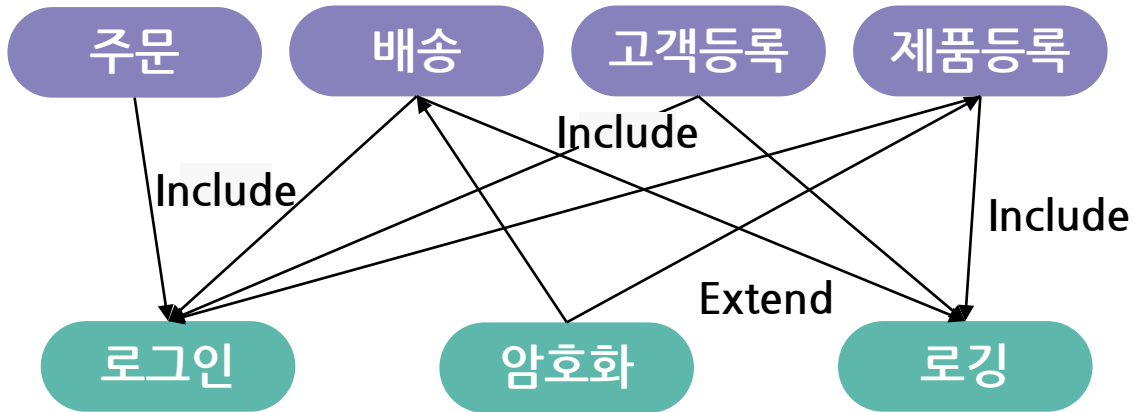
Use Case  
Diagram

다른 Use Case에 의해서 공통적으로  
사용되는 모듈

# 공통 모듈 설계 단계별 내용

## 공통 모듈 분리

### 공통 모듈 분리의 예 - ① Use Case Diagram



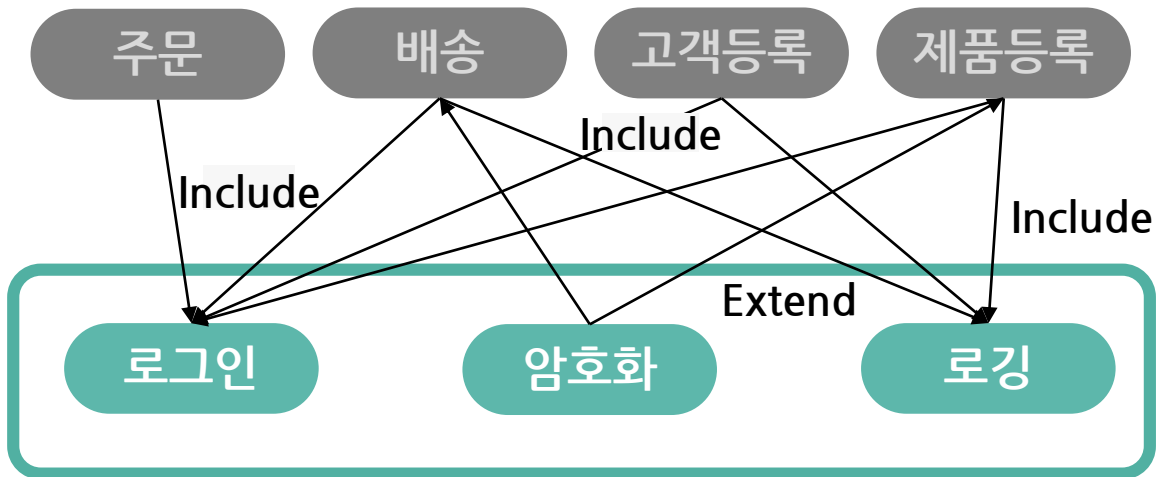
⟨⟨Include⟩⟩ : 다른 Use Case 를 반드시 포함하는 관계

⟨⟨Extend⟩⟩ : 다른 Use Case를 선택적으로 확장하는 관계

# 공통 모듈 설계 단계별 내용

## 공통 모듈 분리

### 공통 모듈 분리의 예 - ① Use Case Diagram



공통 모듈 분리

## 공통 모듈 설계 단계별 내용

### 🎯 공통 모듈 분리

#### ✦ 공통 모듈 분리의 예 - ① Use Case Diagram

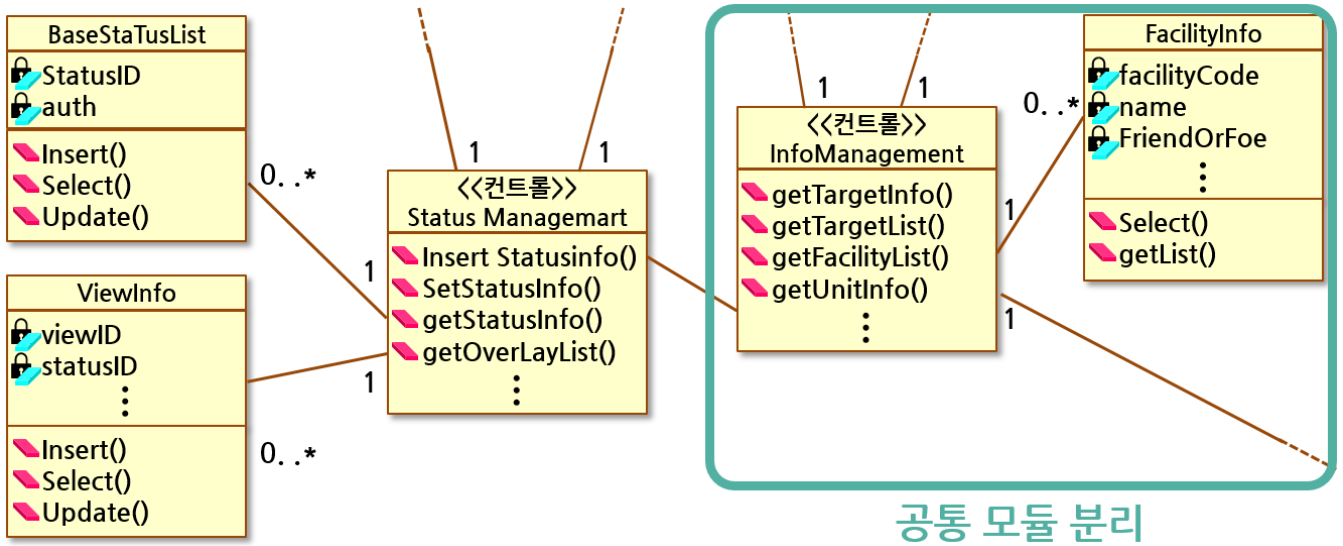
- Use Case는 공통 모듈로 분리하여 별도로 설계



# 공통 모듈 설계 단계별 내용

## 공통 모듈 분리

### 공통 모듈 분리의 예 - ② Class Diagram



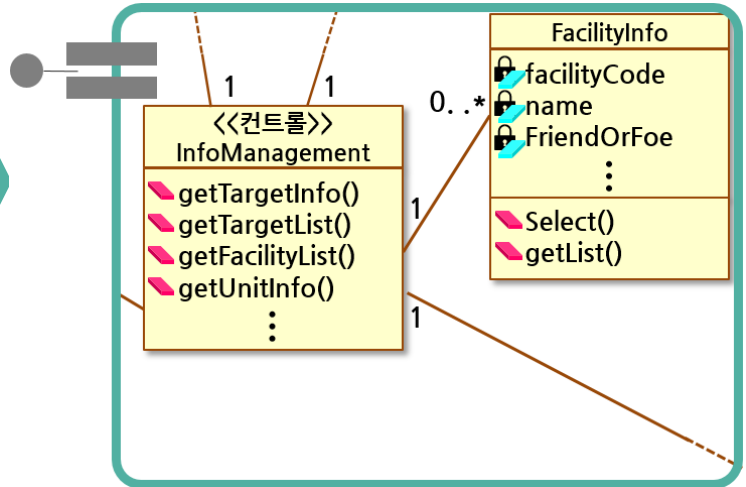
# 공통 모듈 설계 단계별 내용

## 공통 모듈 분리

### 공통 모듈 분리의 예 - ② Class Diagram

#### 공통 모듈 컴포넌트

- 다른 Class에 의해 수시로 참조되는 모듈
- 별도 구성하여 재사용성, 복잡성을 줄임



# 공통 모듈 설계 단계별 내용

## 공통 모듈 정적·동적 설계

### 정적 설계

- 공통 모듈 Use Case에서 Class Diagram 도출

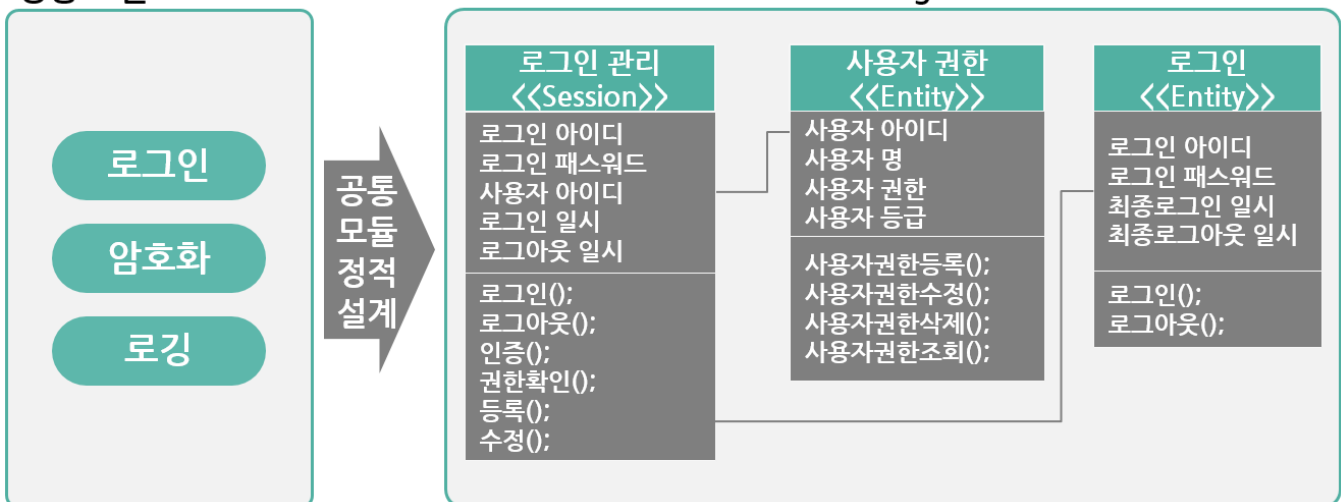
#### 공통 모듈에 대한 Class Diagram 설계 시

- Use Case 시나리오 기반으로 Class 도출
- Control Class와 Entity Class를 도출하여 Use Case 시나리오를 실행

### 정적 설계의 예

공통 모듈 Use Case

Class Diagram



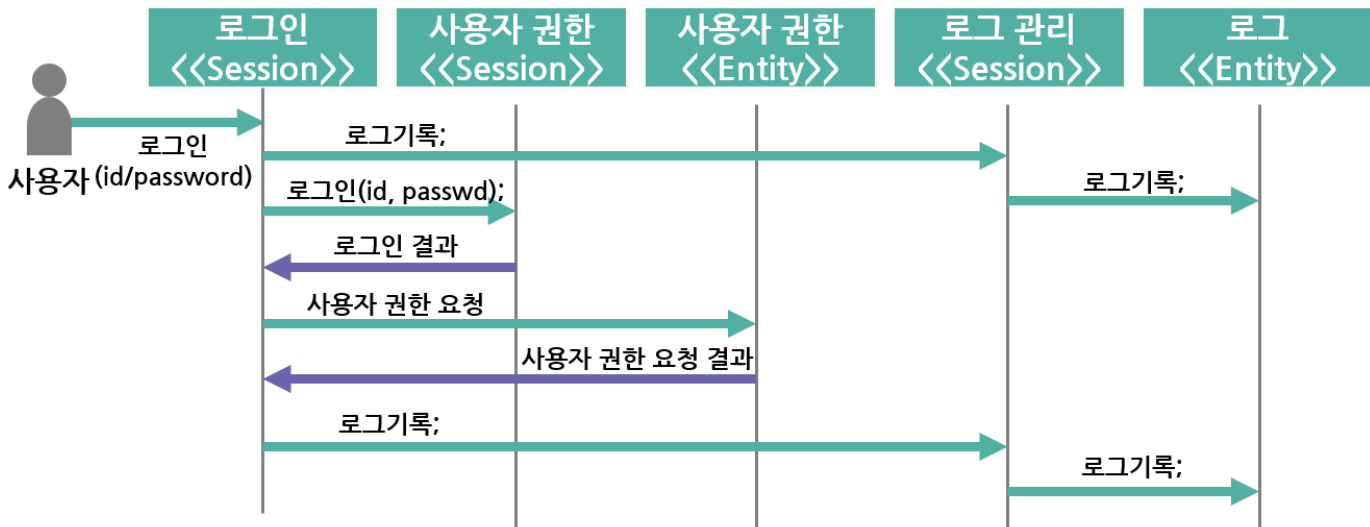
# 공통 모듈 설계 단계별 내용

## 공통 모듈 정적·동적 설계

### ❖ 동적 설계

- Use Case 시나리오를 Sequence Diagram을 활용하여 설계
- 로그인 사용자 권한 로그  
- Class Diagram의 Entity Class 의미
- Entity Class들 간의 상호작용을 Sequence Diagram을 활용하여 제작

### ❖ 동적 설계의 예





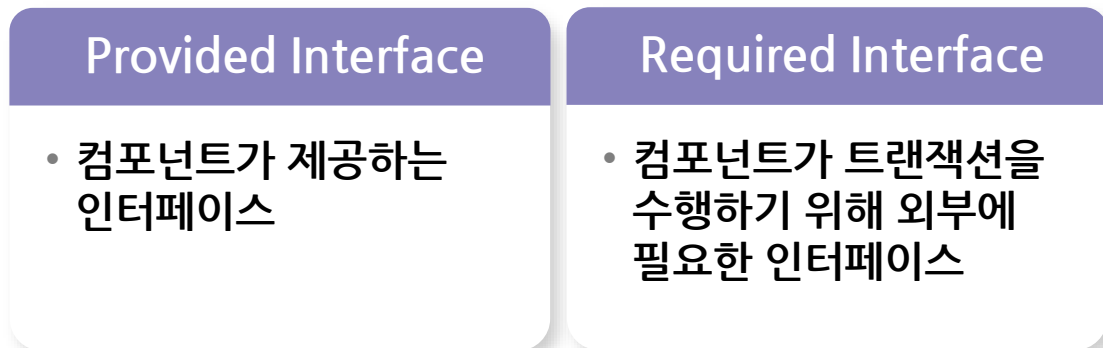
# 공통 모듈 설계 단계별 내용

## 공통 모듈 인터페이스·DB 설계

### ❖ 인터페이스 설계

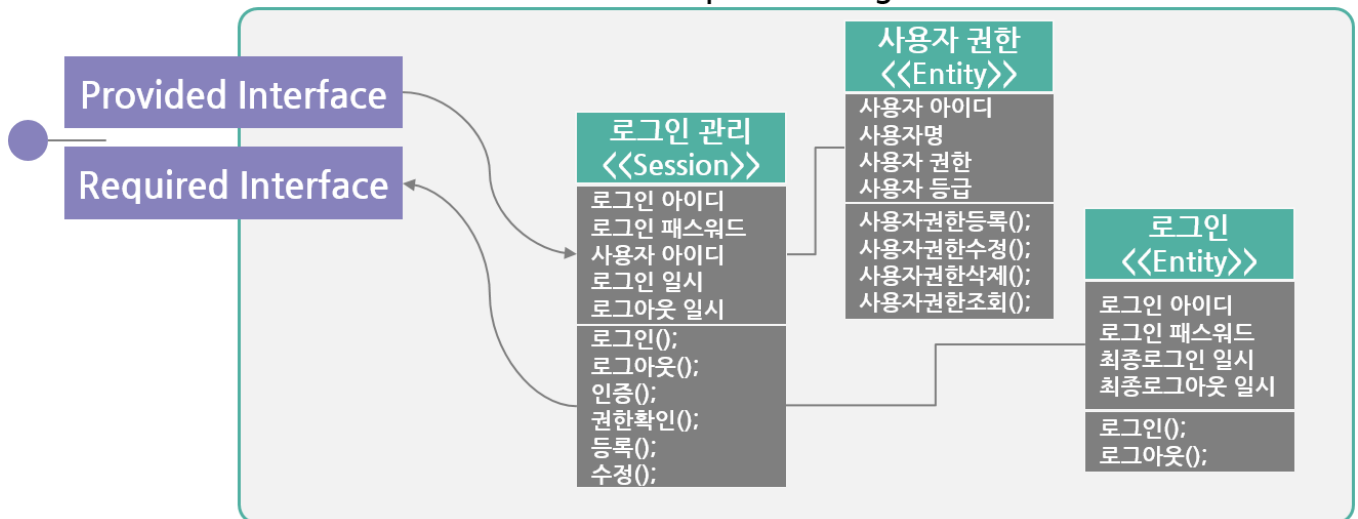
- 컴포넌트 설계 : 공통 모듈 설계 시 가장 많이 활용

### 컴포넌트의 인터페이스



### ❖ 인터페이스 설계의 예

Component Diagram



# 공통 모듈 설계 단계별 내용

## 공통 모듈 인터페이스·DB 설계

### ❖ 인터페이스 명세

- 공통 모듈 인터페이스는 외부에서 공통 모듈을 활용하기 위한 명세
- 오퍼레이션명, 가시성, 반환 타입, 파라미터를 설명

### ❖ 인터페이스 명세의 예

오퍼레이션 명	가시성	반환 타입	파라미터		설명
			명칭	설명	
login	public	SmartResult	id	로그인 사용자 아이디	로그인한 사용자의 아이디와 패스워드를 비교하여 인증된 사용자인지 확인
			password	로그인 사용자 패스워드	
logout	public	SmartResult	id	로그아웃 사용자 아이디	사용자 아이디에 대해 로그아웃 설정
authenticate	public	SmartResult	id	로그인 사용자 아이디	로그인한 사용자의 시스템 사용 권한을 확인
...	...	...	...	...	...

## 공통 모듈 설계 단계별 내용

### 🎯 공통 모듈 인터페이스·DB 설계

#### ➤ 인터페이스 명세서

- 공통 모듈에 대한 설명과 오퍼레이션 혹은 메소드에 대한 설명 포함
- Input/Output 파라미터 설명
- 활용예제를 직접 코드로 설명하기도 함

# 공통 모듈 설계 단계별 내용

## 공통 모듈 인터페이스·DB 설계

### ❖ 인터페이스 명세서의 예 - ①

모듈명	서버 공통 모듈	패키지	rdis.common
클래스명	BaseAction	관련 프로그램 ID	
서버 모듈 정의	컨트롤러의 기반이 되는 추상 클래스		
비고	<p>다음 예와 같이 컨트롤러 선언 시 사용</p> <pre>public class ProductViewAction extends BaseAction {     ..... }</pre>		

### ❖ 인터페이스 명세서의 예 - ②

메소드명	perform	메소드 ID	
메소드 정의	JSP에서 호출할 실행 메소드		
INPUT	HttpServletRequest request HttpServletResponse response	OUTPUT	String
<p>다음 예와 같이 JSP에서 컨트롤러를 사용할 때 호출</p> <pre>ProductViewAction action = new ProductViewAction(); action.perform( request, response );</pre> <p>반환 값은 애플리케이션의 용도에 따라 정의하여 사용</p>			

# 공통 모듈 설계 단계별 내용

## ◎ 공통 모듈 인터페이스·DB 설계

### ✧ 인터페이스 명세서의 예 - ③

메소드명	doAction	메소드 ID	
메소드 정의	업무를 실행하는 컨트롤러 클래스에서 실제로 구현할 메소드		
INPUT	HttpServletRequest request HttpServletResponse response	OUTPUT	String

메소드는 추상 메소드이므로 업무 실행하는 컨트롤러 클래스를 만들 때 BaseAction 클래스를 상속한다면 이 메소드를 구현해야 함

```
protected String doAction(HttpServletRequest request, HttpServletResponse response) throws
Exception {
    String myVal = "test string";
    request.setAttribute( "value", myVal );

    if ( myVal!=null )
        redirect( response, "/test" );

    return null;
}
```

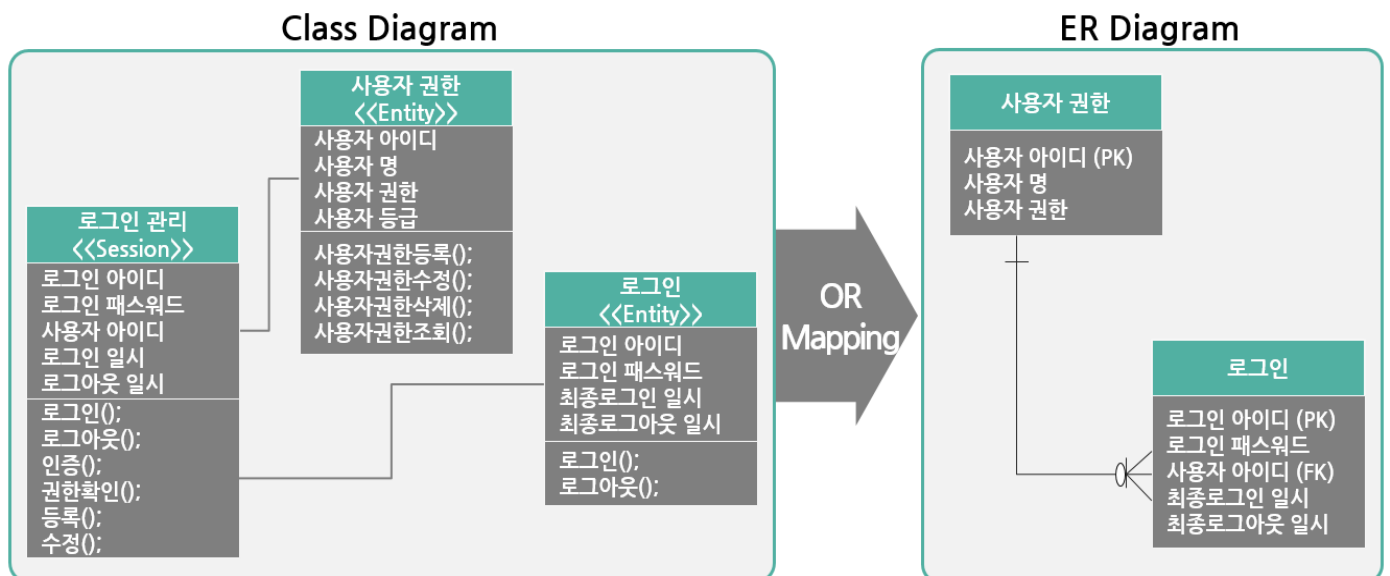
# 공통 모듈 설계 단계별 내용

## 공통 모듈 인터페이스·DB 설계

### DB 설계

- Class Diagram에서 OR Mapping을 활용하여 ER Diagram 설계
- Class Diagram에는 객체가 ER Diagram에서는 Entity로 불림
- 객체가 가지고 있는 속성은 Entity의 속성으로 변환됨
- 객체의 오퍼레이션은 ER Diagram으로는 변환되지 않음
- 객체의 오퍼레이션은 Sequence Diagram이나 Data Flow Diagram으로 변환됨

### DB 설계의 예



## 공통 모듈 설계 단계별 내용

### ◎ 공통 모듈 구현 방식 결정

#### 공통 모듈 구현 방식

클래스  
(Class)

라이브러리  
(Library)

컴포넌트  
(Component)

프레임워크  
(Framework)

# 공통 모듈 설계 단계별 내용

## ◎ 공통 모듈 구현 방식 결정

### ❖ 클래스(Class)

- 공통 모듈을 1~2개 정도의 Class 형태로 구현
- 다른 프로젝트에서 재사용 어려움

### ❖ 라이브러리(Library)

- 공통 모듈을 여러 개의 Class library(.jar) 형태로 구현
- 다른 프로젝트에서 Lib 디렉토리에 넣고 Import하여 사용

### ❖ 컴포넌트(Component)

- 공통 모듈을 컴포넌트 인터페이스를 활용하여 구현
- 다른 프로젝트에서 재사용 용이

### ❖ 프레임워크(Framework)

- 공통 모듈을 구현하여 프레임워크에 추가
- 다른 프로젝트에서 재사용이 쉽고, 사용 확대 가능



## 공통 모듈 설계 단계별 내용

### ◎ 공통 모듈 설계 검증

❖ 검증 지표 : 응집도, 결합도

#### 응집도

$$\frac{\text{공통 모듈 내부에서 처리 가능한 오퍼레이션 수}}{\text{전체 오퍼레이션 수}}$$

#### 결합도

$$\frac{\text{타 모듈 호출이 필요한 오퍼레이션 수}}{\text{전체 오퍼레이션 수}}$$


# 공통 모듈 설계 단계별 내용

## 공통 모듈 설계 검증

### 검증 방법

- 공통 모듈 중 공통 코드 검증은 공통 코드 테이블을 조회하여 개수를 비교
- 엑셀로 관리하는 공통 코드와 실제 데이터베이스에 입력된 공통 코드 간에 개수 비교를 통하여 누락된 공통 코드가 없는지 확인

### 공통 모듈 설계 검증 예

한글 데이터 명	코드상세내역	영문 데이터 명	CN_CODE	프로그램 ID	SP_CN_CODE
프로그램 형태	SP	Work ID	A-02	업무명	공통관리
프로그램 흐름			검증방법		
1. eMIS 코드상세내역(CM_CODE) 정보를 조회함 2. EIS 코드상세내역(CN_CODE) 테이블에 Import함 3. 소스 테이블 CM_CODE 4. 타겟 테이블 CN_CODE			1. eMIS 코드상세내역(CN_CODE) 테이블의 전체 건수와 EIS 코드상세내역(CM_CODE) 테이블의 전체 건수가 일치해야 함 2. 코드ID는 코드그룹(CN_CODE_KIND) 테이블에 존재해야 함 3. 시작일자는 종료일자보다 같거나 작아야 함 4. 검증 쿼리 <pre> SELECT COUNT(*)   FROM CN_CODE A,        CN_CODE_KIND B  WHERE A.CODE_ID = B.CODE_ID; /***** SELECT COUNT(*)   FROM CN_CODE A  WHERE START_DATE &gt; TRMN_DATE; /***** SELECT COUNT(*) FROM CN_CODE;                     </pre>		

## 학습정리

### 1. 공통 모듈 설계 순서



### 2. 공통 모듈 설계 단계별 내용

- 공통 모듈 분리는 Use Case, Class Diagram 등 다양한 설계로부터 분리함
- 분리된 공통 모듈은 Class, Component, Sequence Diagram 등의 정적·동적 설계를 함
- 공통 모듈을 Component로 설계 시 인터페이스 명세를 상세하게 작성함
- 공통 모듈의 Class Diagram에서 OR Mapping을 활용하여 ER Diagram을 설계함
- 공통 모듈 구현 방식에는 클래스, 라이브러리, 컴포넌트, 프레임워크가 있음