

# 10 주차

## 모듈화 설계

1. 모듈화 설계 개요
2. 결합력의 이해
3. 응집력의 이해
4. 모듈화 응용 기술 (그닥?)

## 1. 모듈화 설계 개요

**중요한 원칙 - 응집력은 높이고, 결합력을 줄이도록 설계한다.**

소프트웨어 품질을 향상시키기 위해서 소프트웨어 공학 기법을 적용하는데 공학 원리 중 가장 근간이 되는 원리가 모듈화.

독립적인 기능이 있는 논리적 묶음에 해당하는 모듈로 소프트웨어 시스템을 구성할 수 있도록 개발한다는 의미

### 모듈화 장점

이해하기 쉽다. 팀 단위의 개발 작업이 쉬워진다. 변경에 의한 수정 사항 반영이 쉽다. 재 사용 가능성이 높아진다. 추적성이 높아진다.

### 좋은 설계를 수행한 소프트웨어의 특성들(알아둘 것)

설계가 계층적 구조를 나타낸다. 즉, 시스템은 서브 시스템으로 구성되고, 각 서브 시스템은 하위의 기능 영역으로, 각 기능 영역은 모듈들로 구성되는 계층적 구조를 나타낸다.

설계 결과가 모듈로 구성되어야 하고, 독립적인 특성이 있는 기능 단위로 구현할 수 있어야 한다.

데이터와 처리 절차가 구분 가능하고 분리된 표현으로 나타나야 한다.

## 2. 결합력

결합력은 모듈 간의 의존성 관계를 나타낸다. 결합력이 작을수록 모듈간의 의존성이 떨어지므로 좋은 설계이다. 모듈이 독립적인 기능을 갖도록 설계하고 불필요한 중복을 제거해야 한다.

가능한 모듈 간 상호작용을 줄인다.

### 결합력을 최소화할 때 장점

시스템의 구성 요소 간에 결합이 느슨해진다.

변경에 의한 파동 효과를 막을 수 있다.

SW에 대한 이해도를 높인다.

모듈 인터페이스가 단순해진다.

### 결합력 유형

읽어볼 것 타이핑하기 힘들, 결국엔 결합력을 줄이자가 목표임

메시지, 데이터, 스탬프, 제어, 외부, 공유, 내용 결합력

#### 메시지 결합력

결합의 정도가 가장 느슨한 유형이다. 객체 간 상호작용은 최소화되며, 상호작용은 메시지 전달이라는 한가지 개념에 의해서만 이루어지는 경우에 해당한다.

#### 데이터 결합력

2개의 모듈이 정수형, 문자형 등의 단순한 기본 데이터 타입을 갖는 변수들에 의해 상호작용 한다면 두 모듈은 데이터 결합력 관계에 있다고 한다.

#### 스탬프 결합력

두 모듈이 구조체와 같은 복합 데이터를 이용하여 상호작용 한다면 스탬프 결합력을 갖는다고 정의한다.

## 제어 결합력

한 모듈에서 다른 모듈로  $n$ 개 변수를 전달할 때, 그 매개변수가 호출되는 함수의 내부 행위를 제어하는 역할을 한다면 이 두 모듈은 제어 결합력 관계에 있다고 정의한다. 이 경우 대체로 변수 flag를 사용하여 함수의 내부 행위를 제어한다.

## 외부 결합력

2개의 모듈이 외부에 존재하는 다른 정보를 공유하고 있다면 이 모듈은 외부 결합 관계에 있다고 정의한다. 여기서 외부 정보는 파일, 디바이스 인터페이스, 프로토콜을 의미하며 한 모듈과 외부에서 제공하는 장치들 간 상호작용과 관련되어 있다.

## 공유 결합력

2개의 모듈이 광역 변수를 공유하고 있다면 이 모듈은 공유 결합력 관계에 있다고 할 수 있다. 이는 외부 결합력과 비슷하게 한 모듈의 공유 변수에 대한 수정이 다른 모듈에 영향을 줄 수 있음을 의미한다. 공유 변수를 사용시 메모리 사용량을 줄이고 프로그램을 간단히 작성할 수 있지만, 모듈 간의 의존성을 높이는 문제가 있다.

## 내용 결합력

한 모듈에서 다른 모듈의 내부를 직접 참조한다면 내용 결합 관계에 있다고 한다. 내용 결합력은 거의 발생하지 않는다.

# 3. 응집력

모듈을 구성하는 내적 요소 간의 기능적 관련성 강도를 측정하는 척도이다.  
모듈을 구성하는 내적 요소는 명령어, 변수 정의, 함수 호출 등이 있다.

## 3.1 응집력 유형

기능, 순차, 교환, 절차, 시간, 논리, 우연

### 기능 응집력

모듈을 구성하는 모든 요소가 단지 하나의 기능을 구현하기 위해 구성되었다면 기능 응집력이 있다고 정의한다.

### 순차 응집력

모듈을 구성하는 문장 관계에서, 한 문장의 실행 결과가 다음 문장의 입력으로 사용되는 경우에 순차 응집력에 해당한다. 순차 응집력은 매우 좋은 설계에 해당하며, 유지 보수가 용이하다는 장점이 있다.

### 교환 응집력

모듈을 구성하는 모든 요소가 동일한 입력 또는 출력을 사용한다면 교환 응집력이 있다고 정의한다.

### 절차 응집력

모듈을 구성하는 문장들이 의미상 서로 관련 없지만 제어 흐름의 순서가 있는 경우에 해당한다. 절차 응집력이 있는 한 가지 기능을 수행하기 위한 목적보다는 실행 순서와 관련성이 있도록 구성한 경우이다.

### 시간 응집력

모듈을 구성하는 각 문장이 SW 실행의 특정 시간과 관련이 있는 것으로만 구성된 경우에 해당한다. 즉, 모든 요소가 처리되는 시간과 관련성이 있을 경우에 해당한다.

### 논리 응집력

모듈을 구성하는 모든 요소가 논리적으로 같은 유형의 외부 동작들로 구성되는 경우에 해당한다. 즉, 다수의 유사한 행위를 포함하고 있으며 이 행위들은 flag 변수에 의해 실행 여부가 결정되는 일이 많다.

### 우연 응집력

가장 나쁜 형태의 응집력이며, 절대 작성되어서는 안되는 모듈 구성이다.  
모듈을 구성하는 모든 요소가 아무런 관련성이 없는 것으로 묶인 경우에 해당한다.

# 모듈화와 기술 부채

모듈화의 척도 관점에서 좋은 아키텍처일수록 결합력 보다는 응집력에 더 집중해야 한다.  
결합력의 경우 인터페이스 종속성보다 아키텍처 컴포넌트 간에 존재할 수 있는 기능 의존성 문제를 해결하는 데 집중해야 한다.