

7주차 2차시 모듈 응집도

【학습목표】

1. 응집도 및 모듈 응집도의 개념을 설명할 수 있다.
2. 응집도 유형별 정의를 구분할 수 있으며, 판정 기준을 설명할 수 있다.

학습내용1 : 응집도 개념

* 응집도 : (Cohesion) 모듈의 구성요소 사이의 관련성을 나타내는 척도임

모듈 응집도  최대한 강하게 만들어야함

<응집도의 유형은 7 가지가 있음>

- 교재 「P. 395」의 「표 12-2」 참조

- ① 우연적 응집도
- ② 논리적 응집도
- ③ 시간적 응집도
- ④ 절차적 응집도
- ⑤ 통신적 응집도
- ⑥ 순차적 응집도
- ⑦ 기능적 응집도

<가장 나쁜 경우부터 차례대로 나열>

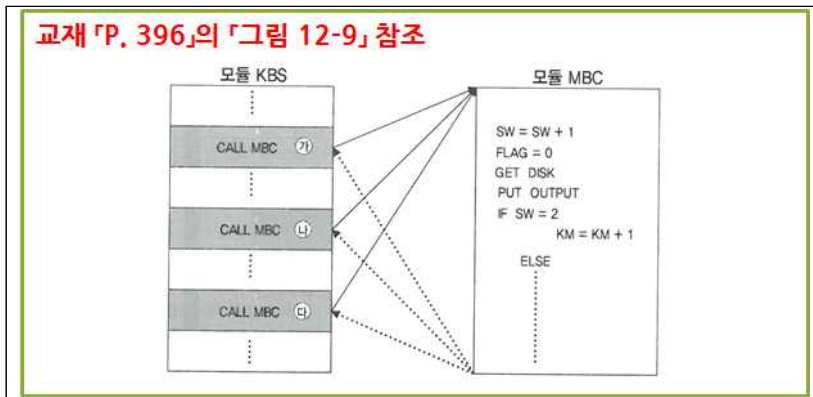
학습내용2 : 응집도 유형 및 판정 기준

1. 응집도 유형

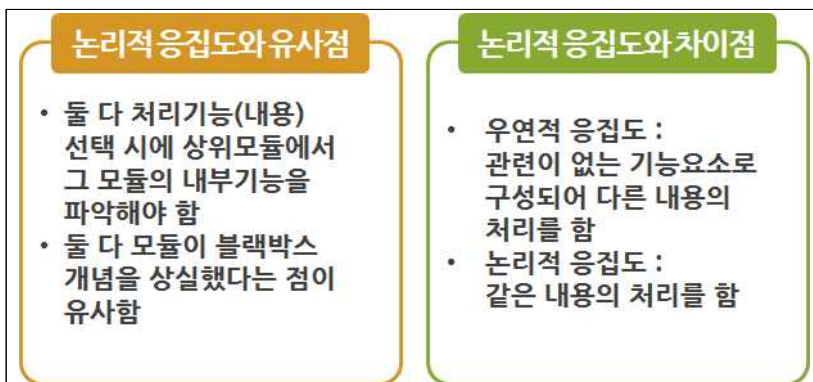
1) 우연적 응집도(암합적 응집도, 부합적 응집도 : (Coincidental Cohesion))

① 정의 : 모듈을 구성하는 요소들 사이에 관련이 없는 것끼리 모인 형태의 응집을 의미함

- 교재 'P. 396'의 「그림 12-9」 참조



② 발생원인 : 기억장치의 용량 때문에 기존 모델을 여러 개로 분할하는 경우, 우연히 중복된 내용을 통합해 하나의 모듈로 만들 때 발생함



2) 논리적 응집도(Logical Cohesion)

* 원리 : 논리적으로 관련이 있는 요소들을 모아 모듈로 구성하는 응집형태임

* 예시 :

- 모듈 전체가 입출력을 조작하는 요소로 구성된 경우
- 특정한 업무의 추가 · 삭제 · 갱신 등의 갱신처리 요소로 구성된 경우

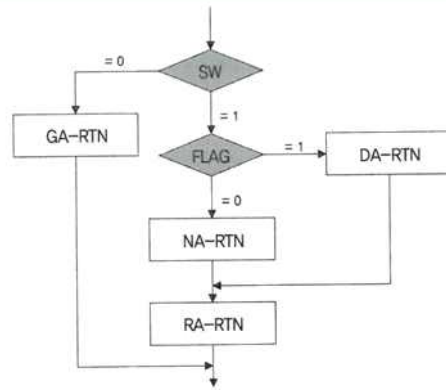
<논리적 응집도를 가진 모듈에서 필요한 내용만을 선택하여 사용하기 위해서는 파라미터 값으로 조정해야 함>

- 교재 「P. 397」의 「그림 12-10」 참조

* 논리적 응집도를 포함한 모듈의 문제점

- 모듈 내의 특정한 기능을 변경할 경우 많은 문제를 유발할 가능성이 존재함
- 동일한 모듈 내에 다양한 기능요소가 존재하므로 입출력 변수, 버퍼, 코드, 상수를 공유하기 때문에 유지관리가 어려움

교재 「P. 397」의
「그림 12-10」 참조

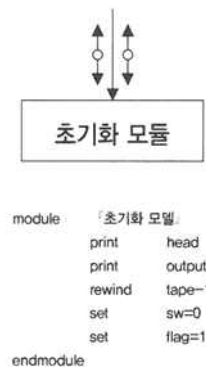


3) 시간적 응집도(Temporal Cohesion)

* 정의 : 초기화 및 종료 모듈처럼 특정한 시간에 처리되는 몇 가지 기능을 하나의 모듈로 만드는 경우에 나타나는 응집의 형태임

- 교재 「P. 398」의 「그림 12.11」 참조

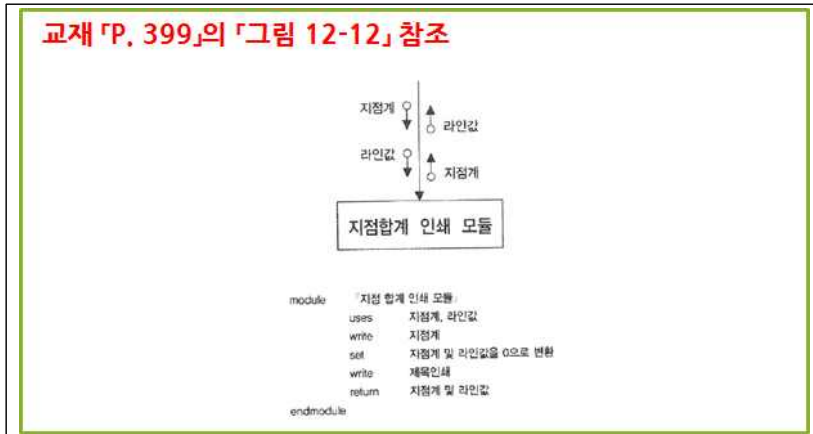
교재 「P. 398」의 「그림 12.11」 참조



4) 절차적 응집도(Procedural Cohesion)

* 정의 : 모듈 구성요소가 일정한 시간에 특정한 순서에 따라서 실행된다는 사실 이외에, 기능이 서로 다른 요소로 구성되어 있고, 기능이 상호 관계가 없는 처리에 포함된 응집형태임

- 교재 「P. 399」의 「그림 12-12」 참조



- 절차적 응집도의 모듈 내에서 하나의 처리를 종료하고 다음 처리로 이동은 「제어신호」에 의해서 이루어짐

- 기능의 일부를 모듈화 시에(순서도의 일부를 모듈화 할 때) 절차적 응집 도가 발생함

* 절차적 응집도의 특성

- 여러 개의 다른 기능 요소들로 모듈이 구성됨
- 각 기능들은 순차적으로 수행됨
- 각 기능 요소들 사이에는 데이터가 이동되지 않음
- 해당 모듈로 보내는 데이터(전달 데이터)와, 그 모듈로부터 돌려 받는 데이터(변환 데이터) 사이에는 거의 관련성이 없음
- 부분적인 결과(부분적으로 편집된 입력 데이터), 스위치, 플래그 등을 전달함

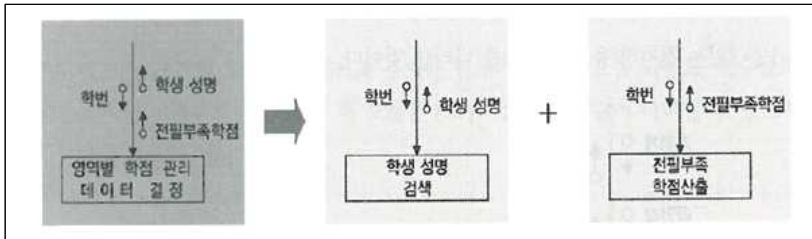
5) 통신적 응집도(Communicational Cohesion)

* 정의 : 한 모듈 내에서 서로 다른 기능(Task) 요소들이 동일한 데이터를 사용하여 처리하지만, 그 처리 절차가 다르고, 관련이 없는 응집형태임

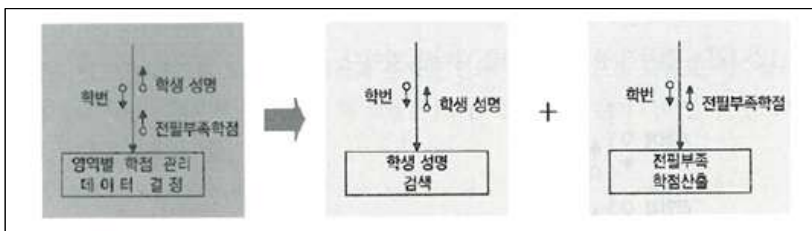
- 위의 특성 때문에 각 기능들을 분할 가능함
- 이처럼 분할하면「순차적 응집도」와 유사하게 보임
- 그러나 동일한 데이터를 처리순서에 관계없이 사용 가능하다는 점이 차이점임

* 통신적 응집도의 문제점

- 처리 기능 코드가 중복되는 경우가 많음(교재 「P. 400」의 「그림 12-13」의 왼쪽 참조)

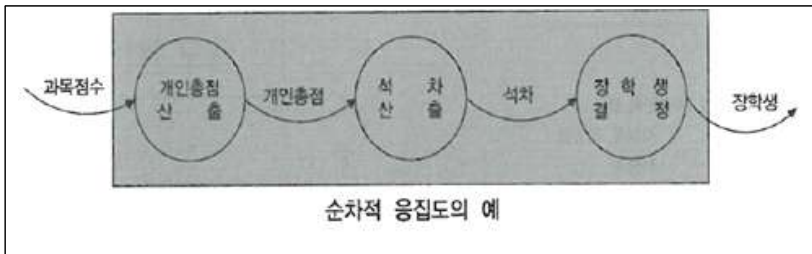


- 「처리요소」들 사이에 ⇨「코드의 공유」를 하는 경우가 발생함
- 문제점 해결방안
 - 통신적 응집도 모듈을 기능적으로 응집된 모듈로 분리함 (교재 「P. 400」의 「그림 12-13」의 중앙 및 오른쪽 참조)

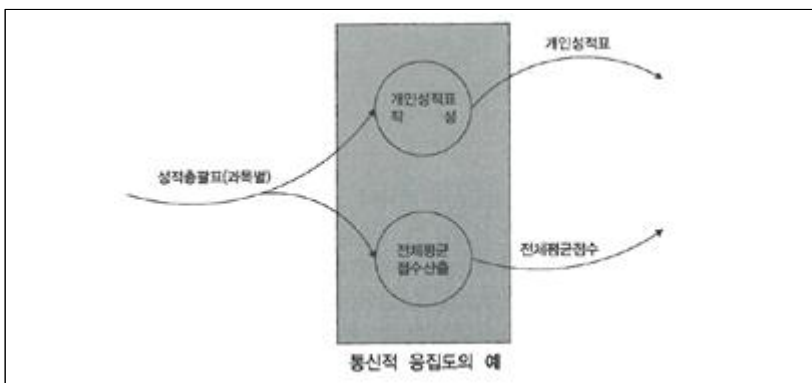


* 통신적 응집도와 순차적 응집도 차이점

- 순차적 응집도 모듈 : 기능요소들이 반드시 순차적으로 수행됨 (교재 「P. 400」의 「그림 12-14」의 위쪽 참조)



- 통신적 응집도 모듈 : 몇 가지 기능요소 중에서 어느 것이 먼저 수행되어도 무방함 (교재 「P. 400」의 「그림 12-14」의 아래쪽 참조)

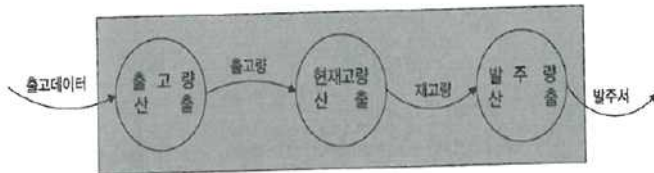


6) 순차적 응집도(Sequential Cohesion)

* 정의 : 동일 모듈 내에서 모듈의 구성요소가 「하나의 활동(Activity)의 결과로 산출된 출력 데이터」를 「그 다음에 연속되는 활동의 입력 데이터」로 사용하는 응집형태임

- 순차적 응집도의 예 : 교재 「PP. 401~402」의 「그림 12-15 및 16」 참조

순차적 응집도의 예 : 교재 「PP. 401~402」의 「그림 12-15」 참조



순차적 응집도의 예 : 교재 「PP. 401~402」의 「그림 12-16」 참조

```

module : 「array 내용 판독」
    DIMENSION      KM(2, 3, 4)
    READ(5, 10)KM
    :
endmodule

```

7) 기능적 응집도(Functional Cohesion)

* 정의 : 모듈의 모든 구성요소들이 한 가지 기능을 수행하기 위해서 존재하는 경우를 의미함

<응집도 형태 중에서 응집도가 가장 강함(최선의 응집도임)>

- 기능적 응집도의 예 : 교재「PP. 402~403」의「그림 12-17 및 18」 참조

기능적 응집도의 예 : 교재「PP. 402~403」의「그림 12-17」 참조

```
module : 『M의 사인값 계산 및 인쇄』
M=0
RAD=3.14159/180
DO 30 K=1, 89
M=M+1
RADN=M*RAD
SS=SIN(RADN)
30 WRITE(6, 77)SS
endmodule
```

기능적 응집도의 예 : 교재「PP. 402~403」의「그림 12-18」 참조

```
module : 『행렬의 가산』
DIMENSION KBS(5, 5), MBC(5, 5), KAL(5, 5)
DO 35 I=1, 5
DO 35 J=1, 5
35 KAL(I, J)=KBS(I, J)+MBC(I, J)
:
endmodule
```

2. 응집도 판정 기준

- 1) 교재 「P. 403」의 「그림 12-19」의「디시전 트리」 참조



- 2) 설계 트레이드 오프(Design Trade Off)

- 모듈설계 시에 모듈의 응집도에 강도를 낮추고 다른 이점을 얻기 위한 조치를 의미함

【학습정리】

1. 응집도 개념을 학습한다.
2. 응집도 유형을 파악한다.
3. 응집도 판정 기준을 설명할 수 있다.