

정보처리기사 필기

소프트웨어 설계 6 애플리케이션설계 ①

양문자 선생님

출처 : ncs 학습모듈(NCS능력단위 애플리케이션설계)

참조 : 객체지향소프트웨어공학/최은만/한빛아카데미

참조 : 객체 지향 설계와 분석을 위한 UML 기초와 응용/한정수/한빛아카데미

소프트웨어 설계

차례

1 요구사항 확인

2 화면 설계

3 애플리케이션 설계

1) 공통 모듈 설계

2) 객체지향 설계

4 인터페이스 설계

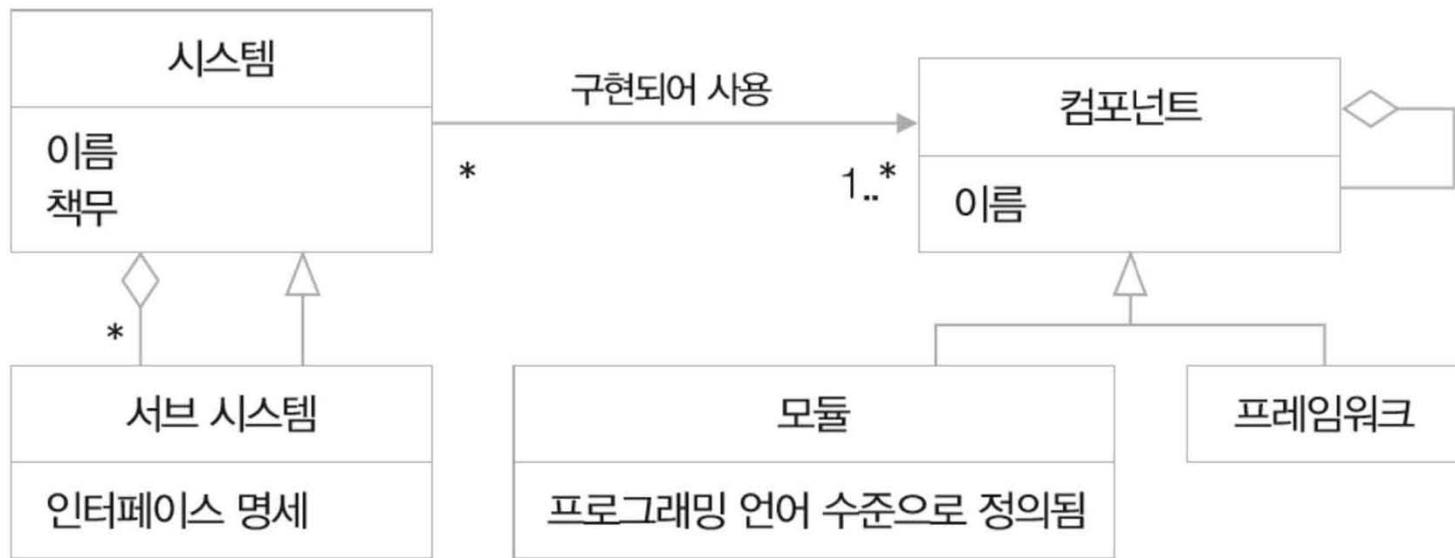
소프트웨어 설계

■ 학습목표

- 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 이에 따른 애플리케이션 구현을 수행하기 위해 공통 모듈 설계, 타 시스템 연동에 대하여 상세 설계할 수 있다.
- 재사용성 확보와 중복 개발을 회피하기 위하여, 전체 시스템 차원과 단위 시스템 차원의 공통부분을 식별하여 이에 대한 상세 명세를 작성할 수 있다.
- 개발할 응용 소프트웨어의 전반적인 기능과 구조를 이해하기 쉬운 크기로 공통 모듈을 설계할 수 있다.
- 소프트웨어 측정지표 중 모듈 간의 결합도는 줄이고 개별 모듈들의 내부 응집도는 높이기 위한 공통 모듈을 설계할 수 있다.

소프트웨어 설계

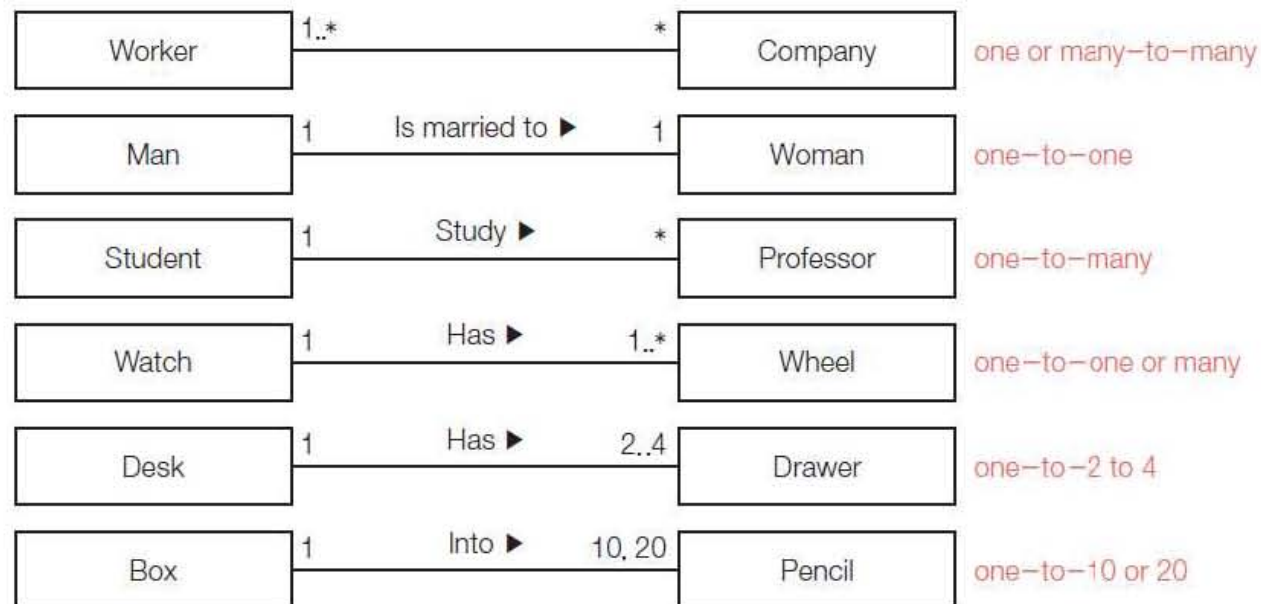
〈 시스템을 이루는 요소들 〉



소프트웨어 설계

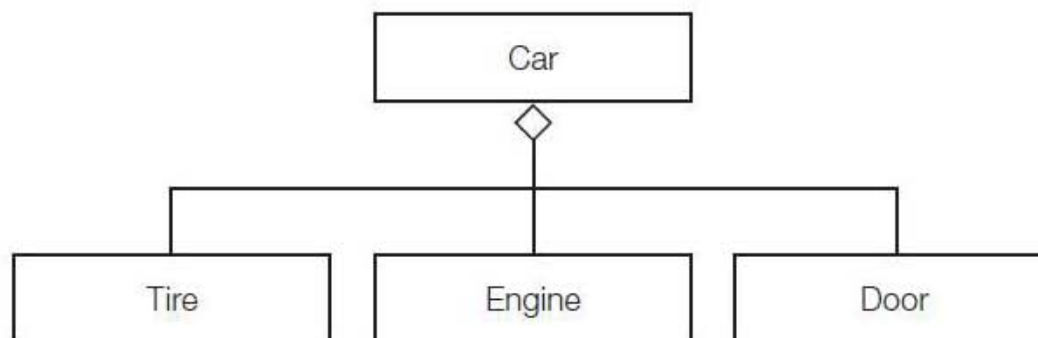
UML 구성요소 - 연관관계

: 다중성 - 객체 하나에 몇 개의 객체가 연결되어 있는지를 밝히는 것



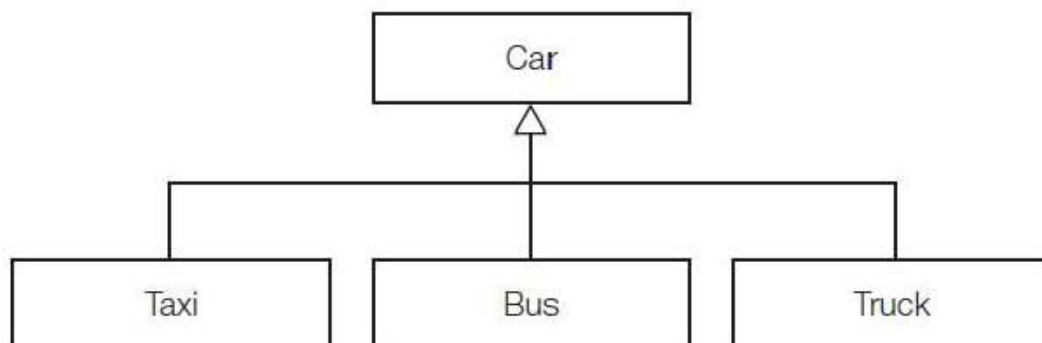
UML 구성요소 - 연관관계

: 집합연관 - 전체 쪽 객체 하나가 부분 쪽 객체들을 소유, has-a 관계



UML 구성요소 – 일반화 관계

: 일반화된 사물과 좀 더 특수화된 사물 사이의 관계



소프트웨어 설계

<컴포넌트>

：분명한 역할을 가지고 있는 하드웨어 또는 소프트웨어 조각

－독존할 수 있으며 같은 기능을 가진 다른 컴포넌트로 대체 시킬 수 있어야함

－대부분의 컴포넌트는 재사용 가능 하도록 설계

ex) 사용자인터페이스, 백킹업무, 참고관리등

소프트웨어 설계

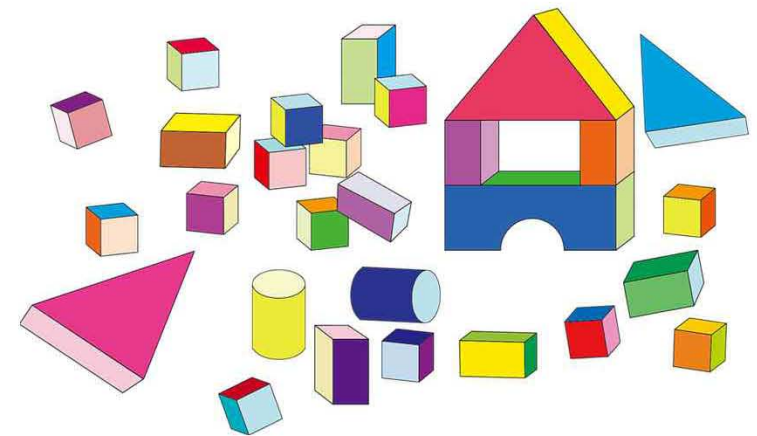
(1) 공통 모듈 설계

1. 모듈의 개념

전체 프로그램의 기능 중 특정 기능을 처리할 수 있는 실행 코드.
자체적으로 컴파일 가능하고 다른 프로그램에서 재사용이 가능하다.

2. 공통 모듈

여러 기능 및 프로그램에서 공통적으로 사용할 수 있는 모듈
ex > 날짜 처리를 위한 유틸리티 모듈



3. 공통 모듈에 대한 명세 작성의 원칙

- (1) 정확성(Correctness) : 해당 기능이 실제 시스템 구현 시 필요한지 여부를 알 수 있도록 정확하게 작성한다.
- (2) 명확성(Clarity) : 해당 기능에 대해 일관되게 이해되고 한 가지로 해석될 수 있도록 작성한다.
- (3) 완전성(Completeness) : 시스템이 구현될 때 필요하고 요구되는 모든 것을 기술한다.
- (4) 일관성(Consistency) : 공통 기능들 간에 상호 충돌이 없도록 작성한다.
- (5) 추적성(Traceability) : 공통 기능에 대한 요구사항 출처와 관련 시스템 등의 유기적 관계에 대한 식별이 가능하도록 작성한다.

<재사용(Reuse)>

: 개발 시간 및 비용 절감을 위하여 이미 검증된 기능을 파악하고 재구성하여 시스템에 응용하기 위해 적합하게 최적화시키는 작업이다.

1. 함수와 객체 재사용
2. 컴포넌트 재사용
3. 애플리케이션 재사용

〈재사용을 위한 필요 항목〉

- (1) 재사용과 관련된 내용이 이해하기 쉽고, 누구나 사용 가능하도록 사용법이 공개되어야 한다.
- (2) 외부 모듈과의 연관성(결합도)은 적어야 하며, 자체적인 완성도(응집도)는 높아야 한다.

소프트웨어 설계

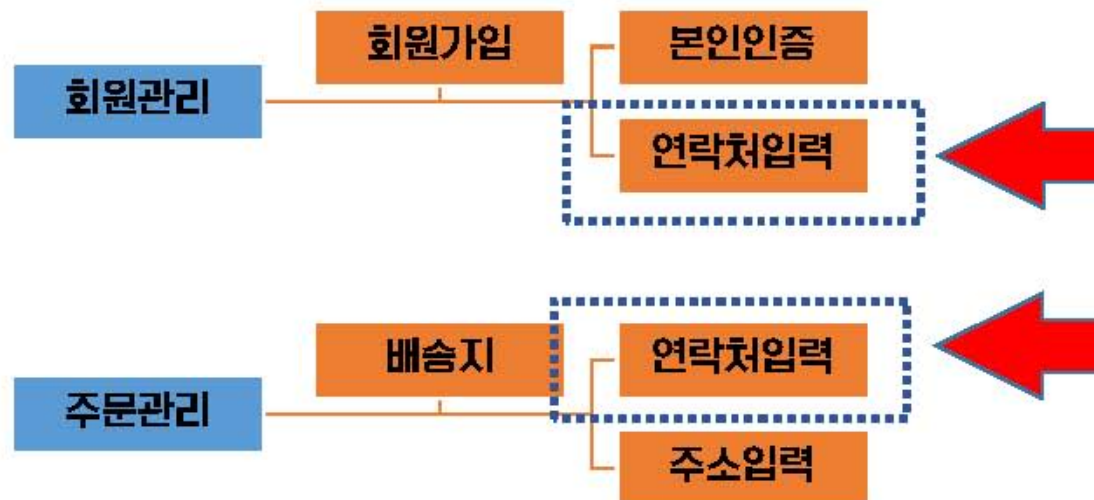
<공통 모듈 식별 및 명세하기>

- I. 전체 시스템 차원과 단위 시스템 차원의 공통부분을 식별한다.



소프트웨어 설계

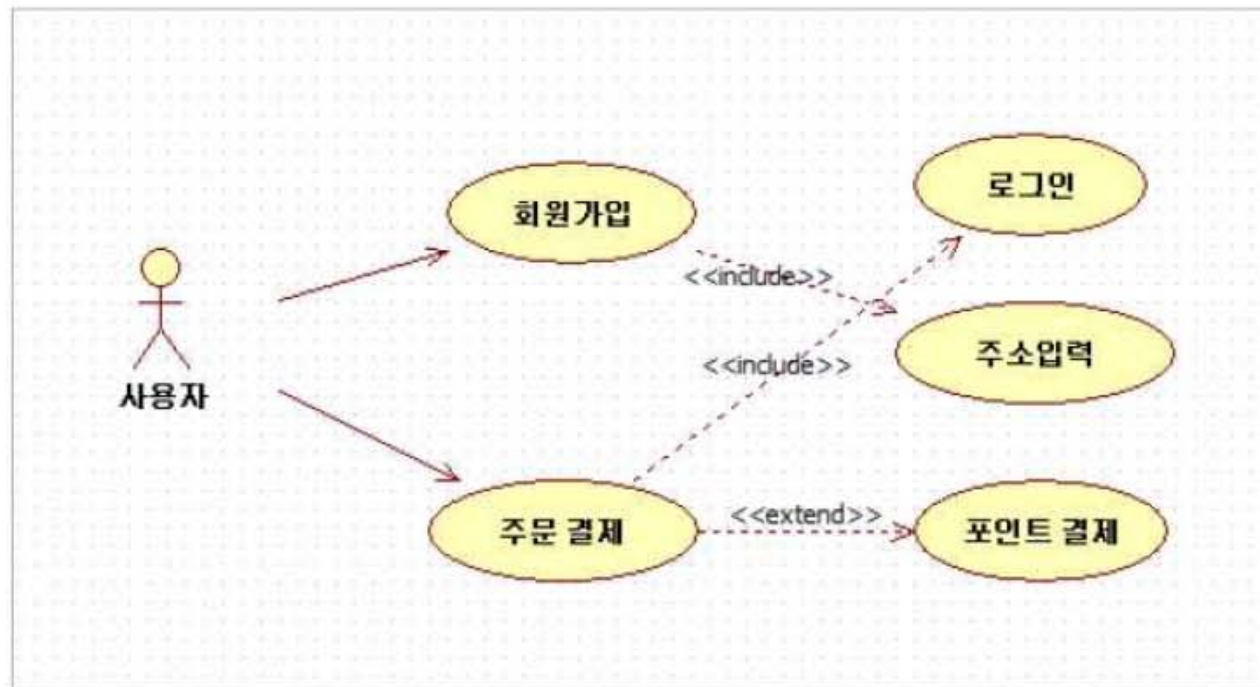
1. 단위 시스템의 업무 기능을 분석하여 공통부분을 식별한다.



소프트웨어 설계

2. 유스케이스를 분석하여 공통부분을 식별한다.

: <<include>>관계로 식별된 기능에 대해 공통 기능으로 적용 가능한지 분석한다.



소프트웨어 설계

3. 단위 시스템의 공통부분에 대한 검토 회의를 진행한다.
 - 1) 단위 시스템 이해관계자와의 검토 일정을 수립한다.
 - 2) 공통 기능에 대한 적용 여부를 검토한다.
 - 3) 공통 기능으로 적용 가능한 다른 기능에 대한 의견을 취합한다.
 - 4) 공통 기능에 대한 담당자를 정의한다.
4. 단위 시스템에 대해 공통부분으로 식별된 기능에 대한 상세 기능을 기록한다.
5. 전체 시스템 차원의 공통부분을 식별한다.

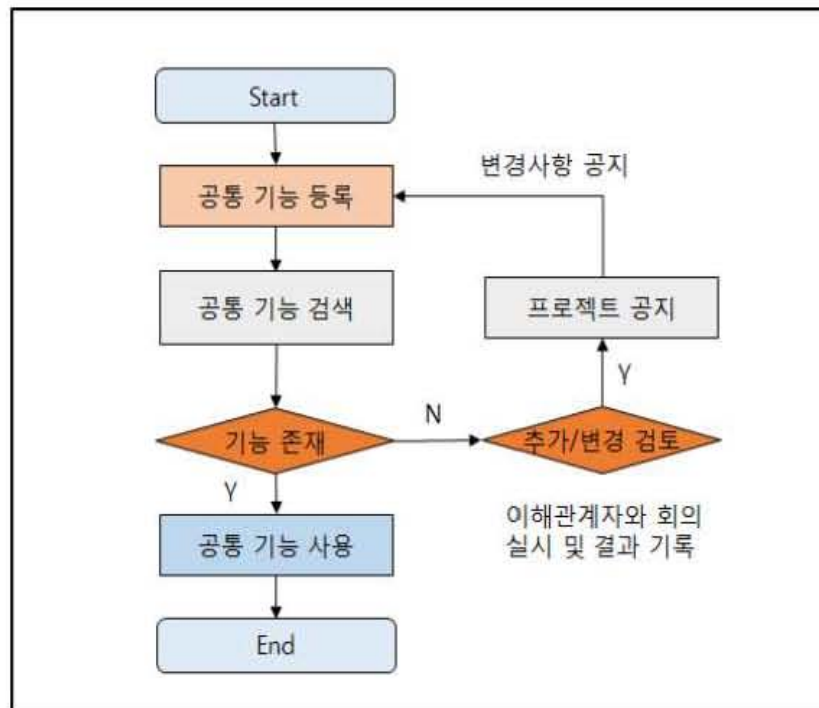
소프트웨어 설계

6. 공통 기능 관리 프로세스를 수립한다.

1) 공통 기능 관리와 관련된 책임과 역할을 정의한다.

ex> 프로젝트 관리자, 공통 개발 프로젝트 리더, 개발자, 공통 모듈 담당자

2) 재사용 및 공통 기능 관리 절차를 수립한다.



<모듈화(Modularity)>

1. 모듈화 개념

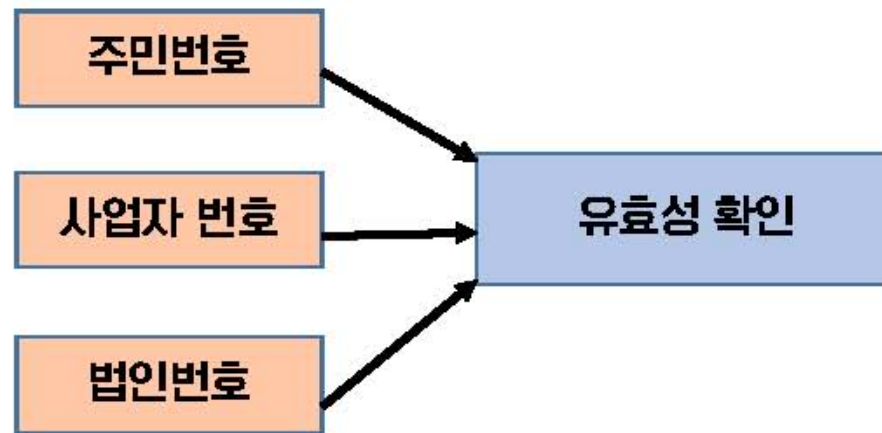
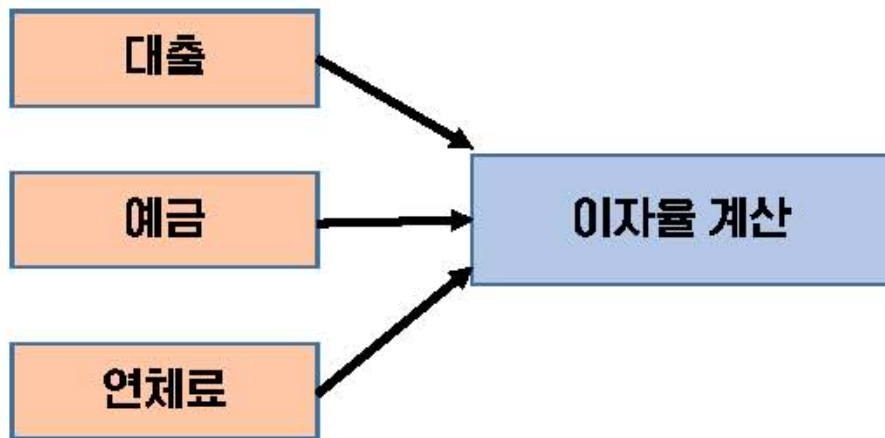
프로그램이 효율적으로 관리될 수 있도록 시스템을 분해하고 추상화함으로써 소프트웨어 제품의 성능을 향상시키거나 시스템의 수정 및 재사용, 유지 관리를 용이하게 하는 기법을 의미한다.

2. 모듈화 필요성

모듈의 크기가 너무 작아서 모듈 개수가 많아지면 모듈 간의 통합 비용이 많이 들고, 모듈의 크기가 너무 크면 모듈 간의 통합 비용이 상대적으로 줄어드는 대신 모듈 하나를 개발하는 데 드는 비용이 커진다.

소프트웨어 설계

<공통 기능에 대한 모듈화>



〈응집도〉

응집도는 모듈 내부에서 구성 요소 간에 밀접한 관계를 맺고 있는 정도로 평가되며, 응집도가 높을수록 필요한 요소들로 구성되어 있고 낮을수록 관련이 적은 요소들로 구성되어 있다.

소프트웨어 설계

1. 응집도의 유형

구분	설명
기능적 응집도(Functional Cohesion)	<ul style="list-style-type: none">• 모듈 내부의 모든 기능이 단일한 목적을 위해 수행• 예) 원넓이 계산
순차적 응집도(Sequential Cohesion)	<ul style="list-style-type: none">• 한 활동으로부터 나온 출력값을 다른 활동이 사용할 경우• 예) 파일에서 데이터를 읽고, 데이터를 처리
통신적 응집도 (Communication Cohesion)	<ul style="list-style-type: none">• 동일한 자료를 사용하여 다른 기능을 수행하는 활동들이 모여 있을 경우• 처리 순서는 상관없다.
절차적 응집도 (Procedural Cohesion)	<ul style="list-style-type: none">• 관련없는 기능요소가 배열된 순서로 수행됨.• 요소들이 서로 관련이 없다는 것만 빼면 순차적 응집도와 유사함.

소프트웨어 설계

1. 응집도의 유형

구분	설명
시간적 응집도(Temporal Cohesion)	<ul style="list-style-type: none">특정 시간에 처리되어야 하는 활동들을 한 모듈에서 처리할 경우ex> 한 시스템의 시작이나 초기화 하는 동안 사용 되는 코드들을 같이 모아 둔 경우, 예외사항 발생시
논리적 응집도 (Logical Cohesion)	<ul style="list-style-type: none">유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들이 한 모듈들에서 처리되는 경우예) 모든 마우스 및 키보드 입력 처리 루틴 그룹화
우연적 응집도(coincidental Cohesion)	<ul style="list-style-type: none">모듈 내부의 각 구성 요소들이 연관이 없을 경우

소프트웨어 설계

2. 응집도와 품질

다양한 기준으로 모듈을 구성할 수 있으나 품질 측면에서 기능적 응집도가 가장 품질이 높고, 우연적 응집도가 가장 낮다.



〈결합도〉

모듈과 모듈 간에 어느 정도 관련성이 있는지를 나타내며, 관련이 적을수록 모듈의 독립성이 높아 모듈 간 영향이 적어지게 된다.

소프트웨어 설계

1. 결합도의 유형

구분	설명
자료 결합도(Data Coupling)	<ul style="list-style-type: none">모듈 간의 인터페이스로 전달되는 파라미터를 통해서만 모듈 간의 상호 작용이 일어나는 경우
스탬프 결합도(Stamp Coupling)	<ul style="list-style-type: none">모듈 간의 인터페이스로 배열이나 오브젝트, 스트럭처 등이 전달되는 경우데이터 구조 결합도
제어 결합도(Control Coupling)	<ul style="list-style-type: none">단순 처리할 대상인 값만 전달되는 게 아니라 어떻게 처리를 해야 한다는 제어 요소가 전달되어 다른 모듈의 흐름을 제어
외부 결합도(External Coupling)	<ul style="list-style-type: none">두 개의 모듈이 외부에서 도입된 데이터 포맷, 통신 프로토콜, 디바이스 인터페이스를 공유할 때 발생.
공통 결합도(Common Coupling)	<ul style="list-style-type: none">파라미터가 아닌 모듈 밖에 선언되어 있는 전역 변수를 참조하고 전역 변수를 갱신하는 식으로 상호 작용하는 경우
내용 결합도(Content Coupling)	<ul style="list-style-type: none">다른 모듈 내부에 있는 변수나 기능을 다른 모듈에서 사용하는 경우

소프트웨어 설계

2. 결합도와 품질

다양한 결합으로 모듈을 구성할 수 있으나 품질 측면에서 자료 결합이 가장 품질이 높고, 내용 결합이 가장 낮다.



결합도 낮음(좋은 품질) ←

문제풀이

1. 바람직한 소프트웨어 설계 지침이 아닌 것은?

- ① 적당한 모듈의 크기를 유지한다.
- ② 모듈 간의 접속 관계를 분석하여 복잡도와 중복을 줄인다.
- ③ 모듈 간의 결합도는 강할수록 바람직하다.
- ④ 모듈 간의 효과적인 제어를 위해 설계에서 계층적 자료 조직이 제시되어야 한다.

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 설계)

2. 응집도의 종류 중 서로 간에 어떠한 의미 있는 연관관계도 지니지 않은 기능 요소로 구성되는 경우이며, 서로 다른 상위 모듈에 의해 호출되어 처리상의 연관성이 없는 서로 다른 기능을 수행하는 경우의 응집도는?

- | | |
|-----------------------|-------------------------|
| ① Functional Cohesion | ② Sequential Cohesion |
| ③ Logical Cohesion | ④ Coincidental Cohesion |

(2020년 4회 정보처리기사 필기 기출문제 프로그래밍언어 활용)

문제풀이

3. 결합도(Coupling)에 대한 설명으로 틀린 것은?

- ① 데이터 결합도(Data Coupling)는 두 모듈이 매개변수로 자료를 전달할 때, 자료구조 형태로 전달되어 이용 될 때 데이터가 결합되어 있다고 한다.
- ② 내용 결합도(Content Coupling)는 하나의 모듈이 직접적으로 다른 모듈의 내용을 참조할 때 두 모듈은 내용적으로 결합되어 있다고 한다.
- ③ 공통 결합도(Common Coupling)는 두 모듈이 동일한 전역 데이터를 접근한다면 공통결합되어 있다고 한다.
- ④ 결합도(Coupling)는 두 모듈간의 상호작용, 또는 의존도 정도를 나타내는 것이다.

(2020년 4회 정보처리기사 필기 기출문제 프로그래밍언어 활용)

문제풀이

4. 공통 모듈에 대한 명세 기법 중 해당 기능에 대해 일관되게 이해하고 한 가지로 해석될 수 있도록 작성하는 원칙은?

- | | |
|---------|-------|
| ① 상호작용성 | ② 명확성 |
| ③ 독립성 | ④ 내용성 |

(2020년 1, 2회 정보처리기사 필기 기출문제 소프트웨어 설계)

문제풀이

5. 응집도(cohesion)는 모듈 내의 구성요소들 간의 연관 정도를 나타낸다. 아래는 각 구성요소가 응집된 형태에 따라 분류한 것인데, 이 중 응집도가 가장 낮은 것과 가장 높은 것을 순서대로 옳게 짝지은 것은?

- ㄱ. 우연적(coincidental) 응집도
- ㄴ. 절차적(procedural) 응집도
- ㄷ. 순차적(sequential) 응집도
- ㄹ. 교환적(communucational) 응집도
- ㅁ. 기능적(functional) 응집도
- ㅂ. 논리적(logical) 응집도
- ㅅ. 시간적(temporal) 응집도

- ① ㄱ - ㄷ
- ③ ㄹ - ㅅ

- ② ㄱ - ㅁ
- ④ ㅂ - ㄴ

(2020_7급_서울시_소프트웨어공학(A)_2020년 10월 17일 소프트웨어공학)

문제풀이

6. 구조적 설계의 평가 기준 중 모듈 응집도가 강한 것에서 약한 것의 순서로 옳게 나열된 것은?

- ① 절차적 응집도 → 통신적 응집도 → 순차적 응집도 → 기능적 응집도
- ② 통신적 응집도 → 절차적 응집도 → 순차적 응집도 → 기능적 응집도
- ③ 절차적 응집도 → 통신적 응집도 → 기능적 응집도 → 순차적 응집도
- ④ 기능적 응집도 → 순차적 응집도 → 통신적 응집도 → 절차적 응집도

(2020년 3회 정보처리산업기사 필기 기출문제 시스템분석설계)

7. 모듈 내부의 모든 기능 요소들이 단일한 목적을 위해 수행하는 경우의 응집도는?

- ① Coincidental cohesion ② Functional cohesion
- ③ Procedural cohesion ④ Temporal cohesion

(2020년 1, 2회 정보처리산업기사 필기 기출문제 시스템분석설계)

소프트웨어 설계

문제풀이

8. <보기>는 모듈화를 중심으로 한 소프트웨어 설계방법에 대한 설명이다. 빈칸의 내용을 올바르게 나열한 것은

- 결합도(coupling)와 응집도(cohesion)는 모듈의 (㉠)을 판단하는 기준이다.
- 결합도란 모듈 (㉡)의 관련성을 의미하며, 응집도란 모듈 (㉢)의 관련성을 의미한다.
- 좋은 설계를 위해서는 결합도는 (㉣), 응집도는 (㉤) 방향으로 설계해야 한다.

	㉠	㉡	㉢	㉣	㉤
①	독립성	사이	내부	작게	큰
②	독립성	내부	사이	크게	작은
③	추상성	사이	내부	작게	큰
④	추상성	내부	사이	크게	작은

(2012년 우정사업본부 계리직 기출문제 컴퓨터일반)