

정보처리기사 필기

소프트웨어 설계  요구사항 확인 ③

양문자 선생님

출처 : ncs 학습모듈(NCS능력단위 요구사항확인)
참고 : 시스템분석과설계/허원실/한빛아카데미
참고 : 소프트웨어 공학의 소개/한혁수/홍릉과학출판사

소프트웨어 설계

차례

1 요구사항 확인

- 1) 현행 시스템 분석
- 2) 요구사항 확인
- 3) 분석모델 확인

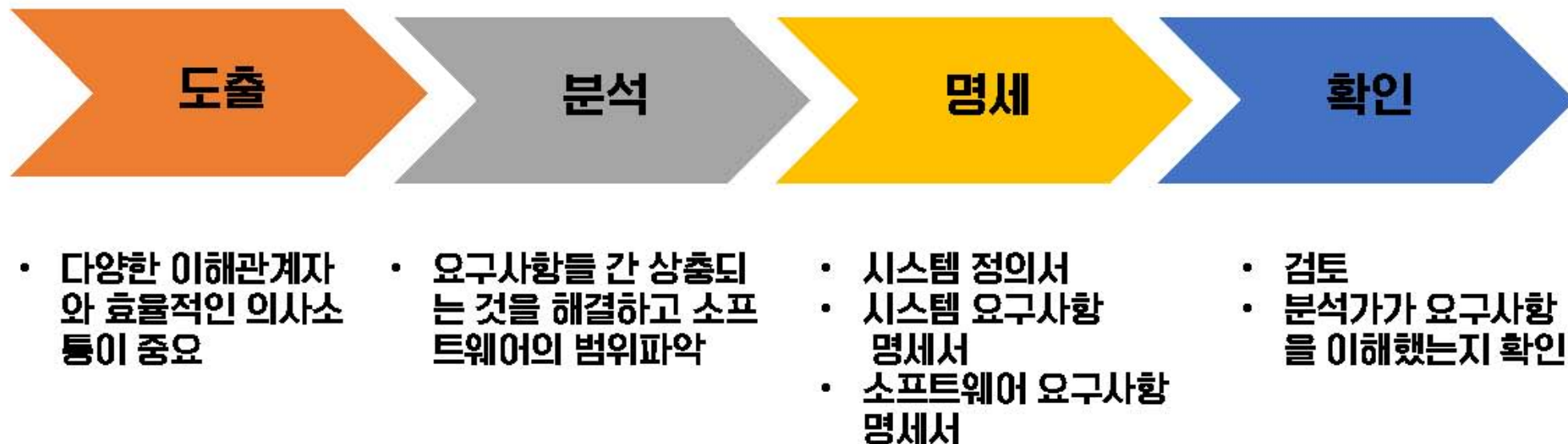
2 화면 설계

3 애플리케이션 설계

4 인터페이스 설계

소프트웨어 설계

요구사항 개발 프로세스



〈요구사항 분석 기법〉

1. 요구사항 분류(Requirement Classification)

- 요구사항이 기능인지 비기능(시스템의 속성들과 제약사항들을 정의하는 요구사항)인지
- 요구사항이 하나 이상의 고수준 요구사항으로부터 유도된 것인지 또는 이해관계자나 다른 원천(Source)으로부터 직접 발생한 것인지
- 요구사항이 제품에 관한 것인지 프로세스에 관한 것인지
- 우선순위가 더 높은 것인지 여부
- 요구사항의 범위(요구사항이 소프트웨어에 미치는 영향의 범위)
- 요구사항이 소프트웨어 생명 주기 동안에 변경이 발생하는지 여부

2. 개념 모델링(Conceptual Modeling)

- 실세계 문제에 대한 모델링이 소프트웨어 요구사항 분석의 핵심이다.
- 모델은 문제가 발생하는 상황에 대한 이해를 증진시키고 해결책을 설명한다.
- 개념 모델은 문제 도메인의 엔티티(entity)들과 그들의 관계 및 종속성을 반영한다.
- 대부분의 모델링 표기법은 UML(Unified Modeling Language)을 사용한다.
- 사용 시나리오를 나타내기 위하여 유스케이스 다이어그램이 많이 사용되고 있다.

3. 요구사항 할당(Requirement Allocation)

- (1) 요구사항을 만족시키기 위한 아키텍처 구성 요소를 식별하는 것.
- (2) 다른 구성 요소와 어떻게 상호 작용하는지 분석을 통하여 추가적인 요구사항을 발견할 수 있다.

4. 요구사항 협상(Requirement Negotiation)

- (1) 두 명의 이해관계자가 서로 상충되는 내용을 요구하거나, 요구사항과 리소스, 기능과 비기능 요구사항들이 서로 상충되는 경우, 어느 한 쪽을 지지하기보다는 적절한 트레이드 오프 지점에서 합의가 중요하다.
- (2) 요구사항에 우선순위를 부여하는 것은 중요한 요구사항을 필터링할 수 있으며, 요구사항들 간 상충되는 문제를 해결하는 데 사용될 수 있다.

5. 정형 분석(Formal Analysis)

- (1) 형식적으로 정의된 시맨틱(Semantics)을 지닌 언어로 요구사항을 표현한다.
- (2) 정확하고 명확하게 표현하여 오해를 최소화시킬 수 있다.
- (3) 정형 분석(Formal Analysis)은 요구사항 분석의 마지막 단계에서 이루어진다.

〈UML(Unified Modeling Language) : 표준화된 범용 모델링 언어〉

- 객체지향 설계를 위한 표준언어
- 시스템을 시각적으로 모델링하기 위한 모델링 언어
- 시스템 개발 과정의 광범위한 분야에 활용 가능

< UML의 구성요소 >

사물(Things) - 기본요소

- 시스템의 구조, 시스템의 행위, 개념의 그룹화, 부가적인 개념설명

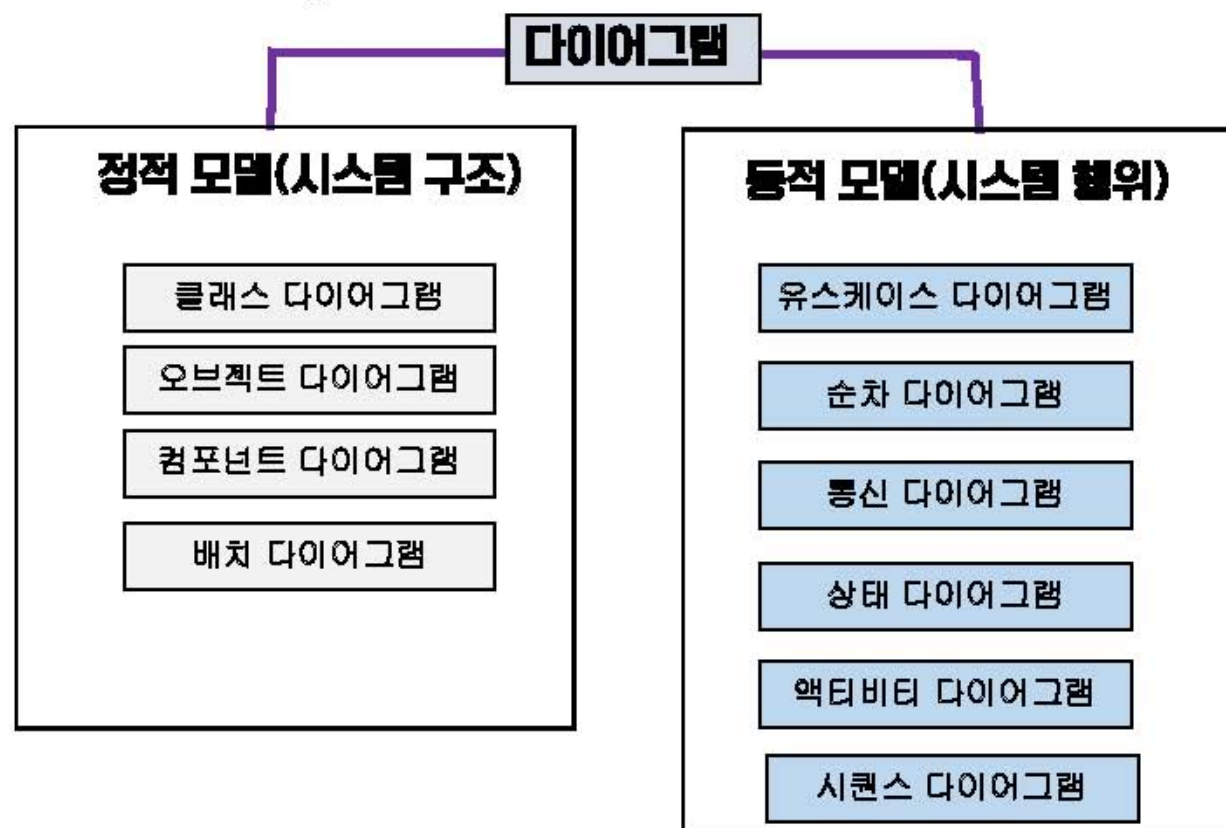
관계(Relationship) - 사물관의 관계

- 의존관계, 연관관계, 일반화관계, 실체화관계

다이아그램(Diagram) - 사물과 관계를 도형으로 표현

- 구조 다이어그램(클래스, 오브젝트, 컴포넌트, 배치)
- 행위다이어그램(유스케이스, 순차, 통신, 상태, 활동)

- UML에서 활용되는 다이어그램(diagrams)



소프트웨어 설계


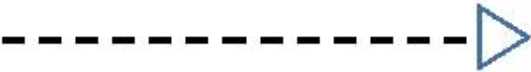


<클래스 다이어그램 예시 - 공공도서관 도서대출>

1. 도서를 대출하기 위해서는 회원가입을 해야한다.
2. 회원증이 발급되면 도서를 대출할 수 있다.
3. 회원은 한 번에 최대 10권의 도서를 1주일동안 대출 받을 수 있다.
4. 반납창구에 제출하여 반납한다.



소프트웨어 설계

- Stereo Type (스테레오 타입)
 - UML에서 제공하는 기본 요소 외에 추가적인 확장요소를 나타내는 것
 - 길러멧(guillemet, « ») 사이에 적는다
 - «interface», «utility», «abstract», «enumeration»

일반화	
실제화	
의존	
연관	

〈유스케이스 다이어그램〉

- 시스템을 사용하는 목적을 사용자 관점에서 기술한 다이어그램
- 목적 달성을 위한 사용자와 시스템 사이의 상호작용을 알수 있다.
- 시스템이 제공하는 기능과 서비스등을 정의하고 시스템의 범위를 결정한다.

<유스케이스 다이어그램 - 시스템>

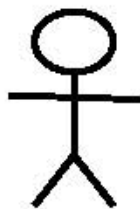
- 만들려는 어플리케이션
- 사각형을 그리고 시스템 명칭을 쓴다.



게시판

<유스케이스 다이어그램 - 액터>

- 시스템의 외부에서 시스템과 상호작용을 하는 사람 또는 다른 시스템
- 사람 모양으로 표현하고, 액터명을 표시한다.



사용자

<유스케이스 다이어그램 - 유스케이스>

- 시스템이 액터에게 제공하는 기능
- 타원으로 표시하고 그 안쪽에 유스케이스명을 기술
- “~한다” 형태의 동사로 표현
- 각 유스케이스가 개발될 기능 하나와 연결됨



금을 등록한다

〈유스케이스 다이어그램 - 관계〉

- 액터와 유스케이스 사이의 관계

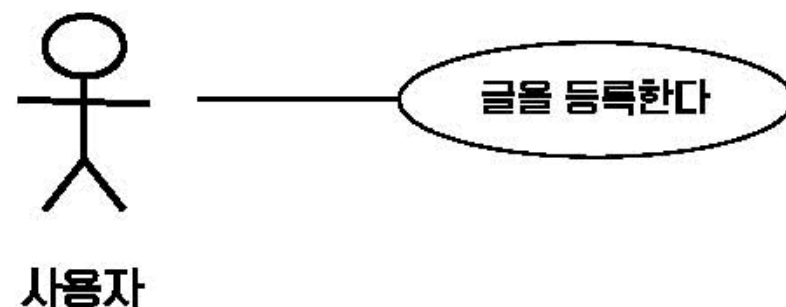
- ① 연관 관계
- ② 의존 관계(포함 관계, 확장 관계)
- ③ 일반화 관계

소프트웨어 설계

<유스케이스 다이어그램 - 관계>

<연관 관계>

- 유스케이스와 액터간의 상호작용을 표현
- 실선으로 연결



<포함 관계>

- 하나의 유스케이스가 다른 유스케이스의 실행을 전제로 할 때
- '포함하는 유스케이스' 에서 '포함되는 유스케이스' 방향으로 화살표를 점선으로 연결하고 <<include>>라고 표기



소프트웨어 설계

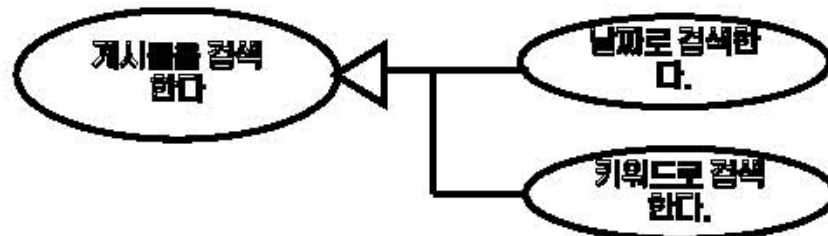
<확장 관계>

- 특정 조건에 따라 확장 기능 유스케이스를 수행하기도 하는 경우
- '확장 기능 유스케이스' 에서 '확장 대상 유스케이스' 방향으로 화살표를 점선으로 연결하고 <<extend>>라고 표기

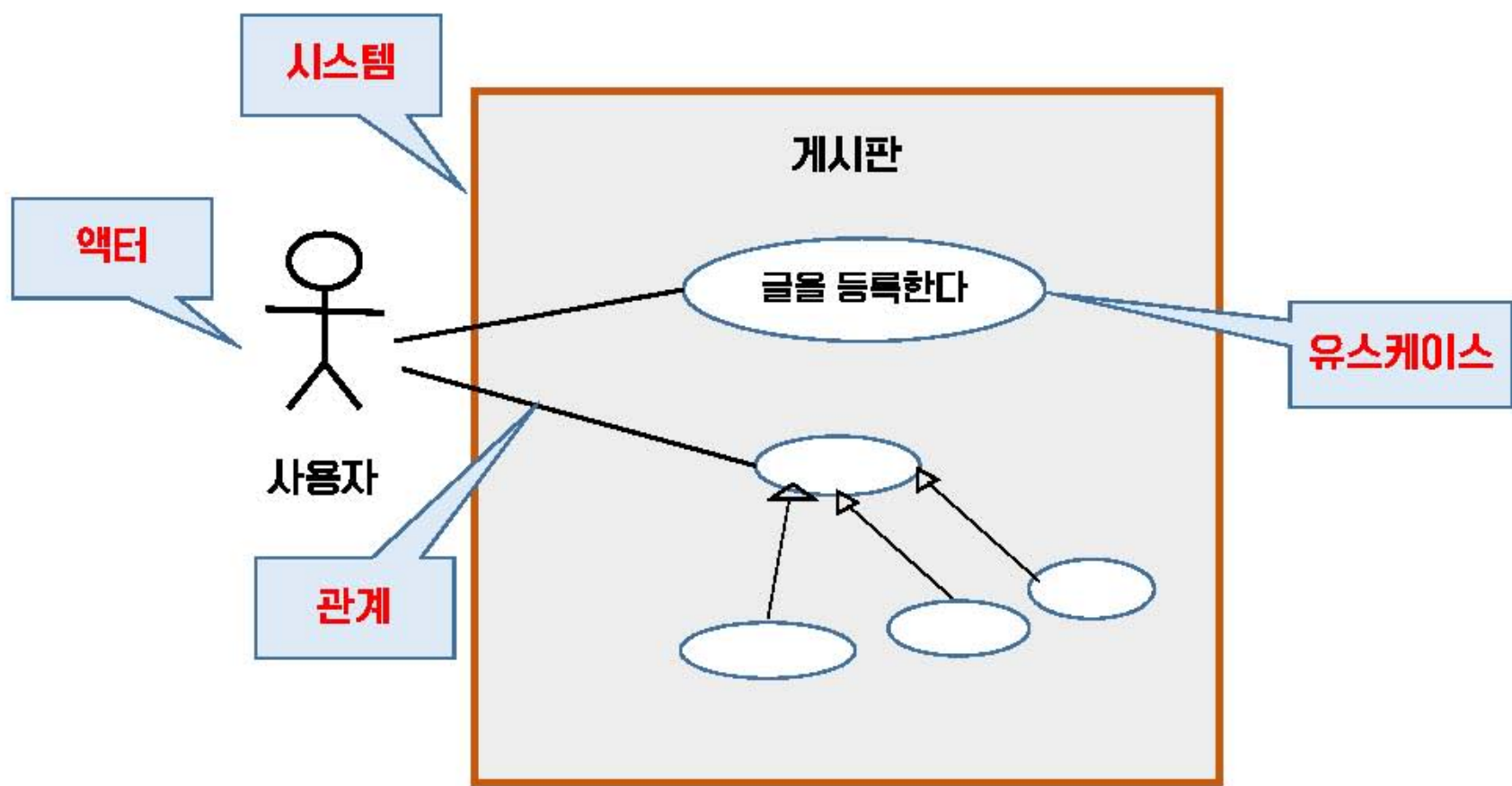


<일반화 관계>

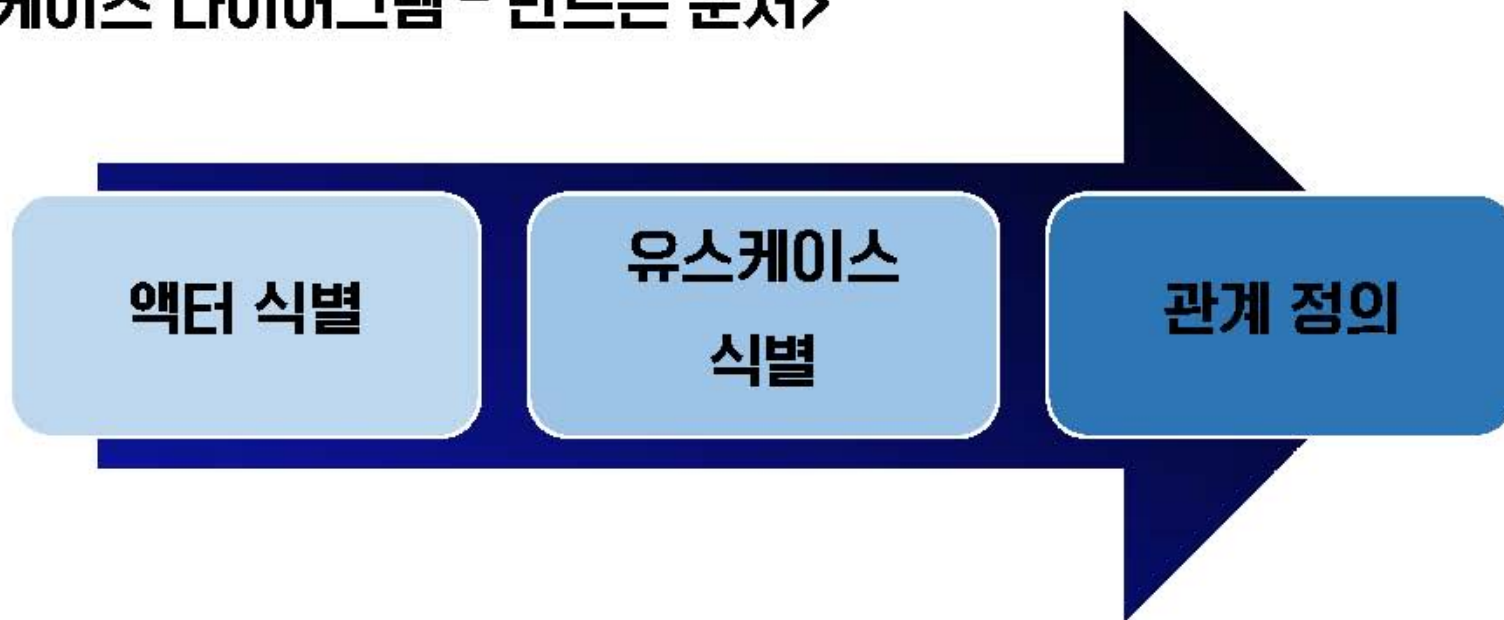
- 유사한 유스케이스들 또는 액터들을 모아 추상화한 유스케이스 액터와 연결시켜 그룹핑
- '구체적인 유스케이스' 에서 '추상적인 유스케이스' 방향으로 끝부분이 삼각형의 테두리인 화살표를 실선으로 연결



소프트웨어 설계



<유스케이스 다이어그램 - 만드는 순서>



<유스케이스 다이어그램 - 예제>

[상품리뷰 게시판 만들기]

- A사가 B고객에게 받은 내용

=> 글을 등록, 수정, 삭제할 수 있는 상품리뷰 게시판을 개발할 것.

- 상품리뷰 게시판의 모든 기능은 로그인 후에만 사용할 수 있다.
- 리뷰를 조회하여 읽을 수 있다.
- 리뷰를 등록할 때에는 사진을 첨부할 수 있다.
- 리뷰를 수정, 삭제할 수 있다.
- 등록된 리뷰는 키워드 또는 날짜별로 검색할 수 있다.

소프트웨어 설계

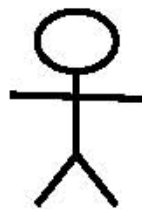
1. 시스템 찾기
: 만들고자 하는 시스템은 “상품리뷰게시판” 이다.



상품리뷰 게시판

2. 액터 찾기

: “상품리뷰 게시판”에 글을 등록하고 조회하는 ‘사용자’ 를 찾음

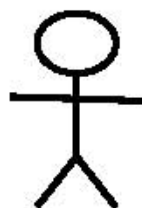


사용자

상품리뷰 게시판

3. 유스케이스 찾기

: '사용자'는 "상품리뷰 게시판"에 리뷰를 조회, 검색, 등록, 수정, 삭제할 수 있다.



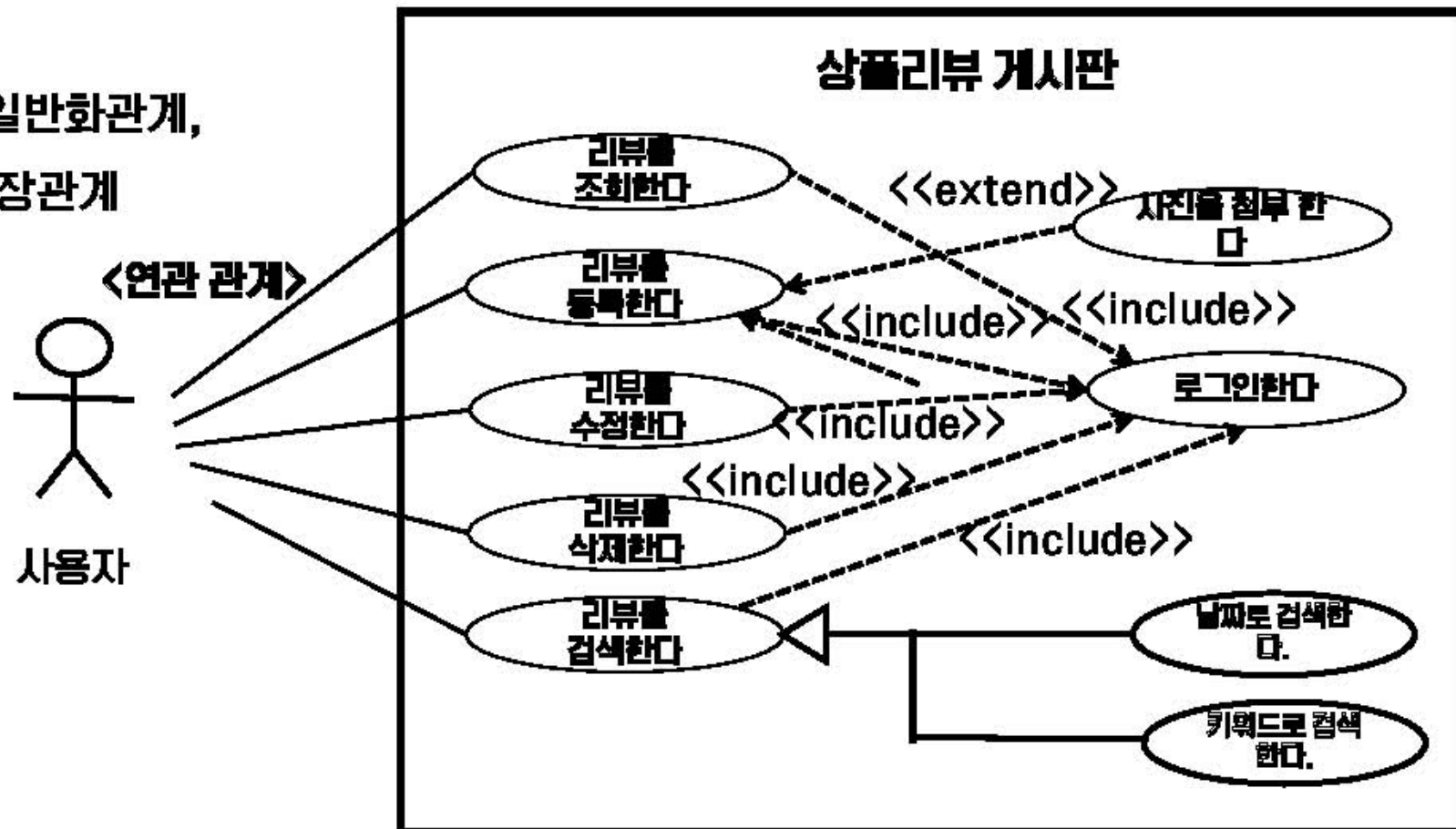
사용자



소프트웨어 설계

4. 관계 찾기

: 연관관계, 일반화관계,
포함관계, 확장관계



〈요구사항 검증하기〉

1. 요구사항 목록에 업무 기능에 대한 요구사항이 모두 반영되었는지 확인한다.
2. 요구사항 정의서의 작성 여부를 확인한다.
3. 비기능적 요구사항이 명확하게 도출되었는지 확인한다.
4. 타 시스템과 연계 및 인터페이스에 대한 요구사항이 적절히 도출되었는지 확인한다.

<요구사항의 시스템화 타당성 분석>

성능 및 용량산정
적정성

시스템 간
상호 운용성

IT 시장 성숙도
및
트렌드 부합성

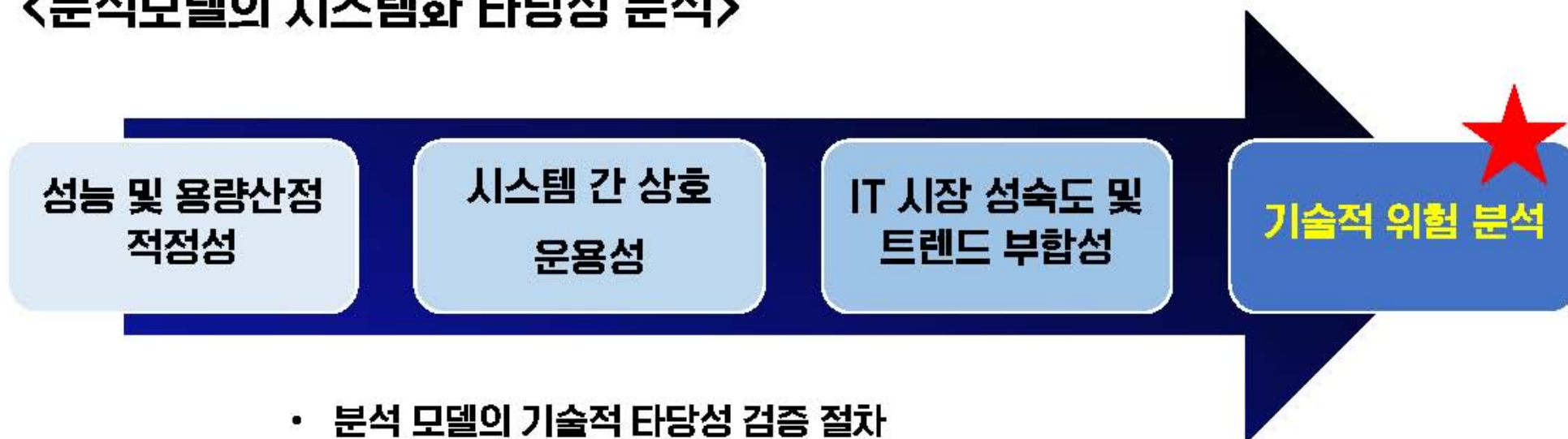
기술적 위험
분석

요구사항 기술적 타당성 검토 4단계

<분석모델 검증하기>

1. 분석모델까지 요구사항 추적표를 작성하고 검토 의견 컬럼을 추가한다.
2. 작성된 요구사항 추적표에 검토 의견을 작성한다.
3. 요구사항 추적표에서 요구사항에 대한 검토 의견을 정제한다.

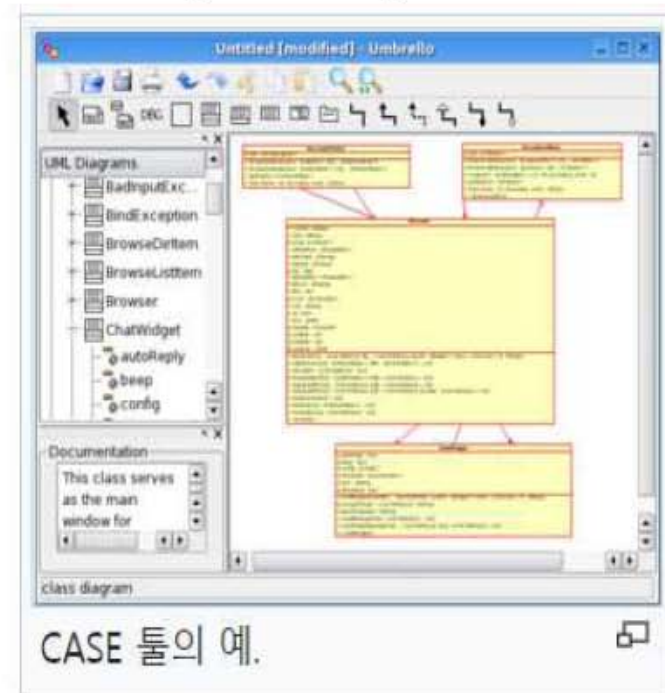
〈분석모델의 시스템화 타당성 분석〉



소프트웨어 설계

<분석 자동화 도구 – CASE(Computer Aided Software Engineering)>

- 컴퓨터 지원 시스템 공학
- **시스템** 개발 방법론들의 자동화를 지원하는 소프트웨어 도구를 제공해 **개발자**의 반복적인 작업량을 줄이도록 한다.
- CASE 도구들은 문서의 생성과 개발 팀 간의 협업을 돕는다. 작업된 내용을 검토하고 수정하기 위해 서로 다른 사람의 파일에 접근하도록 허용해 팀 구성원들은 그들의 작업을 손쉽게 공유할 수 있다.
- CASE 도구들은 강력한 **그래픽 기능**이 있으며 PC 기반에서 운영된다. CASE 도구들은 차트와 **다이아그램**을 자동으로 생성하는 그래픽 기능, 화면과 리포트 생성기, 데이터사전, 분석과 검사 도구, 코드 생성기, 문서 생성기 등을 제공한다.



- 위키백과

< CASE의 장점>

- 구조적인 것들을 그대로 활용할 수 있다.
- 요구 정보를 추출하고 분석하는 것을 도와준다.
- 원형(Prototype)이나 프로그램의 개발 및 유지가 용이하다.
- 개발자들이 반복적인 업무에서 벗어나 창의적 업무에서 몰두하게 해 준다.
- 소프트웨어의 점진적 개발이 가능하다.
- 소프트웨어의 재활용성을 재고시켜 준다.
- 모든 것들이 그림으로 표현되어 있기 때문에 개발자들 간에 정보시스템의 공유가 쉽다.

- 위키백과

문제풀이

1. CASE(Computer Aided Software Engineering)의 주요 기능으로 옳지 않은 것은?

- ① S/W 라이프 사이클 전 단계의 연결
- ② 그래픽 지원
- ③ 다양한 소프트웨어 개발 모형 지원
- ④ 언어 번역

2. UML의 기본 구성요소가 아닌 것은?

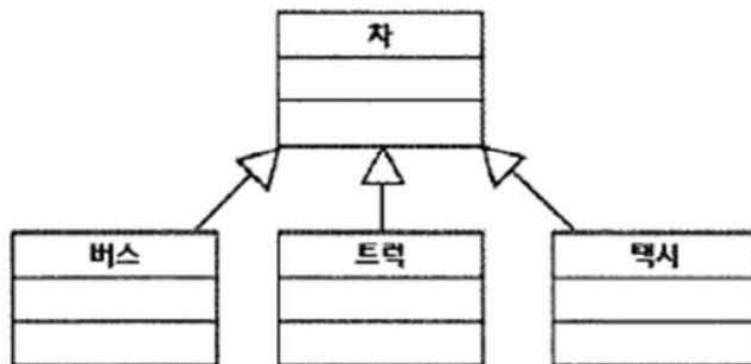
- ① Things ② Terminal
- ③ Relationship ④ Diagram

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 설계)

문제풀이

3. 아래의 UML 모델에서 '차' 클래스와 각 클래스의 관계로 옳은 것은?

- ① 추상화 관계 ② 의존 관계 ③ 일반화 관계 ④ 그룹 관계



(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 설계)

문제풀이

4. UML에서 활용되는 다이어그램 중, 시스템의 동작을 표현하는 행위(Behavioral) 다이어그램에 해당하지 않는 것은?
- ① 유스케이스 다이어그램(Use Case Diagram) ② 시퀀스 다이어그램(Sequence Diagram)
③ 활동 다이어그램(Activity Diagram) ④ 배치 다이어그램(Deployment Diagram)
5. CASE(Computer-Aided Software Engineering)도구에 대한 설명으로 거리가 먼 것은?
- ① 소프트웨어 개발 과정의 일부 또는 전체를 자동화하기 위한 도구이다.
② 표준화된 개발 환경 구축 및 문서 자동화 기능을 제공한다.
③ 작업 과정 및 데이터 공유를 통해 작업자간 커뮤니케이션을 증대한다.
④ 2000년대 이후 소개되었으며, 객체지향 시스템에 한해 효과적으로 활용된다.

(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 설계)

문제풀이

6. CASE가 갖고 있는 주요 기능이 아닌 것은?

- ① 그래픽 지원 ② 소프트웨어 생명주기 전 단계의 연결 ③ 언어 번역 ④ 다양한 소프트웨어 개발 모형 지원

7. UML 확장 모델에서 스테레오 타입 객체를 표현할 때 사용 하는 기호로 맞는 것은?

- ① << >> ② (()) ③ {{ }} ④ [[]]

8. UML 모델에서 사용하는 Structural Diagram에 속하지 않은 것은?

- ① Class Diagram
② Object Diagram
③ Component Diagram
④ Activity Diagram

(2020년 1,2회 정보처리기사 필기 기출문제 소프트웨어 설계)

문제풀이

9. 소프트웨어 개발 과정에서 사용되는 요구분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용의 소프트웨어 도구를 사용하여 자동화하는 것은?

- ① CAD(Computer Aided Design)
- ② CAI(Computer Aided Instruction)
- ③ CAT(Computer Aided Testing)
- ④ CASE(Computer Aided Software Engineering)

(2019년 3회 정보처리기사 필기 기출문제 소프트웨어공학)

문제풀이

10. 소프트웨어 생명주기 모형에서 프로토타입 모형의 장점이 아닌 것은?

- ① 단기간 제작 목적으로 인하여 비효율적인 언어나 알고리즘을 사용할 수 있다.
- ② 개발과정에서 사용자의 요구를 충분히 반영한다.
- ③ 최종결과물이 만들어지기 전에 의뢰자가 최종결과물의 일부 혹은 모형을 볼 수 있다.
- ④ 의뢰자나 개발자 모두에게 공동의 참조 모델을 제공한다.

11. CASE가 제공하는 기능으로 거리가 먼 것은?

- ① 개발을 신속하게 할 수 있다.
- ② 개발 방법론을 생성할 수 있다.
- ③ 오류 수정이 쉬워 S/W 품질이 향상된다.
- ④ S/W개발 단계의 표준화를 기할 수 있다.

(2019년 1회 정보처리기사 필기 기출문제 소프트웨어공학)