

소프트웨어공학 활용



데이터 모델 및 객체 모델



한국기술교육대학교
온라인평생교육원

학습내용

- 데이터 모델
- 객체 모델

학습목표

- 데이터 모델의 개념을 이해하고 설명할 수 있다.
- 객체 모델의 개념을 이해하고 설명할 수 있다.

데이터 모델



1 데이터 모델 개념

1 시맨틱 데이터 모델(Semantic Data Models)

1 개념



시맨틱 데이터 모델이란?

- 시스템에 의해 처리되는 데이터의 논리적인 구조를 묘사하기 위해 사용되는 모델
- 데이터의 관계, 접근과 그 흐름에 필요한 **처리 과정에 관한 추상화된 모형**
- 데이터 모델은 **데이터 구조를 결정함**
- 데이터 모델은 그래픽 형태로 설명되는 데이터 모델링 개념에 정의

2 데이터 모델링

주어진 개념으로부터 **논리적인 데이터 모델을 구성**하는 작업

물리적인 데이터베이스 모델로 변환하여 고객의 요구에 따라 **특정 정보시스템의 데이터베이스**로 반영

데이터 모델링의 주요 목적

- ➡ 업무 정보를 구성하는데 기초가 되는 정보를 일정한 표기법으로 표현, 정보시스템 구축 대상이 되는 업무 내용을 정확하게 분석
- ➡ 분석된 모델을 가지고 실제 데이터베이스를 생성하여 개발 및 데이터관리에 사용

데이터 모델



1 데이터 모델 개념

1 시맨틱 데이터 모델(Semantic Data Models)

3 데이터 모델에 표시할 요소

구조
(S : Structure)

연산
(O : Operation)

제약조건
(C : Constraint)

데이터의 정적 성질로서 데이터베이스에서 표현될 대상으로서의
엔티티 타입과 이들 간의 관계를 명세

구조
(S : Structure)

연산
(O : Operation)

제약조건
(C : Constraint)

데이터의 동적 성질로서 연산은 엔티티 인스턴스에 적용 가능한
연산에 대한 명세

데이터 모델



1 데이터 모델 개념

1 시맨틱 데이터 모델(Semantic Data Models)

3 데이터 모델에 표시할 요소



데이터베이스에 저장하는 데이터가 무결하게 하기 위해 사용

4 데이터 모델의 구성요소

개체(Entity)

속성(Attribute)

관계(Relationship)

- 유형, 무형의 정보로서 데이터베이스에 표현하려는 것
- 독립적으로 존재하거나 그 자체로서 구별이 가능

데이터 모델



1 데이터 모델 개념

1 시맨틱 데이터 모델(Semantic Data Models)

4 데이터 모델의 구성요소

개체(Entity)

속성(Attribute)

관계(Relationship)

- 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드
- 개체를 구성하는 항목

개체(Entity)

속성(Attribute)

관계(Relationship)

- 개체 간의 관계 또는 속성 간의 관계
- 형태 : 일대일(1:1), 일대다(1:n), 다대다(n:n)

2 데이터 모델 관점

개념적 데이터
모델

논리적 데이터
모델

물리적 데이터
모델

데이터 모델



1 데이터 모델 개념

2 데이터 모델 관점

1 개념적 데이터 모델

현실 세계에 존재하는 데이터를 의미 있는 엔티티, 공통된 속성의 엔티티와 엔티티들 사이의 관계를 정의하는 **추상화 과정**

현실 세계를 추상화하여 개념적으로 표현하기 때문에
쉽게 이해할 수 있고 **의사소통을 원활하게** 해주는 과정



객체 관계 모델
(E-R : Entity-Relationship Model)

기술된 엔티티 타입과 이 엔티티 타입 간의 관계를 이용하여 현실 세계를 표현하는 가장 대표적인 개념적 데이터 모델

데이터 모델



1 데이터 모델 개념

2 데이터 모델 관점

2 논리적 데이터 모델

1

개념적 데이터 모델은 DBMS(Database Management System)가 직접 이해할 수 없음

2

컴퓨터가 이해할 수 있도록 **논리적 데이터 모델로 변환**해야 함

3

논리적 데이터 모델은 **데이터 모델링**이라고도 함

4

관계 데이터 모델, 계층 데이터 모델, 네트워크 데이터 모델 등이 있음

3 물리적 데이터 모델



물리적 데이터 모델

- 논리 데이터 모델을 DBMS의 특성 및 성능을 고려하여 구체화시킨 모델
- 논리적 데이터 모델을 각각의 관계형 데이터베이스 관리 시스템 특성, 기능, 성능 등을 고려하여 **데이터베이스의 물리적인 구조(Schema)와 제약조건을 작성**해 가는 과정

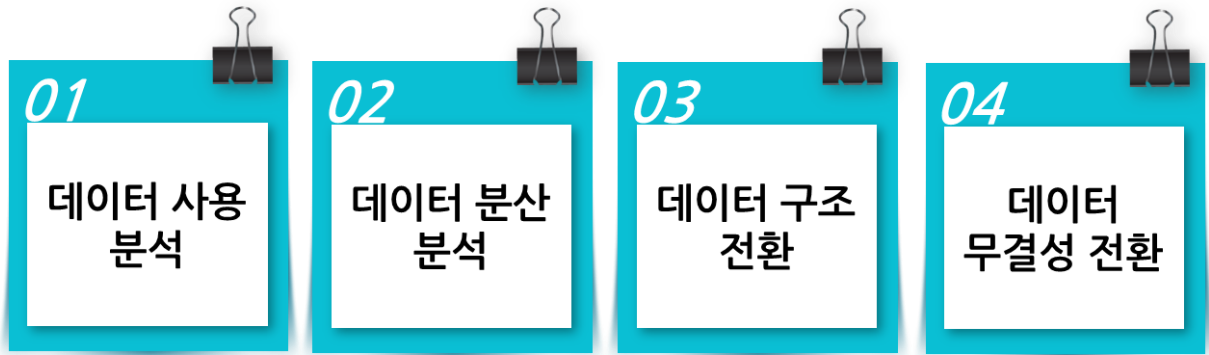
데이터 모델

1 데이터 모델 개념

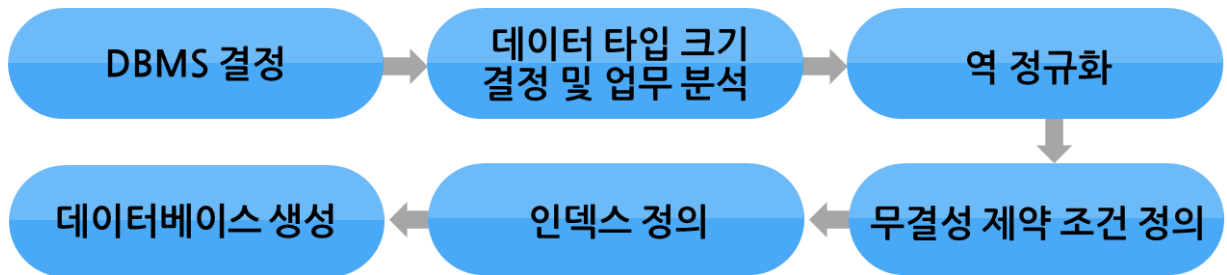
2 데이터 모델 관점

3 물리적 데이터 모델

- 물리적 데이터 모델링 고려사항



- 모델링 과정



데이터 모델



2 데이터 사전

1 데이터 사전 개념

1 개념

- 1 시스템모델에서 사용되는 모든 이름에 대한 리스트이며, 엔티티, 관계 및 속성에 대한 자세한 설명이 포함됨
- 2 이름을 관리할 수 있기 때문에 중복을 피할 수 있음
- 3 분석, 설계 및 구현을 연관 지을 수 있는 통합 정보의 저장소 기능
- 4 대부분의 CASE 워크벤치가 데이터 사전을 지원함
- 5 데이터베이스의 스키마 및 시스템 카탈로그를 저장하는 곳

2 데이터 사전 표기법

정의

- 표기 : =
- ex) 주민등록번호 = XXXXXXX-XXXXXXX

구성

- 표기 : +
- ex) 개인정보 = 주민등록번호 + 전화번호 + 주소

택일

- 표기 : []
- ex) 배송 주소 = [최근배송지 | 집 | 회사]

데이터 모델



2 데이터 사전

1 데이터 사전 개념

2 데이터 사전 표기법

반복

• 표기: {}

ex) 영수증 = 날짜 + {상품명 + 수량 + 가격} + 총액

생략가능

• 표기: ()

ex) 영수증 = (회원 고객명) + 날짜 + (상품 구입내역) + 총액

설명

• 표기 : * *

ex) 영수증 = *XX 고객의 구매 영수증*

객체 모델



1 객체 모델 개념

1 객체 모델(Object Model)

1 개념

1 시스템을 객체의 클래스 용어로 기술(Class = Attribute + Method)

2 객체 클래스는 각각의 객체가 제공하는 공통의 속성들과 동작들을 가진 객체들의 집합에 대한 추상화

3 상호작용적(Interactive) 시스템 개발에 널리 사용

4 현실 세계를 묘사하기 적합한 모델

2 객체특성 적용 모델 종류

상속 모델
(Inheritance
Models)

집합연관 모델
(Aggregation
Models)

상호작용 모델
(Interaction
Models)

객체 모델



1 객체 모델 개념

1 객체 모델(Object Model)

3 객체 모델링 정의

1

시스템의 정적인 구조를 표현

2

객체 지향 분석에서 가장 중요시되며 선행되어야 할 모델링

3

시스템의 기능에 초점을 맞추기보다는 시스템을 구성하고 있는 객체를 강조

4

객체와 클래스로 구성

2 객체 모델링

1 클래스 후보 요소

구조

디바이스

운용 절차

절차

외부시스템

역할

장소 등

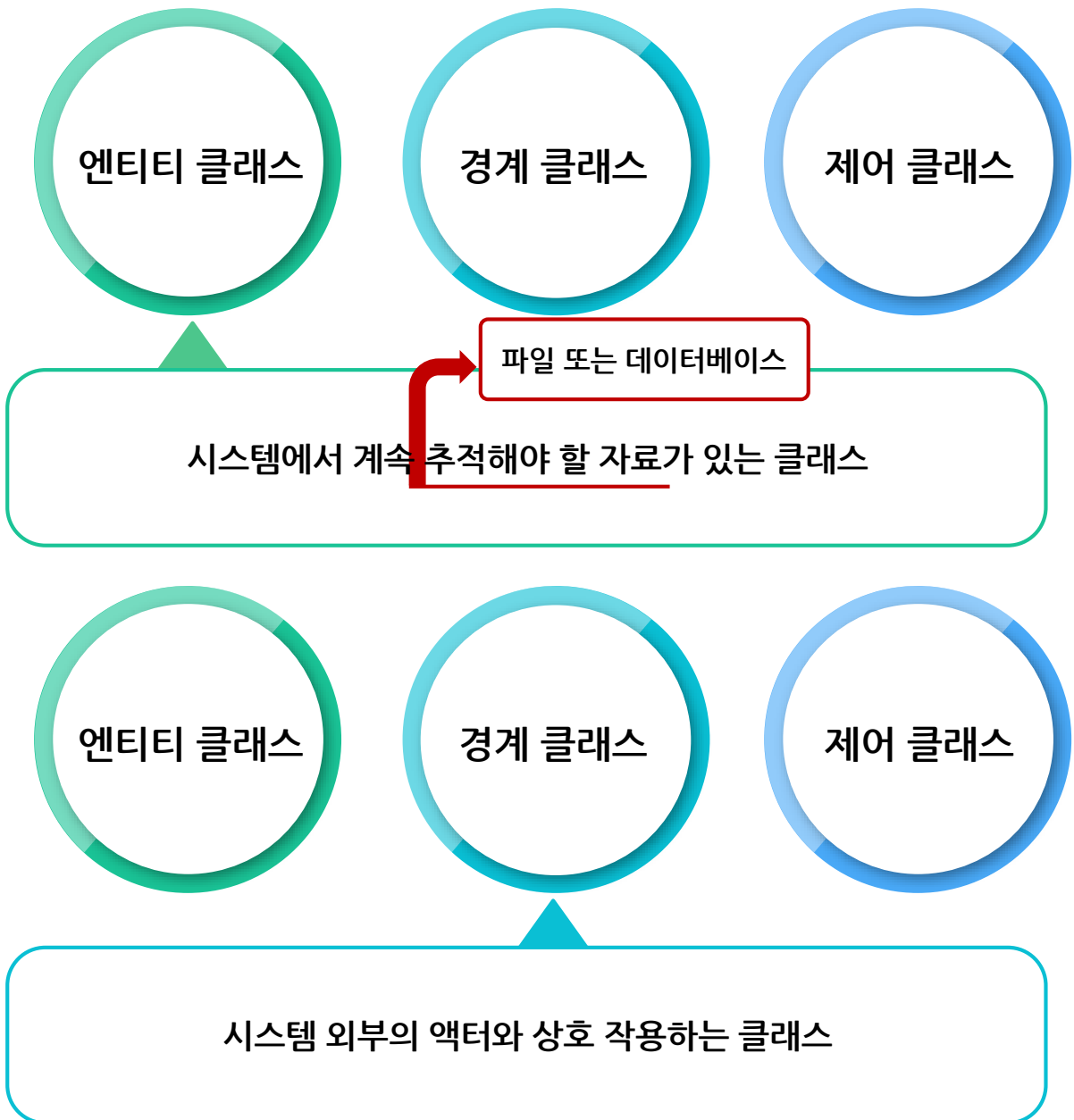
객체 모델



1 객체 모델 개념

2 객체 모델링

2 클래스 후보 유형



객체 모델



1 객체 모델 개념

2 객체 모델링

2 클래스 후보 유형

엔티티 클래스

경계 클래스

제어 클래스

경계 클래스와 엔티티 클래스 사이에 중간 역할 클래스

객체 모델



1 객체 모델 개념

2 객체 모델링

3 부치 방법론

1 설계 중심의 방법론으로 시스템을 몇 개의 뷰(View)로 분석할 수 있다고 봄

2 뷰를 모델 다이어그램으로 나타냄

3 거시적 개발 프로세스와 미시적 개발 프로세스 모두 포함

4 단계적 접근과 자동화 도구를 지원

5 객체 지향 방법론에 대한 광범위한 이론적 배경을 제시

4 야콥슨의 OOSE(Object Oriented Software Engineering)

OOSE는 유스케이스를 강조한 방법론

- 외부 행위자와 상호 작용하는 시스템의 요구사항을 정의
- 개발, 테스트, 검증 단계에서 사용됨



OOSE 방법론이 복잡하여 초보자에게 어려움이 있지만, 큰 규모의 시스템 개발에 효율적

객체 모델



1 객체 모델 개념

2 객체 모델링

5 럼바의 OMT

하나의 시스템을 기술하기 위해 객체 모델(Object Model), 동적 모델(Dynamic Model), 기능 모델(Functional Model) 3가지를 사용

객체모델

시스템에서 필요한 모델을 찾아내고, 객체의 속성과 객체들의 사이 관계 규명

동적모델

객체 모델에서 나타난 객체들의 행위와 상태를 포함하는 생명주기를 나타냄

기능 모델

각 객체의 변화로 인해 다른 상태로 전이가 되었을 때 수행되는 동작들을 기술함

객체 모델

2 객체 지향

1 객체 지향 개념

1 특징

1

구조적 기법의 문제점으로 인한 소프트웨어 위기의 해결책으로 제시

2

소프트웨어의 재사용 및 확장성을 높여 고품질의 빠른 개발, 유지보수가 용이

3

복잡한 구조를 단계적(계층적)으로 표현

2 구성요소

객체(Object)

클래스(Class)

메시지(Message)

- 객체 = 데이터 + 데이터 처리 함수, 두 가지를 하나로 모은 모듈

객체가 처리하는 기능, 객체의 상태를 참조하거나 변경(메소드, 서비스, 동작 연산)

객체가 가지고 있는 정보, 속성, 상태, 분류

객체 모델



2 객체 지향

1 객체 지향 개념

2 구성요소

객체(Object)

클래스(Class)

메시지(Message)

- 공통된 속성과 연산을 갖는 객체의 집합
- 객체들이 갖는 속성과 연산을 정의
- 인스턴스(Instance) : 클래스에 속한 객체
- 인스턴스화 : 클래스로 세운 객체를 생성
요구사항 명세서가 클래스로 새로운 객체 생성
- 최상위 클래스는 유일하게 상위 클래스를 갖지 않음
- 슈퍼 클래스 : 특정 클래스의 상위(부모)클래스
- 서브 클래스 : 특정 클래스의 하위(자식) 클래스

객체(Object)

클래스(Class)

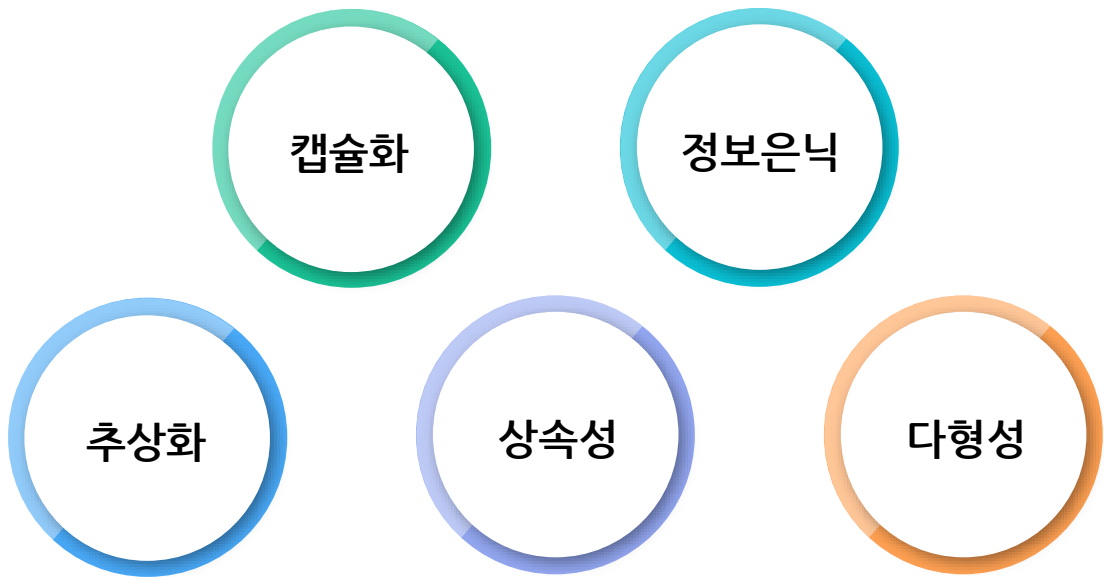
메시지(Message)

- 객체 간의 상호작용할 때 사용하는 수단
- 외부에서 메소드를 일으키는 요구사항
- 메시지를 수신 받은 객체는 요구된 메소드를 수행하여 결과를 반환

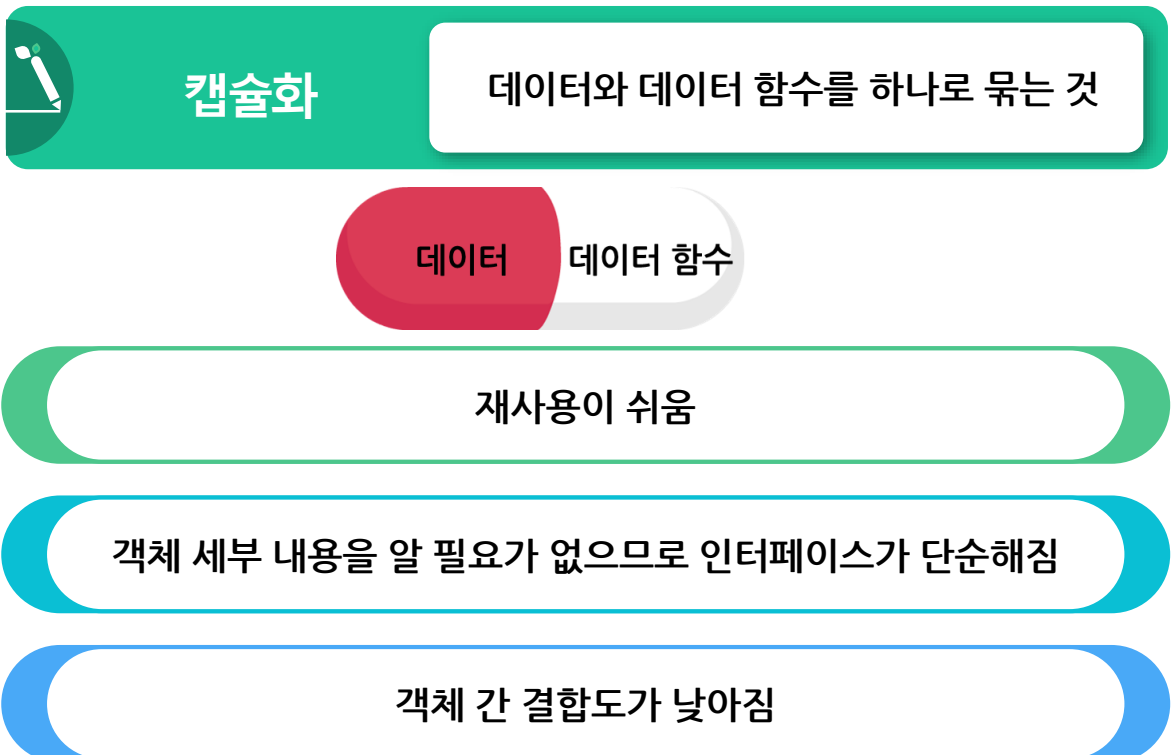
객체 모델

2 객체 지향

2 객체 지향 기법의 원칙



1 캡슐화(Encapsulation)



객체 모델

2 객체 지향

2 객체 지향 기법의 원칙

2 캡슐화(Encapsulation)



정보 은닉

- 다른 객체에 자신의 정보를 숨김
- 연산을 통하여 접근 허용

객체의 수정이 다른 객체에 주는 영향이 적음

유지보수 및 소프트웨어 확장 시 오류 최소화

3 추상화(Abstraction)

01

불필요한 부분 생략

02

객체의 중요한 속성에
중점을 두어 모델화

객체 모델



2 객체 지향

2 객체 지향 기법의 원칙

4 상속성(Inheritance)



상속성

상위 클래스의 모든 속성과 연산을 하위 클래스가 물려받음

하위 클래스는 상위 클래스의 속성과 연산을 정의하지 않고서 사용할 수 있음

하위 클래스는 상위 클래스의 속성 및 연산 외에 새로운 속성과 연산 추가 가능

5 다형성(Polymorphism)



다형성

메시지에 의해 객체가 수행될 때, 각 객체가 가지고 있는 고유한 방법으로 응답

객체는 동일한 메소드를 사용하여 같은 의미를 갖는 결과를 내놓음

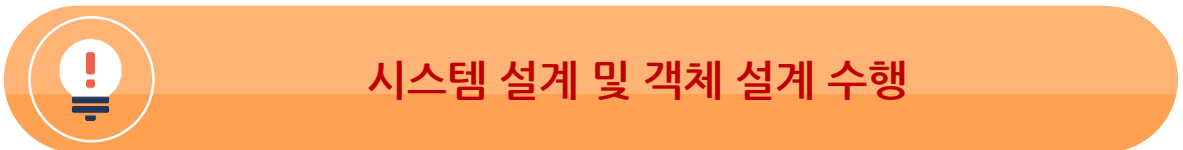
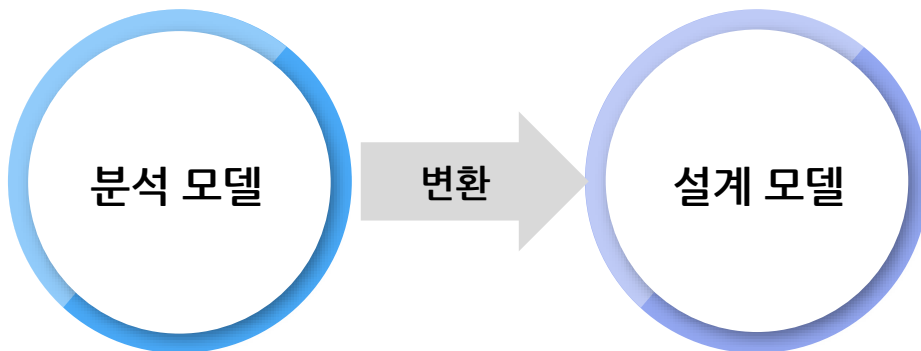
객체 모델



2 객체 지향

3 객체 지향 설계(Object Oriented Design)

1 개념



▪ 일반적인 설계 순서



학습정리

1. 데이터 모델



- 데이터 모델은 시스템에 의해 처리되는 데이터의 논리적인 구조를 묘사하기 위해 사용되는 모델임
- 데이터 구조를 결정하며, 데이터의 관계, 접근과 그 흐름에 필요한 처리 과정에 관한 추상화된 모형임
- 데이터 모델링의 주요 목적은 업무 정보를 구성하는데 기초가 되는 정보를 일정한 표기법으로 표현하며 정보시스템 구축 대상이 되는 업무 내용을 정확하게 분석함
- 데이터 모델은 개체, 속성, 관계로 구성됨
- 개념적 데이터 모델의 장점 : 현실 세계를 추상화하여 개념적으로 표현하기 때문에 이해가 쉽고 의사소통을 원활하게 해줌
- 컴퓨터가 이해할 수 있도록 논리적 모델로 변환하는 모델을 논리적 데이터 모델이라고 함
- 논리적 데이터 모델은 관계 데이터 모델, 계층 데이터 모델, 네트워크 데이터 모델 등이 있음
- 물리적 데이터 모델은 DBMS의 특성 및 성능을 고려하여 구체화 시킨 모델임
- 데이터 사전은 데이터베이스의 스키마 및 시스템 카탈로그를 저장하는 곳으로 이름으로 관리하므로 중복을 피할 수 있음
- 대부분의 CASE 워크벤치에서 데이터 사전을 지원하기 때문에 널리 쓰임

학습정리

2. 객체 모델



- 객체 모델은 시스템을 객체의 클래스 용어로 나타낸 것임
- 객체 모델링은 객체 지향 분석에서 가장 중요시되며 선행되어야 할 모델링 기법이고, 객체와 클래스로 구성되어 있음
- 클래스의 3가지 후보 유형이 있으며 엔티티 클래스, 경계 클래스, 제어 클래스로 각각의 역할에 따라 구분됨
- 객체 지향은 현실 세계의 복잡한 구조를 단계적으로 표현할 수 있으며 구조적인 기법의 문제점으로 인한 소프트웨어 위기의 해결책으로 제시되었음
- 객체, 클래스, 메시지 3가지의 구성요소로 되어있음
- 객체 지향에서 중요한 원칙 : 캡슐화, 정보은닉, 추상화, 상속성, 다형성
- 객체 지향적 설계 순서 : 문제정의→요구명세화→객체 연산자 정의→객체 인터페이스 결정→객체 구현