

SYSTEM 개발...

(몇 개의 독립적 요소가 결합, 전체가 유기적으로 하나로 동작하는 것을 총칭)

IT : 정보를 처리하는 데 사용되는 프로그램(Application)들 절차들, 데이터 및 기기들의 다양한 집합이나 조합
(학사관리 System, 물류 System, 사내 System....)

System 개발 절차

요구사항 정의

분석

설계

1.모델링

2.프로그래밍

구현

1. 모델링

=> System 구축 시 구현 전 결정하는 모든 활동

- 구축할 System 정의
- 요구사항정의
- 구체적인 계획(분석, 설계)
- 결과물 : 모델
- 표 기 법 : UML

2. 구현

=> System 실제구현

- 소스코드, 컴파일, 디버깅
- 결과물 : 코드를 포함한 System
- 프로그래밍 언어 : Java

→ OOP

- Encapsulation
- Abstraction
- Modularity
- Hierarchy
- Object
- Class
- Polymorphism
- Relationship

1. 객체지향의 기본원리

- 추상화 (Abstraction)
- 캡슐화 (Encapsulation)
- 모듈화 (Modularity)
- 계층화 (Hierarchy)

2. 객체지향의 기본개념

- 객 체 (Object)
- 클래스 (Class)
- 다형성 (Polymorphism)
- 관 계 (RelationShips)

○ 객 체(Object)

: 물리적인 객체 / 개념적인 객체 / 소프트웨어적 객체

객체 = 식별성(identity) + 상 태(state) + 행위(Behavior)

=> 식별성 : 같은 상태를 갖더라도 구분 할 수 있다.

=> 상 태 : 객체가 가질 수 있는 조건,

일반적으로 시간에 따라 변한다 / 행위를 통한 변경

=> 행 위 : 객체가 반응 할 수 있는 Message의 집합

○ 클래스(Class)

: 공통된 특성(속성), 행위, 관계, 의미를 갖는 객체의 모임

=> 객체의 공통점을 찾아서 class로 정의한다.

=> 객체를 주상화해서 표현 / 정의한 것

=> 객체지향의 원리 → 추상화

○ 추 상 화(Abstraction)

=> 객체를 구분되는 핵심적인 특징으로 분류 / 집중

=> 문제영역에 의존적이다.

(교육시스템 → 사 람 ← 판매시스템)

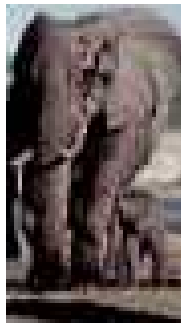
=> 객체지향의 추상화를 이용 System을 modeling할 수 있다.

예) 보더마커를 추상화를 통해 class로 표현하면..

=> 추상화 연습

○ 아래의 객체의 수는 ?

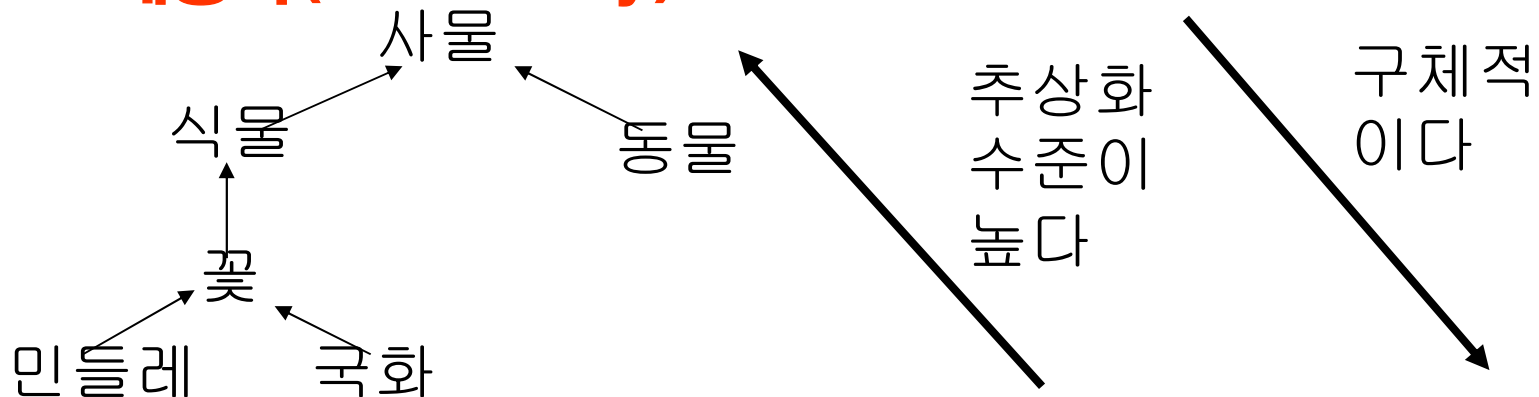
○ 추상화를 통해 class로 표현한다면 class의 수는 ? (문제영역은 없음)



○ 관 계 (RelationShip)

- Generalization(추상화 수준이 높다)
 - 속성과 행위의 공유
 - ~ is a ~ 관계 성립 (하위는 상위를 대신 할 수 있다)
(상위는 하위를 대신 할 수 없다)
- Specialization(구체적이다)
- Realization (추상화 된 것의 구현)
- Association
- dependency

○ 계층화(Hierarchy)



○ 다형성 (Polymorphism)

: 동일한 인터페이스를 사용할 수 있도록 한다.

<< 구현언어 지원 문법 >>

: Method OverLoading / Method OverRiding

: Abstract Method / abstract Class / interface

○ 캡슐화(Encapsulation)

: 공개적인 interface두로 구현을 숨긴다.

→ 구현을 숨기는 절차 은닉

(사용자에게 사용 방법만 open)

→ 객체의 정보를 숨기는 information hiding

(객체의 중요한 정보(상태정보)는 open 하지 않는다.)

객체 지향 설계의 3단계

< 1단계 > 객체 모델링 (Object Modeling)

객체의 **속성, 기능** 을 정리한다.

예) 홍길동 모델링하자.

속성값(attribute)

이름 : 홍길동

직업 : 의 적

선행 행한 횟수 : 횟수

등등...

기능(behavior)

물건을 훔친다.

훔친 물건을 나누어준다.

탐관오리를 혼낸다.

등등.....



< 2단계 > 클래스 정의 모델링 과정에서

속성 은 Field →

전역변수(member variable)

기능 은 method 로 정의한다.

예)

속성값(attribute)

String name = "홍길동"

String job = "의적"

int help = 0;

등등...

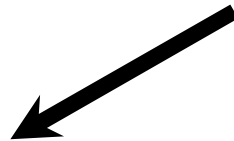
기능(behavior)

public void rent() {}

public void service()

탐관오리를 혼낸다.

등등.....



< 3단계 > 인스턴스의 생성과 사용

main Method 내부에서

new라는 keyword를 사용하여
객체의 **Field, Method**에 접근한다.

예)

Iam iam = new Iam()

iam.name

iam.rent()