

섹션1. 2. 자바의 핵심 - 객체지향 프로그래밍

클래스와 객체1 (1)

객체지향 프로그래밍

- **객체(Object)** : 의사나 행위가 미치는 대상
구체적, 추상적 데이터 단위
- **객체지향 프로그래밍(OOP, Object Oriented Programming)** : 객체를 기반으로 하는 프로그래밍, 코딩 시 객체간의 **협력**(협동) cf. 절차지향프로그래밍
- **클래스** : 객체를 생성하기 위한 코드, 멤버변수, 생성자, 메소드로 구성 ex) 과자틀

클래스 정의 하기

```
(접근 제어자) class 클래스 이름{  
    멤버 변수;  
    메서드;  
}
```

학생 클래스의 예

속성 : 학번, 이름, 학년, 사는 곳 등등...

기능 : 수강신청, 수업듣기, 시험 보기 등등...

cf. 클래스와 메서드 선언 시 모양

클래스 선언 : (접근 제어자) (static) class 클래스 이름

메서드 선언 : (접근 제어자) (static) 리턴타입 함수 이름 매개변수

- **인스턴스** : 클래스 new 키워드 사용 생성한 객체 ex)과자틀로 찍어낸 과자들

cf. 실행 위한 메소드(실행 시작점):

JAVA VM -> **main()** 호출

main() 다른 클래스로 분리 가능

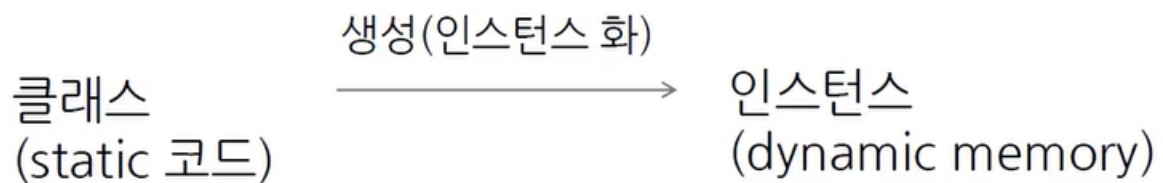
```
public static void main(String[] args) {  
  
}
```

클래스와 객체1 (2)

- 함수 : 하나의 기능 수행, 일련의 코드
 - 함수 이름, 매개 변수, return, 함수 반환형
- 메소드 : 함수의 일종, 클래스 내부 구현된 함수

!!! 하나의 함수, 하나의 기능

클래스와 객체1 (3)



- 클래스 : 개
- 인스턴스(객체) : 닥스훈트 뽀빠, 힙메모리에 생성
- 클래스 생성하기 :
 - 클래스형 변수이름 = new 생성자;
 - ex) Student studentLee = new Student();
 - 프로퍼티 접근법
 - studentLee.studentID = "이순신";
- 생성자 : 처음 객체 생성 시, 해야될 일(인스턴스 초기화) 담당
 - 디폴트 생성자 : 생성자 없을때만,
 - 생성자 오버로드 : 하나의 클래스 여러개 생성자, 파라미터 다름

클래스, 인스턴스, 참조변수, 참조 값

용어	설명
객체	객체 지향 프로그램의 대상, 생성된 인스턴스
클래스	객체를 프로그래밍하기 위해 코드로 만든 상태
인스턴스	클래스가 메모리에 생성된 상태
멤버 변수	클래스의 속성, 특성
메서드	멤버 변수를 이용하여 클래스의 기능을 구현
참조 변수	메모리에 생성된 인스턴스를 가리키는 변수
참조 값	생성된 인스턴스의 메모리 주소 값

클래스와 객체1 (4)

참조 자료형 :

변수의 자료형



클래스 형으로 선언하는 자료형

cf. 자바의 자료형

기본 자료형(8개) : boolean, byte, short, int, long, float, double, char

참조 자료형(무한개) : String, 배열, 클래스 형으로 선언하는 자료형 <- 힙메모리에 생성, new 키워드 사용

변수 vs 참조변수

: 변수는 값 자체, 참조변수는 주소값(값 자체는 heap메모리에 저장) 가지고 있음

정보은닉 :

- **public** : 외부 클래스에서 다 가져다 쓸 수 있음
- **private** : 객체 멤버변수 private 부 클래스에서 못가져다씀, Getter/Setter사용 값 제어, 클래스 내부에서만 가져다 쓸 수 있음
- **(default)** : 같은 패키지 내에서 가져다 쓸 수 있음
- **protected** : 상속 관계에서만 가져다 쓸 수 있음

클래스와 객체2 (1)

`this` 의 의미:

```
public Person(String name , int age){
    this.name = name;
    this.age = age;
}
```

- 객체 자신을 가리킴(this.name = name; vs name = name;; this안쓰면 가장 가까운 name 가리킴)

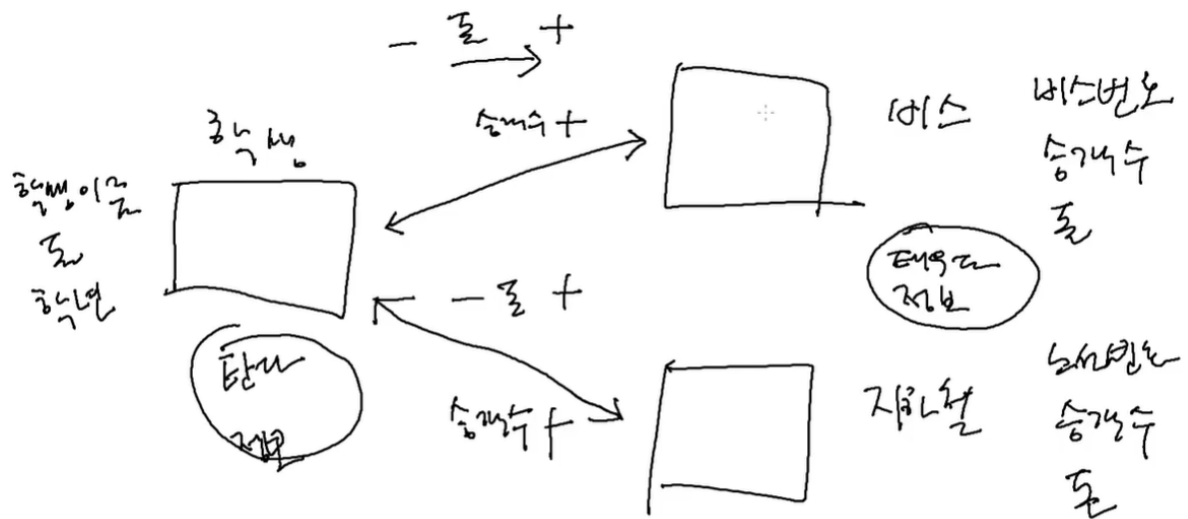
```
public Person(){
    this("이름없음", 1);
}
```

```
public Person(String name , int age){
    this.name = name;
    this.age = age;
}
```

- 생성자에서 같은 클래스의 다른 생성자를 호출(맨 윗줄에다 써야함)
- 자신의 주소를 반환 함

클래스와 객체2 (2)

객체지향 프로그래밍 == 객체 간의 협력



(객체 4개)

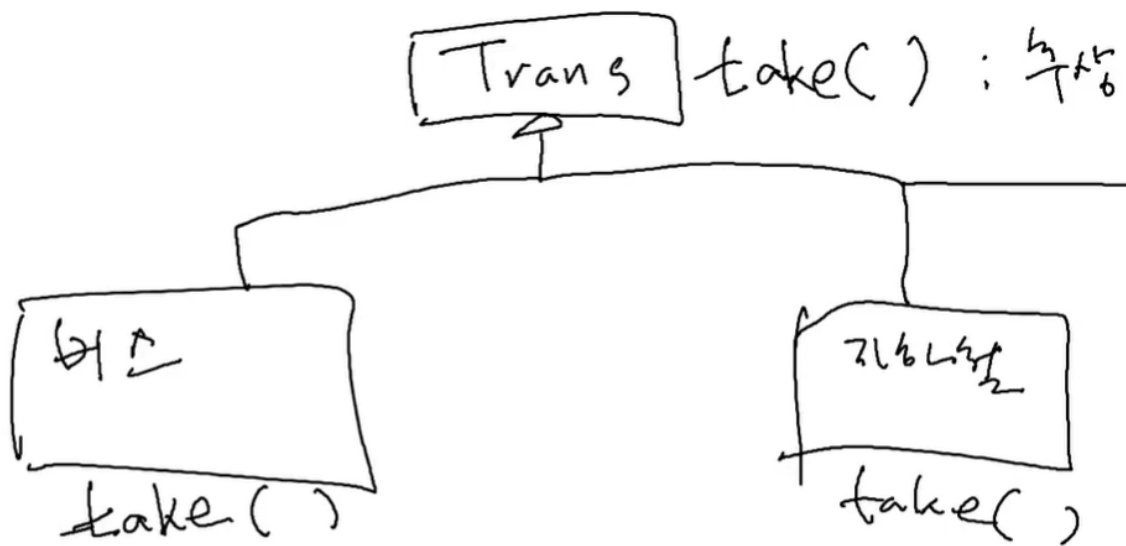
Bus.java

Subway.java

Student.java

TakeTrans.java

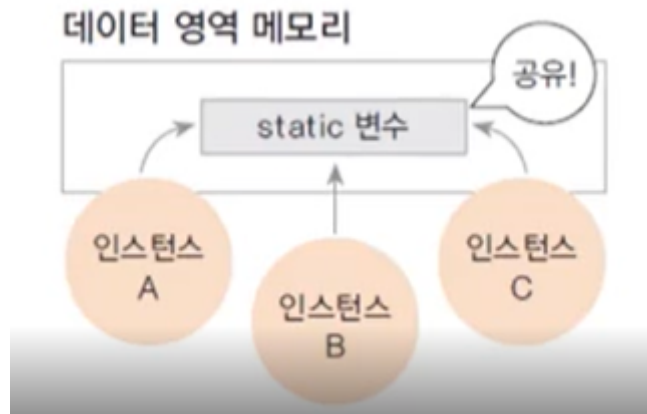
인스턴스 - 속성(멤버변수)/메소드



클래스와 객체2 (3) - static 변수

1. static 변수

: 자료형 변수이름 앞에 static 예약어를 붙임 ex. static int studentNum;

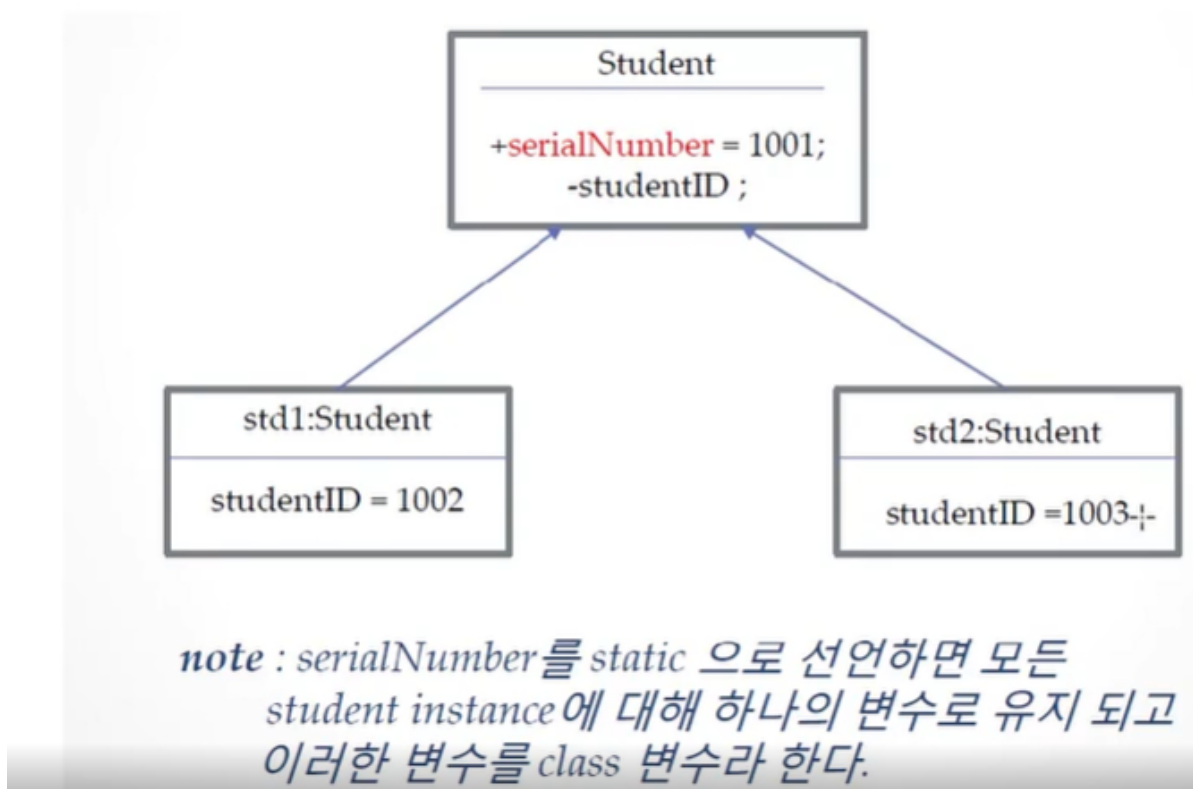


여러개의 인스턴스가 같은 메모리의 값을 공유하기 위해 사용

인스턴스 생성 유무와 상관 없이 메모리에 적재(static 영역)

static 변수는 참조변수로 사용할 때 인스턴스명 대신 클래스명 사용 <- 인스턴스명.변수명(X) 클래스명.변수명(O)

static 변수 VS. 인스턴스 변수

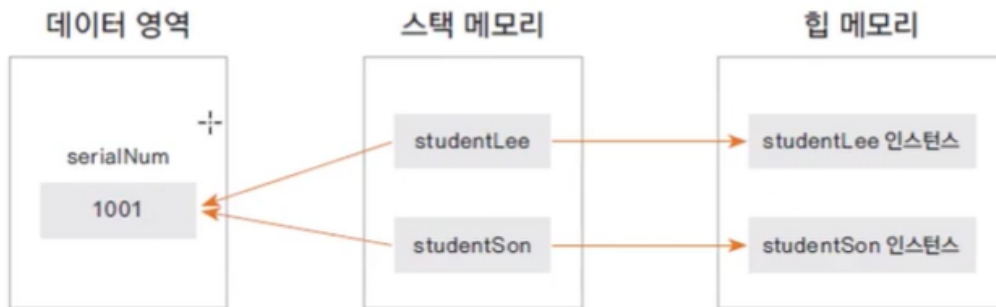


인스턴스(힙메모리 영역에 각각의 메모리 사용) + static변수(static 영역 공유)

static 변수 예

여러 인스턴스가 하나의 메모리 값을 공유 할 때 필요
학생이 생성될 때마다 학번이 증가해야 하는 경우
기준이 되는 값은 **static** 변수로 생성하여 유지 함

각 학생이 생성될 때 마다 **static** 변수 값을 복사해 와서 하나 증가시킨 값을 생성된 인스턴스의 학번 변수에 저장해 줌



자바의 메모리 구조 : static 영역 / Stack 영역 / Heap 영역

class 영역 static 영역 final 영역 메서드 영역	Stack 영역 (기본 자료형, 참조 자료형 상관없이 모든 지역 변수가 위치하는 영역)	Heap 영역 (객체(참조 자료형 변수가 가리키는 실제 데이터)가 위치하는 공간)
---	---	--

데이터 영역 (static 영역) : static, 상수, 리터럴 등 <- 프로그램의 시작부터 종료가 될 때까지 메모리에 남아있음

Stack 영역 : 지역변수 <- 메소드가 호출될 때 메모리에 할당되고 종료되면 메모리가 해제

Heap 영역 : 객체/인스턴스(키워드 new), 동적 메모리 <- Heap에 저장된 데이터가 더 이상 사용이 불필요하다면 메모리 관리를 위해 JVM(자바 가상머신)에 의해 알아서 해제, 가비지컬렉션(GC, 쓰레기 수집)

변수의 유효범위 :

변수의 유효 범위

변수 유형	선언 위치	사용 범위	메모리	생성과 소멸
지역 변수 (로컬 변수)	함수 내부에 선언	함수 내부에서만 사용	스택	함수가 호출될 때 생성되고 함수가 끝나면 소멸함
멤버 변수 (인스턴스 변수)	클래스 멤버 변수로 선언	클래스 내부에서 사용하고 private이 아니면 참조 변수로 다른 클래스에서 사용 가능	힙	인스턴스가 생성될 때 힙에 생성되고, 가비지 컬렉터가 메모리를 수거할 때 소멸됨
static 변수 (클래스 변수)	static 예약어를 사용하여 클래스 내부에 선언	클래스 내부에서 사용하고 private이 아니면 클래스 이름으로 다른 클래스에서 사용 가능	데이터 영역	프로그램이 처음 시작할 때 상수와 함께 데이터 영역에 생성되고 프로그램이 끝나고 메모리를 해제할 때 소멸됨

static 변수 -> static 함수

2. static 메서드 == 클래스 메서드

: 메서드에 static 키워드를 사용하여 구현, 주로 static 변수를 위한 기능 제공

static 메서드도 인스턴스의 생성과 관계없이 클래스 이름으로 직접 메서드 호출, 인스턴스의 변수의 경우 꼭 인스턴스가 먼저 생성되어야 하므로 static 메서드에서는 생성이 불확실한 인스턴스 변수를 사용할 수 없음

클래스와 객체2 (4) - Singleton 패턴

: 인스턴스가 1개만 생성되도록 클래스 설계하자!

객체를 생성한다X, 객체1개 생성해놓고 가져다 쓴다.O

1)private 생성자 <- 외부에서 생성자 호출X, 객체 생성 불가능!!!


2)클래스 내부에서 객체 1개 생성해주기

3)getInstance()메서드 생성 <- new 키워드로 객체를 생성하지 못하고, getInstance()로 가져다 씀

static 응용 : singleton 패턴

자동차 회사가 있고
자동차 회사에는 직원들이 있고
자동차 회사에는 여러 개의 공장들이
있고 생산된 자동차를 운반하는
유반차들이 있다

위에서 객체는 무엇이고, 이 중에
자동차 회사 시스템을 만들 때
단 한 개만이 존재하는 객체는 무엇인가?



-> 자동차 회사

ex) Calendar 등(javaUtil 기본 제공 클래스)

Calendar cal = new Calendar(); (X) -> Calendar cal = Calendar.getInstance(); (O)

cf. C언어 global 변수

JAVA 클래스 외부에 아무것도 선언 불가 패키지명, 임포트 제외