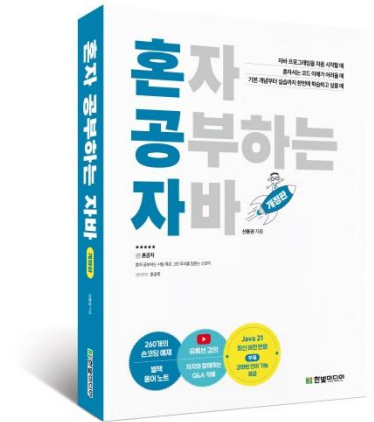


Chapter

# 06

클래스



## 06-1. 객체 지향 프로그래밍

혼자 공부하는 자바(개정판) (신용권 저)

## ❖ 목차

- 시작하기 전에
- 객체의 상호작용
- 객체 간의 관계
- 객체와 클래스
- 클래스 선언
- 객체 생성과 클래스 변수
- 클래스의 구성 멤버
- 키워드로 끝내는 핵심 포인트



# 시작하기 전에

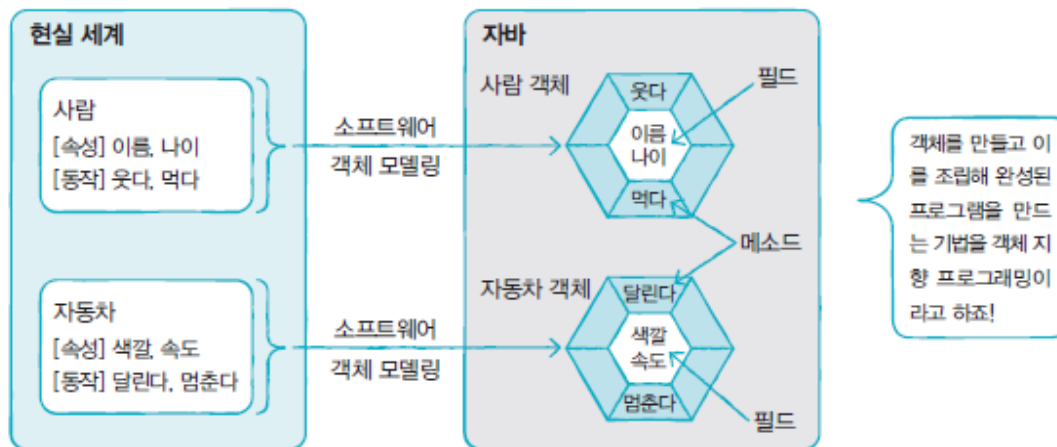
[핵심 키워드] : 클래스, 객체, new 연산자, 클래스 변수, 인스턴스, 클래스 멤버

## [핵심 포인트]

객체의 개념과 객체의 상호작용에 대해 알아본다.  
클래스로부터 객체를 생성하고 변수로 참조한다.

## ❖ 객체 (Object)

- 물리적으로 존재하거나 추상적으로 생각할 수 있는 것 중에서 자신의 속성을 가지며 식별 가능한 것
- 속성 (필드(field)) + 동작(메소드(method))로 구성



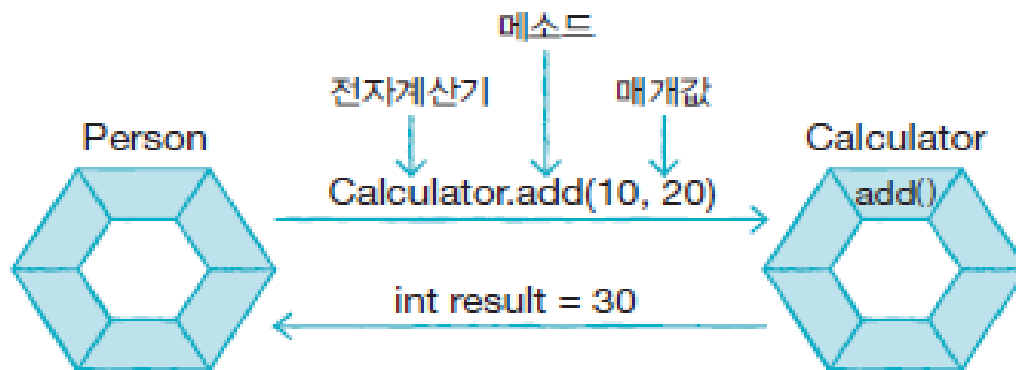
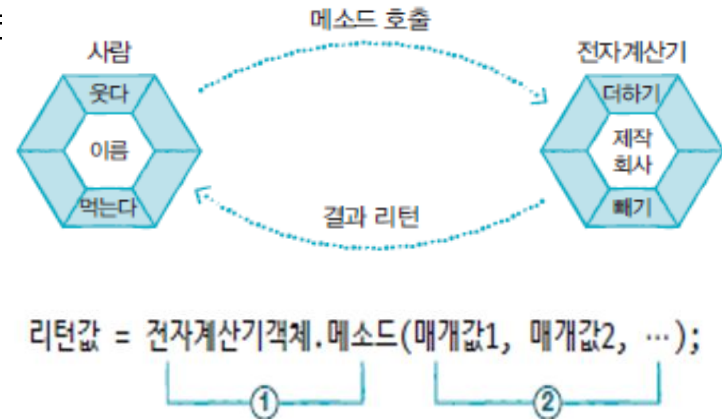
# 객체의 상호작용

## ❖ 객체와 객체 간의 상호작용

- 메소드를 통해 객체들이 상호작용
- **메소드 호출** : 객체가 다른 객체의 기능을 이용하는

```
int result = Calculator.add(10, 20);
```

리턴한 값을 int 변수에 저장



# 객체 간의 관계

## ❖ 객체 간의 관계

### ■ 집합 관계 (has a)

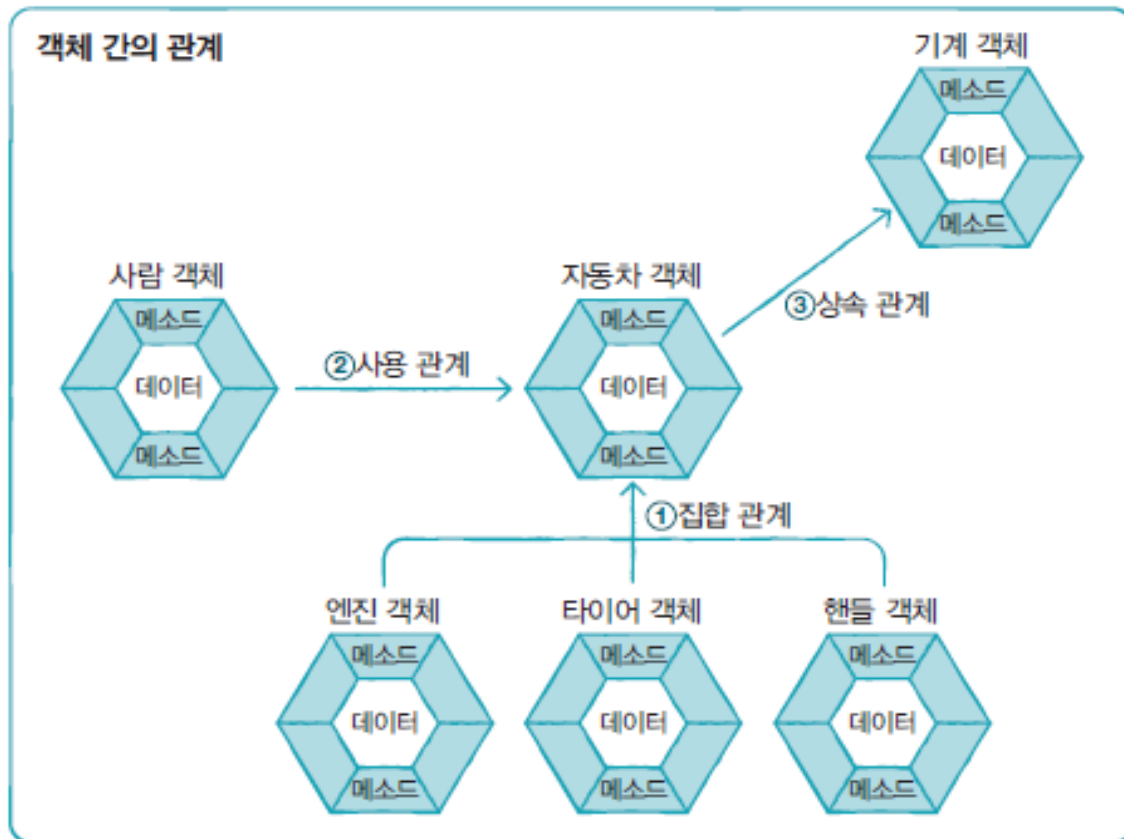
- 부품과 완성품의 관계  
ex.

### ■ 사용 관계 (use)

- 객체 간의 상호작용  
가

### ■ 상속 관계 (is a)

- 상위(부모) 객체를 기반으로
- 하위(자식) 객체를 생성  
ex.



### ■ 객체 지향 프로그래밍

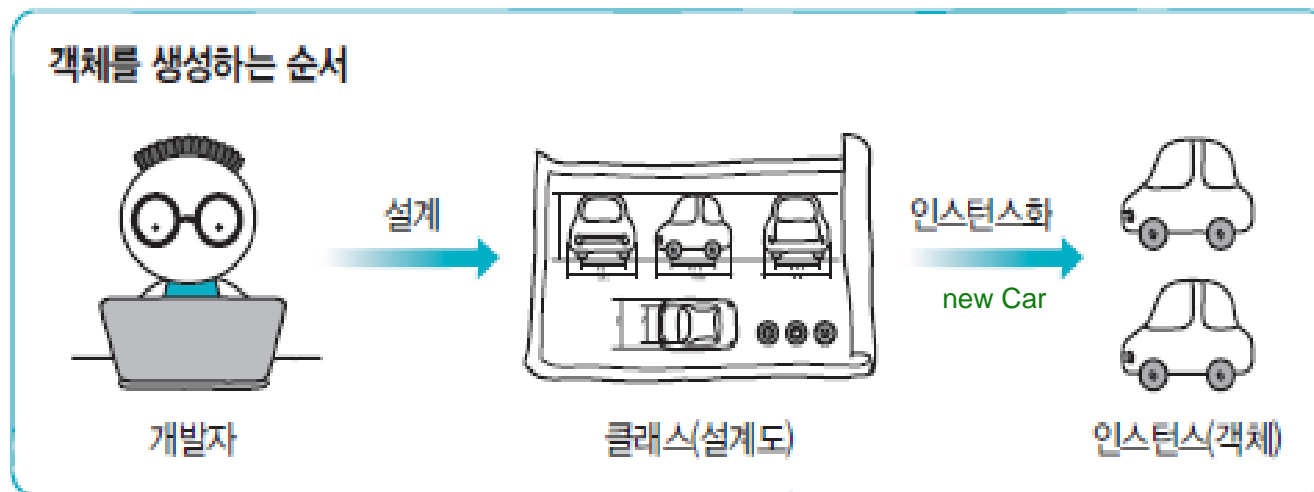
- 집합/사용 관계에 있는 객체를 하나씩 설계한 후 조립하여 프로그램 개발



## 객체와 클래스

## ❖ 클래스 (class)

- 자바의 설계도 ( )
- **인스턴스** (instance): 클래스로부터 만들어진 객체 ( = )
- 객체지향 프로그래밍 단계
  - 클래스 설계 -> 설계된 클래스로 사용할 객체 생성 -> 객체 이용



# 클래스 선언

## ❖ 클래스 선언

- 객체 구상 후 클래스 이름을 결정
  - 식별자 작성 규칙에 따라야 함
    - 하나 이상의 문자로 이루어질 것
    - 첫 글자에는 숫자 올 수 없음
    - \$, \_ 외의 특수 문자는 사용할 수 없음
    - 자바 키워드는 사용할 수 없음
- '클래스 이름.java'로 소스 파일 생성

Calculator, Car, Member, ChatClient, ChatServer, Web\_Browser

```
public class 클래스이름 {  
  
}
```



# 객체 생성과 클래스 변수

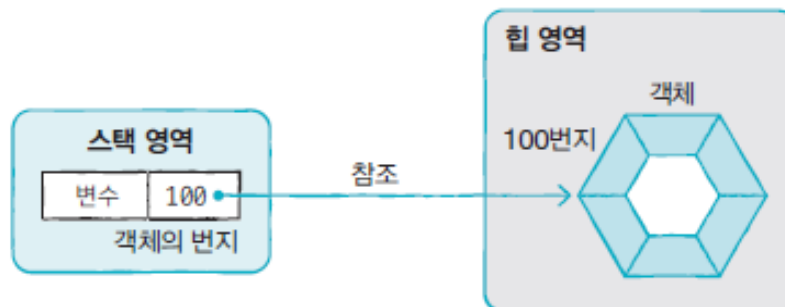
## ❖ 클래스로부터 객체를 생성

- new 클래스();
- new 연산자로 메모리 힙 영역에 객체 생성
- 객체 생성 후 객체 번지가 리턴
  - 클래스 변수에 저장하여 변수 통해 객체 사용 가능



```
클래스 변수;  
변수 = new 클래스();
```

```
클래스 변수 = new 클래스();
```

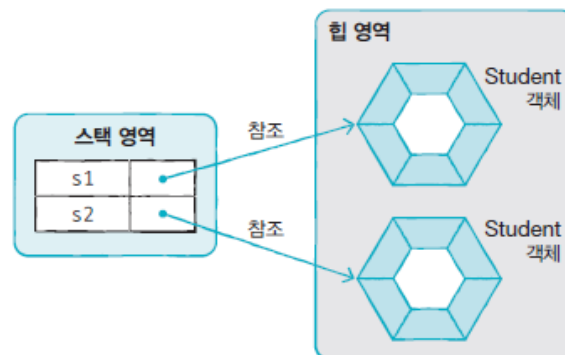




# 객체 생성과 클래스 변수

```
public class Student {  
}
```

```
• public class StudentExample {  
    public static void main(String[] args) {  
        Student s1 = new Student();  
        System.out.println("s1 변수가 Student 객체를 참조합니다.")  
  
        Student s2 = new Student();  
        System.out.println("s2 변수가 또 다른 Student 객체를 참조합니다.");  
    }  
}
```



## ❖ 클래스의 두 용도

- 라이브러리(API : Application Program Interface) 클래스
  - 객체 생성 및 메소드 제공 역할 - Student.java
- 실행 클래스
  - main() 메소드 제공 역할 - StudentExample.java



# 클래스의 구성 멤버

## ❖ 클래스 멤버

• 필드(Field)  
객체의 데이터가 저장되는 곳

• 생성자(Constructor)  
객체 생성 시 초기화 역할 담당

• 메소드(Method)  
객체의 동작에 해당하는 실행 블록

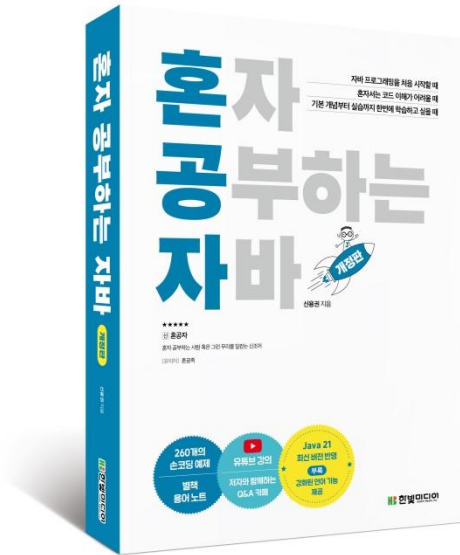
```
public class ClassName {  
  
    //필드  
    int fieldname;  
  
    //생성자  
    ClassName() { ... }  
  
    //메소드  
    void methodName() { ... }  
  
}
```



# 키워드로 끝내는 핵심 포인트

- **클래스**: 객체를 만들기 위한 설계도
- **객체** : 클래스로부터 생성되며 'new 클래스()'로 생성
- **new 연산자** : 객체 생성 연산자이며 생성자 호출하고 객체 생성 번지를 리턴
- **클래스 변수** : 클래스로 선언한 변수이며 해당 클래스의 객체 번지가 저장됨
- **인스턴스** : 객체는 클래스의 인스턴스
- **클래스 멤버** : 클래스에 선언되는 멤버로 필드, 생성자, 메소드가 있음





Thank You!