클래스
(객체 지향 프로그램)

목차

1. 기본

- 핵심 포인트
  - 생성자는 new 연산자로 호출되는 중괄호{} 블록
  - 객체 생성 시 초기화를 담당
- 생성자 (constructor)
  - 클래스로부터 new 연산자로 객체를 생성할 때 호출되어 객체의 초기화를 담당
- 객체 초기화
  - 필드를 초기화하거나 메소드를 호출해서 객체를 사용할 준비를 하는 것
- 생성자가 성공적으로 실행
  - 힙 영역에 객체 생성되고 객체 번지가 리턴

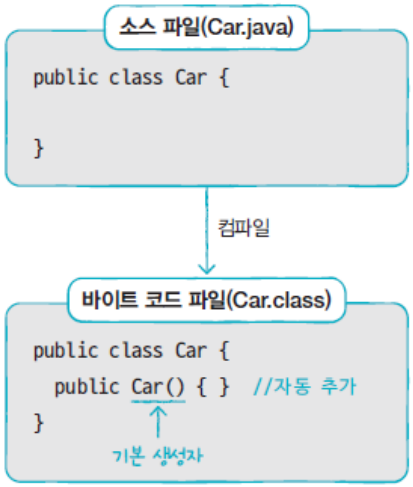
1.1 기본 생성자

- 기본 생성자 (default constructor)
  - 클래스 내부에 생성자 선언 생략할 경우 바이트 코드에 자동 추가

```
[public] 클래스() { }
```

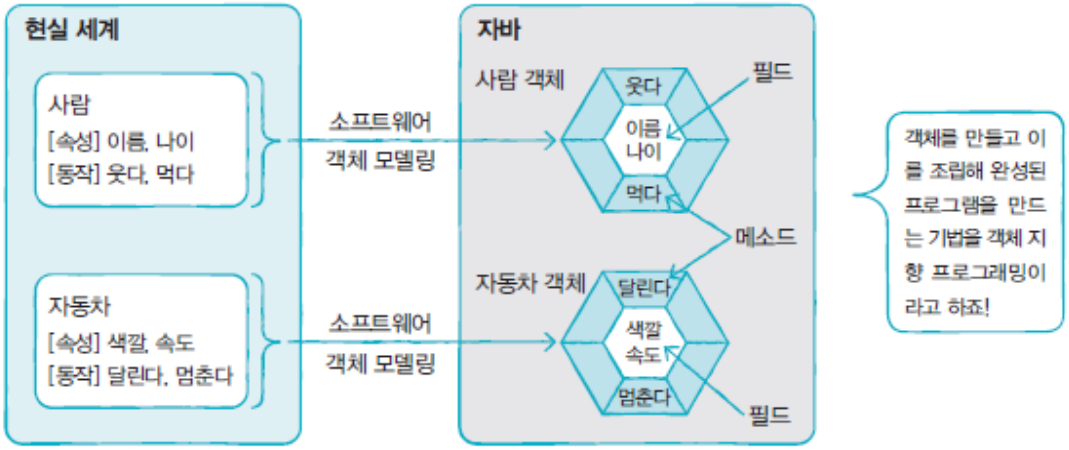
  - 클래스에 생성자 선언하지 않아도 new 생성자()로 객체 생성 가능

```
Car myCar = new Car();
            ↑
          기본 생성자
```



1.1 객체 (Object)

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

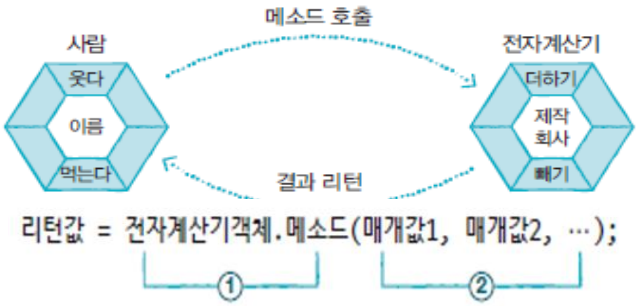


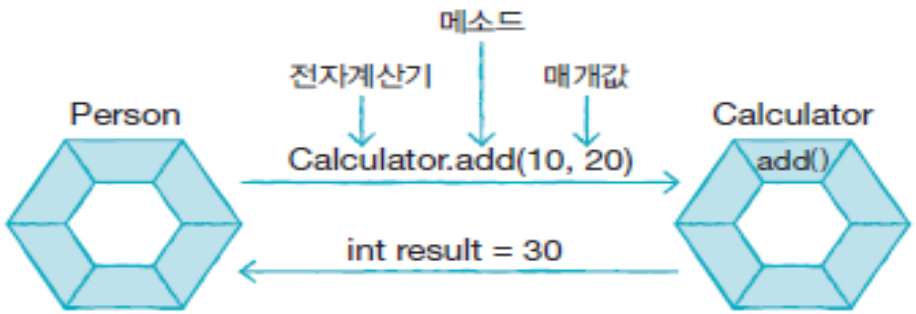
1.2 객체와 객체 간의 상호작용

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

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

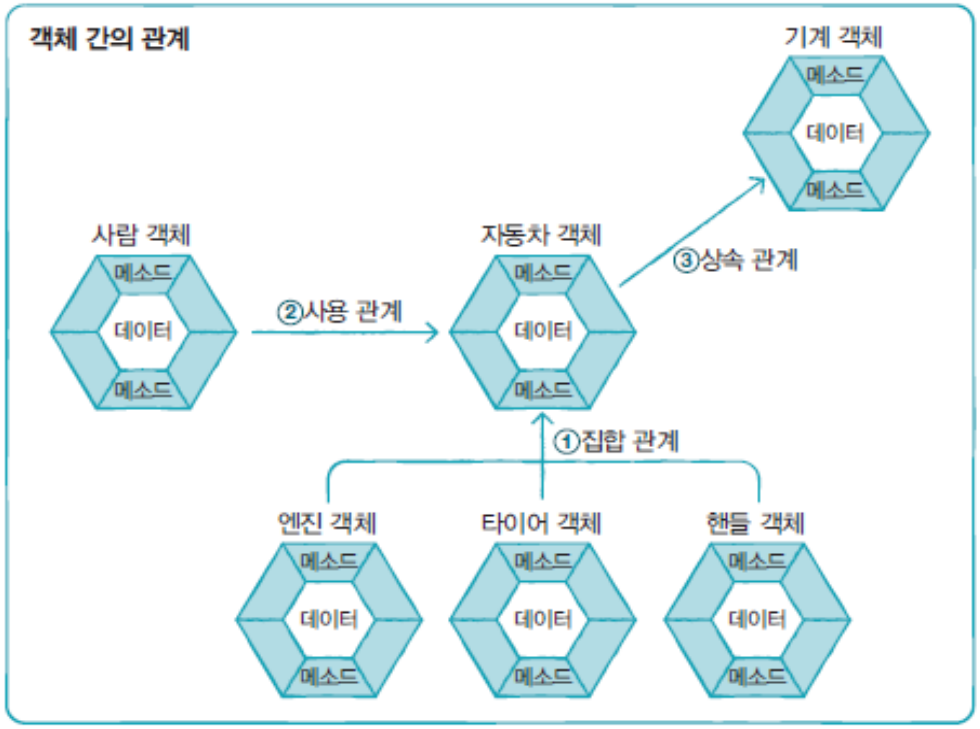
리턴한 값을 int 변수에 저장





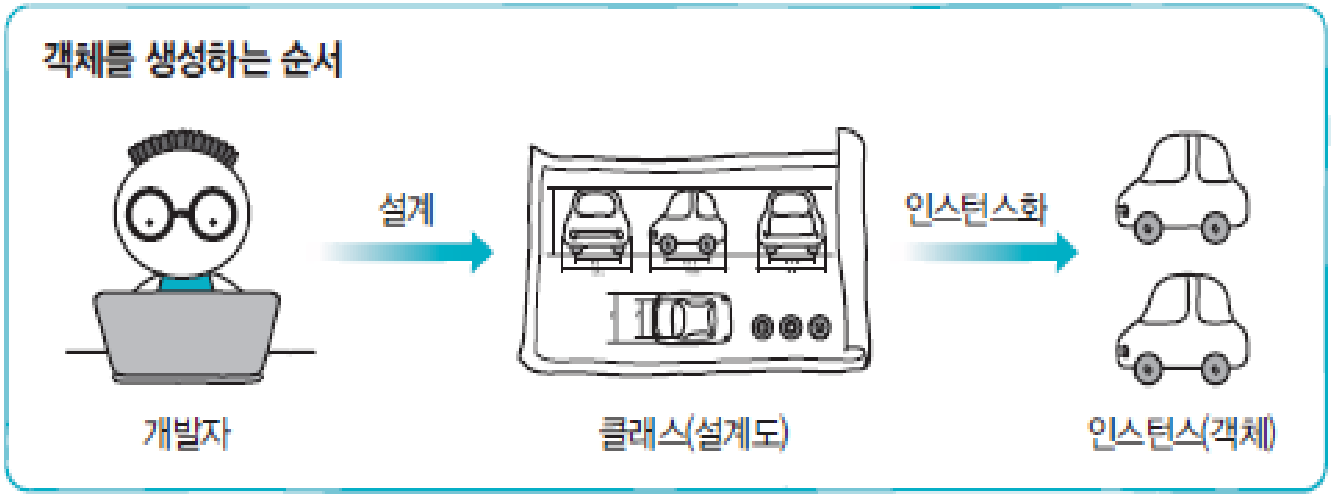
1.3 객체 간의 관계

- 집합 관계
  - 부품과 완성품의 관계
- 사용 관계
  - 객체 간의 상호작용
- 상속 관계
  - 상위(부모) 객체를 기반으로
  - 하위(자식) 객체를 생성
- 객체 지향 프로그래밍
  - 집합/사용 관계에 있는 객체를 하나씩 설계한 후 조립하여 프로그램 개발



1.4 객체와 클래스

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



1.5 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

```
Calculator, Car, Member, ChatClient, ChatServer, Web_Browser
```

- '클래스 이름.java'로 소스 파일 생성

```
public class 클래스이름 {


}
```

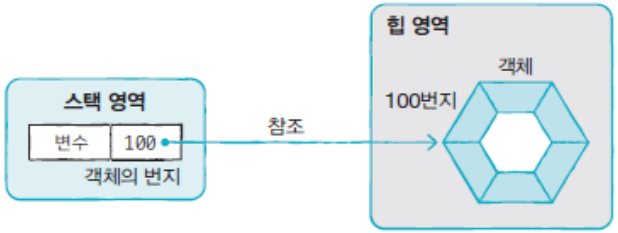
1.6 객체 생성과 클래스 변수

- 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();
```

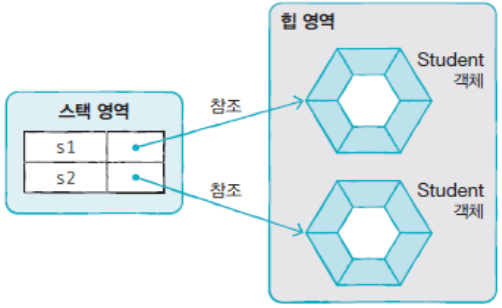
```
클래스 변수 = new 클래스();
```



1.6 객체 생성과 클래스 변수

```
public class Student {
}
```

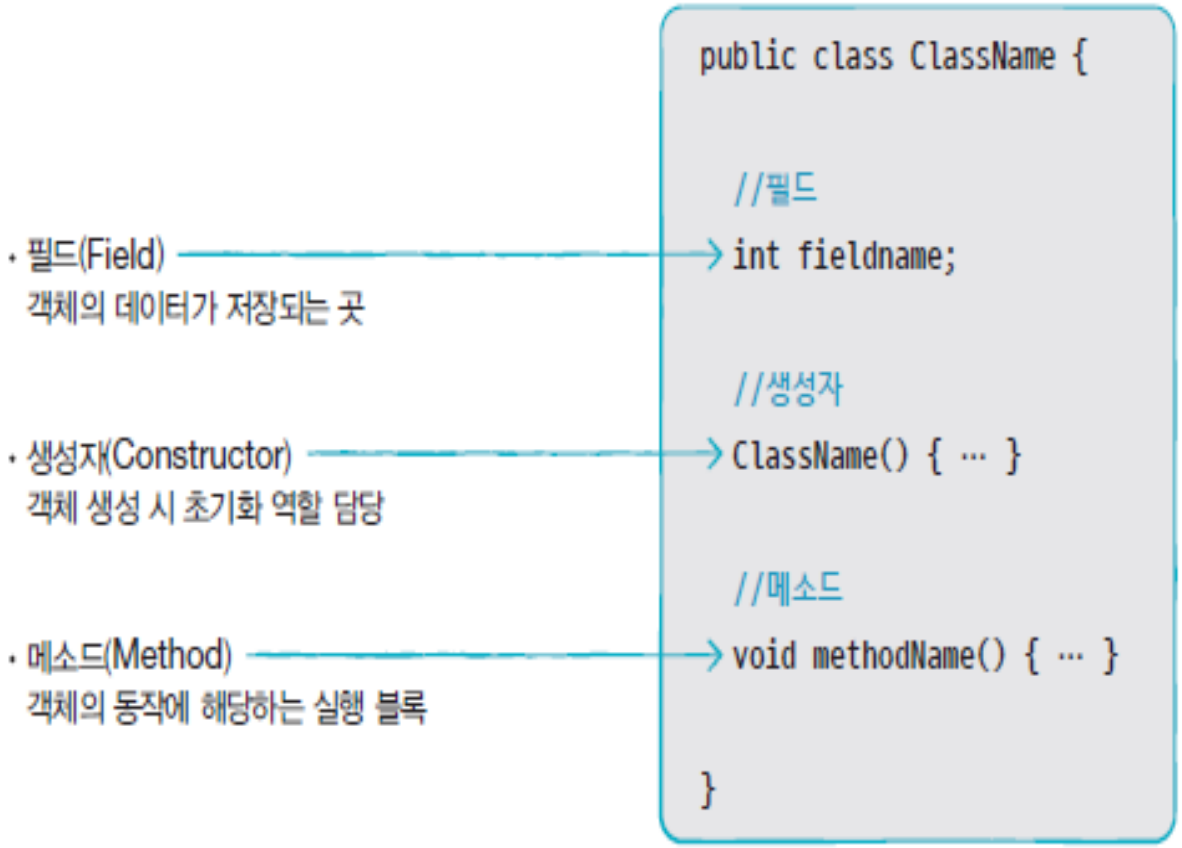
```
public class StudentExample {
  public static void main(String[] args) {
    Student s1 = new Student();
    System.out.println("s1 변수가 Student 객체를 참조합니다.");

    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

1.7 클래스의 구성 멤버

- 클래스 멤버

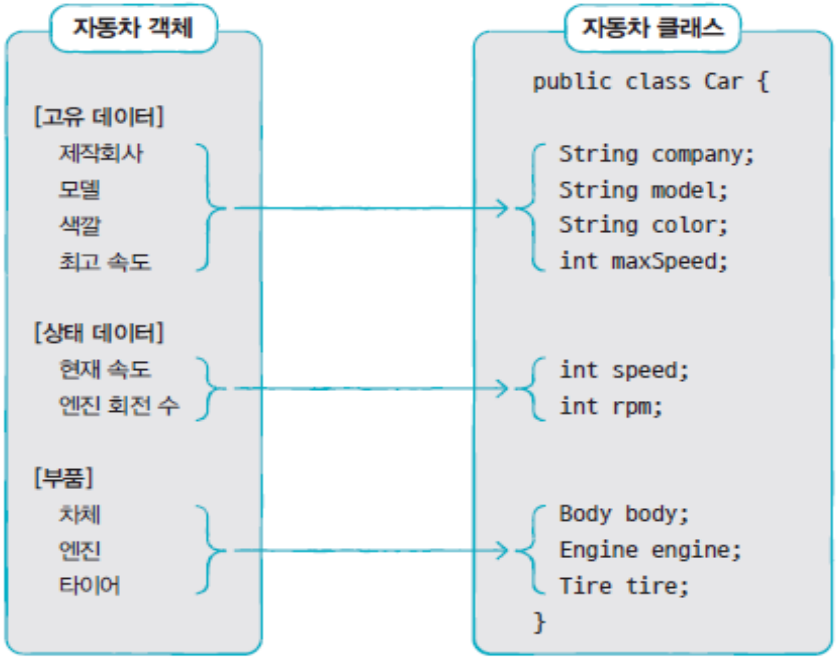


1.8 용어 정리

- 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

1.9 필드

- 필드
  - 객체의 고유 데이터,
  - 객체가 가져야 할 부품,
  - 객체의 현재 상태 데이터 등을 저장



1.10 필드 선언

- 필드 선언

```
타입 필드 [ = 초기값] ;
```

  - 클래스 중괄호 블록 어디서든 존재 가능
  - 생성자와 메소드 중괄호 블록 내부에는 선언 불가
  - 변수와 선언 형태 유사하나 변수 아님에 주의

  - class XXX {

```
String company = "현대자동차";
String model = "그랜저";
int maxSpeed = 300;
int productionYear;
int currentSpeed;
boolean engineStart;
```

  - }

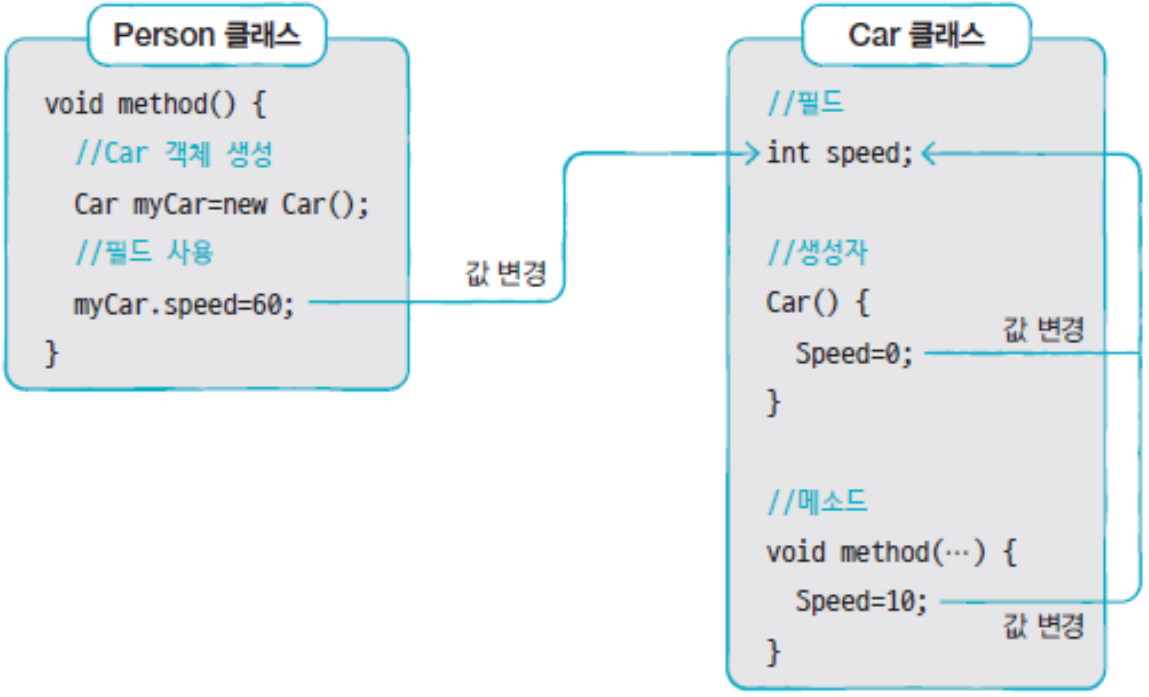
1.10 필드 선언

- 초기값은 주어질 수도, 생략할 수도 있음
  - 초기값 지정되지 않은 필드는 객체 생성 시 자동으로 기본 초기값 설정

분류		타입	초기값
기본 타입	정수 타입	byte char short int long	0 \u0000 (빈 공백) 0 0 0L
	실수 타입	float double	0.0F 0.0
	논리 타입	boolean	false
참조 타입		배열 클래스(String 포함) 인터페이스	null null null

1.11 필드 사용

- 필드 사용
  - 필드값 읽고 변경하는 작업
  - 클래스 내부 생성자 및 메소드에서 사용하는 경우 : 필드 이름으로 읽고 변경
  - 클래스 외부에서 사용하는 경우 : 클래스로부터 객체 생성한 뒤 필드 사용



sec02.exam01
sec02.exam02

1.12 용어 정리

- 필드 선언
  - 클래스 중괄호 블록 어디서든 선언하나 생성자나 메소드 내부에서는
    사용 불가

- 필드 사용
  - 클래스 내부의 생성자와 메소드에서 바로 사용 가능
  - 클래스 외부에서 사용할 경우 반드시 객체 생성하고 참조 변수 통해 사용

질의 응답

- Q & A ???