

IT@COOKBOOK



그림으로 설명하고 실생활 예제로 실습하는

자바를 부탁해

① 자바 기초

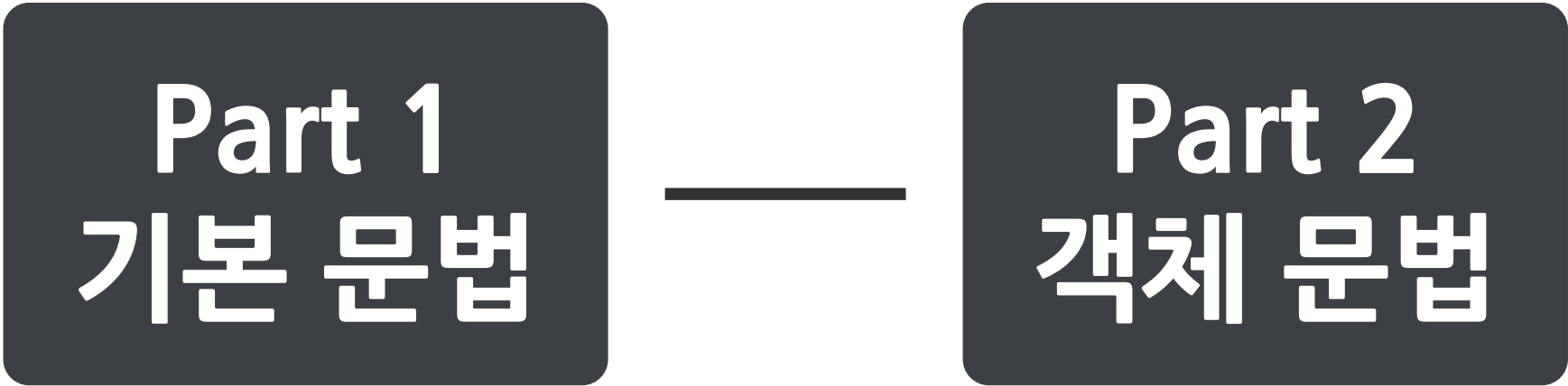
- 01 자바 시작하기
- 02 변수, 타입, 연산자

② 메소드와 조건문

- 03 메소드
- 04 조건문

③ 반복문과 배열

- 05 반복문
- 06 배열



④ OOP 기초

- 07 객체 지향 프로그래밍
- 08 객체 지향 프로그래밍의 활용
- 09 자바 API

⑤ OOP 심화

- 10 상속
- 11 인터페이스
- 12 예외 처리와 파일 입출력

⑥ 프로젝트

- 13 실전 프로젝트: 영화 예매
관리 프로그램

CH07

객체 지향 프로그래밍의 기초

01 객체 지향 프로그래밍 개요

02 클래스 설계와 구현

03 객체의 활용

객체 지향 프로그래밍을 이해한다.

클래스와 **객체**를 구분한다.

클래스 설계와 **구현**을 연습한다.

객체를 **활용**한 프로그램 작성법을 익힌다.

CH07

객체 지향 프로그래밍의 기초

01 객체 지향 프로그래밍 개요

02 클래스 설계와 구현

03 객체의 활용

I . 객체 지향 프로그래밍이란

객체 지향 프로그래밍(Object-Oriented Programming, **OOP**)이란

- 프로그램을 조립하여 만드는 방식
- 프로그램을 부분별로 만든 뒤 전체를 구성하는 형식

자전거를 예로 들면,

- 자전거는 바퀴, 몸체, 손잡이 등의 부품으로 구성
- 자전거를 프로그램이라 한다면 각 부품은 객체가 됨
- 이렇게 각 객체가 모여 전체 프로그램을 구성하는 방식을 OOP라 함

I. 객체 지향 프로그래밍이란

객체 지향 프로그래밍(Object-Oriented Programming, **OOP**)이란

- 프로그램을 조립하여 만드는 방식
- 프로그램을 부분별로 만든 뒤 전체를 구성하는 형식

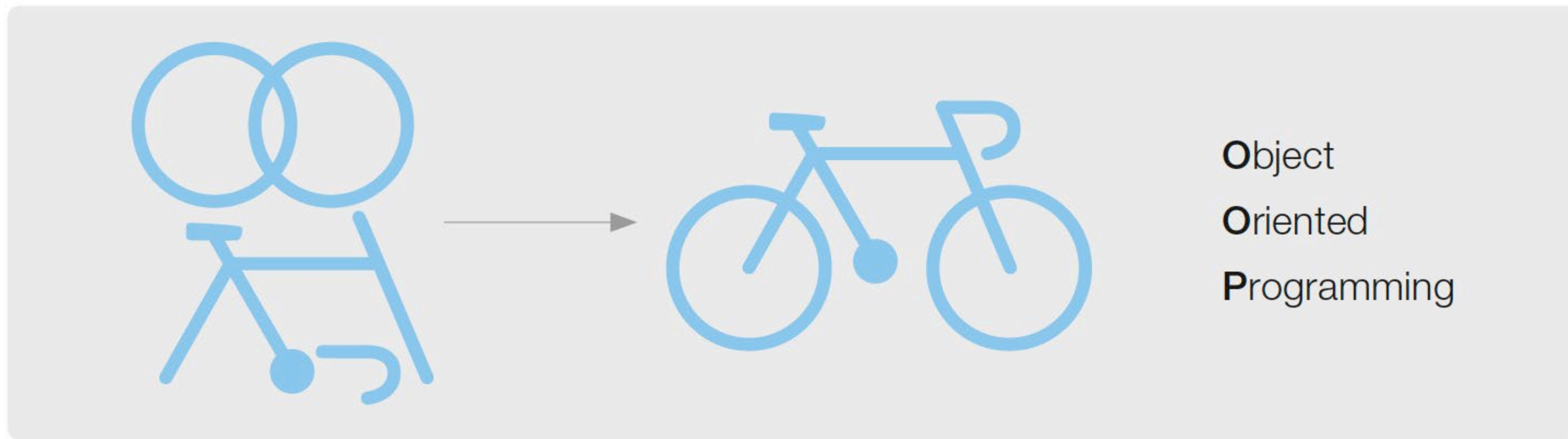


그림 7-1 OOP

II. 객체 지향 프로그래밍의 장점

유지 보수성

- OOP 기반 프로그램은 객체별 관리 가능
- 문제가 생기더라도 특정 객체의 코드만 고치면 됨

코드 재사용성

- 다양한 코드뿐 아니라 다른 프로그램에서까지 손쉽게 사용할 수 있음

프로그램 확장성

- 처음부터 조립 방식을 고려해 만들기 때문에 새로운 코드로 확장하기 편리



III. 클래스와 객체의 관계

클래스(class): 객체의 설계도

- 자전거 부품의 설계도가 필요하듯 객체를 만들려면 클래스가 필요

객체(object): 클래스를 토대로 만들어진 프로그램의 구성 요소

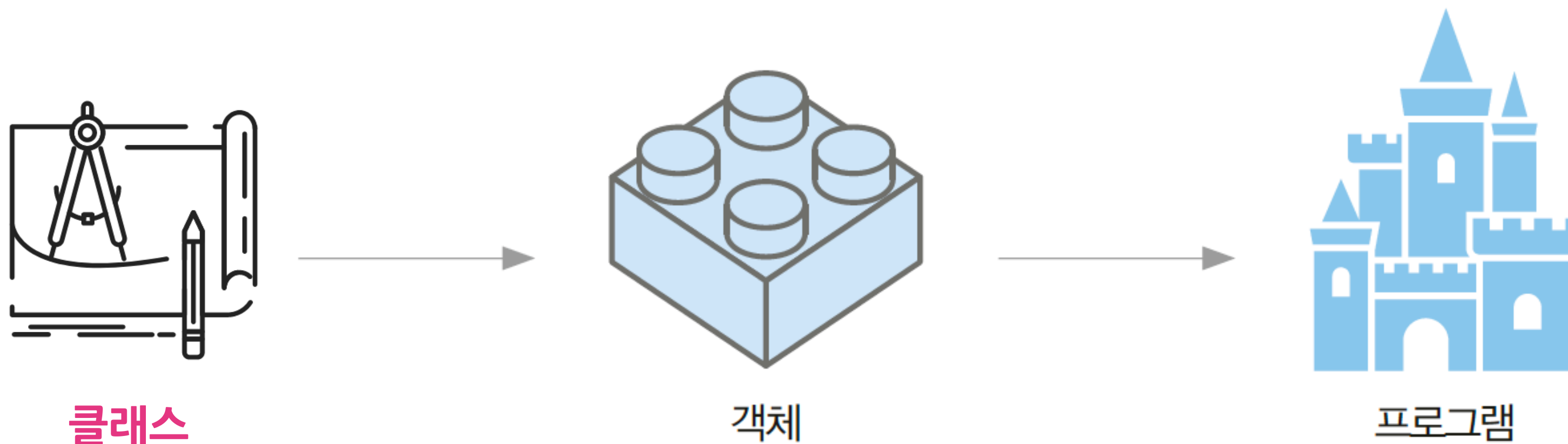


그림 7-3 객체는 어떻게 만들까?

III. 클래스와 객체의 관계

클래스를 토대로 객체를 만드는 과정을 **인스턴스화**(instantiation)라 함

- 이러한 이유로 객체를 **인스턴스**(instance)라고 부르기도 함

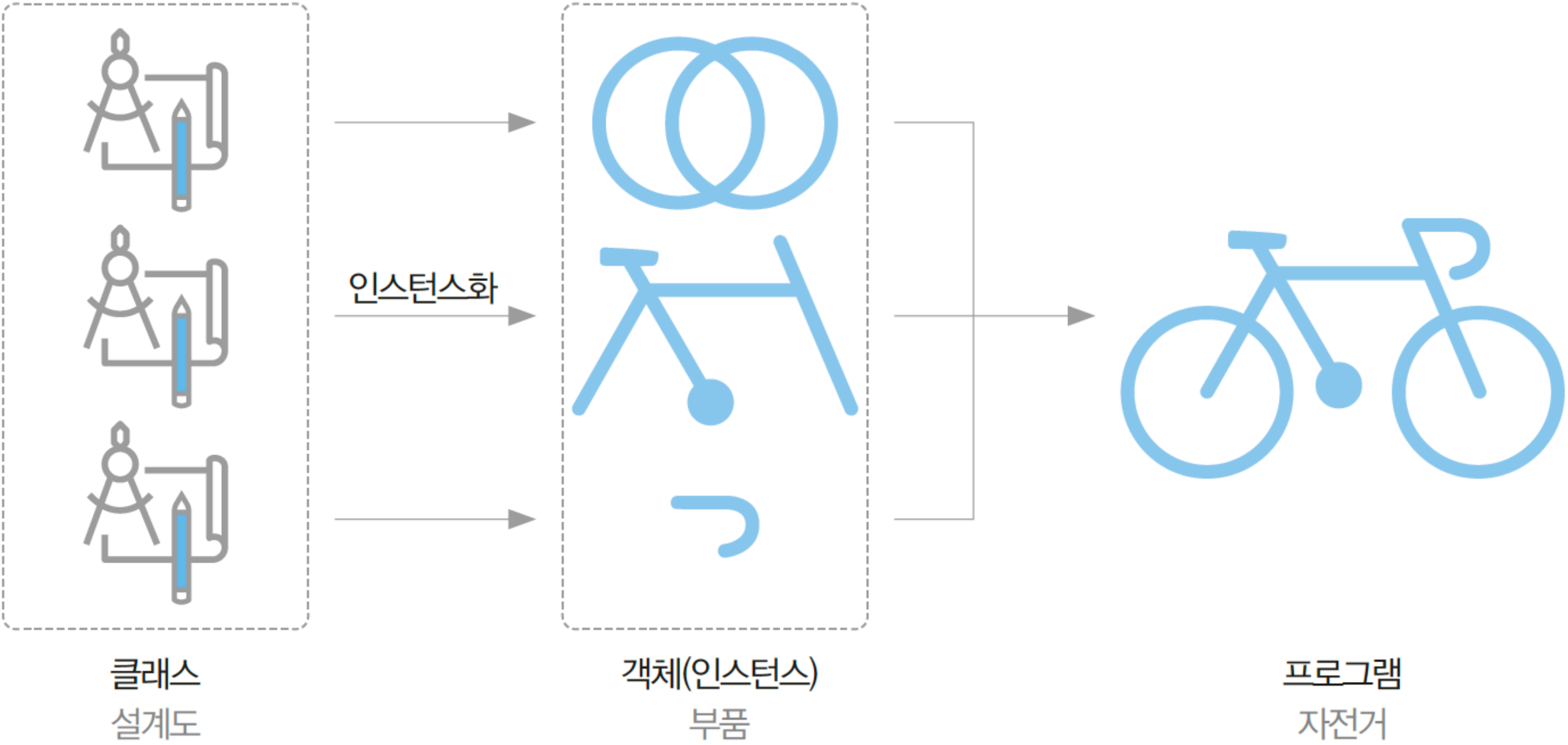


그림 7-4 클래스와 객체, 프로그램의 관계

CH07

객체 지향 프로그래밍의 기초

01 객체 지향 프로그래밍 개요

02 클래스 설계와 구현

03 객체의 활용

I. 클래스 설계하기

클래스 설계란

- 객체를 미리 구상하는 과정으로
- 객체의 상태(state)와 동작(behavior)을 정의하는 것
 - 이때 상태는 변수, 동작은 메소드로 만들 수 있는데
- 클래스는 객체가 가질 상태를 **필드**(field), 동작을 **메소드**(method)로 정의

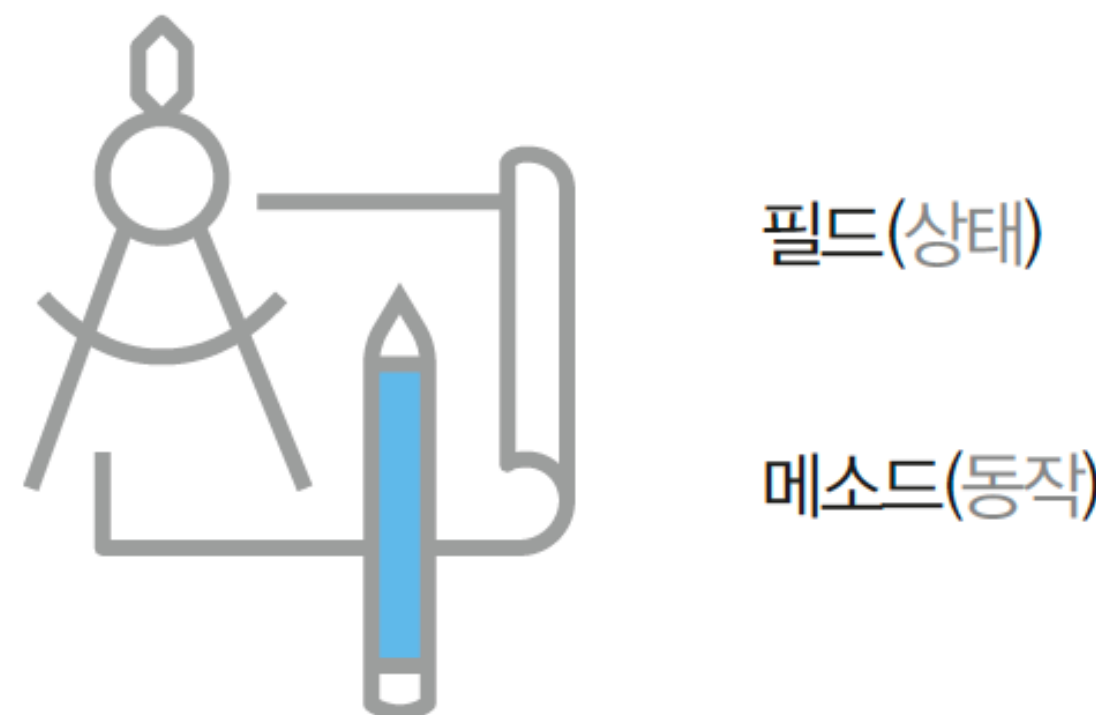
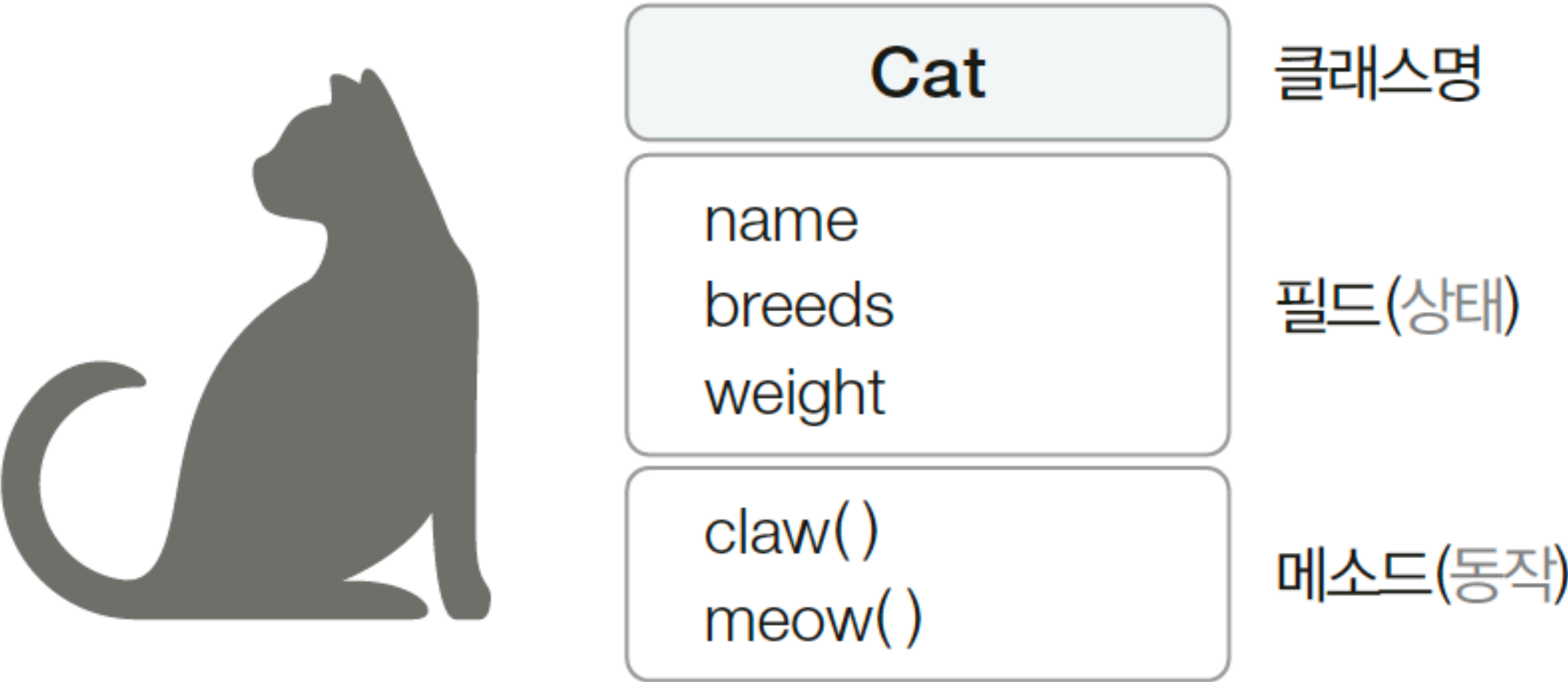


그림 7-5 클래스의 구성 요소, 필드와 메소드

I . 클래스 설계하기

클래스를 도식화한 것을 **클래스 다이어그램**(class diagram)이라고 함

- 고양이를 클래스를 설계해보자!



! TIP 특정 대상의 상세 정보가 아닌 핵심을 정의하는 것을 **추상화**(abstraction)라 함. 클래스 설계는 추상화의 대표적 예.

그림 7-6 Cat 클래스의 필드와 메소드 설계

II. 클래스 구현하기

앞서 설계한 고양이(Cat) 클래스 다이어그램을 실제 코드로 구현한 예

```
public class Cat {
    String name;    // 이름
    String breeds;  // 품종
    double weight;  // 체중

    /* 메소드(동작)를 정의한다. */
    void claw() {
        System.out.println("할퀴기!!");
    }

    void meow() {
        System.out.println("야옹~");
    }
}
```

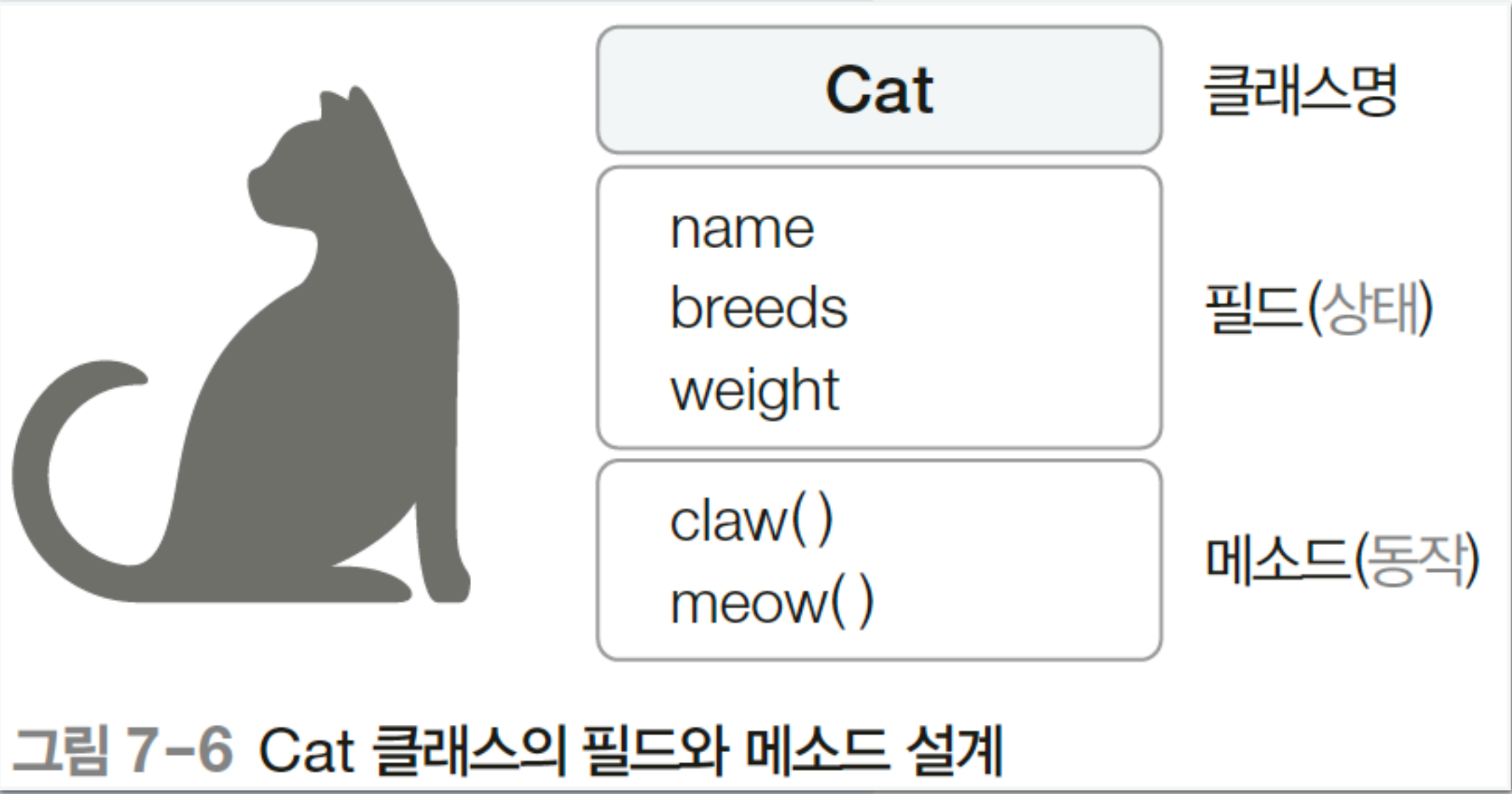


II. 클래스 구현하기

앞서 설계한 고양이(Cat) 클래스 다이어그램을 실제 코드로 구현해보자!

```
public class Cat {  
    String name;    // 이름  
    String breeds; // 품종  
    double weight; // 체중  
  
    /* 메소드(동작)를 정의한다. */  
    void claw() {  
        System.out.println("할퀴기!!");  
    }  
  
    void meow() {  
        System.out.println("야옹~");  
    }  
}
```

① 클래스를 만들고



II. 클래스 구현하기

앞서 설계한 고양이(Cat) 클래스 다이어그램을 실제 코드로 구현해보자!

```
public class Cat {
    String name;    // 이름
    String breeds;  // 품종
    double weight;  // 체중

    /* 메소드(동작)를 정의한다. */
    void claw() {
        System.out.println("할퀴기!!");
    }

    void meow() {
        System.out.println("야옹~");
    }
}
```

② 클래스 안쪽 필드를 변수로 작성한 뒤

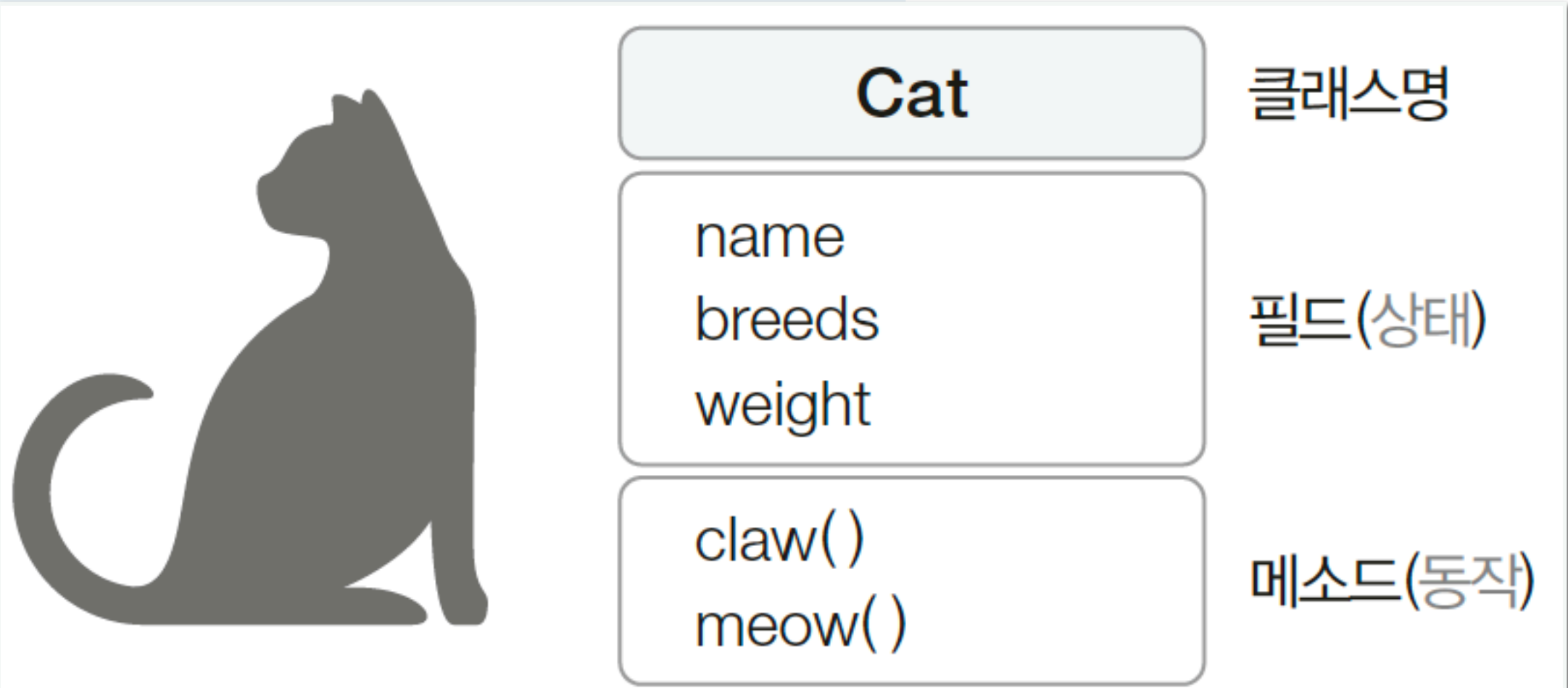


그림 7-6 Cat 클래스의 필드와 메소드 설계

II. 클래스 구현하기

앞서 설계한 고양이(Cat) 클래스 다이어그램을 실제 코드로 구현해보자!

```
public class Cat {
    String name;    // 이름
    String breeds;  // 품종
    double weight;  // 체중

    ③ 메소드 영역을 정의한다

    void claw() {
        System.out.println("할퀴기!!");
    }

    void meow() {
        System.out.println("야옹~");
    }
}
```



II. 클래스 구현하기

앞서 설계한 고양이(Cat) 클래스 다이어그램을 실제 코드로 구현해보자!

```
public class Cat {  
    String name;    // 이름  
    String breeds; // 품종  
    double weight; // 체중
```

③ 메소드 영역을 정의한다

```
void claw() {  
    System.out.println("할퀴기!!");  
}
```

메소드에 public static 키워드는 당분간 사용하지 않음

```
void meow() {  
    System.out.println("야옹~");  
}  
}
```

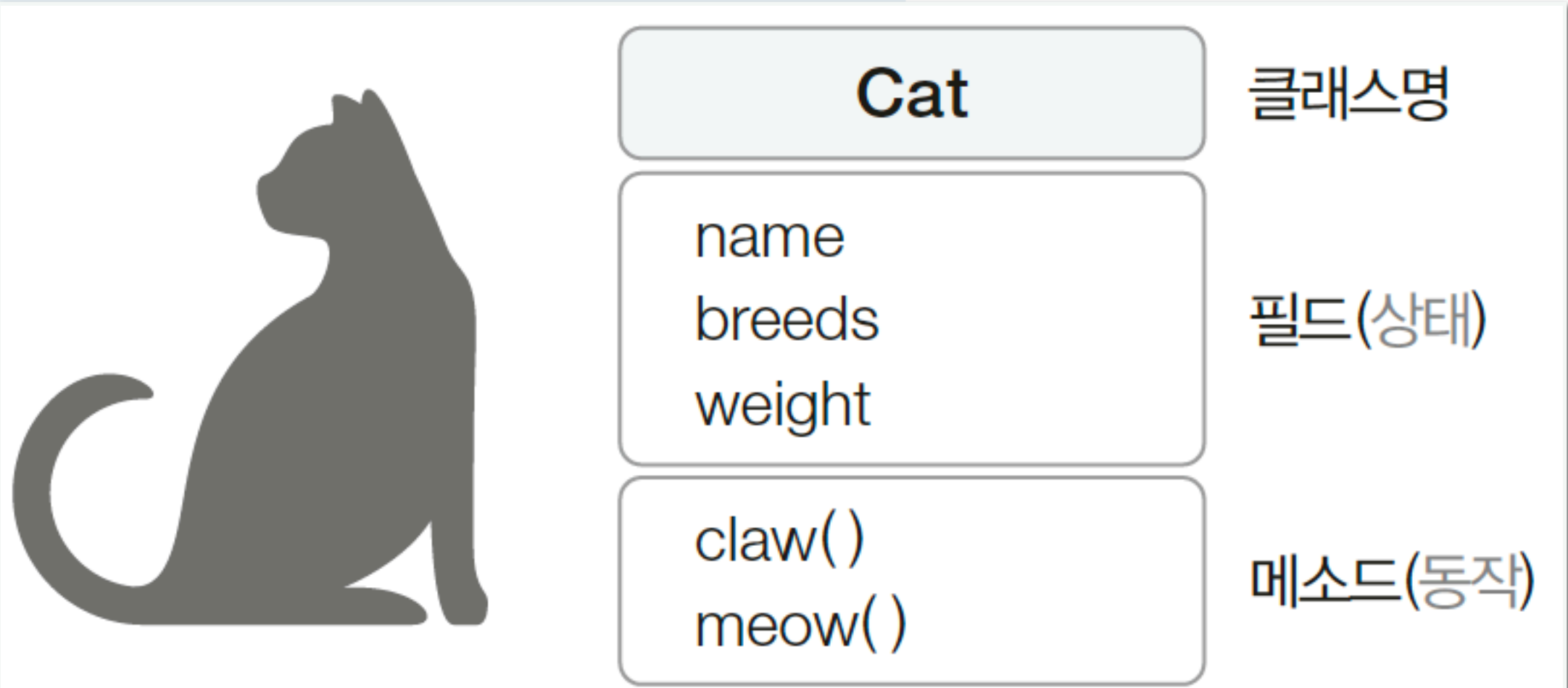



그림 7-6 Cat 클래스의 필드와 메소드 설계

II. 클래스 구현하기

앞서 설계한 고양이(Cat) 클래스 다이어그램을 실제 코드로 구현해보자!

```
public class Cat {  
    String name;    // 이름  
    String breeds; // 품종  
    double weight; // 체중  
  
    /* 메소드(동작)를 정의한다. */  
    void claw() {  
        System.out.println("할퀴기!!");  
    }  
  
    void meow() {  
        System.out.println("야옹~");  
    }  
}
```

 **TIP** 관련 데이터와 코드를 하나의 단위로 묶는 것을 **캡슐화**(encapsulation)라 함. 클래스는 필드와 메소드를 묶는 캡슐화의 대표적 예.

Quiz

01 _____란 객체의 설계도이다.

02 클래스를 토대로 객체를 만드는 과정을 _____라고 한다.

03 클래스는 상태를 의미하는 _____와 동작을 의미하는 _____로 구성된다.

CH07

객체 지향 프로그래밍의 기초

01 객체 지향 프로그래밍 개요

02 클래스 설계와 구현

03 객체의 활용

I. 객체 생성하기

객체 생성 형식

- **new** 키워드와 함께 클래스명과 소괄호를 적어 생성

```
new 클래스명();    // 객체 생성 형식
```

- 생성 객체를 다시 사용하기 위해 레퍼런스 변수를 활용할 수 있음

특정 대상(객체)를 가리키는 변수

```
클래스_타입 레퍼런스변수명 = new 클래스명();  
// 객체 생성 후, 해당 객체를 변수에 연결
```

I . 객체 생성하기

객체 생성 과정

- Cat 클래스를 객체로 만드는 예

② 레퍼런스변수의 생성

① 객체 생성

③ 생성 객체를 변수에 대입

Cat c = new Cat(); // Cat 객체 생성 후, 해당 객체를 변수 c에 연결

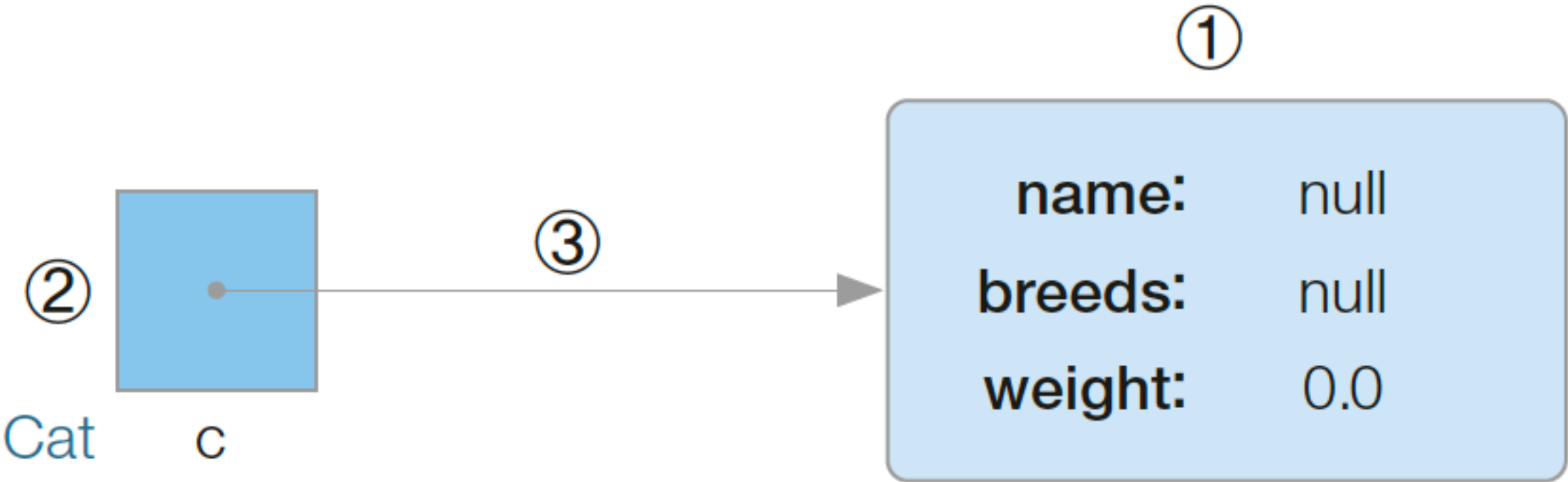


그림 7-7 Cat 객체의 생성 과정

I . 객체 생성하기

객체 생성 과정

- Cat 클래스를 객체로 만드는 예

② 레퍼런스변수의 생성

① 객체 생성 코드 실행

Cat c = new Cat(); // Cat 객체 생성 후, 해당 객체를 변수 c에 연결

③ 생성 객체를 변수에 대입



그림 7-7 Cat 객체의 생성 과정

I . 객체 생성하기

객체 생성 과정

- Cat 클래스를 객체로 만드는 예

② 레퍼런스변수의 생성

① 객체 생성 코드 실행

③ 생성 객체를 변수에 대입

Cat c = new Cat(); // Cat 객체 생성 후, 해당 객체를 변수 c에 연결



그림 7-7 Cat 객체의 생성 과정

I . 객체 생성하기

객체 생성 과정

- Cat 클래스를 객체로 만드는 예

② 레퍼런스변수의 생성

① 객체 생성 코드 실행

Cat c = new Cat(); // Cat 객체 생성 후, 해당 객체를 변수 c에 연결

③ 생성 객체를 변수에 대입

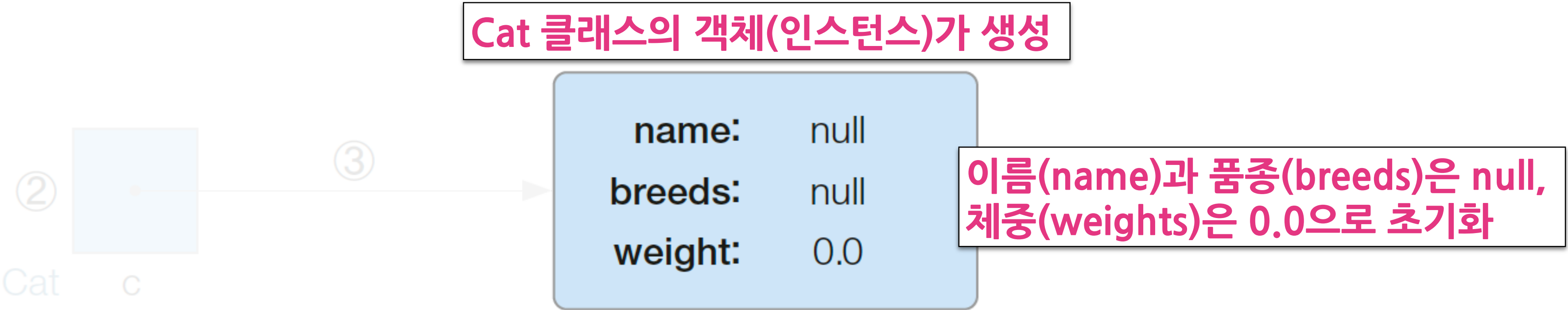


그림 7-7 Cat 객체의 생성 과정

I . 객체 생성하기

객체 생성 과정

- Cat 클래스를 객체로 만드는 예

② 레퍼런스변수의 생성

① 객체 생성 코드 실행

③ 생성 객체를 변수에 대입

Cat c = new Cat(); // Cat 객체 생성 후, 해당 객체를 변수 c에 연결

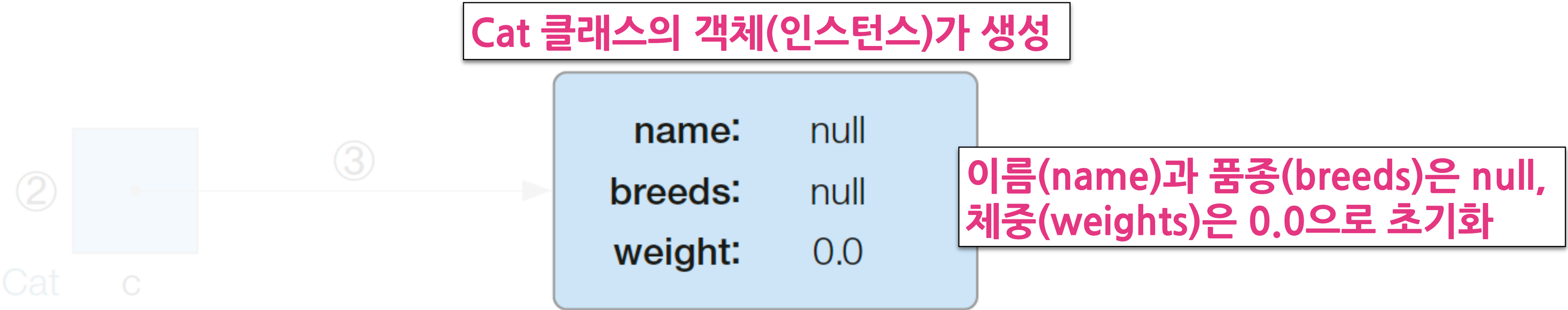


그림 7-7 Cat 객체의 생성 과정

! TIP

변수의 초기값은 타입에 따라 다름.

I . 객체 생성하기

객체 생성 과정

- Cat 클래스를 객체로 만드는 예

② Cat 타입의 레퍼런스변수 c 생성

```
Cat c = new Cat(); // Cat 객체 생성 후, 해당 객체를 변수 c에 연결
```

③ 생성 객체를 변수에 대입

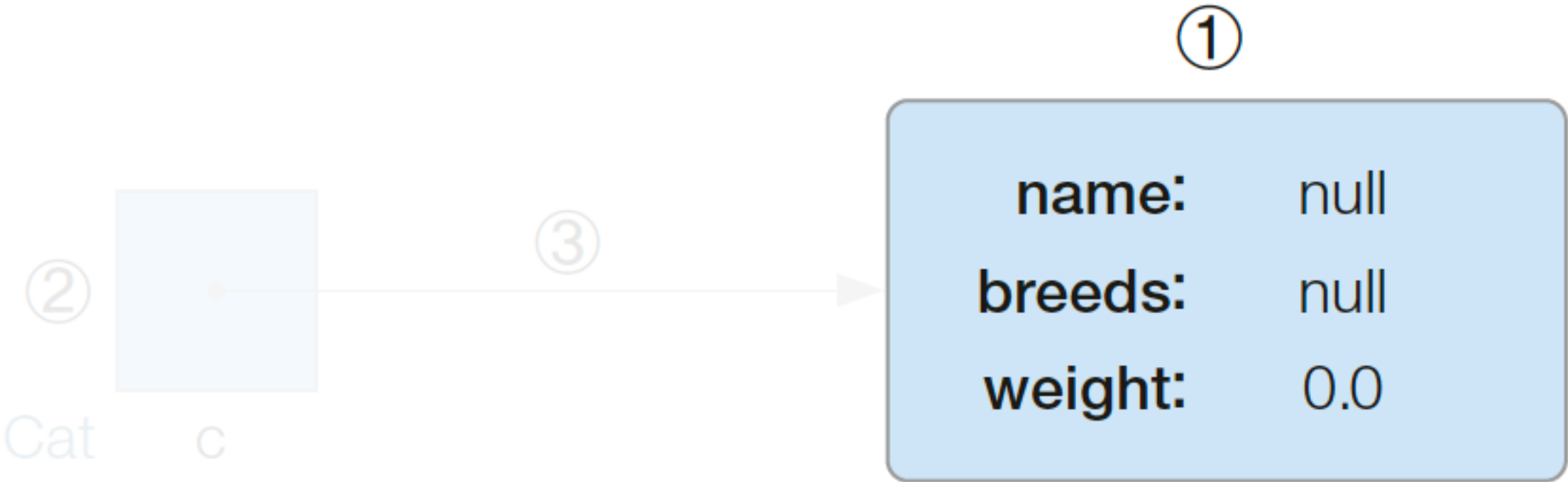


그림 7-7 Cat 객체의 생성 과정

I . 객체 생성하기

객체 생성 과정

- Cat 클래스를 객체로 만드는 예

② Cat 타입의 레퍼런스변수 c 생성

Cat c = new Cat(); // Cat 객체 생성 후, 해당 객체를 변수 c에 연결

③ 생성 객체를 변수에 대입

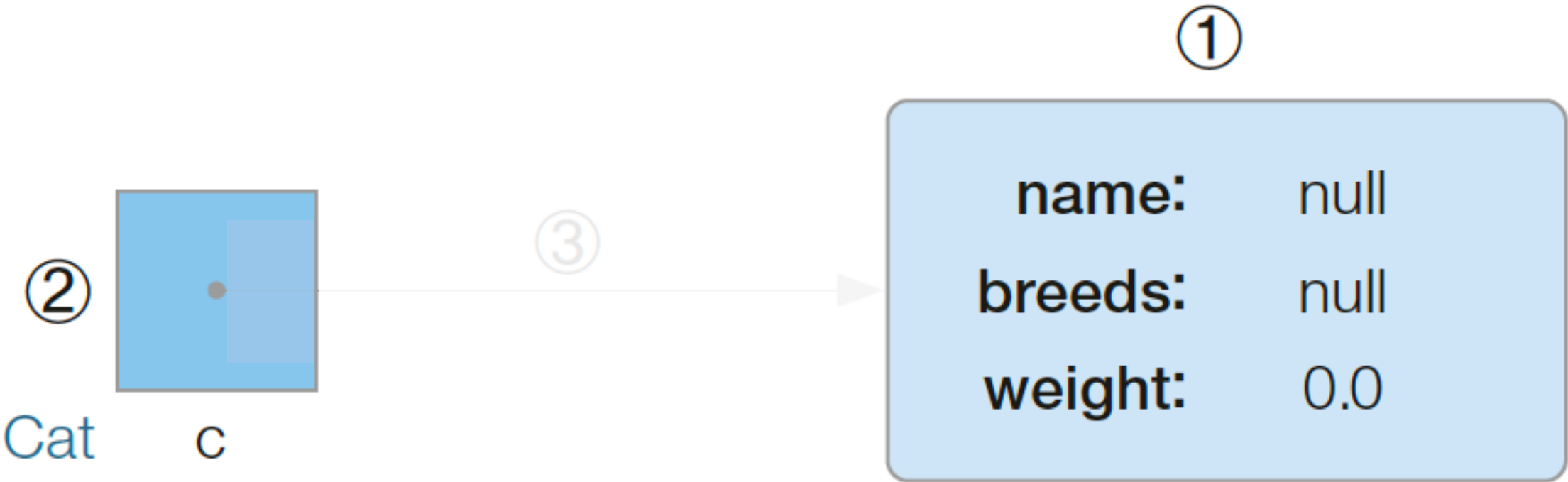


그림 7-7 Cat 객체의 생성 과정

I . 객체 생성하기

객체 생성 과정

- Cat 클래스를 객체로 만드는 예

② 레퍼런스변수의 생성

① 객체 생성

```
Cat c = new Cat(); // Cat 객체 생성 후, 해당 객체를 변수 c에 연결
```

③ 대입 연산(=)을 통해 c가 생성 객체를 가리킴

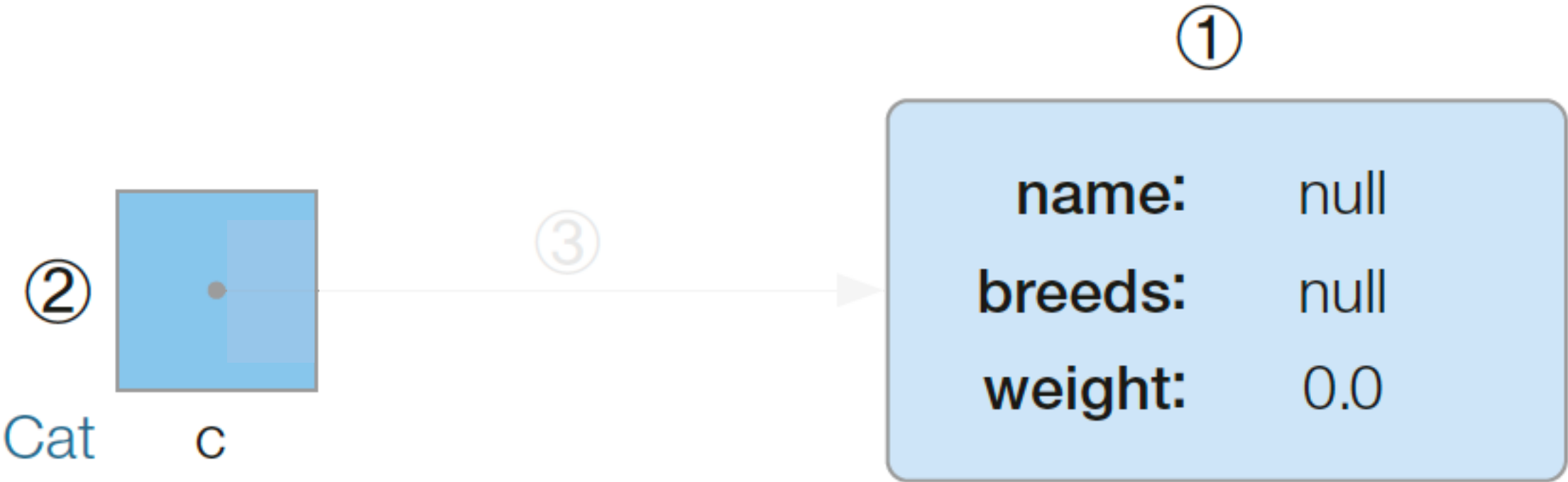


그림 7-7 Cat 객체의 생성 과정

I . 객체 생성하기

객체 생성 과정

- Cat 클래스를 객체로 만드는 예

② 레퍼런스변수의 생성

① 객체 생성

```
Cat c = new Cat(); // Cat 객체 생성 후, 해당 객체를 변수 c에 연결
```

③ 대입 연산(=)을 통해 c가 생성 객체를 가리킴

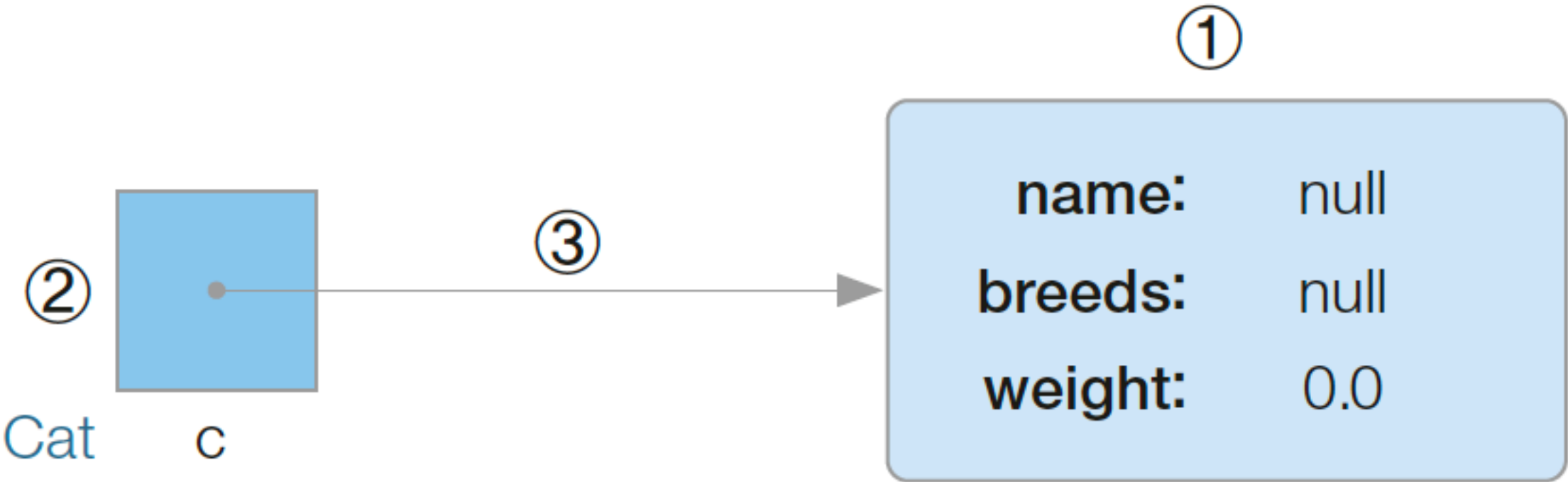


그림 7-7 Cat 객체의 생성 과정

II. 필드 사용하기

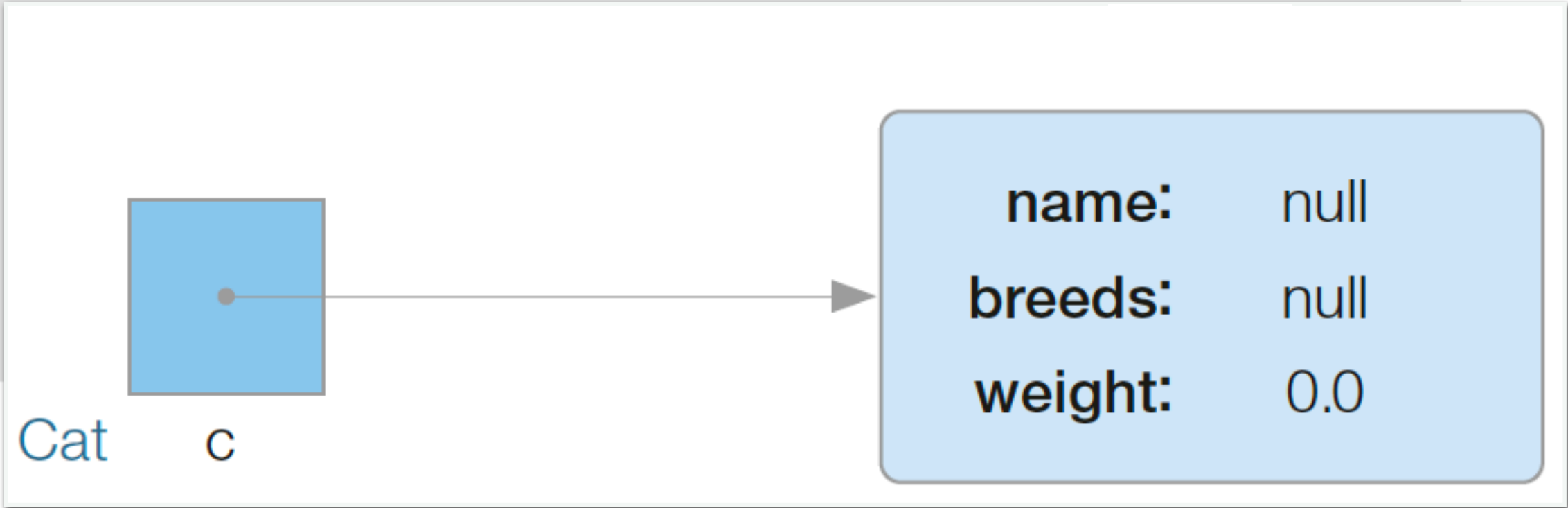
인스턴스변수

- 객체 속에 존재하는 필드를 **인스턴스변수**(instance variable)라 함
- Cat 객체의 인스턴스변수 name, breeds, weight를 출력하는 예

```
System.out.printf("이름: %s\n", c.name); ..... c(가 가리키는) 객체의 name
System.out.printf("품종: %s\n", c.breeds); ..... c 객체의 breeds
System.out.printf("체중: %.2fkg\n", c.weight); ..... c 객체의 weight
```

실행 결과

이름: null
품종: null
체중: 0.00kg



II. 필드 사용하기

닷 연산자

- 닷(.) 연산자는 객체 접근을 위한 연산자
- c.name은 다음의 두 단계로 동작

```
System.out.printf("이름: %s\n", c.name); ..... c(가 가리키는) 객체의 name
System.out.printf("품종: %s\n", c.breeds); ..... c 객체의 breeds
System.out.printf("체중: %.2fkg\n", c.weight); ..... c 객체의 weight
```

① 레퍼런스변수 c에 연결된 객체로 접근

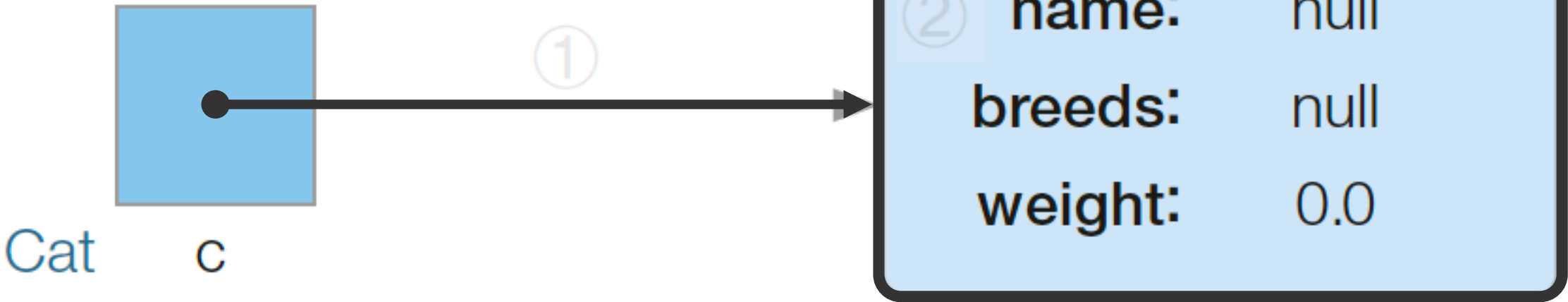


그림 7-8 인스턴스변수로의 접근

II. 필드 사용하기

닷 연산자

- 닷(.) 연산자는 객체 접근을 위한 연산자
- c.name은 다음의 두 단계로 동작

```
System.out.printf("이름: %s\n", c.name); ..... c(가 가리키는) 객체의 name
System.out.printf("품종: %s\n", c.breeds); ..... c 객체의 breeds
System.out.printf("체중: %.2fkg\n", c.weight); ..... c 객체의 weight
```

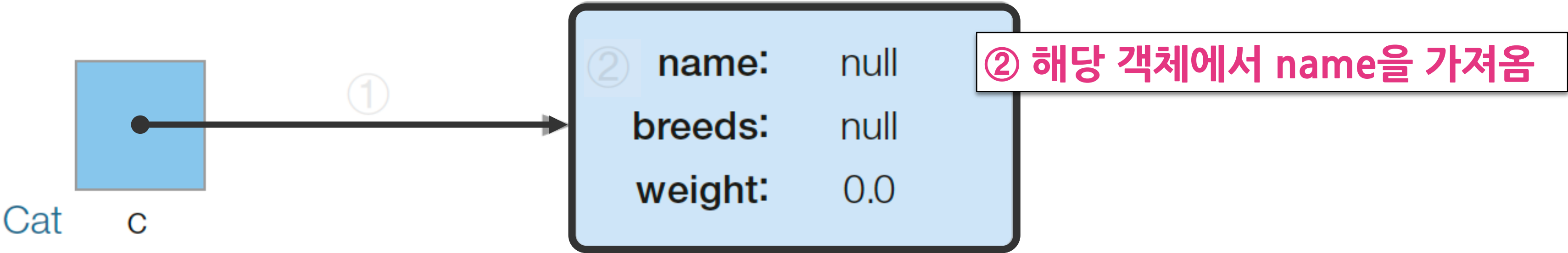


그림 7-8 인스턴스변수로의 접근

II. 필드 사용하기

닷 연산자

- 다음은 인스턴스변수를 변경하는 코드

```
c.name = "네로";           // 이름을 "네로"로 변경  
c.breeds = "페르시안";     // 품종을 "페르시안"으로 변경  
c.weight = 4.37;           // 체중을 4.37로 변경
```

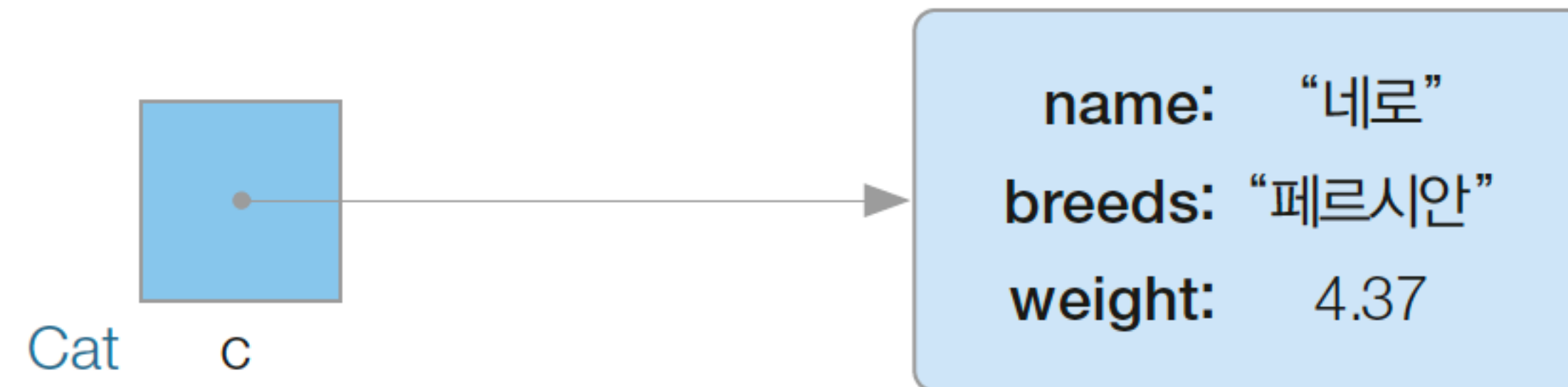


그림 7-9 변경된 인스턴스변수

실습 7-1

객체 생성과 인스턴스변수의 사용

ch07/ex7_1/Cat.java

```
01 public class Cat {
02     String name;    // 이름
03     String breeds;  // 품종
04     double weight;  // 체중
05
06     void claw() {
07         System.out.println("할퀴기!!");
08     }
09     void meow() {
10         System.out.println("야옹~");
11     }
12
13     public static void main(String[] args) {
14         // 객체 생성
15         Cat c = new Cat();
16
17         // 객체 상태(필드) 변경
18         c.name = "네로";
19         c.breeds = "페르시안";
20         c.weight = 4.37;
21
22         // 객체 상태(필드) 출력
23         System.out.printf("이름: %s\n", c.name);
24         System.out.printf("품종: %s\n", c.breeds);
25         System.out.printf("체중: %.2fkg\n", c.weight);
26     }
27 }
```

필드(인스턴스변수)

메소드(인스턴스 메소드)

메인 메소드

실습 코드

https://github.com/cloudstudying-kr/JavaPlz/blob/master/src/part2/ch07/ex7_1/Cat.java

실행 결과

이름: 네로

품종: 페르시안

체중: 4.37kg

III. 여러 객체 만들기

하나의 클래스로 여러 객체를 만들 수 있음

- 강아지(Dog) 클래스 다이어그램을 설계 및 구현해 이를 확인해보자!

클래스 설계와 구현

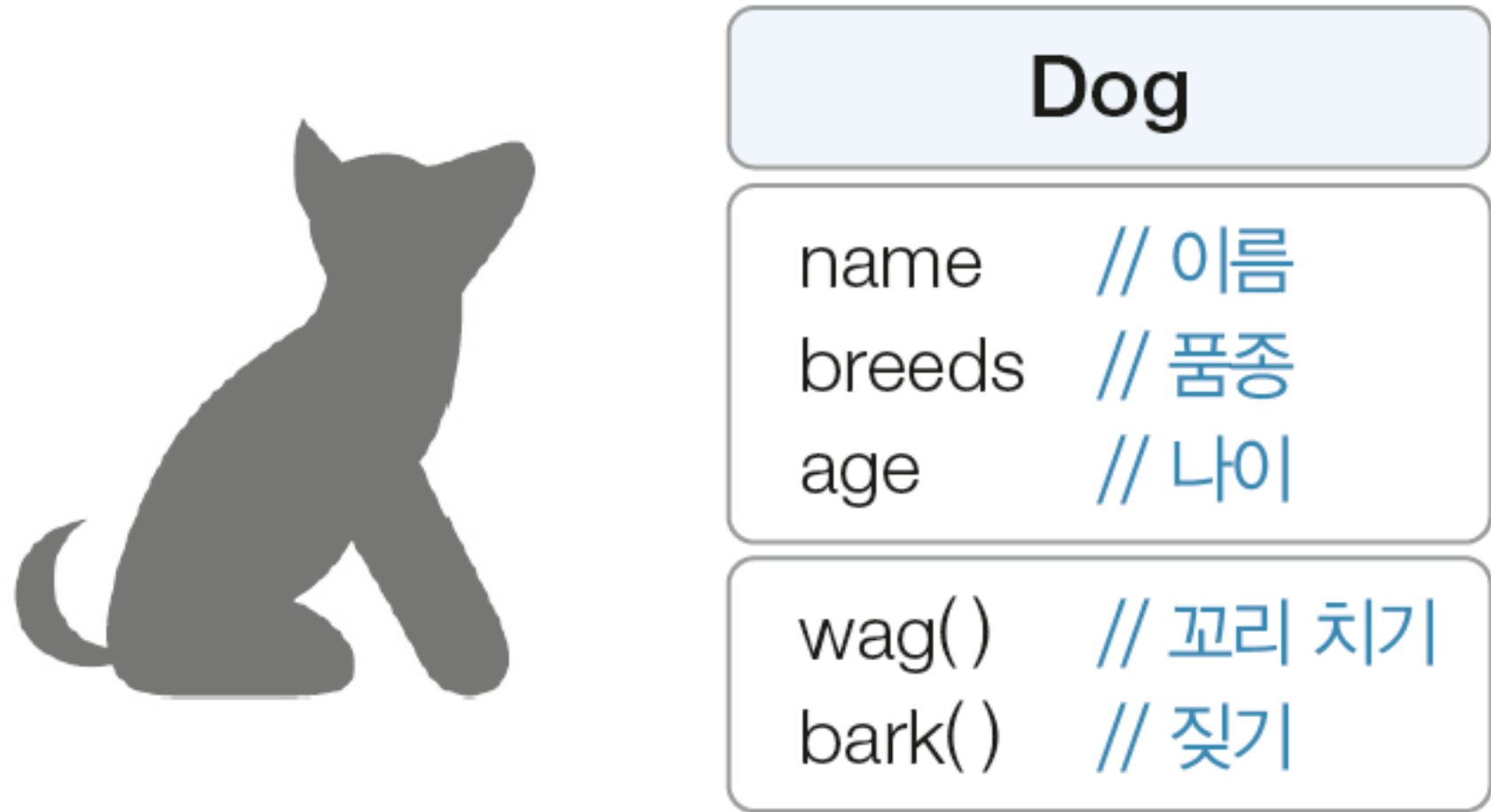


그림 7-10 Dog 클래스의 필드와 메소드 설계

```
public class Dog {
    String name; // 이름
    String breeds; // 품종
    int age; // 나이

    void wag() {
        System.out.println("살랑살랑~");
    }
    void bark() {
        System.out.println("멍멍!");
    }
}
```

III. 여러 객체 만들기

객체 생성과 필드의 사용

- 두 강아지 객체를 레퍼런스변수 d1과 d2에 연결하는 코드 예

```
Dog d1 = new Dog(); // Dog 객체 생성 후, d1 변수에 연결  
Dog d2 = new Dog(); // Dog 객체 생성 후, d2 변수에 연결
```



그림 7-11 두 객체 d1과 d2의 초기 상태

III. 여러 객체 만들기

객체 생성과 필드의 사용

- 두 강아지의 상태 변경 예(레퍼런스변수와 닷 연산자를 사용)

```
// d1 객체의 상태 변경
```

```
d1.name = "망고";
```

```
d1.breeds = "골든리트리버";
```

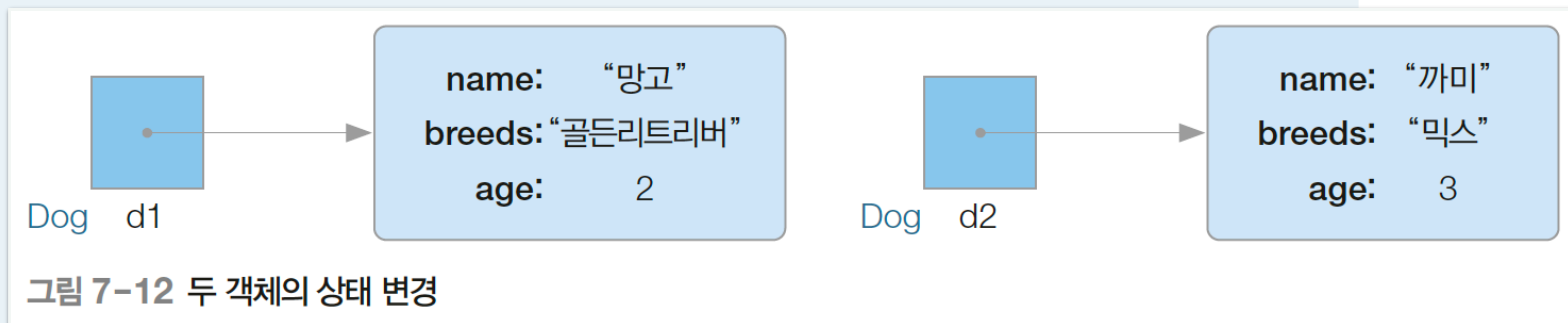
```
d1.age = 2;
```

```
// d2 객체의 상태 변경
```

```
d2.name = "까미";
```

```
d2.breeds = "믹스";
```

```
d2.age = 3;
```



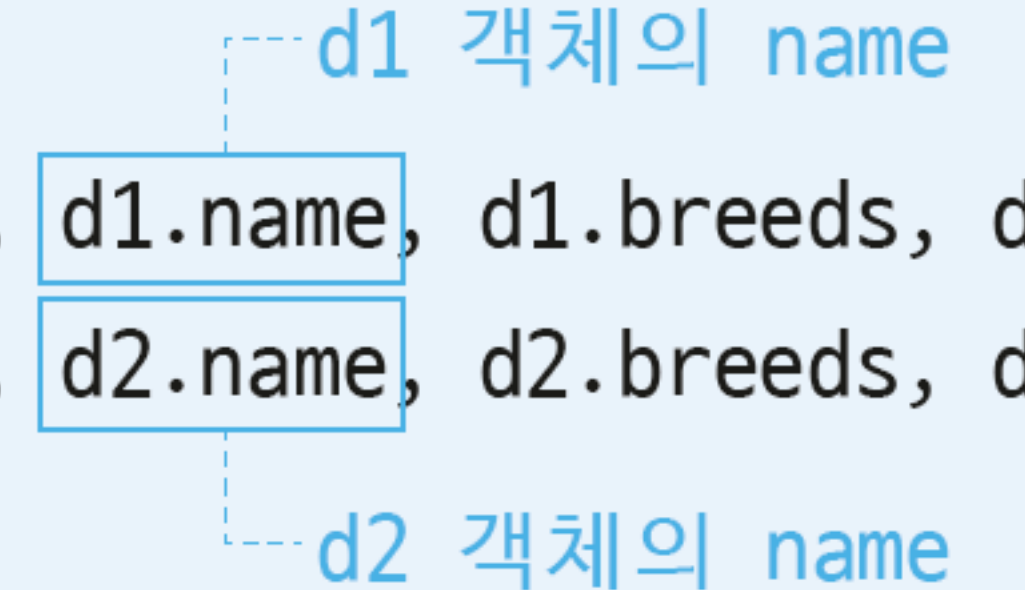
III. 여러 객체 만들기

객체 생성과 필드의 사용

- 각 객체의 상태가 잘 바뀌었는지 출력을 통해 확인

// 결과 출력

```
System.out.printf("d1 => {%s, %s, %d세}\n", d1.name, d1.breeds, d1.age);  
System.out.printf("d2 => {%s, %s, %d세}\n", d2.name, d2.breeds, d2.age);
```



실행 결과

d1 => {망고, 골든리트리버, 2세}

d2 => {까미, 믹스, 3세}

IV. 인스턴스 메소드 호출하기

인스턴스 메소드(instance method)

- 특정 객체가 주체가 되어 수행하는 동작을 뜻함
- 다음은 두 강아지 객체의 인스턴스 메소드의 호출 예

```
d1.wag();    // 꼬리 치기
```

```
d2.bark();   // 짖기
```

IV. 인스턴스 메소드 호출하기

인스턴스 메소드(instance method)

- 특정 객체가 주체가 되어 수행하는 동작을 뜻함
- 다음은 두 강아지 객체의 인스턴스 메소드의 호출 예

```
d1.wag();    // 꼬리 치기
d2.bark();   // 짖기
```

d1 객체가 wag() 메소드를 수행

IV. 인스턴스 메소드 호출하기

인스턴스 메소드(instance method)

- 특정 객체가 주체가 되어 수행하는 동작을 뜻함
- 다음은 두 강아지 객체의 인스턴스 메소드의 호출 예

```
d1.wag();    // 꼬리 치기
d2.bark();   // 짖기
```

d1 객체가 wag() 메소드를 수행

```
public class Dog {
    String name;    // 이름
    String breeds;  // 품종
    int age;        // 나이

    void wag() {
        System.out.println("살랑살랑~");
    }

    void bark() {
        System.out.println("멍멍!");
    }
}
```

IV. 인스턴스 메소드 호출하기

인스턴스 메소드(instance method)

- 특정 객체가 주체가 되어 수행하는 동작을 뜻함
- 다음은 두 강아지 객체의 인스턴스 메소드의 호출 예

```
d1.wag();    // 꼬리 치기
d2.bark();    // 짖기
```

d1 객체가 wag() 메소드를 수행

- 인스턴스 메소드의 호출 결과

살랑살랑~
멍멍!

```
public class Dog {
    String name;    // 이름
    String breeds;  // 품종
    int age;        // 나이

    void wag() {
        System.out.println("살랑살랑~");
    }

    void bark() {
        System.out.println("멍멍!");
    }
}
```

IV. 인스턴스 메소드 호출하기

인스턴스 메소드(instance method)

- 특정 객체가 주체가 되어 수행하는 동작을 뜻함
- 다음은 두 강아지 객체의 인스턴스 메소드의 호출 예

```
d1.wag();    // 꼬리 치기
```

```
d2.bark();   // 짖기
```

d2 객체가 bark() 메소드를 수행

- 인스턴스 메소드의 호출 결과

살랑살랑~

멍멍!

```
public class Dog {  
    String name;    // 이름  
    String breeds; // 품종  
    int age;        // 나이  
  
    void wag() {  
        System.out.println("살랑살랑~");  
    }  
    void bark() {  
        System.out.println("멍멍!");  
    }  
}
```

IV. 인스턴스 메소드 호출하기

인스턴스 메소드(instance method)

- 특정 객체가 주체가 되어 수행하는 동작을 뜻함
- 다음은 두 강아지 객체의 인스턴스 메소드의 호출 예

```
d1.wag();    // 꼬리 치기  
d2.bark();   // 짖기
```

- 인스턴스 메소드의 호출 결과

```
살랑살랑~  
멍멍!
```

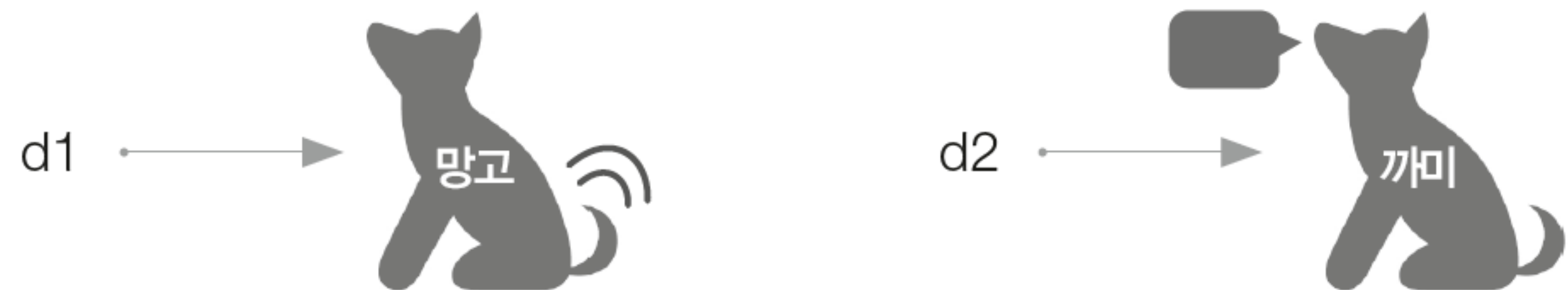


그림 7-13 인스턴스 메소드 호출을 통한 객체의 동작 수행

IV. 인스턴스 메소드 호출하기

인스턴스 메소드

- 누가 꼬리를 쳤는지, 누가 짖었는지를 다음과 같이 구분하려면?

[망고] 사랑사랑~

[까미] 멍멍!

=> 메소드 내부에서 필드를 사용

- 필드의 활동영역이 클래스 전체이기 때문에 메소드 내부에서 사용 가능

IV. 인스턴스 메소드 호출하기

인스턴스 메소드

- 메소드에서 필드값을 사용한 예

```
public class Dog {  
    String name;    // 이름  
    String breeds;  // 품종  
    int age;        // 나이  
  
    void wag() {  
        System.out.printf("[%s] 살랑살랑~\n", name); 필드 name을 사용  
    }  
    void bark() {  
        System.out.printf("[%s] 멍멍!\n", name); 필드 name을 사용  
    }  
}
```

IV. 인스턴스 메소드 호출하기

인스턴스 메소드

- 메소드에서 필드값을 사용한 예

```
public class Dog {  
    String name;      // 이름  
    String breeds;    // 품종  
    int age;          // 나이  
  
    void wag() {  
        System.out.printf("[%s] 살랑살랑~\n", name); 필드 name을 사용  
    }  
    void bark() {  
        System.out.printf("[%s] 멍멍!\n", name); 필드 name을 사용  
    }  
}
```

필드의 활동 영역
(클래스 스코프)

IV. 인스턴스 메소드 호출하기

인스턴스 메소드

- 메소드에서 필드값을 사용한 예

```
public class Dog {  
    String name;    // 이름  
    String breeds; // 품종  
    int age;        // 나이
```

```
    void wag() {  
        System.out.printf("[%s] 살랑살랑~\n", name);  
    }
```

```
    void bark() {  
        System.out.printf("[%s] 멍멍!\n", name);  
    }
```

```
}
```


필드의 활동 영역
(클래스 스코프)

입력변수와 지역변수의 활동 영역
(메소드 스코프)

IV. 인스턴스 메소드 호출하기

인스턴스 메소드

- 메소드에서 필드값을 사용한 예

 **TIP**

필드의 스코프는 클래스 전체이지만, 입력변수와 지역변수의 스코프는 자신이 선언된 메소드 내부.

```
public class Dog {
    String name;      // 이름
    String breeds;    // 품종
    int age;          // 나이

    void wag() {
        System.out.printf("[%s] 살랑살랑~\n", name);
    }

    void bark() {
        System.out.printf("[%s] 멍멍!\n", name);
    }
}
```

필드의 활동 영역
(클래스 스코프)

입력변수와 지역변수의 활동 영역
(메소드 스코프)

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog { ..... 클래스명
02     String name;      // 이름
03     String breeds;    // 품종 ..... 필드(인스턴스변수)
04     int age;          // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     } ..... 이름이 같은 두 메소드
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     } ..... 이름이 같은 두 메소드
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 } ..... 메소드(인스턴스 메소드)
```

실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}
[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)

```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog(); ..... 객체 생성
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2; ..... 객체 상태 변경
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n", ..... 객체
30         d1.name, d1.breeds, d1.age); ..... 정보 출력
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32         d2.name, d2.breeds, d2.age);
33
34     d1.wag();    // 꼬리 치기
35     d2.bark();   // 짖기 ..... 인스턴스 메소드 호출
36     d1.bark(3);  // 짖기 3회
37 }
38 }
```

 **실습 코드**
https://github.com/cloudstudying-kr/JavaPlz/blob/master/src/part2/ch07/ex7_2/Dog.java

실습 7-2

객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용

ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name;    // 이름
03     String breeds;  // 품종
04     int age;        // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```

클래스명

필드(인스턴스 변수)

이름이 같은 두 메소드

이름이 같은 두 메소드

메소드(인스턴스 메소드)

해당 프로그램은 강아지(Dog) 클래스를 활용한 것으로,
02~04행의 필드와
06~15행의 메소드를 사용

실행 흐름은
17~37행의 메인 메소드를 통해 동작

```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag();    // 꼬리 치기
35     d2.bark();   // 짖기
36     d1.bark(3);  // 짖기 3회
37 }
38 }
```

객체 생성

객체 상태 변경

객체 정보 출력

인스턴스 메소드 호출

메인 메소드

실습 7-2

객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용

ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name;    // 이름
03     String breeds;  // 품종
04     int age;        // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```

클래스명

필드(인스턴스 변수)

이름이 같은 두 메소드

이름이 같은 두 메소드

메소드(인스턴스 메소드)

```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag();    // 꼬리 치기
35     d2.bark();   // 짖기
36     d1.bark(3);  // 짖기 3회
37 }
38 }
```

두 Dog 객체를 생성

객체 상태 변경

객체 정보 출력

인스턴스 메소드 호출

메인 메소드

실습 7-2 객체 생성과 인스턴스 변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name;    // 이름
03     String breeds;  // 품종
04     int age;        // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```

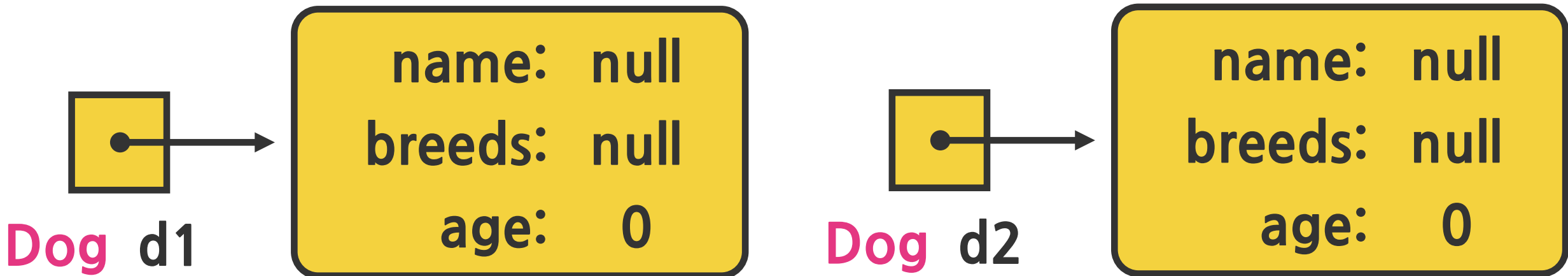
클래스명

필드(인스턴스 변수)

이름이 같은 두 메소드

이름이 같은 두 메소드

메소드(인스턴스 메소드)



```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag();    // 꼬리 치기
35     d2.bark();   // 짖기
36     d1.bark(3);  // 짖기 3회
37
38 }
```

두 Dog 객체를 생성

객체 상태 변경

객체 정보 출력

인스턴스 메소드 호출

메인 메소드

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name; // 이름
03     String breeds; // 품종
04     int age; // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```

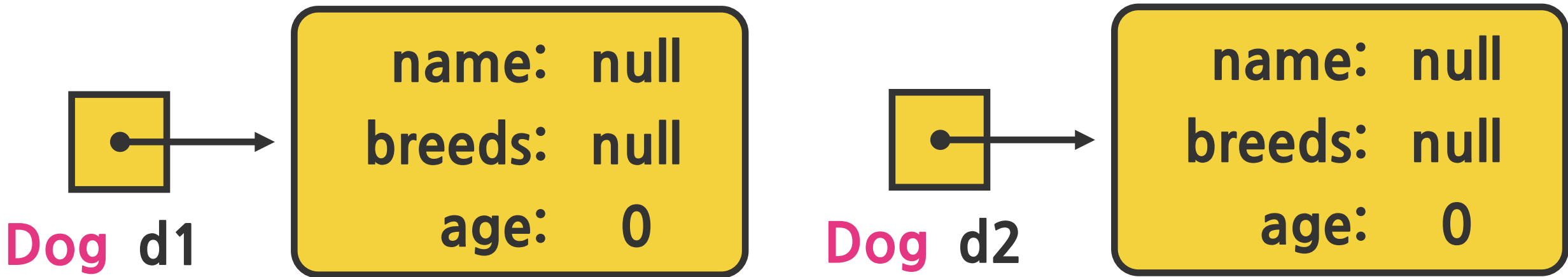
클래스명

필드(인스턴스변수)

이름이 같은 두 메소드

이름이 같은 두 메소드

메소드(인스턴스 메소드)



```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
38 }
```

객체 생성

객체 정보 출력

인스턴스 메소드 호출

메인 메소드

두 객체의 필드(인스턴스 변수)를 변경

실습 7-2 객체 생성과 인스턴스 변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name;    // 이름
03     String breeds;  // 품종
04     int age;        // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```

클래스명

필드(인스턴스 변수)

이름이 같은 두 메소드

이름이 같은 두 메소드

메소드(인스턴스 메소드)



```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag();    // 꼬리 치기
35     d2.bark();   // 짖기
36     d1.bark(3);  // 짖기 3회
37 }
```

객체 생성

객체 정보 출력

인스턴스 메소드 호출

메인 메소드

두 객체의 필드(인스턴스 변수)를 변경

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name; // 이름
03     String breeds; // 품종
04     int age; // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```



```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
```

두 Dog 객체의
정보 출력

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name;    // 이름
03     String breeds;  // 품종
04     int age;        // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```

클래스명

필드(인스턴스변수)

이름이 같은 두 메소드

이름이 같은 두 메소드

메소드(인스턴스 메소드)



```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag();    // 꼬리 치기
35     d2.bark();   // 짖기
36     d1.bark(3);  // 짖기 3회
37 }
```

객체 생성

객체 상태 변경

인스턴스 메소드 호출

두 Dog 객체의
정보 출력

실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}

[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name; // 이름
03     String breeds; // 품종
04     int age; // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```



```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30         d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32         d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
```

인스턴스 메소드 호출

실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}

[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name; // 이름
03     String breeds; // 품종
04     int age; // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```



```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
```

인스턴스 메소드 호출

실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}

[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name; // 이름
03     String breeds; // 품종
04     int age; // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```



```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30         d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32         d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
```

인스턴스 메소드 호출

실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}
[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name; // 이름
03     String breeds; // 품종
04     int age; // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```



```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
```

인스턴스 메소드 호출

실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}
[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog { ..... 클래스명
02     String name;      // 이름
03     String breeds;    // 품종 ..... 필드(인스턴스변수)
04     int age;          // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     } ..... 이름이 같은 두 메소드
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     } ..... 이름이 같은 두 메소드
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 } ..... 메소드(인스턴스 메소드)
```

```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog(); ..... 객체 생성
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2; ..... 객체 상태 변경
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n", ..... 객체
30         d1.name, d1.breeds, d1.age); ..... 정보 출력
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32         d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
```

인스턴스 메소드 호출
실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}
[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)



실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog { ..... 클래스명
02     String name;      // 이름
03     String breeds;    // 품종 ..... 필드(인스턴스변수)
04     int age;          // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     } ..... 이름이 같은 두 메소드
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     } ..... 이름이 같은 두 메소드
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 } ..... 메소드(인스턴스 메소드)
```

```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog(); ..... 객체 생성
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2; ..... 객체 상태 변경
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%, %s, %d세}\n", ..... 객체
30         d1.name, d1.breeds, d1.age); ..... 정보 출력
31     System.out.printf("d2 => {%, %s, %d세}\n",
32         d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
```



인스턴스 메소드 호출 실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}
[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name; // 이름
03     String breeds; // 품종
04     int age; // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```

클래스명

필드(인스턴스변수)

이름이 같은 두 메소드

이름이 같은 두 메소드

메소드(인스턴스 메소드)

```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
```

객체 생성

객체 상태 변경

객체 정보 출력

인스턴스 메소드 호출



실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}
[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)

실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog {
02     String name; // 이름
03     String breeds; // 품종
04     int age; // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     }
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     }
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16 }
```

클래스명

필드(인스턴스변수)

이름이 같은 두 메소드

이름이 같은 두 메소드

메소드(인스턴스 메소드)

```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog();
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3;
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n",
32                       d2.name, d2.breeds, d2.age);
33
34     d1.wag(); // 꼬리 치기
35     d2.bark(); // 짖기
36     d1.bark(3); // 짖기 3회
37 }
```

객체 생성

객체 상태 변경

객체 정보 출력

인스턴스 메소드 호출
실행 결과

d1 => {망고, 골든리트리버, 2세}
d2 => {까미, 믹스, 3세}
[망고] 살랑살랑~
[까미] 멍멍!
[망고] 컹컹!(x3)



실습 7-2 객체 생성과 인스턴스변수 및 인스턴스 메소드의 사용 ch07/ex7_2/Dog.java

```
01 public class Dog { ..... 클래스명
02     String name;      // 이름
03     String breeds;    // 품종 ..... 필드(인스턴스변수)
04     int age;          // 나이
05
06     void wag() {
07         System.out.printf("[%s] 살랑살랑~\n", name);
08     } ..... 이름이 같은 두 메소드
09     void bark() {
10         System.out.printf("[%s] 멍멍!\n", name);
11     } ..... 이름이 같은 두 메소드
12     void bark(int times) {
13         String sound = "컹컹!";
14         System.out.printf("[%s] %s(x%d)\n", name, sound, times);
15     }
16
```

메소드 오버로딩(method overloading)
이름이 같은 메소드를
입력변수의 개수 또는 타입으로 구분하는 기법

```
17 public static void main(String[] args) {
18
19     Dog d1 = new Dog();
20     Dog d2 = new Dog(); ..... 객체 생성
21
22     d1.name = "망고";
23     d1.breeds = "골든리트리버";
24     d1.age = 2;
25     d2.name = "까미";
26     d2.breeds = "믹스";
27     d2.age = 3; ..... 객체 상태 변경
28
29     System.out.printf("d1 => {%s, %s, %d세}\n",
30                       d1.name, d1.breeds, d1.age);
31     System.out.printf("d2 => {%s, %s, %d세}\n", ..... 객체
32                       d2.name, d2.breeds, d2.age); ..... 정보 출력
33
34     d1.wag();      // 꼬리 치기
35     d2.bark();     // 짖기 ..... 인스턴스 메소드 호출
36     d1.bark(3);    // 짖기 3회
37
38 }
```

..... 메인 메소드

Quiz

04 다음 문장 중 옳은 것을 모두 고르시오.

- ① new 키워드는 객체 생성을 위해 사용된다.
- ② 레퍼런스변수와 닷 연산자는 객체의 필드와 메소드 접근에 사용된다.
- ③ 객체 속에 존재하는 필드를 인스턴스변수라고 한다.
- ④ 특정 객체가 주체가 되어 수행하는 동작을 인스턴스 메소드라고 한다.

Thank You!