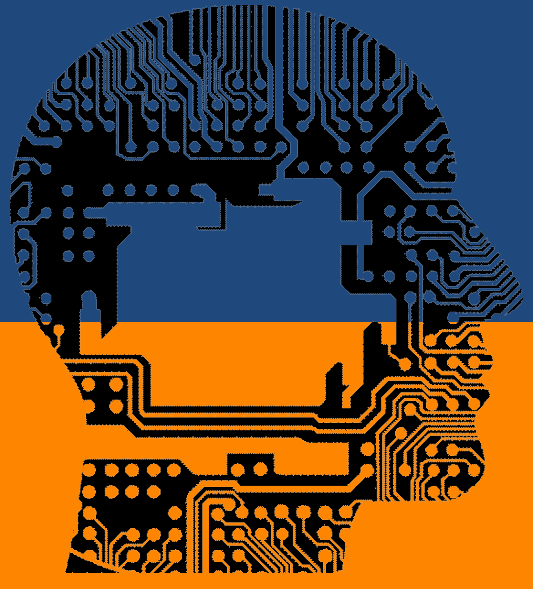
한국IT진흥부설 정보보호교육학원 아이섹

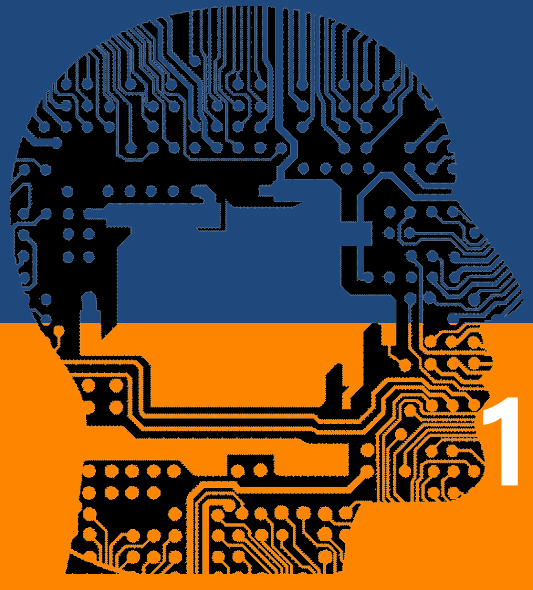


자바 프로그래밍 기초

객체지향 프로그래밍

By SoonGu Hong(Kokono)

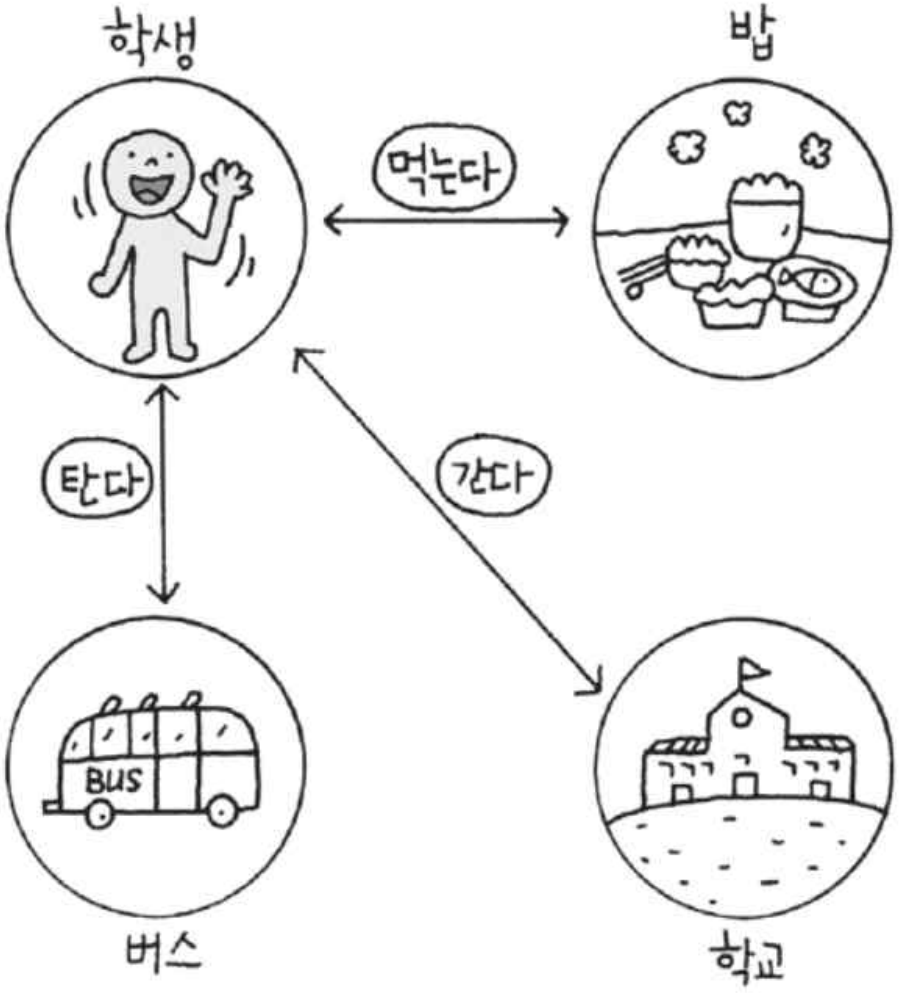
한국IT진흥부설
정보보호교육학원 아이섹



자바 프로그래밍 기초

1. 객체지향 프로그래밍이란

1-1. 객체지향 프로그래밍(Object Oriented Programming)



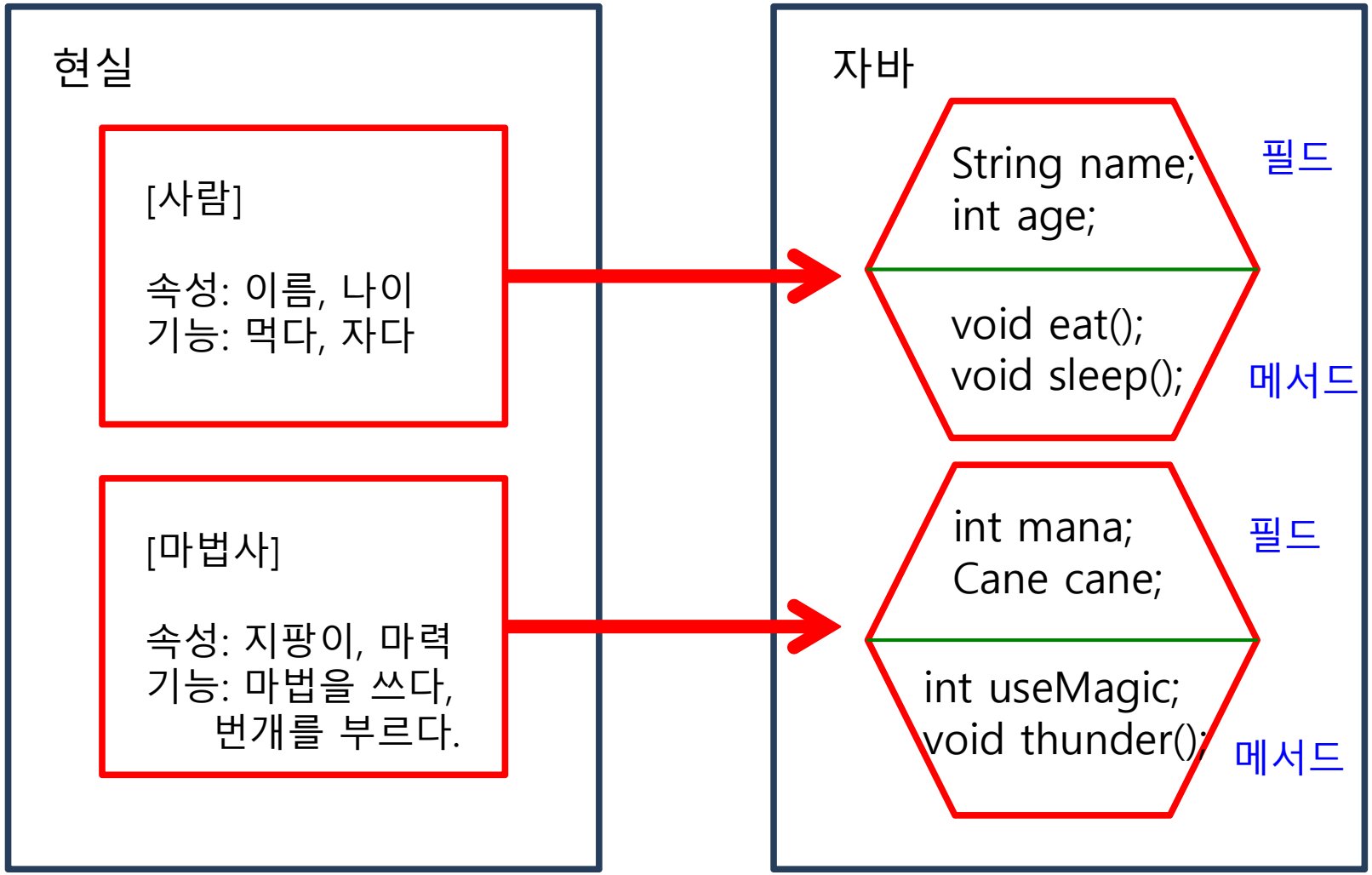
- 객체지향 프로그래밍이란 모든 데이터들을 객체로 표현하고 객체들의 상호작용(책임, 협력, 위임) 등을 프로그램으로 표현하는 프로그래밍 기법입니다.

- 출처: https://m.post.naver.com/viewer/postView.nhn?volumeNo=16885254&memberNo=38386150

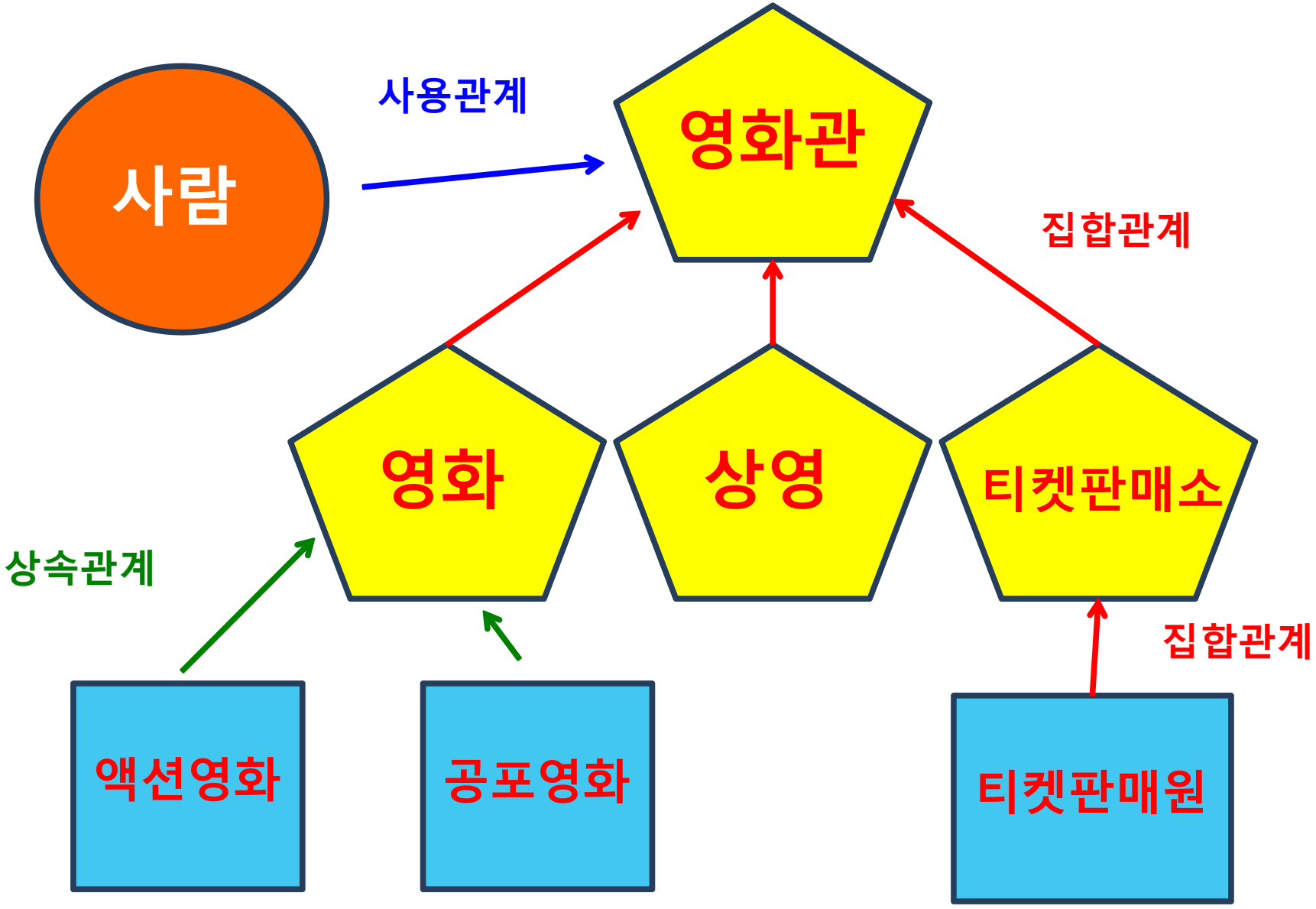
1-2-1. 객체란?

- 객체는 물리적으로 존재하거나 추상적으로 생각할 수 있는 것들을 말합니다.
- 물리적으로 눈에 보이는 펜, 자동차, 사람과 같은 것과 추상적인 주문, 강의 등과 같은 것도 모두 객체가 될 수 있습니다.

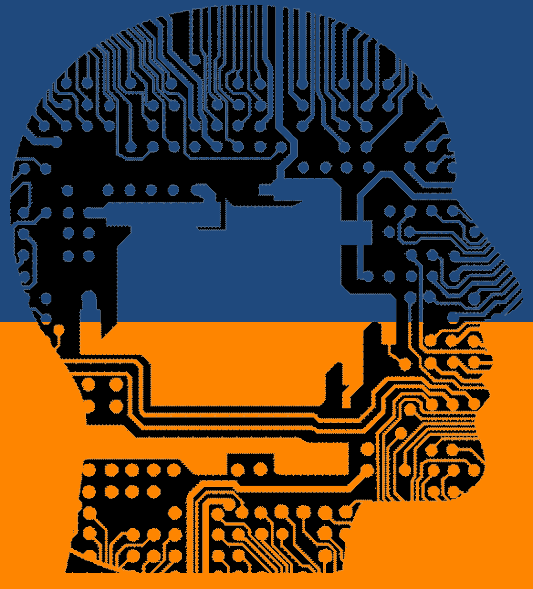
1-2-2. 현실의 객체와 프로그래밍의 객체



1-3. 객체간의 관계



한국IT진흥부설
정보보호교육학원 아이섹



자바 프로그래밍 기초

2. 객체의 기능과 속성

2-1. 객체의 속성과 기능

- 속성이란 객체가 가진 정보들을 의미합니다.

* 강아지의 속성
- 견종, 나이, 이름, 키, 몸무게, 예방접종 여부 등

- 기능이란 객체가 가진 행위들을 의미합니다.

* 강아지의 기능
- 짖다, 놀다, 사료를 먹다, 꼬리를 흔들다 등

2-2. 객체의 속성과 기능을 코드로 표현

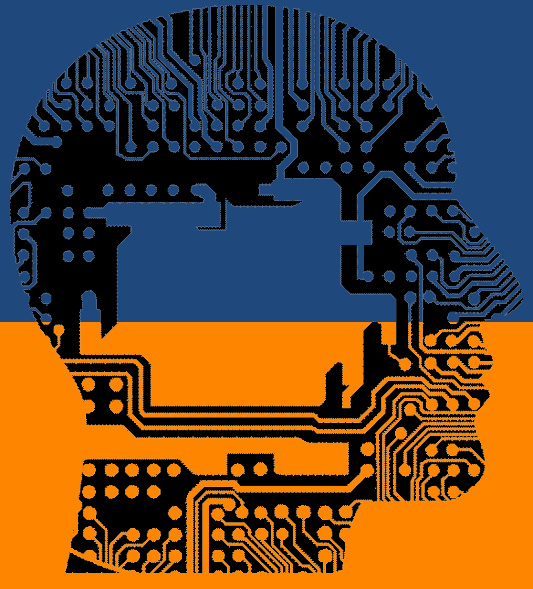
```
class Dog {
    String name;
    int age;
    double height;
    double weight;

    void bite();
    void eat();
    void playWithChild();
}
```

객체의 형태는 **클래스**라는 틀 안에 작성한다!

객체의 속성은 변수형태로 표현하고 필드라 부른다!

객체의 기능은 함수형태로 표현하고 메서드라고 부른다!

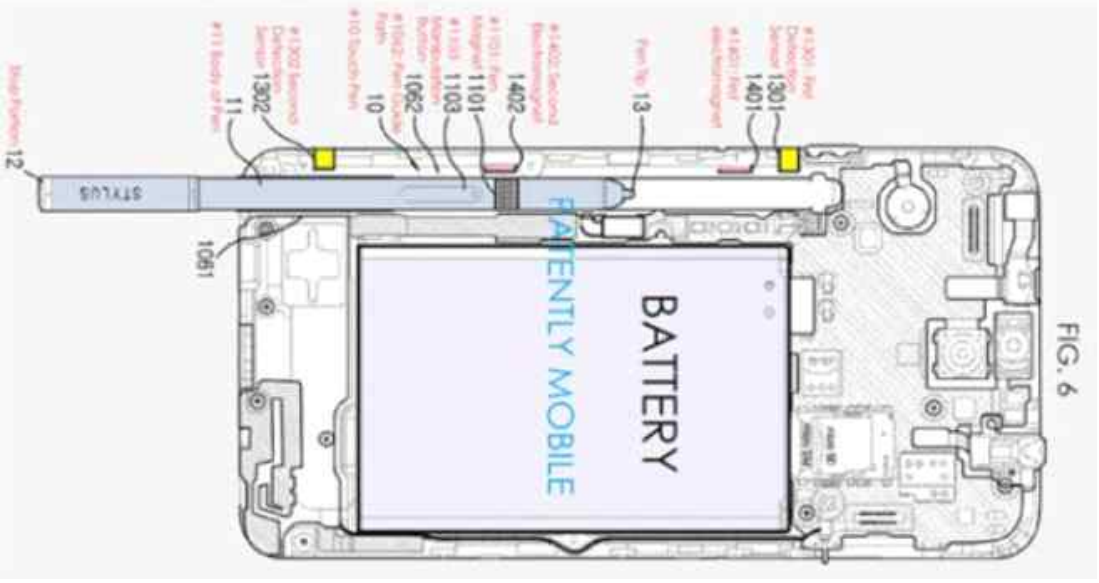
한국IT진흥부설
정보보호교육학원 아이섹



자바 프로그래밍 기초

3. 클래스

3-1. 클래스란?



- 현실에서도 객체는 갑자기 어디선가 나타나는게 아니라 설계도를 바탕으로 만들어집니다.
- 클래스란 바로 프로그래밍에서 객체를 만들기 위한 설계도입니다.
- 클래스로부터 만들어진 객체를 해당 클래스의 **인스턴스(instance)**라고 부릅니다.
- 하나의 클래스로 여러 개의 인스턴스를 만들 수 있습니다.

3-2. 클래스 정의하기

```
<modifiers> class <ClassName> {
    ClassBody;
    //attributes
    //constructors
    //methods
}
```

클래스 구성요소
1. 필드
2. 생성자
3. 메서드

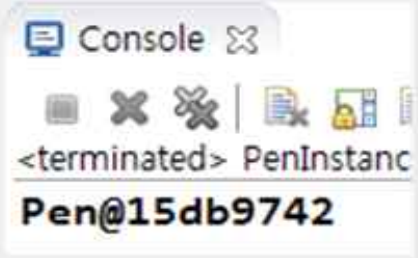
```
ClassName instanceName = new Constructor();
```

Pen.java

```
1:  public class Pen {
2:      //클래스 안의 코드 작성
3:  }
```

PenInstanceExample.java

```
1:  public class PenInstanceExample {
2:      public static void main(String[] args) {
3:          Pen redPen = new Pen();
4:          System.out.println(redPen);
5:      }
6:  }
```



3-3. 필드(field)

- 필드란 객체의 고유 속성데이터, 객체의 상태데이터, 객체의 부품정보를 저장하는 곳입니다.

ex) 자동차 객체의 필드가 될 수 있는 데이터
  a. [고유 데이터] 제작회사, 모델명, 색상, 최고속도 등
  b. [상태 데이터] 현재 속도, 엔진 회전 수, 현재 연료량 등
  c. [부품 데이터] 엔진, 타이어, 핸들, 에어컨 등

- 필드 선언 위치는 클래스 블록 안에서만 가능하며, 메서드나 생성자 블록에서는 불가능합니다.

- 생성자나 메서드안에 선언된 변수는 필드가 아닌 지역변수라고 부릅니다.

3-4-1. 생성자(constructor)

- 생성자란 new 연산자와 같이 사용되어 클래스로부터 객체를 생성할 때 호출되어 객체의 초기화를 담당합니다.

- 객체 초기화란 필드값을 세팅하거나 메서드를 호출하여 객체를 사용할 준비를 하는 것입니다.

- 생성자를 호출하지 않으면 객체를 생성할 수 없으며 생성자가 정상 호출되면 객체가 Heap메모리에 올라가며 객체가 생성되며 그 결과로 객체의 메모리 주소값이 리턴됩니다.

- 생성자는 여러 개 선언할 수 있으며, 하나도 선언하지 않으면 기본생성자(default constructor)가 자동 선언됩니다.

3-4-2. 생성자 선언

```
Phone() {


}
```

```
Phone(String pModel, String pColor, int pPrice) {
    model = pModel;
    color = pColor;
    price = pPrice;
}
```

- 생성자는 클래스 블록 안에 선언하며, 이름은 필드나 메서드처럼 사용자 정의가 아닌 **클래스이름과 반드시 대/소문자까지 동일**하게 작성해야 합니다.

- 생성자는 리턴을 할 수 없으며 따라서 리턴타입도 존재하지 않습니다.

- 매개변수가 없는 생성자를 기본 생성자라고 부릅니다.

3-5. 메서드(method)

- 메서드는 객체의 동작에 해당하는 블록을 말합니다.

- 메서드는 필드를 읽고 수정하는 역할도 하지만, 다른 객체를 생성해서 다양한 기능을 수행하기도 합니다.

- 메서드는 객체간의 메시징을 담당하며 객체 상호간 데이터 전달의 수단으로 사용됩니다.

- 8강에서 배운 메서드의 내용에서 static은 제거하고 선언해야 하며 객체 생성 없이는 다른 클래스에서 호출할 수 없습니다.

(이런 메서드를 인스턴스(instance) 메서드라고 부르며, 정적(static)메서드와의 차이는 뒷 부분에 설명합니다.)

감사합니다
THANK YOU