



# 2장. 클 래 스



# 클래스 개념과 특성



# 개념

## \* 클래스는 명세 장치

- \* 여러 유사 객체들이 공통적으로 갖는 속성이나 행위를 기술하는 명세 장치
- \* 인터넷 쇼핑몰의 고객
- \* 모든 회원들은 공통적으로 '아이디(userId)', '비밀번호(password)', '이름(name)', '주민등록번호(ssn)', '생년월일(birthDate)', '집주소(homeAddress)', '집전화번호(homePhone)', '회사명(companyName)', '부서명(deptName)', '직장주소(officeAddress)', '직장전화번호(officePhone)', '직장우편번호(officeZipCode)', '이메일(email)' 등과 같은 속성들을 갖는다.

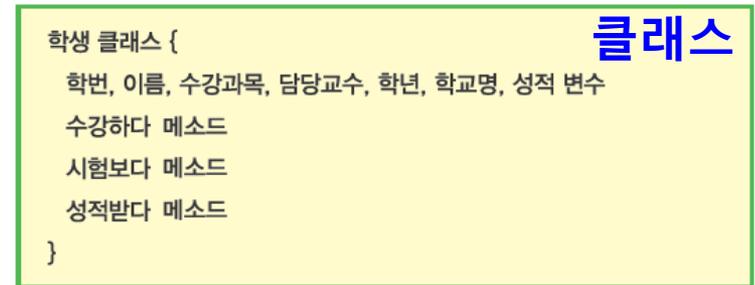


# 클래스와 객체 관계

- \* 클래스는 틀이고 객체는 실사례이다.
- \* 클래스는 상태가 없고 객체는 있다.
- \* 클래스와 객체는 동일한 속성수와 메소드 수를 갖는다

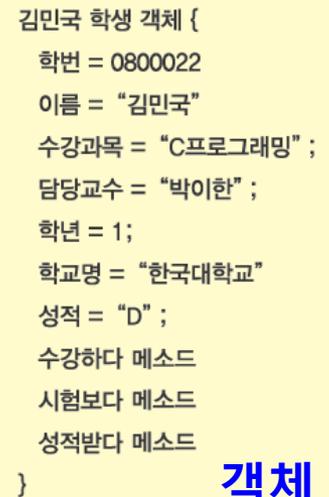
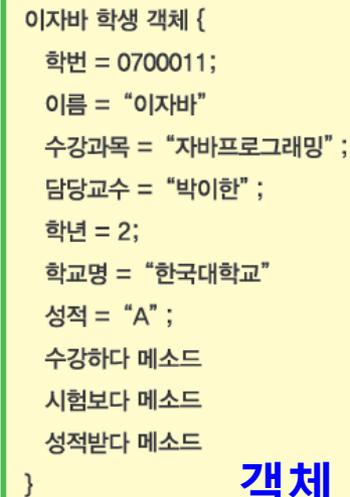
# 클래스와 객체 관계

- \* 클래스는 상태가 없고 객체는 있다.
- \* 클래스는 속성에 대한 선언만 내포하고 있는 반면에 객체는 선언된 각 속성들에 대해 값을 지닌다. 이러한 값을 객체의 **상태**라고 표현한다.



**객체 생성**

**객체 생성**



# 클래스와 객체 관계

- \* 클래스와 객체는 동일한 속성수와 메소드 수를 갖는다.
- \* 한 클래스로부터 생성되는 객체들의 속성이나 메소드 수는 동일하다.

```
학생 클래스 {  
    학번, 이름, 수강과목, 담당교수, 학년, 학교명, 성적 변수  
    수강하다 메소드  
    시험보다 메소드  
    성적받다 메소드  
}
```

**클래스**

```
이자바 학생 객체 {  
    학번 = 0700011;  
    이름 = "이자바"  
    수강과목 = "자바프로그래밍";  
    담당교수 = "박이한";  
    학년 = 2;  
    학교명 = "한국대학교"  
    성적 = "A";  
    수강하다 메소드  
    시험보다 메소드  
    성적받다 메소드  
}
```

**객체**

```
김민국 학생 객체 {  
    학번 = 0800022  
    이름 = "김민국"  
    수강과목 = "C프로그래밍";  
    담당교수 = "박이한";  
    학년 = 1;  
    학교명 = "한국대학교"  
    성적 = "D";  
    수강하다 메소드  
    시험보다 메소드  
    성적받다 메소드  
}
```

**객체**



# 클래스 설계

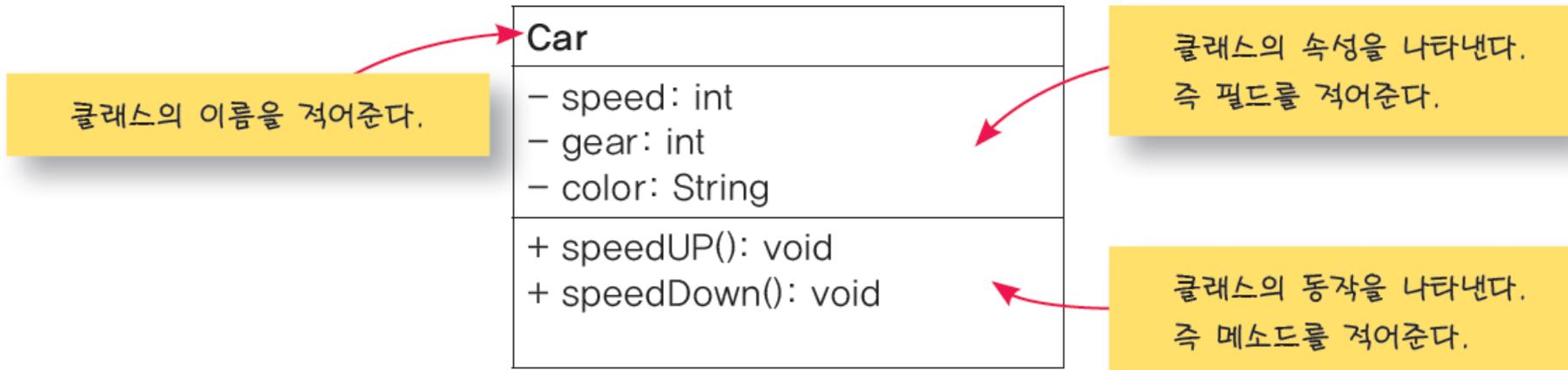


# 클래스 표기법

- \* UML에서 클래스 표기
  - \* 3 행(row)으로 구성된 박스(Box)로 표기한다.
  - \* 클래스 다이어그램의 핵심 구성 요소로서 클래스명, 속성, 행위(함수)로 구성된다.

# UML

- \* UML(Unified Modeling Language): UML은 클래스만을 그리는 도구는 아니고 객체지향설계 시에 사용되는 일반적인 모델링 언어라고 할 수 있다.
- \* UML을 사용하면 소프트웨어를 본격적으로 작성하기 전에 구현하고자 하는 시스템을 시각화하여 검토할 수 있다.





# Television 클래스의 UML



Television
-isOn -volume -channel
+setChannel() +getChannel() +setVolume() +getVolume() +turnOn() +turnOff() +toString()



# 클래스 구현

# 클래스 구현

- \* 자바에서 클래스 코드
  - \* 가시성, class 예약어, 클래스명, 중괄호({ })로 선언하며 중괄호 사이에 속성 및 행위(함수)를 구현한다.

가시성      클래스명  
                  예약어

```
public class Product {  
  
}
```



# 클래스 구현

## \* 속성

Payment 클래스



Delivery 클래스

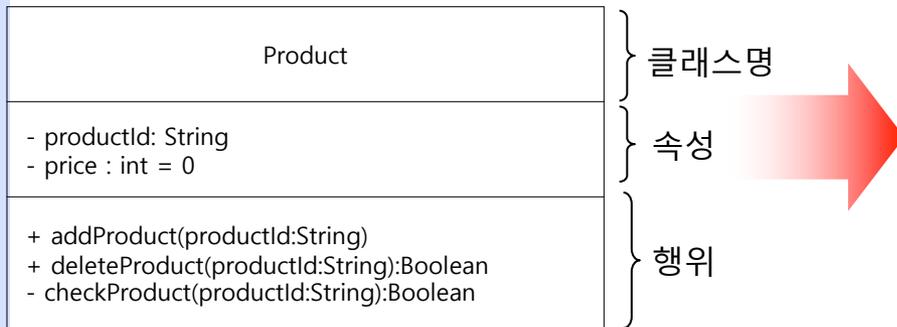


```
public class Order {  
    String    orderId;  
    Payment  paymentType;  
    Delivery  deliveryType;  
}
```

# 클래스 구현

## \* 행위

- \* 가시성, 반환값의 데이터 타입, 행위명(함수명), 매개변수, 중괄호({}) 순으로 구현된다.



(가) UML 클래스의 행위 표기

```
public class Product {  
    String productId;  
    int price = 0;  
  
    public void addProduct(String productId) {  
        ...  
    }  
  
    public Boolean deleteProduct(String productId) {  
        ...  
    }  
  
    private Boolean checkProduct(String productId) {  
        ...  
    }  
}
```

(나) 자바 클래스의 행위 코드



실 습



# 기초 실습

- \* 'Product' 클래스를 설계하시오.

Product
- productId: String - price : int = 0
+ addProduct(productId:String) + deleteProduct(productId:String):Boolean - checkProduct(productId:String):Boolean

- \* 'Product' 클래스를 개발하시오.

# 메소드 overloading

- \* 하나의 클래스 내부에 같은 이름의 메소드 정의 가능
  - \* 매개변수 형식을 다르게 하여 정의
    - \* 매개변수 타입을 다르게
    - \* 매개변수 개수를 다르게
    - \* 매개변수 순서를 다르게
- \* 객체지향의 다형성 (Polymorphism) 지원
  - \* 한 이름의 메소드를 매개변수에 따라 다양한 방법을 사용



메소드  
오버로딩  
(Overloading)

# 메소드 overloading 실습

## \* 프로그램 5-8 실습



- \* OverloadTest.java 작성
- \* + 연산자를 이용한 다양한 연산 수행 메서드 overloading
  - \* `add( String i, String j ); ← add("java", "program");`
  - \* `add( double i, double j ); ← add(2.56, 1.10);`
  - \* `add( int i, int j ); ← add(10,20);`

```
C:\WJAVA>java OverloadTest
문자열 결합의 결과 : javaprogram
실수 덧셈의 결과 : 3.66
정수 덧셈의 결과 : 30
```

'+' 연산자도  
overloading의  
한 형태이군 !

# 메소드 overloading 실습



```
class AddTest {
    int add(int i, int j) {
        return i + j;
    }

    String add(String i, String j) {
        return i + j;
    }

    double add(double i, double j) {
        return i + j;
    }
}
```

```
class OverloadTest {
    public static void main(String args[]) {
        AddTest at = new AddTest();
        String res1 = at.add("java", "program");
        double res2 = at.add(2.56, 1.10);
        int res3 = at.add(10, 20);

        System.out.println("문자열 결합의 결과 : " + res1);
        System.out.println("실수 덧셈의 결과 : " + res2);
        System.out.println("정수 덧셈의 결과 : " + res3);
    }
}
```





# THANK YOU

