3장. 객 체

개념

* 객체는 물리적 사물 또는 논리적 개념이다.
  * **고객**이 **쇼핑몰**에 로그인 한다.
  * 고객이 **상품**을 검색한다.
  * 고객이 상품을 주문한다.
  * 고객이 **신용카드**로 **주문**을 **결제**한다.
  * 고객이 주문을 조회한다.
  * 고객이 주문한 상품을 반송한다.

행위 중심

개념

- 객체는 모듈이다.
  - 객체는 데이터와 해당 데이터를 처리(조작)하는 오퍼레이션들을 포함하는 모듈이다.
  - 객체, 속성, 행위에 대한 다양한 표현

객체지향 개념	객체지향 분석/설계	객체지향
객체	객체	객체
속성(Attribute)	속성, 프로퍼티(Property)	데이터(Data), 멤버 데이터(Member Data), 멤버 변수(Member Variable)
행위(Behavior)	메소드(Method)	오퍼레이션(Operation), 멤버 함수(Member Function)

객체 개념과 특성

특성

* 객체의 모태는 클래스이기 때문에
  클래스의 특성과 유사함.

* 객체는 유일한 식별자를 가져야 한다.
  * 객체 참조(Object Reference) : 메모리 주소 저장

특성

동일 Customer로 부터 생성된
c1,c2는 서로 다른 객체

```
Customer c1 = new ("srkang01", "강사랑", ...);
Customer c2 = new ("swkim02", "김소원", ...);
...
```

그림 3-3 ● 동일 클래스에서 생성된 서로 다른 객체

특성

- 객체는 상태를 가져야 한다.
  - **객체**의 상태란 현재 해당 객체가 지니는 속성(또는 데이터)의 **값**을 의미한다.
  - **클래스**는 각각의 객체가 지녀야 할 속성에 대한 선언만 할 뿐 속성에 대한 값 즉, 상태는 지니지 않는다.
    - 예외) 정적 변수(Static Variable)

특성

```
class Customer{
      String userId;
      String password;
      String name;
      String ssn;
      ...
      public void addCustomer(){...};
      public void updateCustomer(){...};
      ...
      }
```

c1객체

```
useId = "srkang01";
password = "****";
name = "강사랑";
ssn = "980203-******";
      ...
public void addCustomer(){...};
public void updateCustomer(){...};
      ...
}
```

c2객체

```
useId = "swkim02";
password = "****";
name = "김소원";
ssn = "980613-******";
      ...
public void addCustomer(){...};
public void updateCustomer(){...};
      ...
}
```

그림 3-4 ● 객체의 상태

객체 생명 주기

- 객체 생성
  - 생성자(Constructor)에 의해 생성

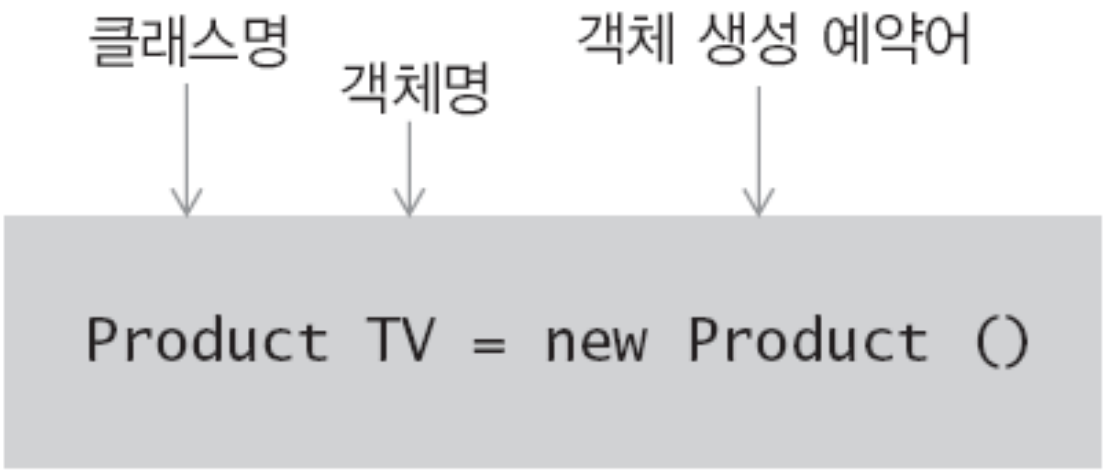
```
Customer Kim = new Customer("kim101", "김솔미", "010-8976-
4657", smkim101@daum.net)
Customer Kang = new Customer("lovekang", "강사랑", "010-4656-
2737", "love88@naver.com")
```

  - 객체 선언
  - 객체 인스턴스화(Instantiation)
  - 객체 초기화
- 객체 사용
- 객체 소멸

객체 구현

객체 구현

- 자바의 객체 코드는 클래스를 기반으로 구현되며, 보통 클래스를 통해 객체가 생성된다 라고 말한다.
- 생성이란 의미는 클래스에 대해 서로 다른 상태를 가진 객체가 생성되는 것을 의미한다.

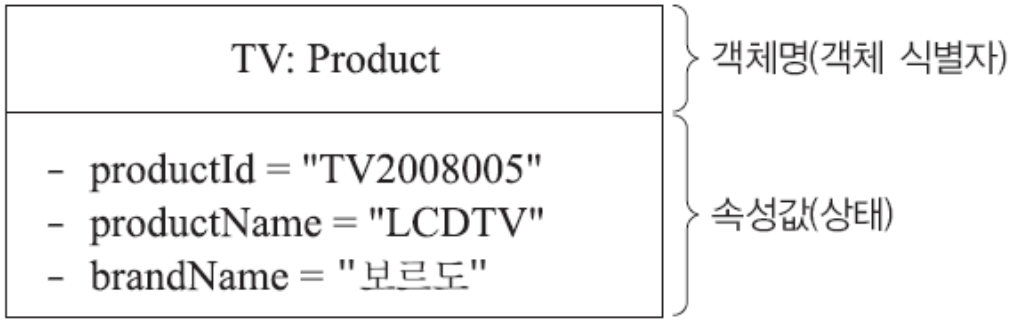


(나) 자바 객체 코드

객체 구현

* 속성값 구현
  * 객체 생성을 통해 객체가 상태를 갖는다.

UML



(가) 객체 구성요소에 의한 객체 표기

자바

```
Product TV = new Product ("TV2008005", "LCDTV", "보르도");
```

(나) 자바 객체의 속성값(상태)

객체 구현

- 속성값 구현
  - 객체 생성 후에 객체 속성 변수들에 속성값을 할당하여 상태를 갖게 할 수 있다.

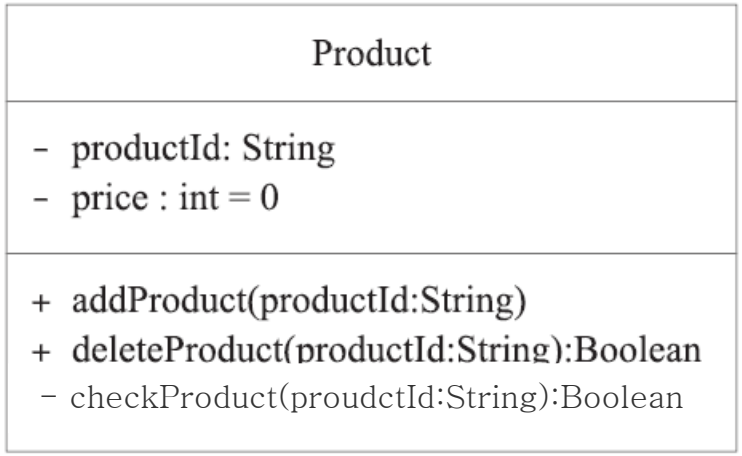
```
Product TV = new Product();
 TV.productId = "TV2008005";
 TV.productName = "LCDTV";
 TV.brandName = "보르도";
```

그림 3-13 ● 객체 속성값 지정

실 습

기초 실습

* 기존 아래와 같이 설계한 'Product' 클래스가 있습니다.



* ProductTest 클래스를 정의하고 main 메소드를 선언한 후 'Product' 클래스의 각 메소드를 활용, 테스트하세요.

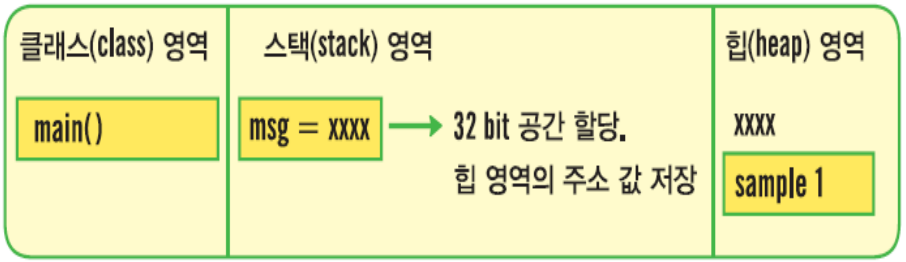
THANK YOU

Appendix

Object Reference

- **참조형 변수**의 메모리 구조
  - 참조형 변수
    - 메모리 크기가 고정되지 않고 가변적인 데이터
    - 데이터 값이 저장되어 있는 **메모리의 주소**를 가짐
      → C에서 포인터
    - 배열, 클래스, 인터페이스

```
class TestVar {
    public static void main(String args[])
    {
        String msg = "sample1";
    }
}
```



Object Reference

* Student 객체 생성시 메모리 구조