



# 8장. 상 속



# 상속 개념

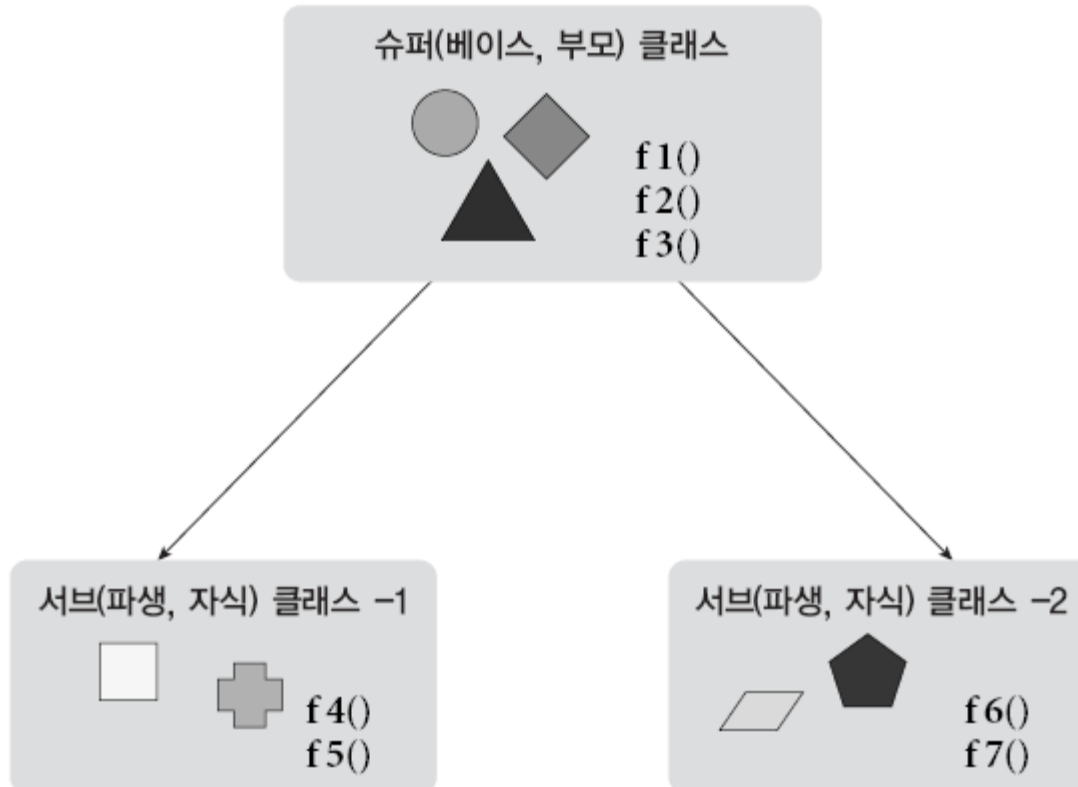


# 개념

- \* 소프트웨어 개발 생산성을 높여주기 위한 방법 → **재사용성**
- \* **재사용성**을 향상시키기 위해 장치 → **상속(Inheritance)**
  - \* **슈퍼 클래스 (Super Class)**
    - \* 재사용될 수 있는 데이터나 함수가 정의된 클래스
    - \* 부모 클래스
  - \* **서브 클래스 (Sub Class)**
    - \* 슈퍼 클래스의 데이터나 함수를 재사용하는 클래스
    - \* 데이터나 함수를 추가적으로 선언할 수 있음.
    - \* 자식 클래스

# 개념

- \* 상속은 데이터나 함수의 재사용을 위한 장치
  - \* 상속은 동일한 데이터나 함수의 중복 정의를 피하고 공통된 데이터나 함수를 한 곳에 정의하여 재사용하는 개념





# 개념

## \* 상속은 Is-A 관계

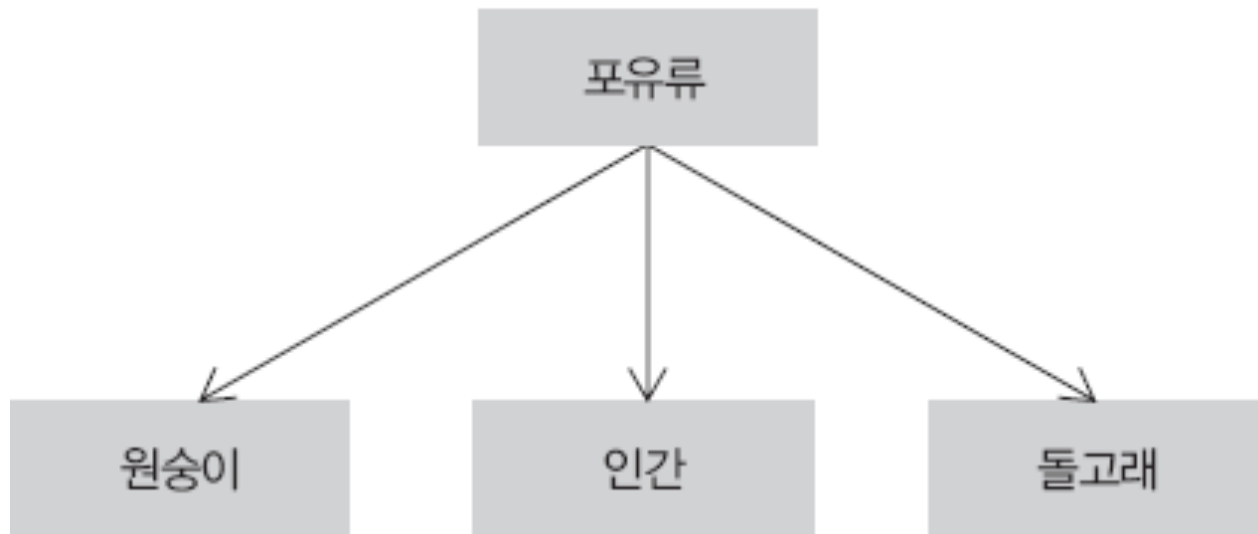
### \* is-a 관계는

서브 클래스에서 슈퍼 클래스를 향한 관계

- \* "서브 클래스 is a 슈퍼 클래스"
- \* "서브 클래스이면 슈퍼 클래스이다"
- \* "서브 클래스 = 슈퍼 클래스"
- \* "슈퍼 클래스 is a 서브 클래스" (X)

# 개념

- \* “인간이면 포유류이다” ( O )
- \* “포유류이면 인간이다” ( X )

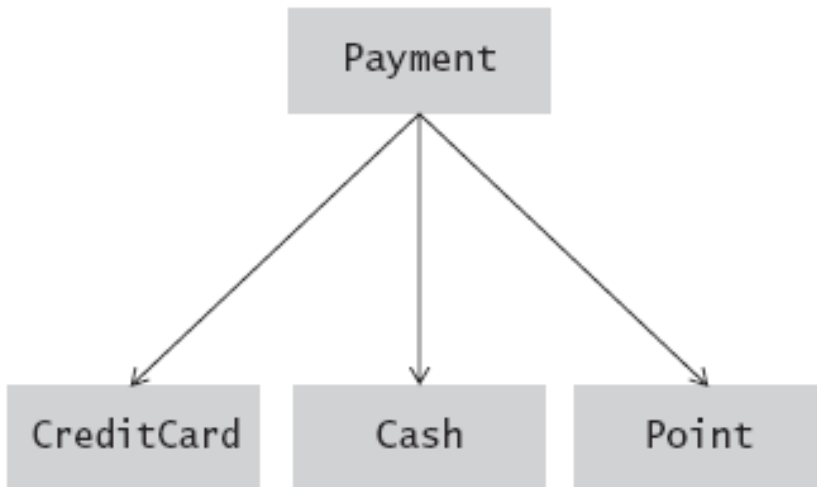




# 개념

- \* is-a 관계를 Java에서는 “upcasting” 메커니즘으로 표현
- \* upcasting 메커니즘을 통해 상속에서 제공하는 재사용성 이외에 “대체성(Substitutability)”이라는 특징을 제공한다.
- \* 대체성이란,  
상속 관계에서 **슈퍼 클래스**의 객체 대신에  
**서브 클래스**의 객체가 대신 대체되어 사용

# 개념



```
class Payment {
    ...
    public void processPayment(Payment p){
        p.pay();
    }
}

class CreditCard extends Payment{
    String cardNum;
    String expireDate;
    ...
    public void pay( ) {
        ... // 신용카드 결제 로직
    }
}

class Cash extends Payment{
    QString account;
    String bankName;
    ...
    public void pay( ) {
        ...// 현금이체
    }
}

class Point extends Payment{
    int point;
    ...
    public void pay() {
        ... // 포인트 결제
    }
}
```





# 상속 특성



# 특성

- \* 데이터와 함수의 중복 제거
- \* 데이터나 함수의 추가 용이
- \* 새로운 클래스의 추가 용이

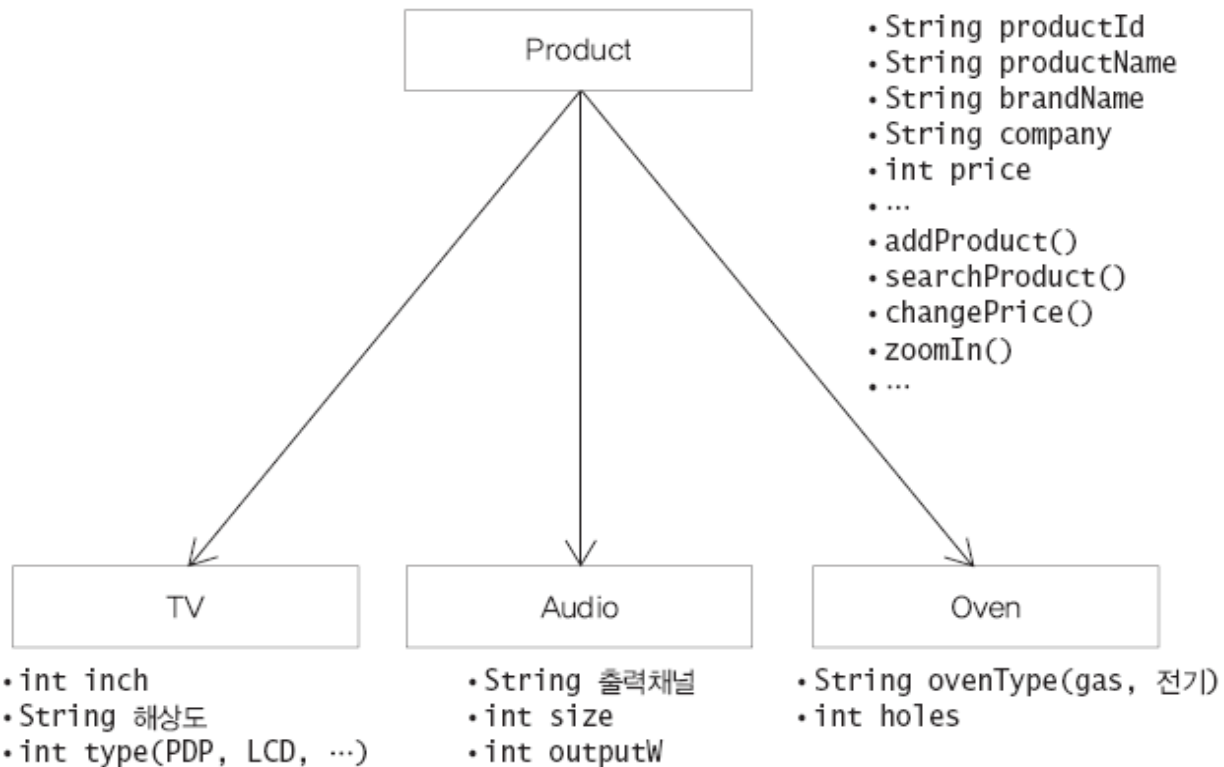


# 특성

- \* 데이터와 함수의 중복 제거
  - \* 슈퍼 클래스에 정의된 데이터나 함수를 서브 클래스에서 정의하지 않아도 그대로 재사용할 수 있음
  - \* 기존의 절차적 프로그래밍 기법에 비해 중복성이 감소

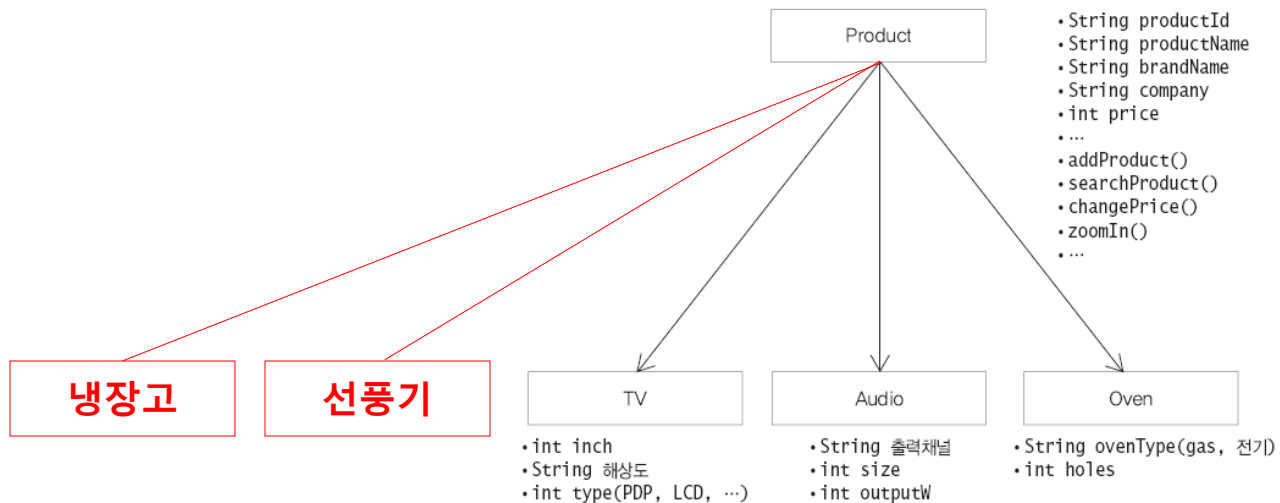
# 특성

- \* 데이터나 함수의 추가 용이
  - \* 서브 클래스에 데이터나 함수를 추가하는 것이 슈퍼 클래스에 영향을 주지 않음.



# 특성

- \* 새로운 클래스의 추가 용이
  - \* 상속 관계는 계층 구조로 형성되어 있기 때문에 새로운 하위 클래스의 추가가 용이
  - \* Product 클래스 밑에 새로운 하위 클래스 '냉장고'나 '선풍기' 등을 쉽게 추가할 수 있음.



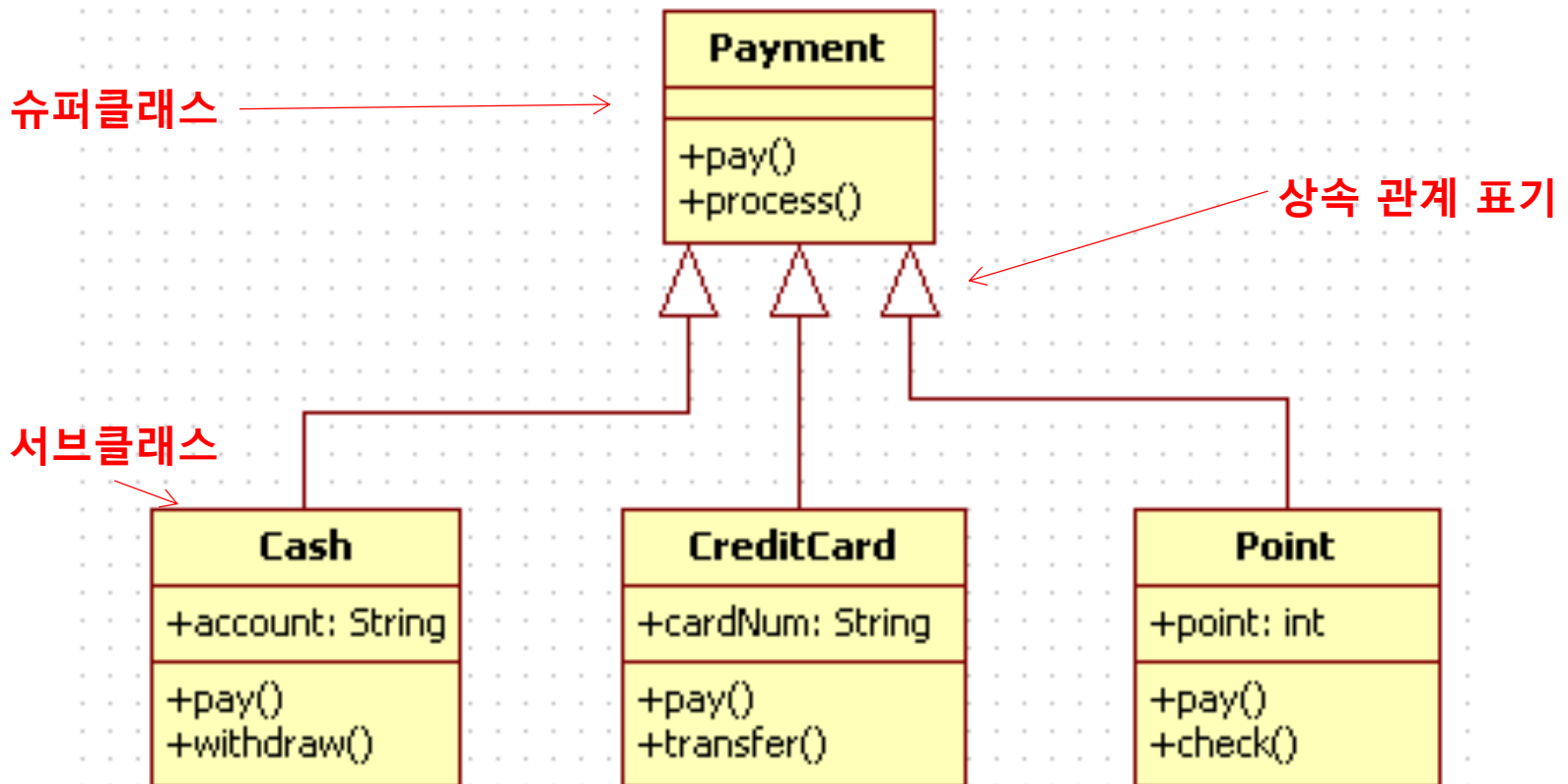


# 상속 설계 및 구현

# 상속 표기법

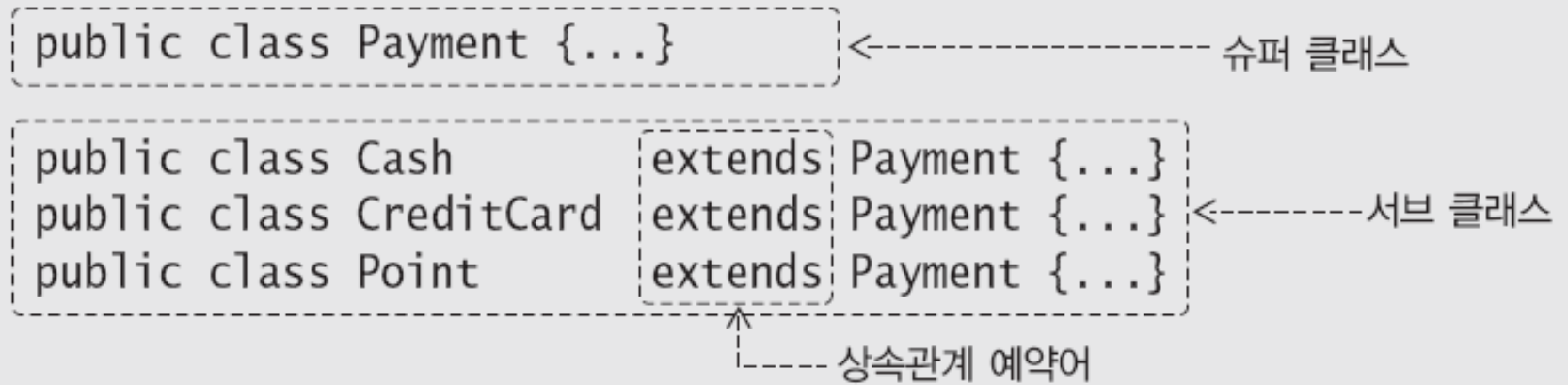
\* UML에서 상속 표기

\* 상속 설계는 클래스 다이어그램(Class Diagram)을 이용



# 상속 구현

- \* 자바에서 상속 코드
  - \* 'extends' 예약어 사용

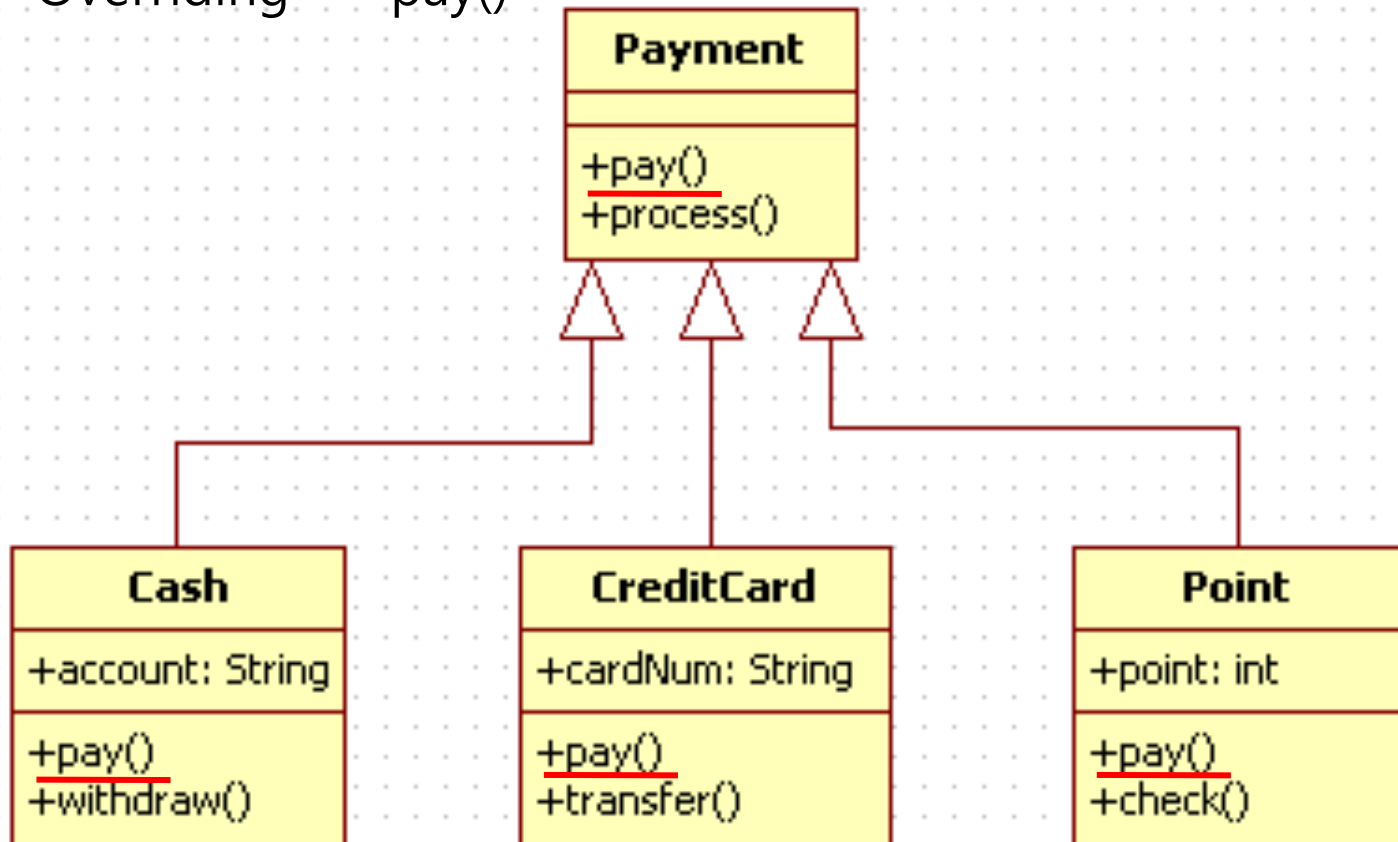




# 상속 구현

\* 자바에서 상속 코드

\* Overriding → 'pay()'





실 습



# 응용 실습

- \* 온라인 쇼핑몰 설계 및 구현

## 온라인 쇼핑몰 요구사항

온라인 쇼핑몰에서 지불(Payment) 방식인 온라인 बैं킹(Bank), 신용카드(CreditCard), 포인트(Point)를 이용하여 지불(pay)과 지불 취소(cancel) 기능을 제공 한다. 또한 신용카드와 포인트는 포인트 적립(save) 기능이 있다.

클래스 다이어그램을 설계하시오. 그리고 코드로 작성하시오.



**THANK YOU**

