



11장. 다형성



다형성 개념



개념

- * 다형성(Polymorphism)
 - * “여러 개의 형태를 가진다”는 의미의 그리스어에서 유래
 - * 특정한 심벌이나 연산자에 대해 상황별로 의미를 달리 부여



개념

- * 객체지향에서의 다형성 형태
 - * 오버로딩(Overloading 혹은 중첩)
 - * 오버라이딩(Overriding 혹은 재정의)



개념

- * 한 클래스에서의 함수 오버로딩
 - * 한 클래스 내에서 동일 이름의 함수가 여러 번 중복 정의되어 있는 경우
 - * 제약사항
 - * 1. 함수 명이 동일해야 함
 - * 2. 중복 정의되는 함수의 매개변수 개수나 혹은 매개변수 타입이 달라야 함



개념

- * 한 클래스 내에서의 함수 오버로딩

```
class function_overloading {  
    int x,y;  
    double a,b;  
    public void f1() {  
        x=10;  
        y=10;  
    }  
    public void f1(int a) {  
        x=10+a;  
        y=10;  
    }  
    public void f1(double c, double d) {  
        a = c;  
        b = d;  
    }  
}
```



개념

* 함수 오버라이딩

- * 부모 클래스 내에 구현된 함수에 대해서 함수명만 상속 받고 함수 구현은 서브 클래스들에서 각각 재정의하는 것
- * 상속 구조에서 부모 클래스가 추상 클래스이든지 일반 클래스이든지 상관이 없음



개념

* 함수 오버라이딩

```
class Arc {  
    void draw( ) {  
        System.out.println("Drawing arc....");  
    }  
    void erase( ) {  
        System.out.println("Erasing arc....");  
    }  
}  
class Oval extends Arc {  
    void draw( ) {  
        System.out.println("Drawing oval....");  
    }  
    void erase( ) {  
        System.out.println("Erasing oval....");  
    }  
}  
class Circle extends Oval {  
    void draw( ) {  
        System.out.println("Drawing circle...");  
    }  
    void erase( ) {  
        System.out.println("Erasing circle...");  
    }  
}
```




다형성 특성



특성

- * 코드의 재사용성을 높여 준다.
 - * 특정 기호를 여러 가지 목적에 사용해야 한다고 가정해보자
 - * 데이터 형에 따라 기호의 목적에 맞게 사용할 수 있도록 할 수 있다
 - * 이로써 프로그램이 간결해 지고 프로그램에서 기호의 재사용성을 높일 수 있다



특성

- * 일관된 인터페이스 제공
 - * 자바의 효율적인 다형성을 활용하기 위해서는 상속 구조를 통해 슈퍼 클래스가 추상 클래스나 인터페이스 형태를 취함
 - * 이렇게 하면, 하위 서브 클래스들이 슈퍼 클래스에 종속되는 것을 피할 수 있음
 - * 슈퍼 클래스에는 간단히 함수 선언 정도만 다루고, 하위 서브 클래스들은 상세한 함수 구현을 하도록 함



특성

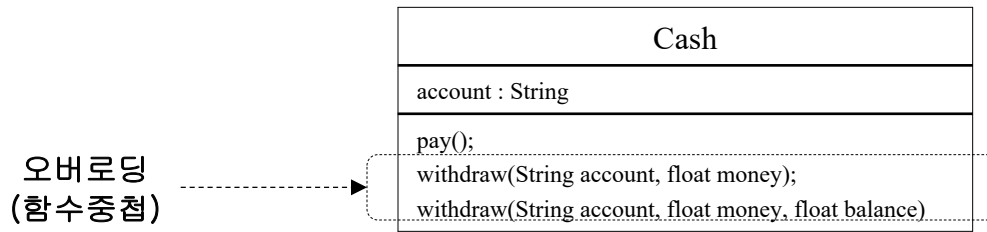
- * 일관된 인터페이스 제공
 - * 이 클래스를 접근하는 다른 객체들은 하나의 통로(인터페이스 메소드나 추상 메소드)로 다양한 유형의 객체들을 접근할 수 있음
 - * 즉, 외부 객체는 해당 메소드를 구현하는 객체들이 누구지 일일이 신경쓰지 않을 수 있음
 - * 또한, 하나의 메소드로 여러 유형의 객체들을 동적으로 사용할 수 있음



다형성 설계 및 구현

다형성 설계

- * UML에서 한 클래스 내에서의 함수 오버로딩 설계



다형성 구현

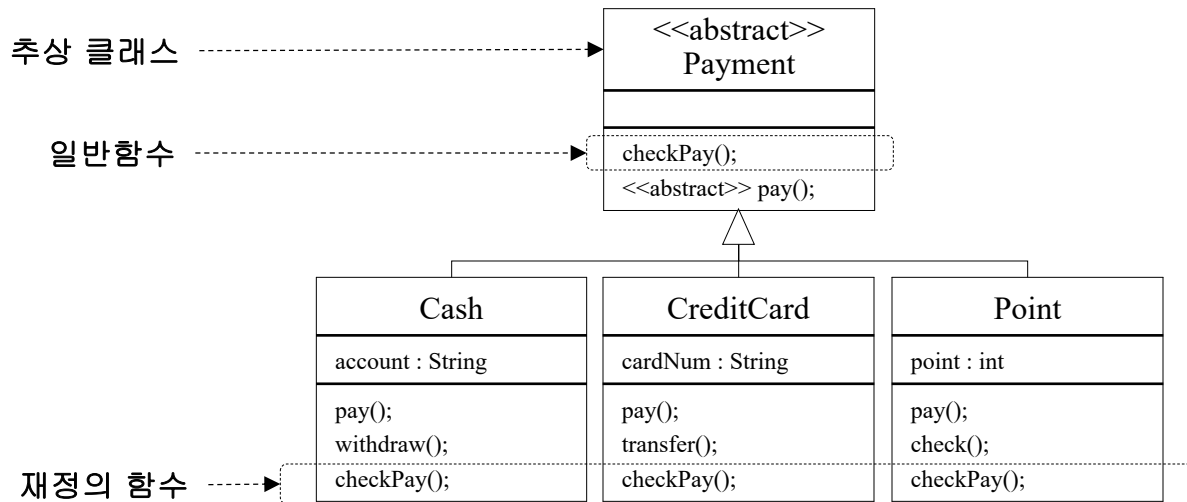
- * 한 클래스 내에서 함수 오버로딩 구현 코드
 - * 함수명은 동일하되, 매개변수 개수나 타입을 달리함

오버로딩
(함수중첩)

```
public class Cash extends Payment {  
    public void pay() { ... }  
  
    public void withdraw(String account, float money) { // 인출 처리 방식 A... }  
    public void withdraw(String account, float money, float balance) { // 인출 처리 방식 B... }
```

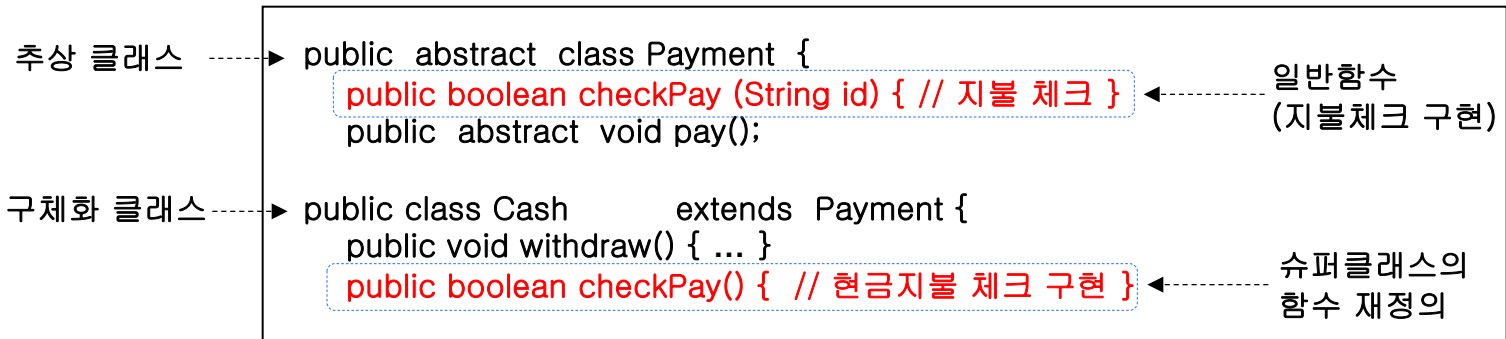
다형성 설계

* 함수 오버라이딩 설계



다형성 구현

* 오버 라이딩 구현 코드





실습



응용 실습 1

- * 온라인 banking 시스템의 계좌 클래스를 이용하여 다형성을 설계하시오.
 - * 계좌 클래스의 다형성 요구사항 (계좌 클래스는 일반 클래스)
 - * 다형성 함수
 - * 이체(이체금액)
 - * 이체(이체금액, 타행계좌)
 - * 이체(이체금액, 타행계좌, 타행명)
 - * 계좌 클래스의 자식클래스들을 이용한 다형성 요구사항
 - * 자식 클래스 : 일반계좌, 정기에금계좌, 자산관리계좌(CMA, Cash Management Account), 증권예탁계좌
 - * 각 클래스의 다형성 함수
 - * 이체(이체금액, 타행계좌, 타행명)



응용 실습 2

- * 2. 온라인 뱅킹 시스템의 계좌 클래스 설계에 적용된 다형성 자바 코드를 정의하시오.
 - * 계좌 클래스의 설계된 3개의 함수를 정의하시오.
 - * 이체 함수 구현부 : return true;
 - * 자식 클래스에서 정의한 다형성 함수를 정의하시오.
 - * 이체 함수 구현부 : return true;
 - * 자식 클래스들의 다형성 함수를 호출하는 코드를 정의하시오.



THANK YOU

