

자바 입문 과제_6

박은종

인터페이스

다음 인터페이스를 활용하여 사칙연산을 수행하는 Calculator 계산기를 만들고

ICalc calc = new Calculator() 로 생성하여 각 사칙연산을 실행해보세요

```
public interface Calc {  
  
    double PI = 3.14;  
    int ERROR = -999999999;  
  
    int add(int num1, int num2);  
    int subtract(int num1, int num2);  
    int times(int num1, int num2);  
    int divide(int num1, int num2);  
}
```

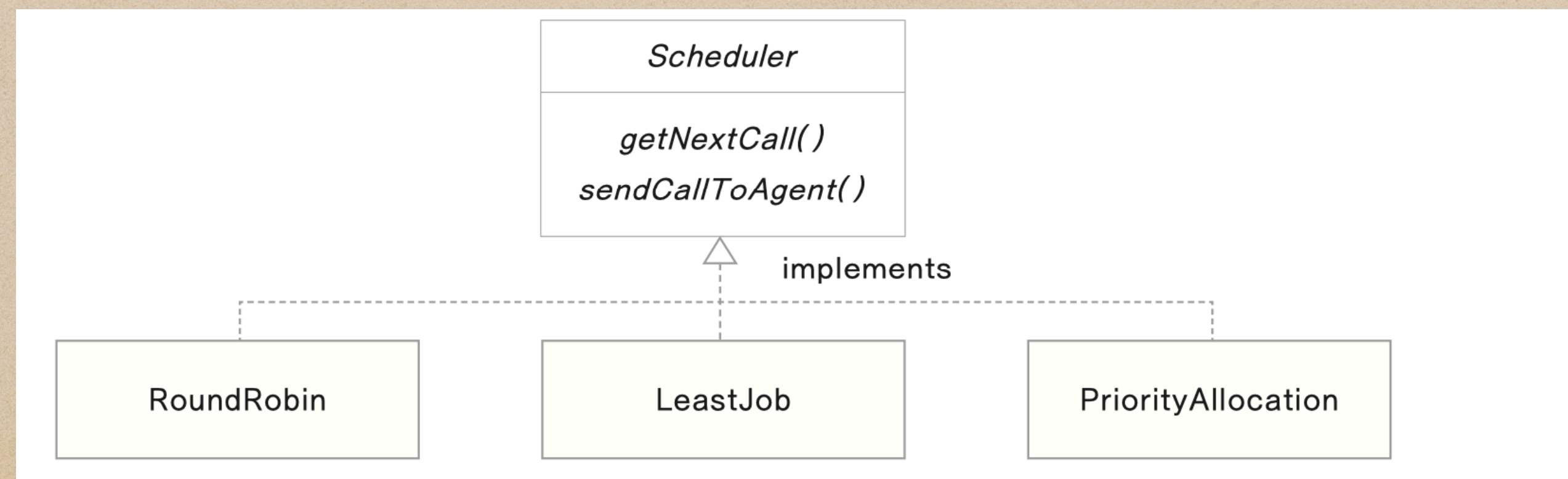
#인터페이스 #인터페이스구현 #인터페이스란무엇인가? #인터페이스에선언된변수는?

인터페이스와 다형성

고객 센터에는 전화 상담을 하는 상담원들이 있습니다. 일단 고객 센터로 전화가 오면 대기열에 저장됩니다.

상담원이 지정되기 전까지 대기 상태가 됩니다. 각 전화(call)를 상담원에게 배분하는 정책은 다음과 같이 여러 방식으로 구현할 수 있습니다.

1. 순서대로 배분하기 : 모든 상담원이 동일한 상담 건수를 처리하도록 들어오는 전화 순서대로 상담원에게 하나씩 배분합니다.
2. 짧은 대기열 찾아 배분하기 : 고객 대기 시간을 줄이기 위해 상담을 하지 않는 상담원이나 가장 짧은 대기열을 보유한 상담원에게 배분합니다.
3. 우선순위에 따라 배분하기 : 고객 등급에 따라 등급이 높은 고객의 전화를 우선 가져와서 업무능력이 좋은 상담원에게 배분합니다.



인터페이스와 다형성

인터페이스를 구현한 각 정책 클래스가 입력에 따라 수행될 수 있도록 구현하세요

RoundRobin, LeastJob, PriorityAllocation 클래스로 생성한 인스턴스는 모두 Scheduler형 변수에 대입할 수 있습니다.

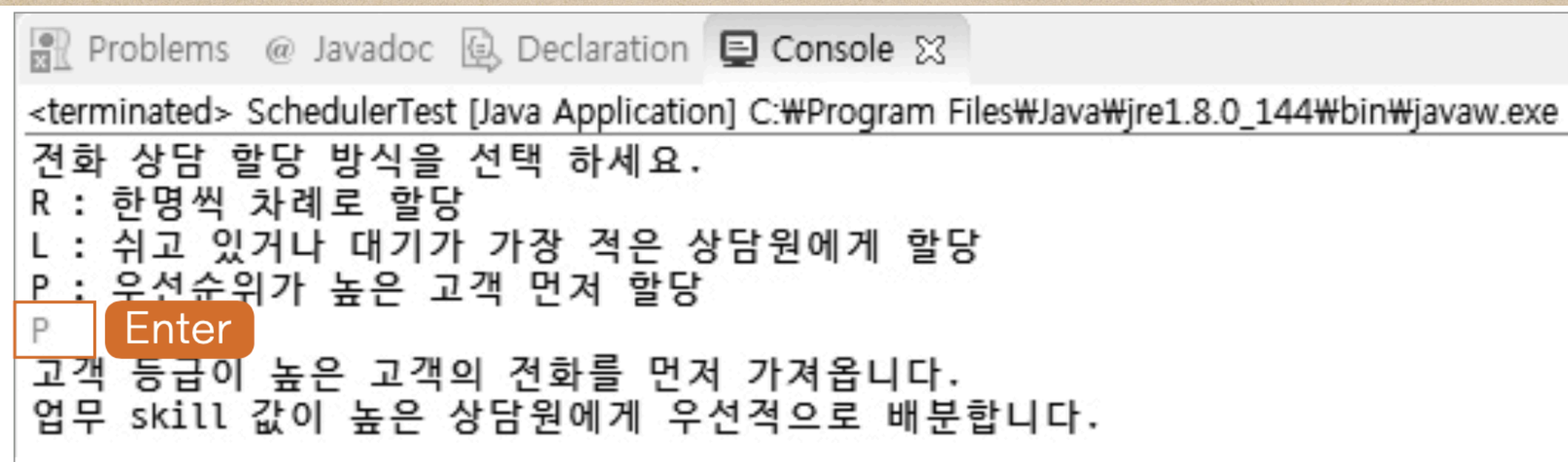
그리고 사용할 인스턴스가 어떤 클래스로 생성되었는지와 상관없이 인터페이스에서 제공하는 메서드를 호출하면 됩니다.

인터페이스를 활용해 다형성을 구현한 것이지요.

실행 화면을 보면 오른쪽에 빨간색 사각형이 있는데 지금 프로그램이 실행 중이라는 뜻입니다.

그리고 화면의 커서 부분에서 R, r, L, l, P, p 문자 중 하나를 입력하고 Enter를 누르면 해당 정책이 수행됩니다.

(system.in을 사용하세요)



```
Problems @ Javadoc Declaration Console X
<terminated> SchedulerTest [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe
전화 상담 할당 방식을 선택 하세요.
R : 한명씩 차례로 할당
L : 쉬고 있거나 대기가 가장 적은 상담원에게 할당
P : 우선순위가 높은 고객 먼저 할당
P Enter
고객 등급이 높은 고객의 전화를 먼저 가져옵니다.
업무 skill 값이 높은 상담원에게 우선적으로 배분합니다.
```


인터페이스의 여러 요소들

#상수

자바 7 이후 구현메서드가 인터페이스에 사용 됨

#디폴트메서드. #디폴트메서드 재정의 하기

#static메서드

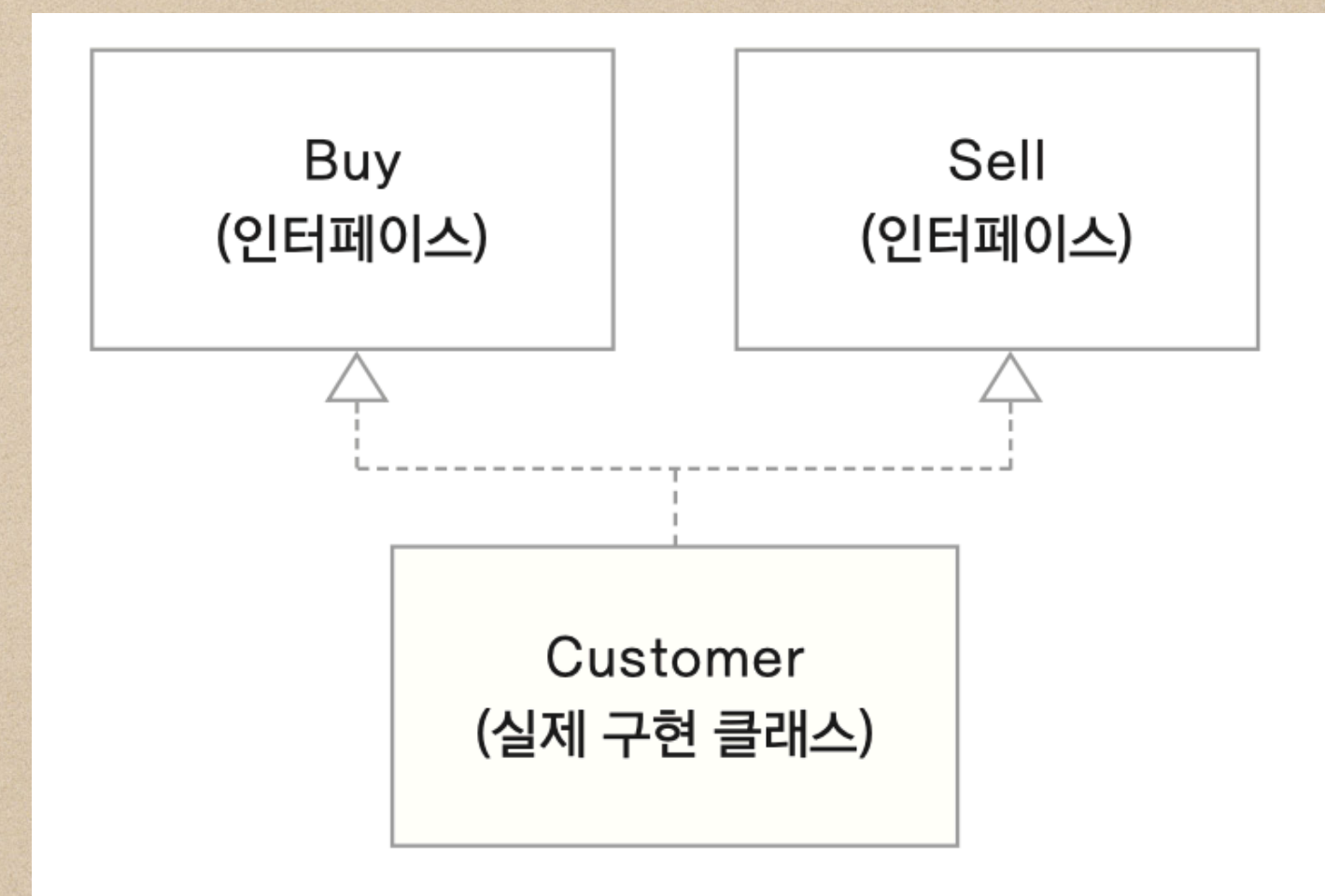
#private메서드

어디에 사용되는지???

여러 인터페이스의 구현

자바에서 클래스는 단일 상속이지만, 인터페이스는 구현코드가 없으므로 여러 인터페이스를 하나의 클래스가 구현 할 수 있습니다. (다중상속 아님...)

```
public interface Buy {  
  
    void buy();  
  
    default void order(){  
        System.out.println("구매 주문");  
    }  
}  
  
public interface Sell {  
  
    void sell();  
  
    default void order(){  
        System.out.println("판매 주문");  
    }  
}
```



두 인터페이스를 구현한 Customer 클래스가 각 Buy와 Sell로 형변환될때 어떻게 사용되는지?
default 메서드가 중복되는경우 Customer에서 어떻게 처리해야하는지?

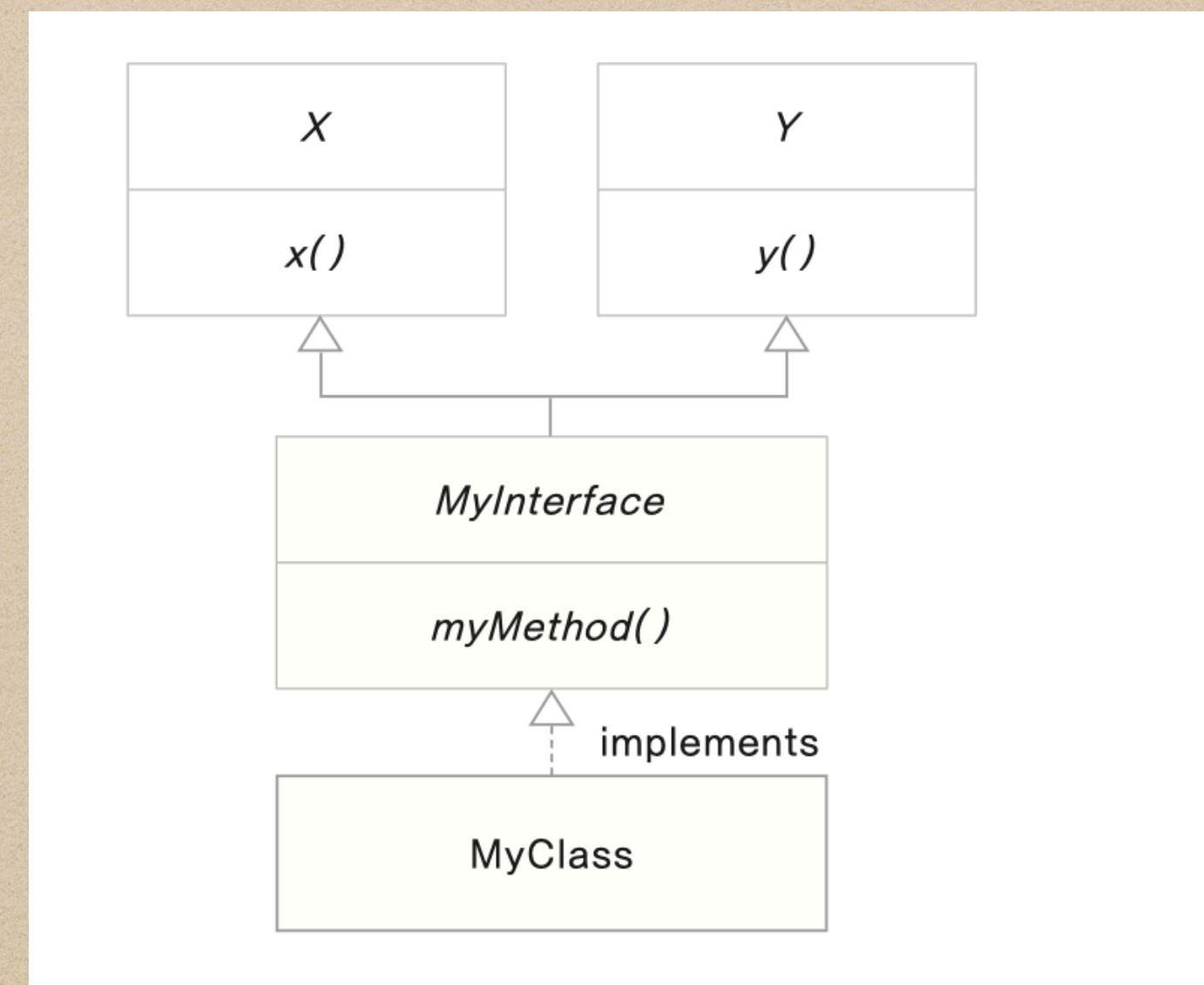
인터페이스간의 상속

인터페이스간의 상속이 가능

구현 상속이 아니므로 type 상속이라고 함

MyClass를 구현하고 다음 테스트 코드를 수행하세요

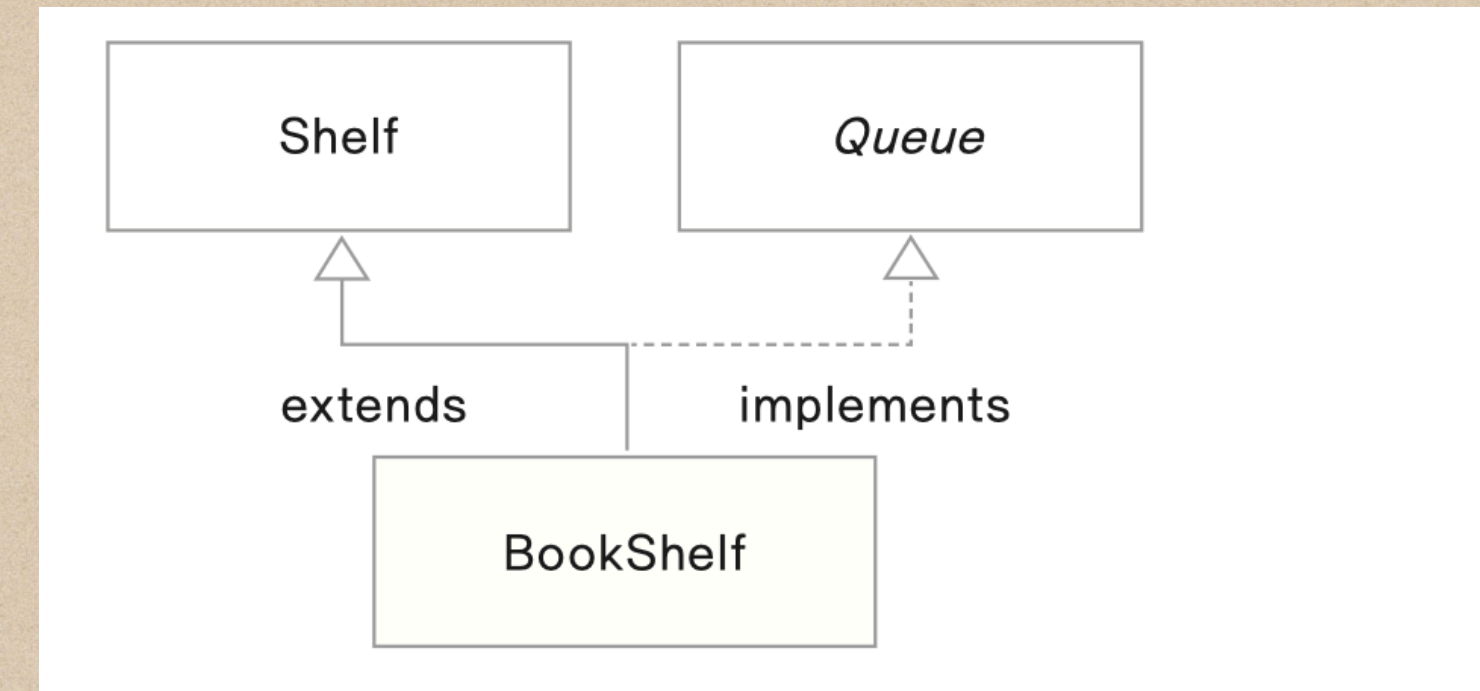
```
public class MyClassTest {  
  
    public static void main(String[] args) {  
  
        MyClass mClass = new MyClass();  
        X xClass = mClass;  
        xClass.x();  
  
        Y yClass = mClass;  
        yClass.y();  
  
        MyInterface iClass = mClass;  
        iClass.myMethod();  
        iClass.x();  
        iClass.y();  
    }  
}
```



인터페이스 구현과 클래스 상속 같이 사용하기

다음 코드가 수행되도록 Shelf와 Queue, BookShelf를 구현하세요

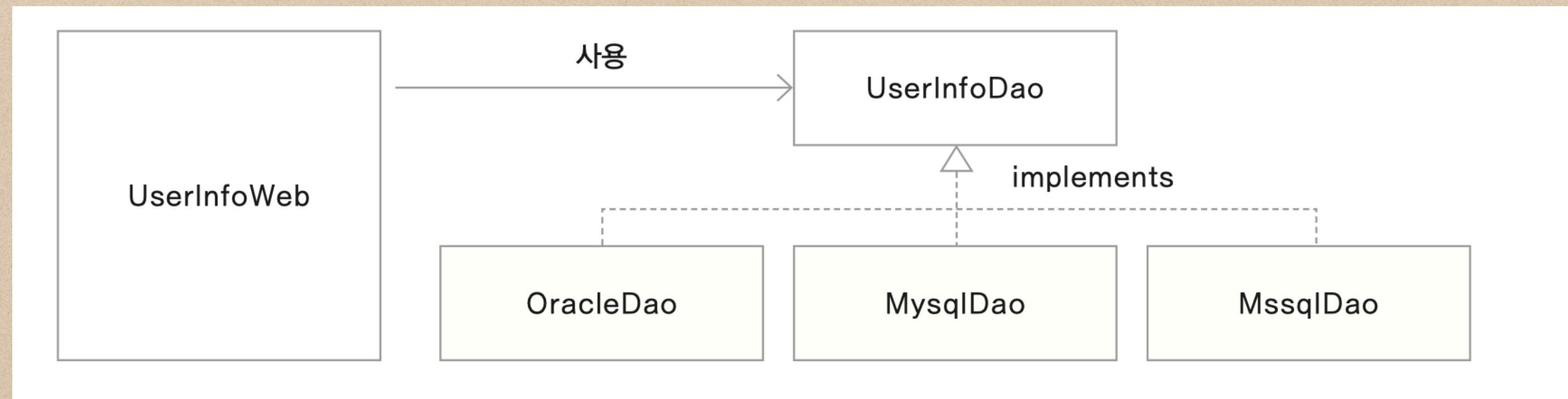
```
public class BookShelfTest {  
  
    public static void main(String[] args) {  
  
        Queue shelfQueue = new BookShelf();  
        shelfQueue.enqueue("태백산맥 1");  
        shelfQueue.enqueue("태백산맥 2");  
        shelfQueue.enqueue("태백산맥 3");  
  
        System.out.println(shelfQueue.dequeue());  
        System.out.println(shelfQueue.dequeue());  
        System.out.println(shelfQueue.dequeue());  
  
    }  
}
```



The screenshot shows a Java IDE's console window. The title bar includes tabs for Problems, Javadoc, Declaration, and Console. The console output shows the execution of BookShelfTest, which prints the titles of three books: 태백산맥 1, 태백산맥 2, and 태백산맥 3, each on a new line.

Think about

구현 코드도 없는 인터페이스는 왜 쓰는걸까요? (아래 그림을 보며 생각해 보세요)



인터페이스와 추상 클래스의 쓰임새의 차이는 무엇일까요? (추상 클래스도 구현코드 없이 메서드 선언만 하는 경우도 종종 있습니다.)

=> 어떤 경우 추상 클래스로 구현하여 상속을 하고, 어떤 경우에 인터페이스로 선언한것을 구현해야 할까요?