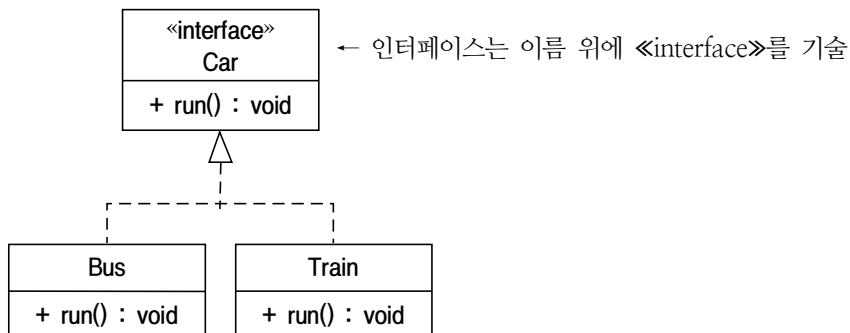


5. 실체화(realization)

- 실체화는 먼저 인터페이스에서 메서드를 선언하고
- 자식클래스에서 상속받아서 메서드의 실제 기능을 구현한다.
- 인터페이스와 자식클래스의 관계는 점선과 인터페이스 쪽으로 빈 삼각형을 연결한다.

〈실체화(realization)〉



↓ Java 원시코드 생성

〈Java 원시코드〉

```

interface Car { public void run(); } //인터페이스
class Bus implements Car //클래스 버스(Bus)
{
    public void run() { System.out.println("버스는 도로를 이용"); } //메서드 구현
}
class Train implements Car //클래스 기차(Train)
{
    public void run() { System.out.println("기차는 철도를 이용"); } //메서드 구현
}
    
```

◆ 스테레오 타입(stereo type)

- 스테레오 타입은 UML에서 기본으로 제공하는 이외에 추가 확장요소를 나타내는 것이다.
- 스테레오 타입은 쌍 꺾쇠와 비슷하게 생긴 길러멧(guillemet, « ») 사이에 적는다.
- 길러멧 기호는 쌍 꺾쇠와는 다른 것으로 폰트 크기가 조금 작다.
- 스테레오 타입 : «interface», «abstract», «utility», «enumeration» 등

2 <http://cafe.daum.net/pass365>(홍재연)

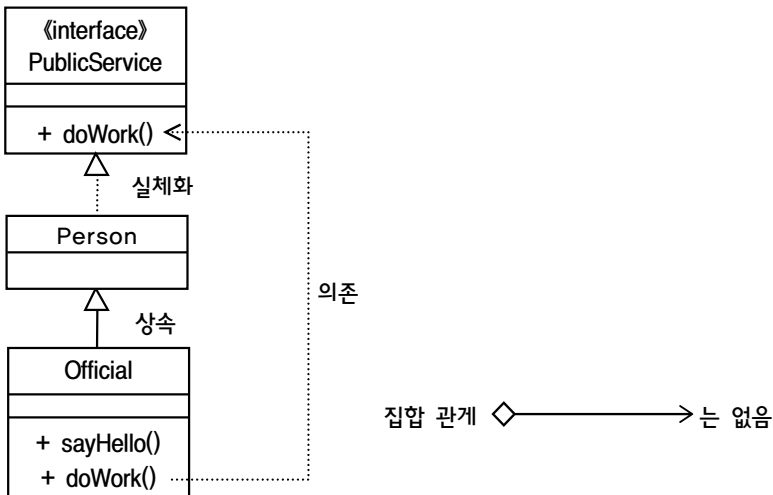
기출문제 분석

1. 다음 Java 코드에서 생략된 부분을 고려하지 않고 명시된 부분만으로 UML 클래스 다이어그램을 작성하였을 때, 다이어그램에 나타나지 않는 관계는? [2022년 국가 7급]

```
class Person { ... }
class Work { ... }
interface PublicService
{
    public void doWork(Work work);
}
class Official extends Person implements PublicService
{
    public void sayHello() { ... }
    public void doWork(Work work) { ... }
    ...
}
```

- ① ----->
- ② ◇----->
- ③ -----▷
- ④ -----▷

☞ 주어진 원시코드를 UML 클래스 다이어그램으로 그리면 다음과 같다.



정답 : ②

2. <보기>는 MyIF 인터페이스를 정의하고 이를 MyImpl 클래스로 구현한 자바코드와 이들의 관계를 UML로 표현한 클래스 다이어그램이다. (가), (나)에 들어갈 내용과 (다)에 들어갈 그림을 옳게 짝 지은 것은? [2022년 서울 7급]

-----<코드>-----

```

(가) MyIF { ... }
class MyImpl (나) MyIF { ... }
    
```

-----<UML 표기>-----

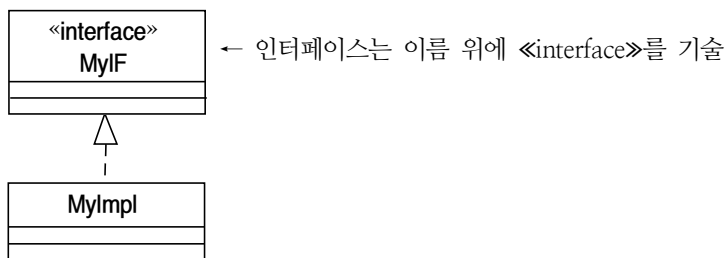
(다)

	(가)	(나)	(다)
①	abstract	extends	
②	interface	extends	
③	interface	implements	
④	interface	implements	

♣ 클래스 다이어그램 : 실체화(realization)

```

interface MyIF { ... }
class MyImpl implements MyIF { ... }
    
```



- 클래스 다이어그램의 실체화는 먼저 인터페이스에서 메서드를 선언하고
- 자식클래스에서 상속받아서 메서드의 실제 기능을 구현한다.
- 인터페이스와 자식클래스의 관계는 점선과 인터페이스 쪽으로 빈 삼각형을 연결한다.

3. 다음 설명에 해당하는 용어는? [2009년 국가 7급]

클래스나 컴포넌트가 제공하는 서비스들을 나타내는 오퍼레이션들의 집합을 나타낸다. 함수들의 시그니처(signature)만 명세할 뿐 함수 구현은 전혀 존재하지 않으며, 속성 값도 존재할 수 없다. 구현은 상속한 클래스에서 하며, 자체는 객체를 생성할 수 없다. 클래스와의 관계는 실체화 관계로 표현한다.

- ① 추상클래스 ② 액티브 클래스
- ③ 인터페이스 ④ 상대머신

☞ 인터페이스

- 메소드 구현은 상속받은 클래스에서 메서드를 완성하는 것이다.(실체화라고 함)
- Java에서 인터페이스 구현(실체화)을 예로 들면 다음과 같다.

```
interface Bird{ void fly(); }           //메소드 시그니처만 명세할 뿐 메소드 구현은 없다.
interface Horse{ void run(); }        //메소드 시그니처만 명세할 뿐 메소드 구현은 없다.
```

```
public class Pegasus implements Bird, Horse    //Pegasus는 날개 달린 말
{
    public void fly(){ System.out.println("새처럼"); } //메소드 구현은 상속받은 클래스에서
    public void run(){ System.out.println("말처럼"); } //메소드 구현은 상속받은 클래스에서
    public static void main(String args[])
    {
        Pegasus p = new Pegasus();
        p.fly();
        p.run();
    }
} //실행 결과
새처럼
말처럼
```

- 인터페이스는 100% 추상화된 클래스이다.
- 인터페이스는 상수와 추상메서드를 선언한 집합이다.
- 인터페이스는 상수와 추상메서드만을 갖는 추상클래스라 할 수 있다.
- 인터페이스는 추상클래스보다 더욱 완벽한 추상화를 제공할 수 있는 요인이다.