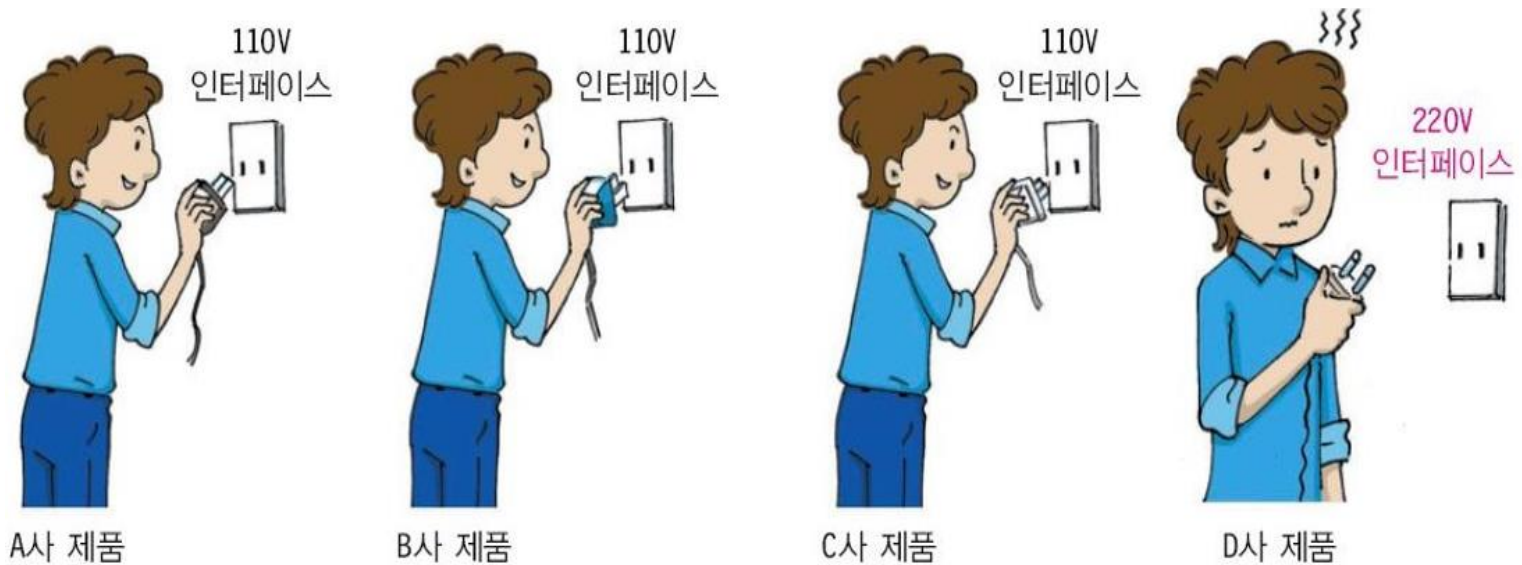


인터페이스와 구현 클래스 선언

▣ 인터페이스 역할

- 인터페이스(interface)는 사전적인 의미로 두 장치를 연결하는 접속기를 말한다.
여기서 두 장치를 서로 다른 객체로 본다면, 인터페이스는 이 두 객체를 연결하는 역할을 한다.
- 상속을 이용해서 다형성을 구현할 수도 있지만, 인터페이스를 이용해서 다형성을 구현하는 경우가 더 많다.



정해진 규격(인터페이스)에 맞기
만 하면 연결 가능.
각 회사마다 구현 방법은 다름

정해진 규격(인터페이스)에 맞지
않으면 연결 불가

	추상화	인터페이스
공통점	- 추상화&인터페이스 메서드는 구현부(바디, {})가 없는 메서드 형태. ex) <code>public abstract void abstract_method();</code> ex) <code>public void interface_method();</code> - 오버라이딩 강제성이 있다.	
선언부	abstract	interface
물려받는 키워드	상속 (extends) ex) <code>class AbClass extends Aclass{ }</code>	구현 (implements) ex) <code>class ImClass implements Aclass{ }</code>
내부 메소드	일반 메소드, 추상 메서드 둘다 가능	추상 메서드만 구현 가능 (JAVA 8부터는 default, static 가능)
다중 상속	불가능	가능

▣ 인터페이스와 구현 클래스 선언

- 인터페이스는 ‘~.java’ 형태의 소스 파일로 작성되고, ‘~.class’ 형태로 컴파일되기 때문에 물리적 형태는 클래스와 동일하다. 단, 소스를 작성할 때 선언하는 방법과 구성 멤버가 클래스와 다르다.

● 인터페이스 선언

- 접근 제한자는 class 와 동일합니다.

```
interface 인터페이스명 { ... }           //default 접근 제한  
public interface 인터페이스명 { ... }    //public 접근 제한
```

```
public interface 인터페이스명 {  
    //public 상수 필드  
    //public 추상 메소드  
    //public 디폴트 메소드  
    //public 정적 메소드  
    //private 메소드  
    //private 정적 메소드  
}
```

※ 추상 메소드란 선언부만 있고 실행부인 중괄호가 없는 메소드를 말한다.

● 구현 클래스 선언

- 인터페이스에 정의된 추상 메소드에 대한, 실행 내용이 구현(작성)된 객체를 구현 객체라고 한다.

```
public class B implements 인터페이스명 { ... }
```

