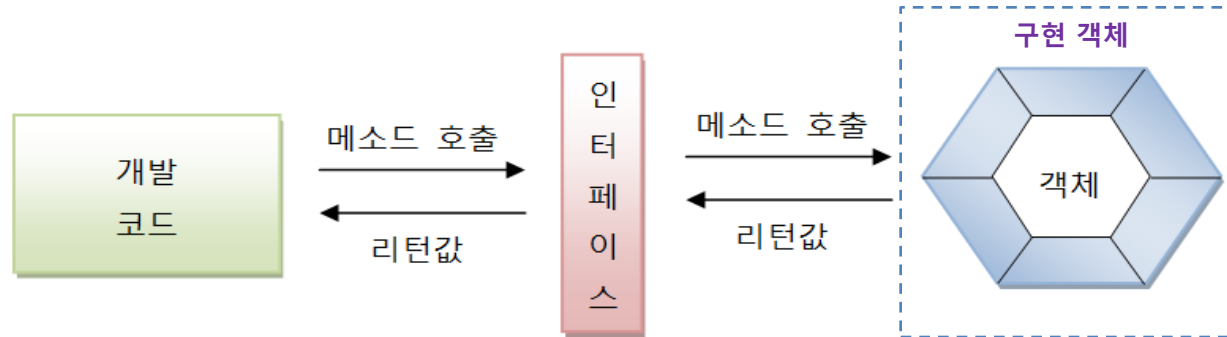


인터페이스 구현 (Implementation)

인터페이스 구현(Implementation)

❖ 구현 객체와 구현 클래스

- 인터페이스의 추상 메소드 대한 실제 메소드를 가진 객체 = 구현 객체



- 구현 객체를 생성하는 클래스 = 구현 클래스

인터페이스 구현(Implementation)

❖ 구현 클래스 선언

- 자신의 객체가 인터페이스 타입으로 사용할 수 있음
 - `implements` 키워드로 명시

```
public class 구현클래스명 implements 인터페이스명 {  
    //인터페이스에 선언된 추상 메소드의 실제 메소드 선언  
}
```

❖ 추상 메소드의 실제 메소드를 작성하는 방법

- 메소드의 선언부가 정확히 일치해야
- 인터페이스의 모든 추상 메소드를 재정의하는 실제 메소드 작성해야
 - 일부만 재정의할 경우, 추상 클래스로 선언 + `abstract` 키워드 붙임

인터페이스 구현(Implementation)

❖ 구현 클래스 : Television.java

```
public class Television implements RemoteControl {  
    // 필드  
    private int volume;  
  
    // turnOn() 추상 메소드의 실제 메소드  
    public void turnOn() {  
        System.out.println("TV를 켭니다.");  
    }  
  
    // turnOff() 추상 메소드의 실제 메소드  
    public void turnOff() {  
        System.out.println("TV를 끕니다.");  
    }  
}
```

인터페이스 구현(Implementation)

❖ 구현 클래스 : Television.java

```
// setVolume() 추상 메소드의 실제 메소드
public void setVolume(int volume) {
    if (volume > RemoteControl.MAX_VOLUME) {
        this.volume = RemoteControl.MAX_VOLUME;
    } else if (volume < RemoteControl.MIN_VOLUME) {
        this.volume = RemoteControl.MIN_VOLUME;
    } else {
        this.volume = volume;
    }
    System.out.println("현재 TV 볼륨: " + volume);
}
}
```

인터페이스 구현(Implementation)

❖ 구현 클래스 : Audio.java

```
public class Audio implements RemoteControl {  
    // 필드  
    private int volume;  
  
    // turnOn() 추상 메소드의 실제 메소드  
    public void turnOn() {  
        System.out.println("Audio를 켭니다.");  
    }  
  
    // turnOff() 추상 메소드의 실제 메소드  
    public void turnOff() {  
        System.out.println("Audio를 끕니다.");  
    }  
}
```

인터페이스 구현(Implementation)

❖ 구현 클래스 : Audio.java

```
// setVolume() 추상 메소드의 실제 메소드
public void setVolume(int volume) {
    if (volume > RemoteControl.MAX_VOLUME) {
        this.volume = RemoteControl.MAX_VOLUME;
    } else if (volume < RemoteControl.MIN_VOLUME) {
        this.volume = RemoteControl.MIN_VOLUME;
    } else {
        this.volume = volume;
    }
    System.out.println("현재 Audio 볼륨: " + volume);
}
}
```

인터페이스 구현(Implementation)

❖ 인터페이스 변수에 구현 객체 대입: RemoteControlExample.java

```
public class RemoteControlExample {  
    public static void main(String[] args) {  
        RemoteControl rc;  
        rc = new Television();  
        rc = new Audio();  
    }  
}
```


인터페이스 구현(Implementation)

❖ 익명 구현 객체

- 명시적인 구현 클래스 작성 생략하고 바로 구현 객체를 얻는 방법
 - 이름 없는 구현 클래스 선언과 동시에 객체 생성

```
인터페이스 변수 = new 인터페이스() {  
    //인터페이스에 선언된 추상 메소드의 실제 메소드 선언  
};
```

- 인터페이스의 추상 메소드들을 모두 재정의하는 실제 메소드가 있어야
- 추가적으로 필드와 메소드 선언 가능하나 익명 객체 안에서만 사용
- 인터페이스 변수로 접근 불가

인터페이스 구현(Implementation)

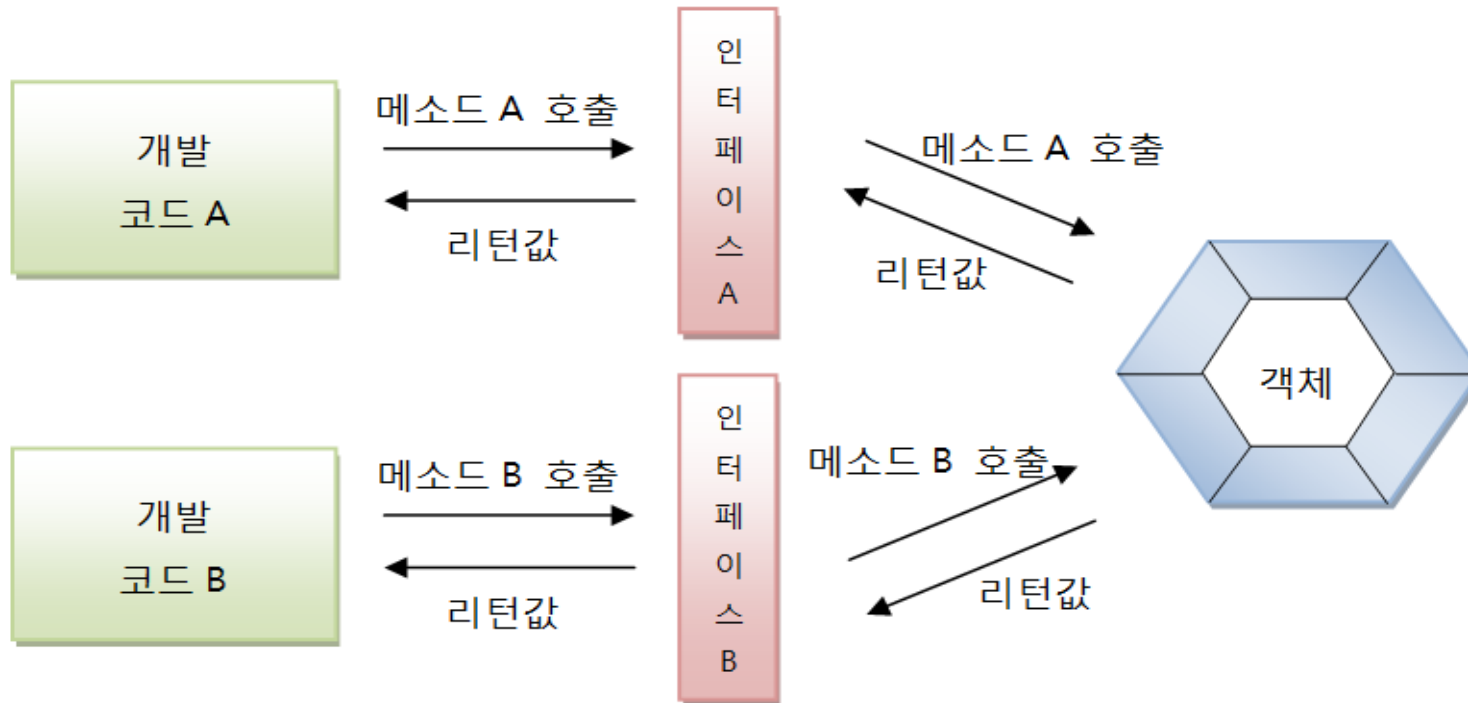
❖ 익명 구현 클래스 : RemoteControlExample.java

```
public class RemoteControlExample {  
    public static void main(String[] args) {  
        RemoteControl rc = new RemoteControl() {  
            public void turnOn() {  
                /* 실행문 */ }  
  
            public void turnOff() {  
                /* 실행문 */ }  
  
            public void setVolume(int volume) {  
                /* 실행문 */ }  
        };  
    }  
}
```

RemoteControlExample\$1.class

인터페이스 구현(Implementation)

❖ 다중 인터페이스 구현 클래스



```
public class 구현클래스명 implements 인터페이스 A, 인터페이스 B {  
    //인터페이스 A 에 선언된 추상 메소드의 실제 메소드 선언  
    //인터페이스 B 에 선언된 추상 메소드의 실제 메소드 선언  
}
```

인터페이스 구현(Implementation)

❖ 인터페이스: Searchable.java

```
public interface Searchable {  
    void search(String url);  
}
```

인터페이스 구현(Implementation)

❖ 다중 인터페이스 구현 클래스: Searchable.java

```
public class SmartTelevision implements RemoteControl, Searchable {
    private int volume;

    public void turnOn() {
        System.out.println("TV를 켭니다.");
    }

    public void turnOff() {
        System.out.println("TV를 끕니다.");
    }

    public void setVolume(int volume) {
        if (volume > RemoteControl.MAX_VOLUME) {
            this.volume = RemoteControl.MAX_VOLUME;
        } else if (volume < RemoteControl.MIN_VOLUME) {
            this.volume = RemoteControl.MIN_VOLUME;
        } else {
            this.volume = volume;
        }
        System.out.println("현재 TV 볼륨: " + volume);
    }

    public void search(String url) {
        System.out.println(url + "을 검색합니다.");
    }
}
```