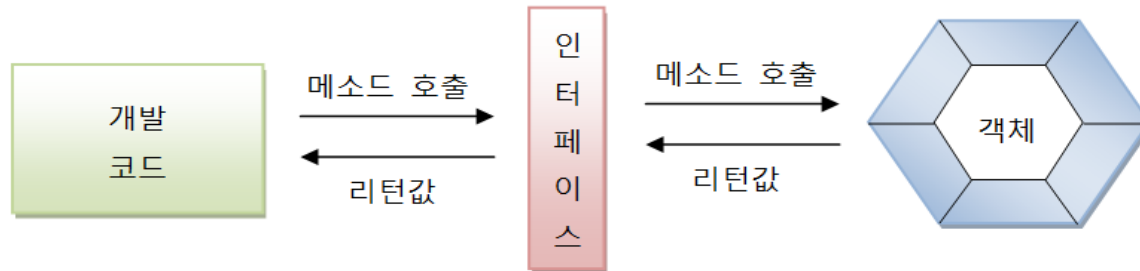


# 인터페이스 (Interface)

# 인터페이스(Interface)

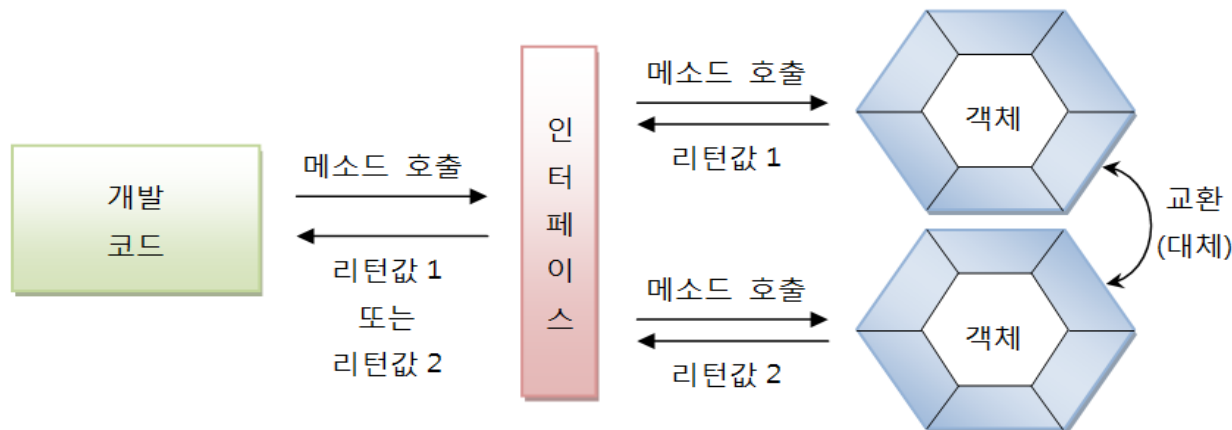
## ❖ 인터페이스란?

- 개발 코드와 객체가 서로 통신하는 접점
- 개발 코드는 인터페이스의 메소드만 알고 있으면 OK



### ○ 인터페이스의 역할

- 개발 코드가 객체에 종속되지 않게 -> 객체 교체할 수 있도록 하는 역할
- 개발 코드 변경 없이 리턴값 또는 실행 내용이 다양해질 수 있음 (다형성)



# 인터페이스(Interface)

---

## ❖ 인터페이스 선언

- 인터페이스 이름 - 자바 식별자 작성 규칙에 따라 작성
- 소스 파일 생성
  - 인터페이스 이름과 대소문자가 동일한 소스 파일 생성
- 인터페이스 선언

```
[ public ] interface 인터페이스명 { ... }
```

## 인터페이스(Interface)

---

### ❖ 인터페이스 선언: RemoteControl.java

```
public interface RemoteControl {  
  
}
```

# 인터페이스(Interface)

---

## ❖ 인터페이스 선언

- 인터페이스의 구성 멤버

```
interface 인터페이스명 {  
    //상수  
    타입 상수명 = 값;  
    //추상 메소드  
    타입 메소드명(매개변수,...);  
    //디폴트 메소드  
    default 타입 메소드명(매개변수,...) {...}  
    //정적 메소드  
    static 타입 메소드명(매개변수) {...}  
}
```

# 인터페이스(Interface)

---

## ❖ 상수 필드 선언

- 인터페이스는 상수 필드만 선언 가능
  - 데이터 저장하지 않음
- 인터페이스에 선언된 필드는 모두 `public static final`
  - 자동적으로 컴파일 과정에서 붙음
- 상수명은 대문자로 작성
  - 서로 다른 단어로 구성되어 있을 경우에는 언더 바(\_)로 연결
- 선언과 동시에 초기값 지정
  - `static { }` 블록 작성 불가 - `static { }` 으로 초기화 불가

# 인터페이스(Interface)

---

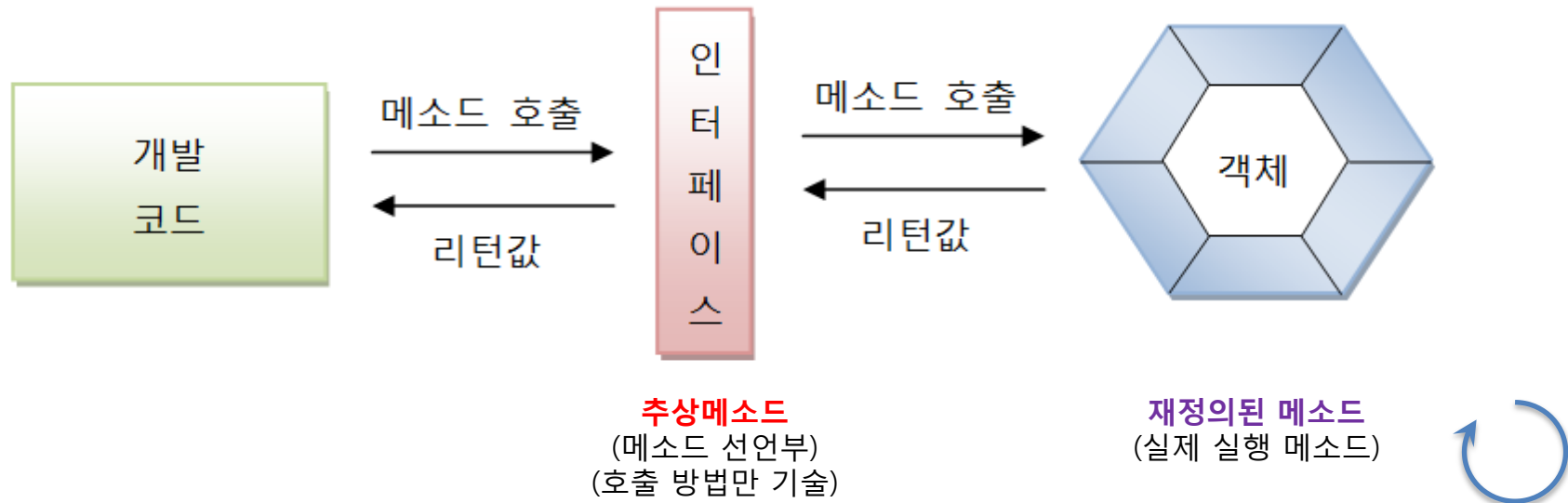
## ❖ 상수 필드 선언: RemoteControl.java

```
public interface RemoteControl {  
    int MAX_VOLUME = 10;  
    int MIN_VOLUME = 0;  
}
```

# 인터페이스(Interface)

## ❖ 추상 메소드 선언

- 인터페이스 통해 호출된 메소드는 최종적으로 객체에서 실행
  - 인터페이스의 메소드는 기본적으로 실행 블록이 없는 추상 메소드로 선언
  - `public abstract`를 생략하더라도 자동적으로 컴파일 과정에서 붙게 됨



```
[ public abstract ] 리턴타입 메소드명(매개변수, ...);
```



# 인터페이스(Interface)

---

## ❖ 추상 메서드 선언: RemoteControl.java

```
public interface RemoteControl {  
    //상수  
    int MAX_VOLUME = 10;  
    int MIN_VOLUME = 0;  
  
    //추상 메소드 - 메서드 선언부만 작성  
    void turnOn();  
    void turnOff();  
    void setVolume(int volume);  
}
```

# 인터페이스(Interface)

---

## ❖ 디폴트 메소드 선언

- 자바8에서 추가된 인터페이스의 새로운 멤버

```
[public] default 리턴타입 메소드명(매개변수, ...) { ... }
```

- 실행 블록을 가지고 있는 메소드
- default 키워드를 반드시 붙여야
- 기본적으로 public 접근 제한
  - 생략하더라도 컴파일 과정에서 자동 붙음

# 인터페이스(Interface)

---

## ❖ 메서드 선언: RemoteControl.java

```
public interface RemoteControl {  
    //상수  
    int MAX_VOLUME = 10;  
    int MIN_VOLUME = 0;  
  
    //추상 메소드  
    void turnOn();  
    void turnOff();  
    void setVolume(int volume);  
  
    //디폴트 메소드  
    default void setMute(boolean mute) {  
        if(mute) {  
            System.out.println("무음 처리합니다.");  
        } else {  
            System.out.println("무음 해제합니다.");  
        }  
    }  
}
```

# 인터페이스(Interface)

---

## ❖ 정적 메소드 선언

- 자바8에서 추가된 인터페이스의 새로운 멤버

```
[public] static 리턴타입 메소드명(매개변수, ...) { ... }
```

```
public interface RemoteControl {  
    static void changeBattery() {  
        System.out.println("건전지를 교환합니다.");  
    }  
}
```

# 인터페이스(Interface)

---

## ❖ 메서드 선언: RemoteControl.java

```
public interface RemoteControl {  
    int MAX_VOLUME = 10;  
    int MIN_VOLUME = 0;  
  
    void turnOn();  
    void turnOff();  
    void setVolume(int volume);  
  
    default void setMute(boolean mute) {  
        if(mute) {  
            System.out.println("무음 처리합니다.");  
        } else {  
            System.out.println("무음 해제합니다.");  
        }  
    }  
  
    // 정적 메소드  
    static void changeBattery() {  
        System.out.println("건전지를 교환합니다.");  
    }  
}
```