

🏠 연습문제 4 – 스마트 홈 기기 제어 시스템 (최상 난이도)

📖 문제 설명

당신은 다양한 스마트 기기를 제어할 수 있는 홈 어시스턴트 프로그램을 개발하고 있습니다. 기기들은 이름과 전원 상태를 가지고 있으며, 공통적으로 켜고 끄는 기능과 상태 출력 기능을 갖습니다. 기기마다 고유한 기능이 있으며, 고유 기능은 전원이 켜져 있어야만 동작합니다.

이번 실습에서는 배열에 저장된 스마트 기기들을 제어하고, 기기마다 고유한 기능을 **다운캐스팅을 통해 호출**해야 합니다.

💡 요구사항

1. **SmartDevice** 클래스를 만든다.

- 멤버 변수: **name**, **isOn**
- 메서드:
 - **turnOn()** → 전원을 켜고 "기기명이 켜졌습니다." 출력
 - **turnOff()** → 전원을 끄고 "기기명이 꺼졌습니다." 출력
 - **showStatus()** → "기기명 상태: ON" 또는 "기기명 상태: OFF" 출력

2. 다음 클래스를 **SmartDevice** 클래스를 상속받아 만든다:

- **SmartLight**
 - 고유 메서드: **changeColor(String color)**
 - 전원이 꺼져 있으면 동작하지 않음
 - 메시지 예: "기기명이 꺼져 있어 색상을 변경할 수 없습니다."
- **SmartSpeaker**
 - 고유 메서드: **playMusic(String song)**
 - 전원이 꺼져 있으면 동작하지 않음
- **SmartThermostat**
 - 고유 메서드: **setTemperature(int degree)**
 - 전원이 꺼져 있으면 동작하지 않음

3. 다음 순서로 동작을 구현한다:

- 배열 **devices**에 3개의 스마트 기기를 순서대로 저장한다
SmartLight, SmartSpeaker, SmartThermostat
- 모든 기기를 켜고 상태를 출력한다
- 배열의 인덱스를 활용하여 각 기기의 고유 기능을 호출한다:
 - 0번 인덱스 → 파란색 조명으로 변경
 - 1번 인덱스 → 'Jazz' 음악 재생
 - 2번 인덱스 → 24도 설정

4. 다운캐스팅을 이용해 각 고유 메서드를 호출해야 한다.

5. 메인 클래스를 제외한 모든 클래스는 default 클래스입니다.

출력 예시

- 스마트 전등이 켜졌습니다.
- 스마트 스피커가 켜졌습니다.
- 스마트 온도조절기가 켜졌습니다.
- 스마트 전등 상태: ON
- 스마트 스피커 상태: ON
- 스마트 온도조절기 상태: ON
- 스마트 전등 조명이 파란색으로 바뀝니다.
- 스마트 스피커에서 'Jazz' 음악을 재생합니다.
- 스마트 온도조절기의 온도를 24도로 설정합니다.

학습 목표

- 상속과 오버라이딩의 활용
- 객체 배열과 다형성
- 배열 인덱스를 이용한 객체 제어
- 다운캐스팅과 고유 기능 호출
- 상태 조건에 따라 기능 실행 여부 제어

메인 클래스는 아래 코드를 활용하세요.

```
public class SmartHomeTest {
    public static void main(String[] args) {
        SmartDevice[] devices = new SmartDevice[3];
        devices[0] = new SmartLight("스마트 전등");
        devices[1] = new SmartSpeaker("스마트 스피커");
        devices[2] = new SmartThermostat("스마트 온도조절기");

        // 모든 기기 전원 켜기
        for (int i = 0; i < devices.length; i++) {
            devices[i].turnOn();
        }

        // 상태 출력
        for (int i = 0; i < devices.length; i++) {
            devices[i].showStatus();
        }

        System.out.println();

        // 각 기기의 고유 기능 실행 (다운캐스팅 사용)
        SmartLight light = (SmartLight) devices[0];
        light.changeColor("파란색");

        SmartSpeaker speaker = (SmartSpeaker) devices[1];
        speaker.playMusic("Jazz");
    }
}
```

```
SmartThermostat thermostat = (SmartThermostat) devices[2];  
thermostat.setTemperature(24);  
    }  
}
```