```
import time
from threading import Timer

# Wi-Fi 네트워크 확인
class WiFi:
    def __init__(self, connected=True):
        self.connected = connected

    def check_connection(self):
        return self.connected

# IoT 장치: 조명
class IoTDevice1:
    def __init__(self):
        self.state = "OFF"
        self.timer = None

    def turn_on(self):
        self.state = "ON"
        print("조명이 켜졌습니다.")

    def turn_off(self):
        self.state = "OFF"
        print("조명이 꺼졌습니다.")

    def set_timer(self, seconds):
        if self.timer:
            self.timer.cancel()
        self.timer = Timer(seconds, self.turn_off)
        self.timer.start()
        print(f"조명이 {seconds}초 후 꺼집니다.")

# IoT 장치: 에어컨
class IoTDevice2:
    def __init__(self):
        self.state = "OFF"
        self.mode = "냉방"
        self.timer = None

    def turn_on(self):
        self.state = "ON"
        print(f"에어컨이 켜졌습니다. 모드: {self.mode}")

    def turn_off(self):
        self.state = "OFF"
        print("에어컨이 꺼졌습니다.")

    def set_mode(self, mode):
        self.mode = mode
        print(f"에어컨 모드가 {self.mode}로 설정되었습니다.")

    def set_timer(self, seconds):
        if self.timer:
            self.timer.cancel()
```

```
            self.timer = Timer(seconds, self.turn_off)
            self.timer.start()
            print(f"에어컨이 {seconds}초 후 꺼집니다.")

# 서버: 명령 처리
class Server:
    def __init__(self):
        self.iot_device1 = IoTDevice1()
        self.iot_device2 = IoTDevice2()

    def handle_request(self, device, action, timer=None, mode=None):
        if device == "조명":
            if action == "ON":
                self.iot_device1.turn_on()
            elif action == "OFF":
                self.iot_device1.turn_off()
            if timer:
                self.iot_device1.set_timer(timer)
        elif device == "에어컨":
            if action == "ON":
                self.iot_device2.turn_on()
            elif action == "OFF":
                self.iot_device2.turn_off()
            if mode:
                self.iot_device2.set_mode(mode)
            if timer:
                self.iot_device2.set_timer(timer)

# 앱: 사용자와 상호작용
class App:
    def __init__(self):
        self.wifi = WiFi()
        self.server = Server()

    def send_command(self, device, action, timer=None, mode=None):
        if self.wifi.check_connection():
            print(f"{device}에 대한 명령을 처리 중...")
            self.server.handle_request(device, action, timer, mode)
        else:
            print("네트워크 연결이 실패했습니다. 같은 Wi-Fi 네트워크에 연결해주세요.")

# 테스트 시나리오
if __name__ == "__main__":
    app = App()

    # 사용자 명령: 조명 켜고 타이머 5초 설정
    app.send_command("조명", "ON", timer=5)

    # 사용자 명령: 에어컨 켜고, 모드 냉방, 타이머 10초 설정
    app.send_command("에어컨", "ON", timer=10, mode="냉방")

    # 잠시 기다린 후에, 타이머가 작동하고 장치들이 꺼지는 것을 확인
    time.sleep(12)  # 타이머가 작동하도록 기다림
```