

테스트 설계 기법에 대한 자세한 설명은 본 가이드의 파트 II '5장 테스트 케이스 구현 가이드'를 참고하기를 바란다.

#### 4.4.5 단위 테스트 케이스 작성

김대리는 [표 I.4.3]의 리스크 항목 중 자신이 개발하고 있는 오븐(oven) 프로그램의 단위 테스트 케이스를 작성하기로 했다. 오븐 프로그램은 리스크가 가장 높은 STA 영역이며 테스트 종료 기준은 결정 커버리지(decision coverage) 100%이다. 김대리는 먼저 오븐 소스 코드를 분석했다. [그림 I.4.12]는 오븐 프로그램의 소스 코드다.

```
public class Oven {

    final int IDLE = 0, SUSPENDED=1, COOKING=2, COMPLETED=3;
    final int OPEN = 1, CLOSE=0;

    int power, dish_qty, food_wgt;
    boolean special_food=false;
    static Oven oven;

    int time=0, leftTime, door_st, currentState=IDLE;
    static String cookingTime;

    public void checkPower()
    {

        if((power<100) || (currentState != COOKING))
            power = 100;

        else if ((power < 200) && (time < 30))
            power = 300;

        else
            power = 200;

        if (dish_qty > 3 || (dish_qty > 0 && food_wgt > 1000) || special_food == true)
            time = 60;

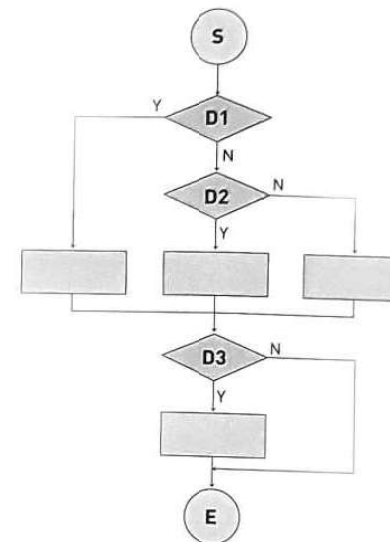
    } // end of public void checkPower()

} // end of public class Oven
```

[그림 I.4.12] 단위 테스트 케이스 설계 - Oven.java 소스 코드

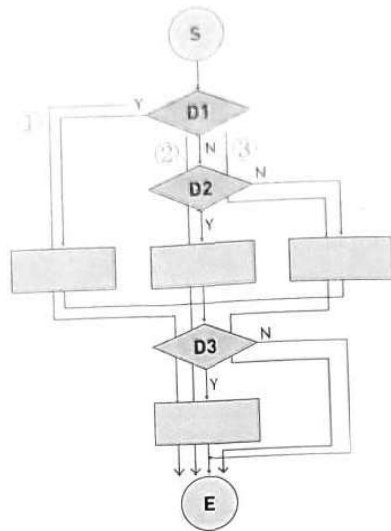
'Oven.java' 소스 코드를 분석한 김대리는 결정 테스트를 위해 [그림 I.4.13]과 같이 'checkPower()' 메소드의 제어 흐름도를 작성했다. D1, D2, D3는 프로그램 소스 코드에서 결정문(분기문)인 if 조건문을 의미한다.

결정 테스트 기법에 대한 자세한 설명은 본 가이드 파트 II의 '5장 테스트 케이스 구현 가이드'를 참고하기를 바란다.



[그림 I.4.13] 단위 테스트 케이스 설계 - checkPower()의 제어 흐름도

김대리는 [그림 I.4.13]의 제어 흐름도를 분석해 결정 커버리지를 100% 만족하는 최소 테스트 경로 3개를 [그림 I.4.14]와 같이 도출했다. 결정 커버리지 100%는 '프로그램 소스 내의 모든 결정문(분기문)에서 생성 가능한 모든 결과(참 - Yes, True, 거짓 - No, False)를 누락 없이 한 번씩은 테스트 한다'는 의미다. 즉, 테스트 케이스를 설계할 때 모든 분기를 최소 한 번은 지나도록 작성한다. 결정 커버리지 100%를 달성하기 위한 최소 테스트 경로가 3개인 경우, 경로에 맞는 3개의 테스트 케이스를 작성하면 결정 커버리지를 100% 만족한다.



[그림 1.4.14] checkPower() 결정 커버리지 100% 테스트 경로

김대리는 [그림 1.4.14]에 도출한 테스트 경로를 근거로 테스트 케이스를 작성하기로 했다. 각 테스트 경로를 분석하면 필요한 입력값을 찾을 수 있다. 먼저 김대리는 결정문과 테스트 경로(테스트 케이스)와 커버리지 관계를 [표 1.4.4]와 같이 정리했다. 3개의 테스트 경로(테스트 케이스)가 모든 결정문의 분기를 커버하는 것을 [표 1.4.4]에서 확인할 수 있다.

결정문 (분기문)	분기 (연산 결과)	테스트 경로 (테스트 케이스) 1	테스트 경로 (테스트 케이스) 2	테스트 경로 (테스트 케이스) 3
D1	Y	✓		✓
	N		✓	
D2	Y			✓
	N		✓	
D3	Y	✓	✓	✓
	N			

[표 1.4.4] checkPower() 결정문과 테스트 경로(테스트 케이스)의 커버리지 관계

다음 단계로 김대리는 [표 1.4.4]를 기반으로 [표 1.4.5]와 같이 논리적 테스트 케이스를 도출했다.

결정문 (분기문)	분기 (연산 결과)	개별 조건식	테스트 케이스 1	테스트 케이스 2	테스트 케이스 3
D1	Y	T or F	✓		✓
	N	F or F		✓	
D2	Y	T and T		✓	
	N	T and F			✓
D3	Y	F or (T and T) or F	✓	✓	
	N	F or (F and T) or F			✓

[표 1.4.5] checkPower() 논리적 테스트 케이스 도출

이제 단위 테스트 설계 마지막 단계로 김대리는 [표 1.4.5]를 기반으로 [표 1.4.6]과 같이 실제 테스트 가능한 입력 데이터를 고려해 물리적 테스트 케이스를 작성했다.

결정문 (분기문)	입력 데이터	테스트 케이스 1	테스트 케이스 2	테스트 케이스 3
D1	power	99	101	101
	currentState	COOKING	COOKING	COOKING
D2	time	-	28	31
D3	dish_qty	1	1	0
	food_wgt	1001	1001	1001
	special_food	false	false	false
기대결과	power	100	300	200
	time	60	60	31

[표 1.4.6] checkPower()의 물리적 테스트 케이스 작성

산출물	체크리스트	필수/선택
테스트 계획서	· 단위 테스트 계획서가 있는가?	필수
	· 단위 테스트의 목적이나 목표를 구체적으로 정의했는가?	필수
	· 단위 테스트의 테스트 대상 품질 특성과 품질 목표를 기술했는가?	선택
	· 단위 테스트 전략을 정의했는가?	필수
	· 단위 테스트 일정 및 노력이 계획됐는가?	필수
	· 단위 테스트에 필요한 역할과 책임을 정의했는가?	필수
테스트 설계 명세서	· 단위 테스트 시작 기준과 종료 기준을 명시했는가?	필수
	· 테스트 계획서에 있는 테스트 전략 내용을 반영했는가?	필수
	· 단위 테스트 케이스 설계 기법이나 방법을 명시했는가?	선택
	· 단위 테스트 케이스에 테스트에 필요한 입력 데이터, 기대결과 값을 명시했는가?	필수
	· 단위 테스트 계획에서 명시한 테스트 커버리지 목표를 만족할 수 있도록 테스트 케이스를 작성했는가?	선택
	· 에러 처리, 예외 처리, 이상 처리 등과 같은 비정상 상황을 위한 테스트 케이스를 포함했는가?	선택
테스트 종료 보고서	· 단위 테스트 종료 시 종료 기준을 만족했는가?	필수
	· 단위 테스트 계획에서 명시한 테스트 커버리지 목표를 달성했는가?	필수
	· 단위 테스트 계획에서 명시한 테스트 활동을 모두 수행하고 테스트를 종료했는가?	선택
	· 단위 테스트 관련 메트릭스(metrics) 및 단위 테스트 활동 내용을 요약해 보고했는가?	선택

[표 1.4.7] 단위 테스트 체크리스트 예시

## 4.5.3 변형 조건/결정 커버리지(MC/DC) 테스트 적용 사례

변형 조건/결정 커버리지(MC/DC, Modified Condition/Decision Coverage)는 항공, 원자력, 자동차, 철도, 의료 등의 분야처럼 기능 안전 우선 소프트웨어에서 요구하는 강도 높은 커버리지가, [그림 1.4.12]의 'Oven.java' 소스 코드를 이용해 변형 조건/결정 커버리지(이하 MC/DC) 테스트 방법을 알아보자.

먼저 MC/DC 테스트를 위해 [표 1.4.8]과 같이 각 분기에서 MC/DC 조건을 만족하는 논리 조합을 도출한다.

결정문 (분기문)	조건/결정 커버리지(MC/DC) 조합
D1: A or B	(1,0), (0,1), (0,0)
D2: A and B	(1,0), (0,1), (1,1)
D3: A or (B and C) or D	(1,1,0,0), (0,1,1,0), (0,0,1,1), (0,0,1,0), (0,1,0,0)

[표 1.4.8] checkPower()의 MC/DC 조합

MC/DC 조합을 도출한 후 [표 1.4.9]와 같이 11개의 모든 MC/DC 조합이 실행될 수 있도록 논리적 테스트 케이스를 설계한다. 5개의 테스트 케이스로 모든 MC/DC 조합을 커버할 수 있다. 더 자세한 내용은 본 가이드 파트 II '5장 테스트 케이스 구현 가이드'를 참고하기를 바란다.

결정문 (분기문)	분기 (연산 결과)	테스트 케이스 1	테스트 케이스 2	테스트 케이스 3	테스트 케이스 4	테스트 케이스 5
D1(1,0)	Y	√				
D12(0,1)	Y		√			
D13(0,0)	N			√	√	√
D21(1,0)	N			√		
D22(0,1)	N				√	
D23(1,1)	Y					√
D31(1,1,0,0)	Y	√				
D32(0,1,1,0)	Y		√			
D33(0,0,1,1)	Y			√		
D34(0,0,1,0)	N				√	
D35(0,1,0,0)	N					√

[표 1.4.9] checkPower()의 MC/DC 논리적 테스트 케이스 설계