

# Software Testing HW3 - 測試覆蓋標準實作

313552024 楊宇盛

## 實驗目標

實作三種不同的測試覆蓋標準於 Thermostat 類別：

- 1. PC (Predicate Coverage) - 謂詞覆蓋
- 2. CC (Clause Coverage) - 子句覆蓋
- 3. CACC (Correlated Active Clause Coverage) - 相關主動子句覆蓋

## 程式碼分析

Thermostat 類別主要邏輯

Thermostat 類別中的 `turnHeaterOn` 方法包含一個複合條件：

```
this.heaterOn = (this.curTemp < targetTemp - this.thresholdDiff) &&
                 (this.timeSinceLastRun >= this.minLag);
```

這個複合條件可以分解為兩個子句：

- P1: `this.curTemp < targetTemp - this.thresholdDiff`
  - 檢查當前溫度是否低於目標溫度減去閾值
- P2: `this.timeSinceLastRun >= this.minLag`
  - 檢查距離上次運行的時間是否超過最小延遲

## 測試案例設計

### 1. Predicate Coverage (PC)

PC 需要測試整個複合條件的真假值。

測試案例	P1 && P2	預期結果	實際結果
PC-1	true	true	✓ 通過
PC-2	false	false	✓ 通過

### 2. Clause Coverage (CC)

CC 需要測試每個子句的所有可能組合。

測試案例	P1	P2	預期結果	實際結果
CC-1	true	true	true	✓ 通過

測試案例	P1	P2	預期結果	實際結果
CC-2	true	false	false	✓ 通過
CC-3	false	true	false	✓ 通過
CC-4	false	false	false	✓ 通過

3. Correlated Active Clause Coverage (CACC)

CACC 需要確保每個子句都能獨立影響最終結果。

測試案例	P1 (主動)	P2	預期結果	實際結果
CACC-1	true	true	true	✓ 通過
CACC-2	false	true	false	✓ 通過
CACC-3	true	true	true	✓ 通過
CACC-4	true	false	false	✓ 通過

測試實作

測試程式碼使用 Jest 框架實現，主要包含以下部分：

1. 環境設置：

```
beforeEach(() => {
  thermostat = new Thermostat();
  settings = new ProgrammedSettings();
  thermostat.period = Period.DAY;
  thermostat.day = DayType.WEEKDAY;
  thermostat.thresholdDiff = 2;
});
```

2. PC 測試案例：

```
test('PC-1: P1 && P2 = true', () => {
  thermostat.setCurrentTemp(60);
  thermostat.timeSinceLastRun = 10;
  const result = thermostat.turnHeaterOn(settings);
  expect(result.heaterOn).toBe(true);
});
```

3. CC 測試案例：

```
test('CC-1: P1=true, P2=true', () => {
  thermostat.setCurrentTemp(60);
```

```

    thermostat.timeSinceLastRun = 10;
    const result = thermostat.turnHeaterOn(settings);
    expect(result.heaterOn).toBe(true);
  });

```

#### 4. CACC 測試案例：

```

test('CACC-1: P1 主動 (true), P2=true', () => {
  thermostat.setCurrentTemp(60);
  thermostat.timeSinceLastRun = 10;
  const result = thermostat.turnHeaterOn(settings);
  expect(result.heaterOn).toBe(true);
});

```

## 測試結果

所有測試案例都成功通過：

```

PASS ./Thermostat.test.js
Predicate Coverage Tests
  ✓ PC-1: P1 && P2 = true
  ✓ PC-2: P1 && P2 = false
Clause Coverage Tests
  ✓ CC-1: P1=true, P2=true
  ✓ CC-2: P1=true, P2=false
  ✓ CC-3: P1=false, P2=true
  ✓ CC-4: P1=false, P2=false
Correlated Active Clause Coverage Tests
  ✓ CACC-1: P1 主動 (true), P2=true
  ✓ CACC-2: P1 主動 (false), P2=true
  ✓ CACC-3: P1=true, P2 主動 (true)
  ✓ CACC-4: P1=true, P2 主動 (false)

```

測試統計：

- 測試套件：1 個通過
- 測試案例：10 個通過
- 執行時間：0.182 秒

## 結論

通過實作這三種不同的測試覆蓋標準，我們可以得出以下結論：

1. PC 提供了最基本的測試覆蓋，但可能會遺漏一些重要的組合情況
2. CC 提供了更全面的測試覆蓋，確保每個子句的所有可能組合都被測試
3. CACC 不僅測試了所有組合，還特別關注每個子句作為主動子句時的情況，提供了最嚴格的測試標準

所有測試都成功通過，證明了 Thermostat 類別的 `turnHeaterOn` 方法在各種不同的條件組合下都能正確運作，符合設計規格。