

1. Sie testen ein elektronisches Verkaufssystem für Automobile. Im Rahmen des Modultests wird die Methode `calculatePrice` betrachtet. Diese Methode berechnet den Gesamtpreis eines Autos, unter Berücksichtigung von Rabattaktionen für Sonderausstattungen.

```
// Elektronisches Verkaufssystem fuer Automobile (Spillner/Linz)
// Rabattberechnung
public class Angebot {
    double calculatePrice(double baseprice, double specialprice,
                          double extraprice, int extras, double discount) {
        double addon_discount;
        double result;
        if (extras >= 3)
            addon_discount = 10;
        else if (extras >= 5)
            addon_discount = 15;
        else
            addon_discount = 0;
        if (discount > addon_discount)
            addon_discount = discount;
        result = baseprice / 100.0 * (100.0 - discount) + specialprice
            + extraprice / 100.00 * (100.0 - addon_discount);
        return result;
    }
}
```

Im Rahmen des Black-Box-Tests wurden die beiden folgenden Testfälle hergeleitet:

1. `calculatePrice(10000.00, 2000.00, 1000.00, 3, 0) = 12900.00`
2. `calculatePrice(25500.00, 3450.00, 6000.00, 6, 0) = 34050.00`

Diese Testfälle sind fehlerfrei absolviert worden. Laut Qualitätssicherungsplan ist für dieses Projekt eine Zweigüberdeckung von > 80 Prozent gefordert. Reichen die beiden Testfälle aus? Falls nicht, stellen Sie weitere Testfälle auf, um die geforderte Zweigüberdeckung zu erreichen! Arbeiten Sie dabei mit einem kollabierten Kontrollflussgraphen.