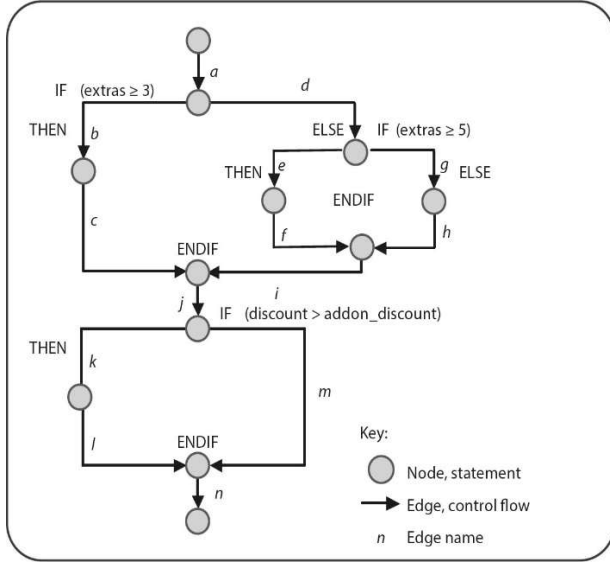


Yazılım Kalite Güvencesi ve Testi

1) Kod parçasına ait graf verilmektedir.



```
double calculate_price (double baseprice, double specialprice,
                        double extraprice, int extras, double discount) {
    double addon_discount, 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-discount) + specialprice +
        extraprice/100.0*(100-addon_discount);
    return result;
}
```

i) Tüm olası "coverage paths" yazın.

ii) "decision coverage" yüzdesini

test case1: $price = calculate_price(10000.00, 2000.00, 1000.00, 3, 0)$

test case2: $price = calculate_price(25500.00, 3450.00, 6000.00, 6, 0)$ için **nedenini kod üzerinden açıklayarak** hesaplayın (cevabınızı sadece değer olarak verilmesi değerlendirilmez).

price değişkeni testin beklenen(expected) değeridir.

iii) "branch coverage" yüzdesini

test case1: $price = calculate_price(10000.00, 2000.00, 1000.00, 3, 0)$

test case2: $price = calculate_price(25500.00, 3450.00, 6000.00, 6, 0)$ için **nedenini kod üzerinden açıklayarak** hesaplayın (cevabınızı sadece değer olarak verilmesi değerlendirilmez).

price değişkeni testin beklenen(expected) değeridir.

iv) Bu problemde decision coverage ~%83, branch coverage ~%86 oranında elde edilebilir. Bunun için test sürecindeki değişikliği nasıl önerirsiniz?

v) Bu probleme "statement coverage" test tekniğini nasıl uygularsınız?

2)

```
public static int soru1 (int[] x) {
    int count = 0;
    for (int i=0; i < x.length; i++)
        { if (x[i]%2 == 1 || x[i] > 0)
            { count++; }
        }
    return count; }
}
```

i) Bu kod parçası için test case $x=[3, -2, 0, 1, 4]$ ve beklenen değer 3 olsun. Bu değeri elde etmede kod parçasında "fault" var mıdır? Eğer varsa nedenini belirterek düzeltin. "fault" yoksa beklenen sonucu nasıl verdiğini gösterin.

ii) Yukarıdaki kod parçası için öyle bir "test case" yazın ki "fault" olmasın (Değerlendirme için sadece test case yeterli değildir, açıklanmalıdır).

3) Aşağıdaki kod parçası TDD örneğidir. Bu geliştirme yöntemini kod üzerinden açıklayınız

```
@Test
public final void whenEmptyStringIsUsedThenReturnValueIs0() {
    Assert.assertEquals(0, StringCalculator.add(""));
}

//Implementation
public static final int add(final String numbers) { // Changed void to int
    String[] numbersArray = numbers.split(",");
    if (numbersArray.length > 2) {
        throw new RuntimeException("Up to 2 numbers separated by comma (,) are allowed");
    } else {
        for (String number : numbersArray) {
            if (!number.isEmpty()) {
                Integer.parseInt(number);
            } } } return 0;
}
```