CS 4472B Assignment 2 - White Box Testing

Department of Computer Science

Western University

March 17th, 2022

Prof. Kostas Kontogiannis

Group 20

Amanpreet Gill

Ositadinma Arimah

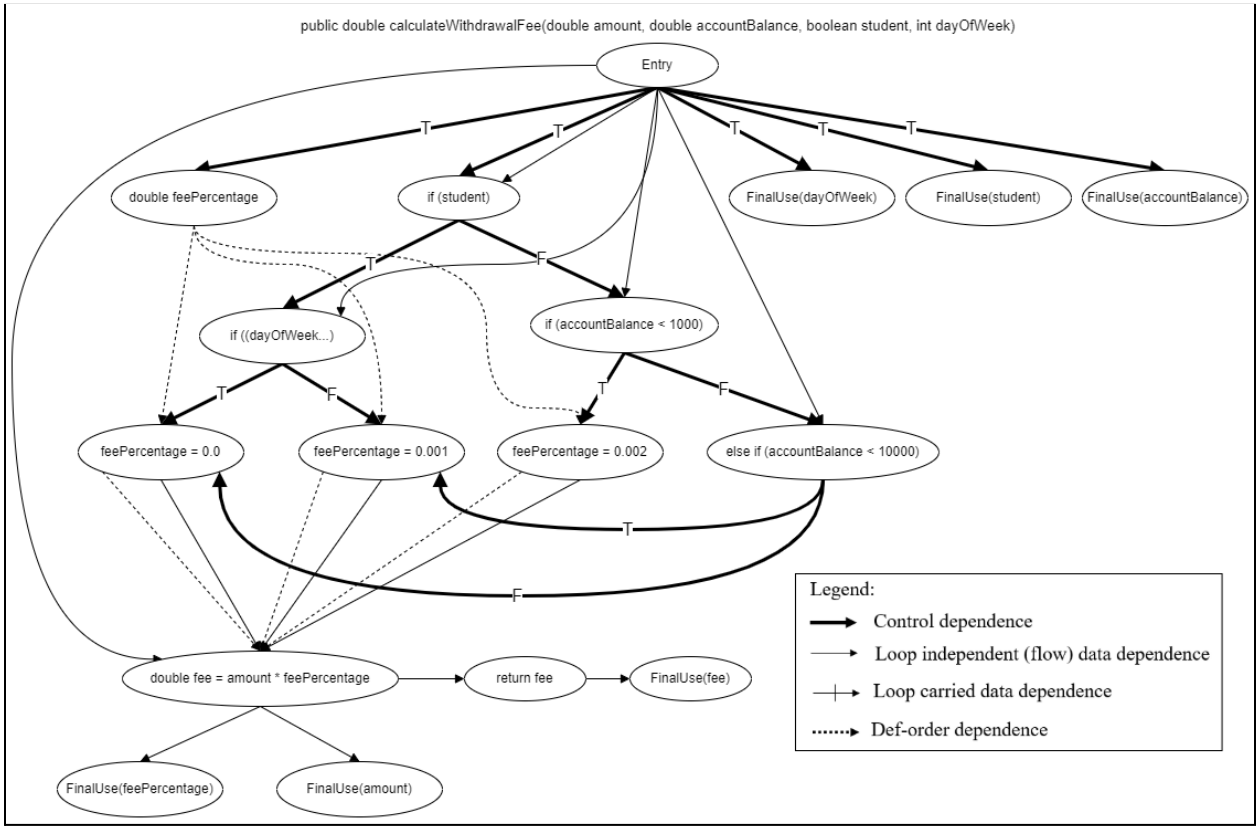
Atulpreet Johar

Bohr Deng

1A - Withdrawal Fee Calculation

File: src/test/java/bank.utils.FeesCalculatorWithdrawalCoverageTest.java

Withdrawal Function PDG



According to the constructed PDG, the slice for FinalUse(fee) is basically the whole program/function except the other final use vertices.

Slice of FinalUse(fee):

```
public double calculateWithdrawalFee(double amount, double
accountBalance, boolean student, int dayOfWeek) {
      1     double feePercentage;
      2     if (student) {
            3     if ((dayOfWeek == Calendar.SATURDAY) || (dayOfWeek ==
Calendar.SUNDAY)) {
                  4     feePercentage = 0.0;
                  } else {
                  5     feePercentage = 0.001;
                  }
            } else {
            6     if (accountBalance < 1000.00) {
                  7     feePercentage = 0.002;
            8     } else if (accountBalance < 10000.00) {
                  9     feePercentage = 0.001;
                  } else {
                  10    feePercentage = 0.0;
                  }
            }

      11    double fee = amount * feePercentage;
      12    return fee;
      }
```

For test statement coverage, we only need enough test cases to exercise all (12) statements:

Test Case	amount	account Balance	student	dayOfWeek	Statements covered	Correct output	Actual Output
1	1	1000	true	Calendar.Saturday	1,2,3,4,11,12	0	0
2	1	1000	true	Calendar.Monday	1,2,3,5,11,12	0.001	0.001
3	1	394	false	Calendar.Tuesday	1,2,6,7,11,12	0.002	0.002
4	1	9001	false	Calendar.Wednesday	1,2,6,8,9,11,12	0.001	0.001
5	1	100000	false	Calendar.Thursday	1,2,6,8,10,11,12	0	0

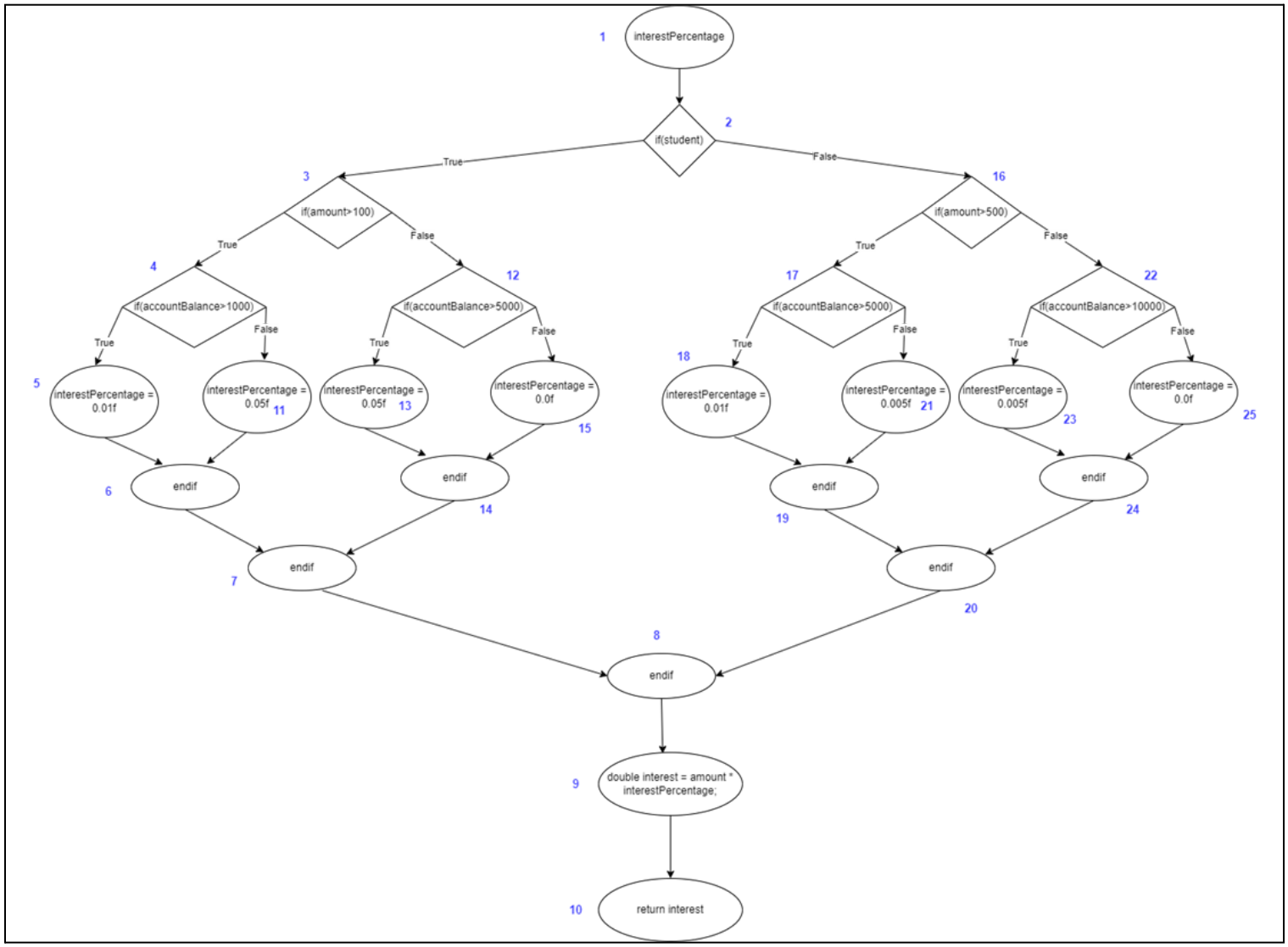
The results from the test case are as follows:

Test Case	**Result**
1	Pass
2	Pass
3	Pass
4	Pass
5	Pass

1B - DU-Path Testing

File: src/test/java/bank.utils.FeesCalculatorDUPathTest.java

The Program Graph for the Deposit class is:



*** Note: Nodes are taken to represent individual statements. This includes conditional expressions and fragments of statements. ***

The definition and usage nodes for the variable “interestPercentage” is:

Variable	**Definition Nodes**	**Usage Nodes**
interestPercentage	1, 5, 11, 13, 15, 18, 21, 23, 25	9

A definition-use path, du-path, with respect to a variable v is a path whose **first node is a defining node** for v, and its **last node is a usage node for v.**

Therefore, the du-paths for interestPercentage are:

Path Number	**Path**
1	**1**-2-3-4-5-6-7-8-**9**-**10**
2	**1**-2-3-4-11-6-7-8-**9**-**10**
3	**1**-2-3-12-13-14-7-8-**9**-**10**
4	**1**-2-3-12-15-14-7-8-**9**-**10**
5	**1**-2-16-17-18-19-20-8-**9**-**10**
6	**1**-2-16-17-21-19-20-8-**9**-**10**
7	**1**-2-16-22-23-24-20-8-**9**-**10**
8	**1**-2-16-22-25-24-20-8-**9**-**10**

For each such path create the test cases that perform condition coverage on the resulting set of the statements in each such DUPath. Therefore, the test cases are as follows:

TC	amount	accountBalance	student
1	>100	>1000	T
2	>100	<=1000	T
3	<=100	>5000	T
4	<=100	<=5000	T
5	>500	>5000	F
6	>500	<=5000	F
7	<=500	>10000	F
8	<=500	<=10000	F

Consequently, the test cases with specific variables are as follows:

TC	amount	accountBalance	student	Expected TC Outcome
1	150	1200	True	Pass
2	150	950	True	Pass

3	50	5500	True	Pass
4	50	4900	True	Fail
5	510	5500	False	Pass
6	510	4900	False	Pass
7	480	12000	False	Pass
8	480	9500	False	Pass

The results from the test case are as follows:

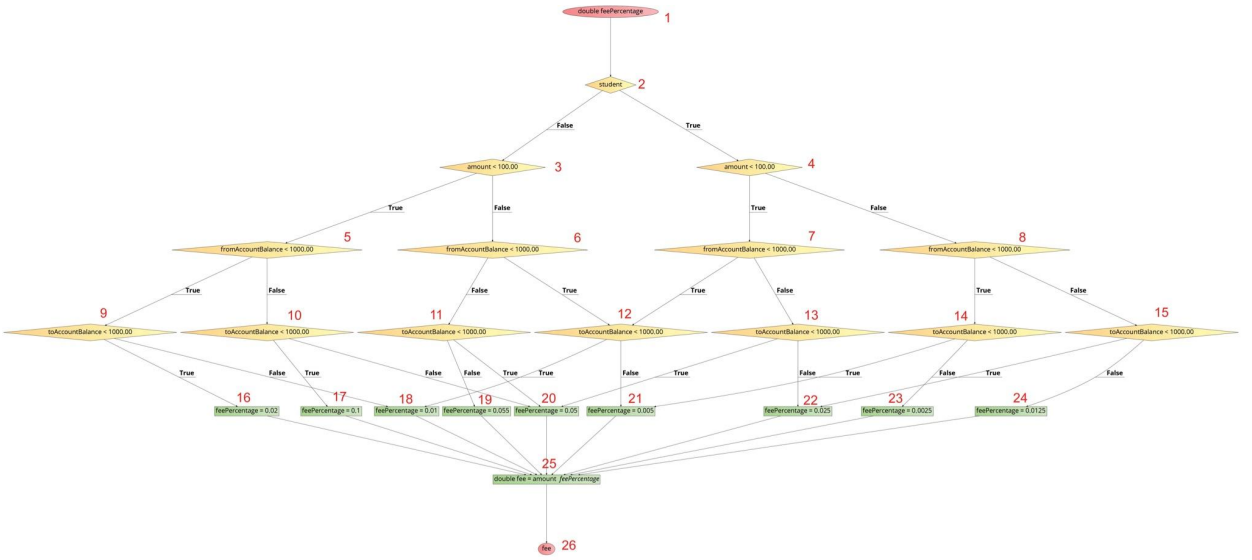
TC	**Result**
1	Pass
2	Pass
3	Pass
4	Fail
5	Pass
6	Pass
7	Pass
8	Pass

1C - Basis Path Testing

File: src/test/java/bank.utils.FeesCalculatorBasisPathTest.java

Control Flow Graph

Attached as a separate file for readability.



Cyclomatic Complexity

$V(G) = E - N + 2$ where E is a graph edge, and N is a graph node.
Based on the CFG we can see that there are 39 edges and 26 nodes, and therefore the complexity is:
$V(G) = 39 - 26 + 2 = 15$

This means we need 15 test cases to cover all basis paths. I have added an extra test path for completeness but path 7-10 could be reduced to 7 and 10 or 8 and 9.

Test Cases

	Path	student	amount	fromAccountBalance	toAccountBalance	feePercentage
1	1,2,3,5,9, 16,25,26	False	50	500	500	0.02
2	1,2,3,5,9, 18,25,26	False	50	500	1500	0.01

3	1,2,3,5,10,17,25,26	False	50	1500	500	0.1
4	1,2,3,5,10,20,25,26	False	50	1500	1500	0.05
5	1,2,3,6,12,18,25,26	False	150	500	500	0.01
6	1,2,3,6,12,21,25,26	False	150	500	1500	0.005
7	1,2,3,6,11,20,25,26	False	150	1500	500	0.05
8	1,2,3,6,11,19,25,26	False	150	1500	1500	0.0055
9	1,2,4,7,12,18,25,26	True	50	500	500	0.01
10	1,2,4,7,12,21,25,26	True	50	500	1500	0.005
11	1,2,4,7,13,20,25,26	True	50	1500	500	0.05
12	1,2,4,7,13,22,25,26	True	50	1500	1500	0.025
13	1,2,4,8,14,21,25,26	True	150	500	500	0.005
14	1,2,4,8,14,23,25,26	True	150	500	1500	0.0025
15	1,2,4,8,15,22,25,26	True	150	1500	500	0.025
16	1,2,4,8,15,24,25,26	True	150	1500	1500	0.0125

Test Results

Test ID	Expected Fee	Actual Fee	Pass/Fail

1	1.00	1.00	P
2	0.50	0.5	P
3	5.00	5.00	P
4	2.50	2.50	P
5	1.50	1.50	P
6	0.75	0.75	P
7	7.50	7.50	P
8	8.25	8.25	P
9	0.50	0.50	P
10	0.25	0.25	P
11	2.50	2.50	P
12	1.25	1.25	P
13	0.75	0.75	P
14	0.375	0.375	P
15	3.75	3.75	P
16	1.875	1.875	P