**public** **class** Problem1 {

**public** **double** calcBalance(**double** balance) {

**int** index;

**double** rate[]= {0.0,0.0,0.0155,0.0255,0.03155,0.0325};

**double** fee[]= {500.0,150.0,0.0,0.0,0.0,0.0};

**double** credit [] = {0.0,0.0,0.0,0.0,0.0,100.0};

**if**(balance<0.0)

index=0;

**else**

**if**(Math.*abs*(balance-0.0)<0.005)

index=1;

**else**

**if**(balance<800.0)

index=2;

**else**

**if**(balance<3\_500.0)

index=3;

**else**

**if**(balance<250\_000.0)

index=4;

**else**

index=5;

**return** balance \* (1.0 + rate[index])-fee[index]+ credit[index];

}

}

Test: ---

**import** **static** junitparams.JUnitParamsRunner.*$*;

**import** **static** org.junit.Assert.\*;

**import** junitparams.JUnitParamsRunner;

**import** junitparams.Parameters;

**import** org.junit.Before;

**import** org.junit.Test;

**import** org.junit.runner.RunWith;

@RunWith(JUnitParamsRunner.**class**)

**public** **class** Problem1Test {

**private** Problem1 prob1;

@Before

**public** **void** setUp () **throws** Exception {

prob1 = **new** Problem1();

}

@SuppressWarnings("unused")

**private** **static** **final** Object[] parametersProblem1Test () {

**return** *$*(

// Parameters are: (1)

// 1=balance, 2=expBal

// Test case 1

*$*(-0.01, -500.01),

// Test case 2

*$*(0.00,-150.00),

// Test case 3

*$*(799.99, 812.38),

// Test case 4

*$*(3\_499.99, 3\_589.23),

// Test case 5

*$*(249\_999.99, 257\_887.48),

// Test case 6

*$*(2\_50\_000.00, 258\_225.00),

// Test case 7

*$*(-1\_000.00, -1\_500.00),

// Test case 8

*$*(0.01, 0.01),

// Test case 9

*$*(800.00, 820.40),

// Test case 10

*$*(3\_500.00, 3\_610.42),

// Test case 11

*$*(3\_00\_000.00, 309\_850.00)

);

}

@Test

@Parameters(method="parametersProblem1Test")

**public** **void** test(**double** balance, **double** expBal) {

*assertEquals*(expBal, prob1.calcBalance(balance),0.01);

}

}