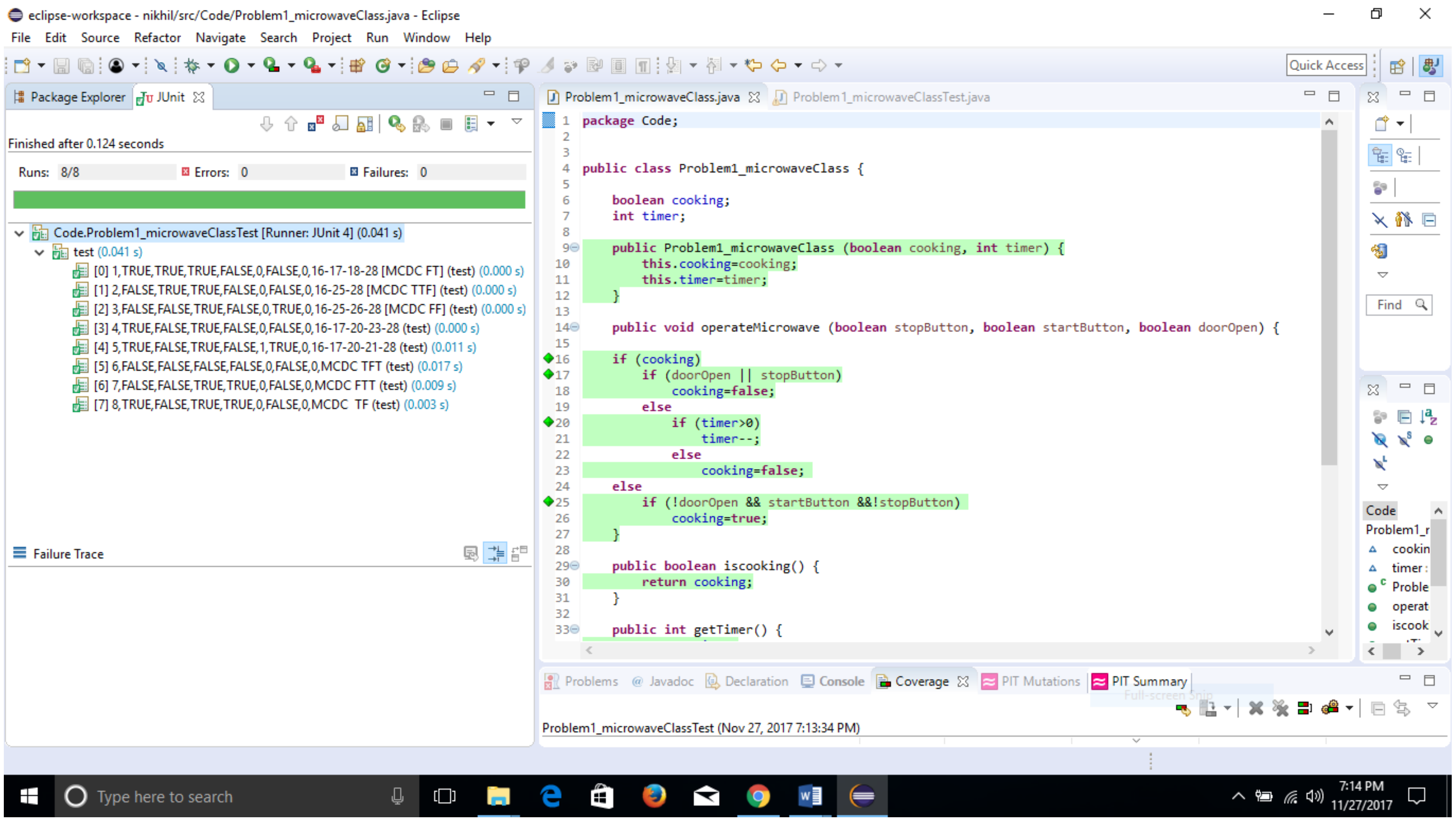
Nikhil Kajrekar, UTA ID: 1001552488

Problem 1:

Test Case Table:

Test Case Number	Inputs					Outputs		Basis Path
	cooking	stopButton	StartButton	doorOpen	timer	cooking	timer	
1	TRUE	TRUE	TRUE	FALSE	0	FALSE	0	16-17-18-28 (MCDC FT)
2	FALSE	TRUE	TRUE	FALSE	0	FALSE	0	16-25-28 (MCDC TTF)
3	FALSE	FALSE	TRUE	FALSE	0	TRUE	0	16-25-26-28 (MCDC FF)
4	TRUE	FALSE	TRUE	FALSE	0	FALSE	0	16-17-20-23-28
5	TRUE	FALSE	TRUE	FALSE	1	TRUE	0	16-17-20-21-28
6	FALSE	FALSE	FALSE	FALSE	0	FALSE	0	MCDC TFT
7	FALSE	FALSE	TRUE	TRUE	0	FALSE	0	MCDC FTT
8	TRUE	FALSE	TRUE	TRUE	0	FALSE	0	MCDC TF

JUnit pass indicator (green bar expanded) and JaCoCo statement green source line annotations:



```
Finished after 0.124 seconds
Runs: 8/8    Errors: 0    Failures: 0

Code.Problem1_microwaveClassTest [Runner: JUnit 4] (0.041 s)
  test (0.041 s)
    [0] 1,TRUE,TRUE,TRUE,FALSE,0,FALSE,0,16-17-18-28 [MCDC FT] (test) (0.000 s)
    [1] 2,FALSE,TRUE,TRUE,FALSE,0,FALSE,0,16-25-28 [MCDC TTF] (test) (0.000 s)
    [2] 3,FALSE,FALSE,TRUE,FALSE,0,TRUE,0,16-25-26-28 [MCDC FF] (test) (0.000 s)
    [3] 4,TRUE,FALSE,TRUE,FALSE,0,FALSE,0,16-17-20-23-28 (test) (0.000 s)
    [4] 5,TRUE,FALSE,TRUE,FALSE,1,TRUE,0,16-17-20-21-28 (test) (0.011 s)
    [5] 6,FALSE,FALSE,FALSE,FALSE,0,FALSE,0,MCDC TFT (test) (0.017 s)
    [6] 7,FALSE,FALSE,TRUE,TRUE,0,FALSE,0,MCDC FTT (test) (0.009 s)
    [7] 8,TRUE,FALSE,TRUE,TRUE,0,FALSE,0,MCDC  TF (test) (0.003 s)
```

```java
1  package Code;
2
3
4  public class Problem1_microwaveClass {
5
6      boolean cooking;
7      int timer;
8
9      public Problem1_microwaveClass (boolean cooking, int timer) {
10         this.cooking=cooking;
11         this.timer=timer;
12     }
13
14     public void operateMicrowave (boolean stopButton, boolean startButton, boolean doorOpen) {
15
16     if (cooking)
17         if (doorOpen || stopButton)
18             cooking=false;
19         else
20             if (timer>0)
21                 timer--;
22             else
23                 cooking=false;
24     else
25         if (!doorOpen && startButton &&!stopButton)
26             cooking=true;
27     }
28
29     public boolean iscooking() {
30         return cooking;
31     }
32
33     public int getTimer() {
```

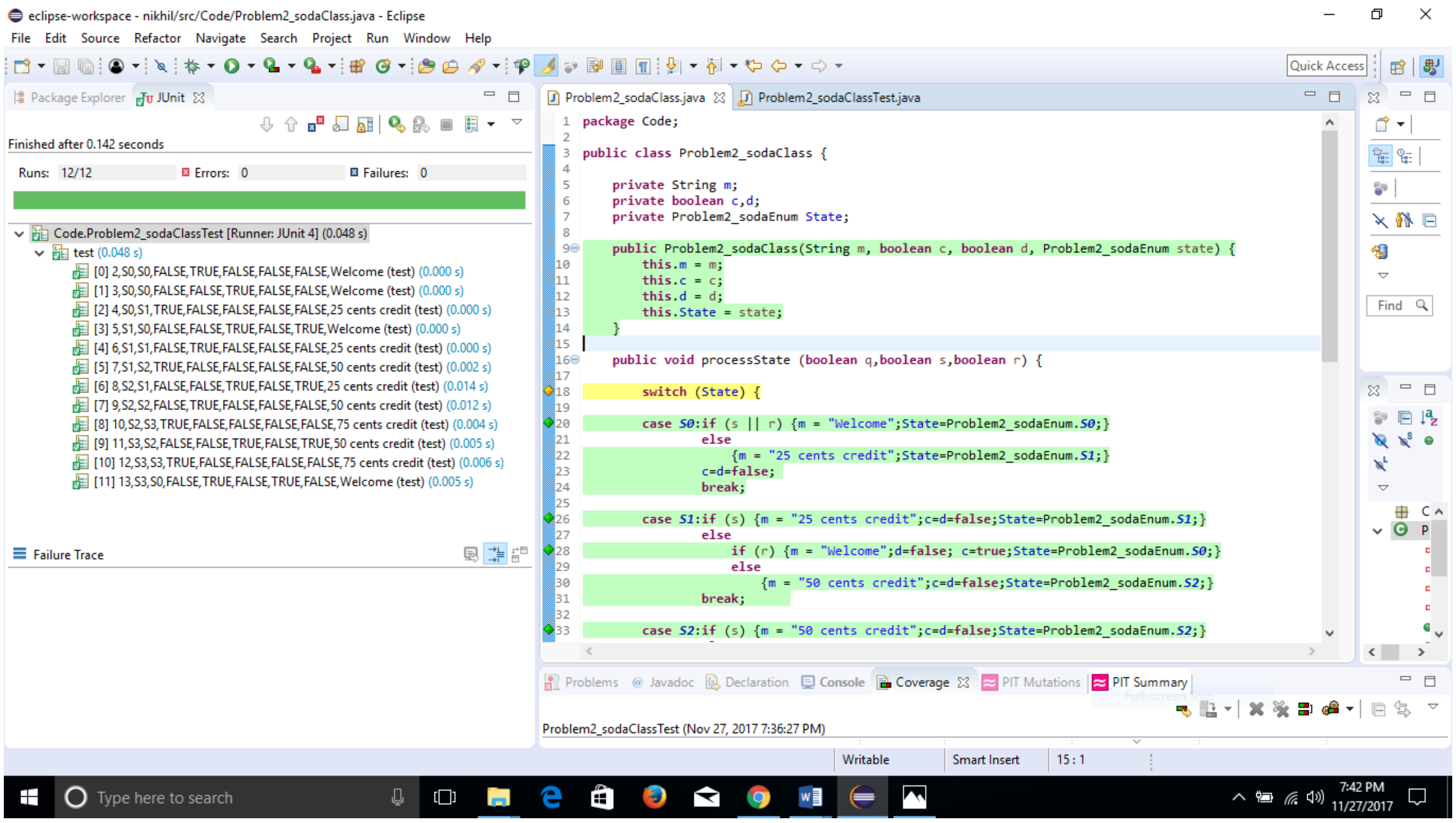
Problem1_microwaveClassTest (Nov 27, 2017 7:13:34 PM)

Problem 2:

Test Case Table:

Test Case	Current State	Next State	Inputs			Expected Outputs		
			Q	S	R	D	C	M
2	S0	S0	FALSE	TRUE	FALSE	FALSE	FALSE	Welcome
3	S0	S0	FALSE	FALSE	TRUE	FALSE	FALSE	Welcome
4	S0	S1	TRUE	FALSE	FALSE	FALSE	FALSE	25 cents credit
5	S1	S0	FALSE	FALSE	TRUE	FALSE	TRUE	Welcome
6	S1	S1	FALSE	TRUE	FALSE	FALSE	FALSE	25 cents credit
7	S1	S2	TRUE	FALSE	FALSE	FALSE	FALSE	50 cents credit
8	S2	S1	FALSE	FALSE	TRUE	FALSE	TRUE	25 cents credit
9	S2	S2	FALSE	TRUE	FALSE	FALSE	FALSE	50 cents credit
10	S2	S3	TRUE	FALSE	FALSE	FALSE	FALSE	75 cents credit
11	S3	S2	FALSE	FALSE	TRUE	FALSE	TRUE	50 cents credit
12	S3	S3	TRUE	FALSE	FALSE	FALSE	FALSE	75 cents credit
13	S3	S0	FALSE	TRUE	FALSE	TRUE	FALSE	Welcome

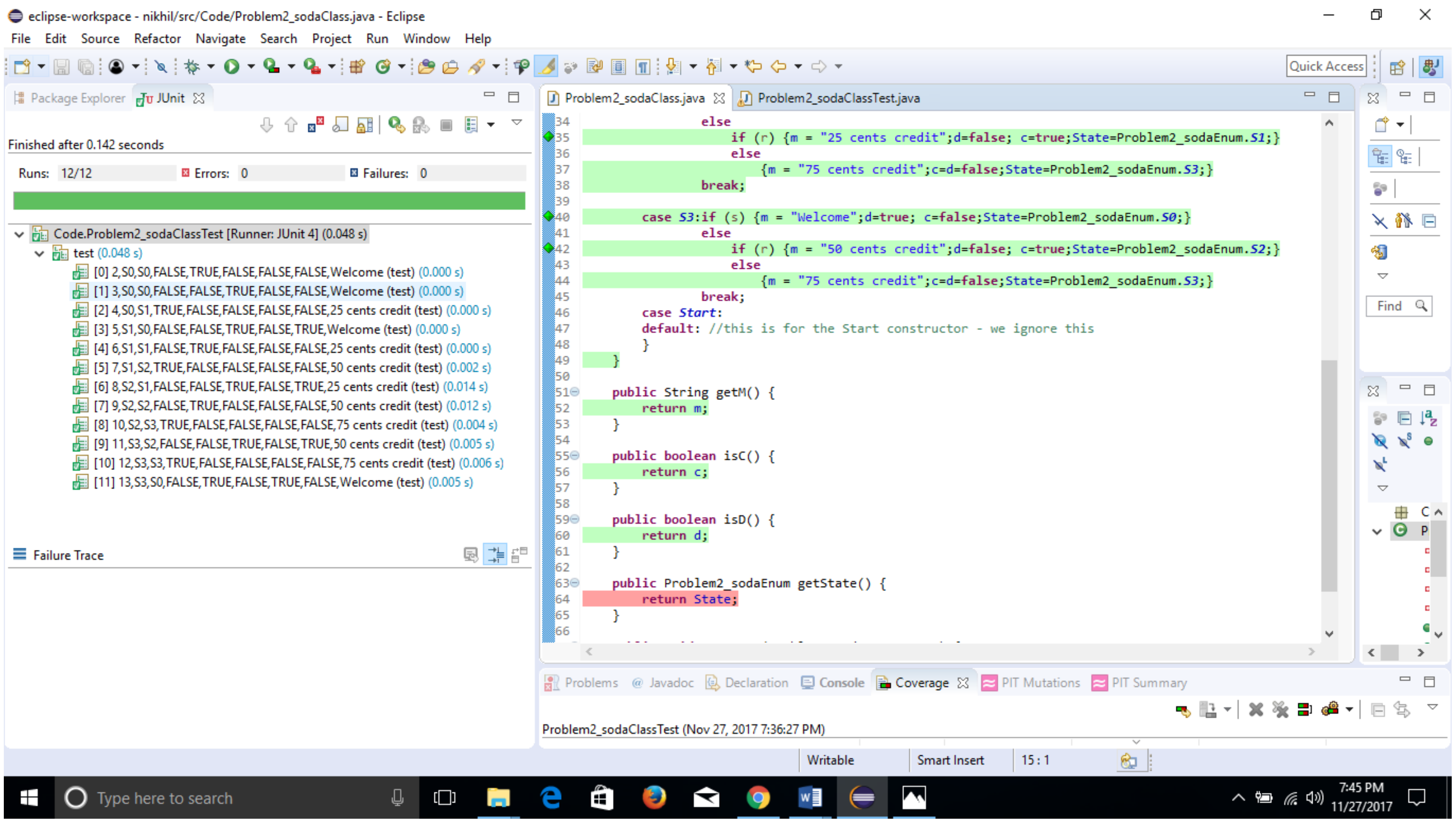
JUnit pass indicator (green bar expanded) and JaCoCo statement green source line annotations:



```
Finished after 0.142 seconds

Runs: 12/12    Errors: 0    Failures: 0

Code.Problem2_sodaClassTest [Runner: JUnit 4] (0.048 s)
  test (0.048 s)
    [0] 2,S0,S0,FALSE,TRUE,FALSE,FALSE,FALSE,Welcome (test) (0.000 s)
    [1] 3,S0,S0,FALSE,FALSE,TRUE,FALSE,FALSE,Welcome (test) (0.000 s)
    [2] 4,S0,S1,TRUE,FALSE,FALSE,FALSE,FALSE,25 cents credit (test) (0.000 s)
    [3] 5,S1,S0,FALSE,FALSE,TRUE,FALSE,TRUE,Welcome (test) (0.000 s)
    [4] 6,S1,S1,FALSE,TRUE,FALSE,FALSE,FALSE,25 cents credit (test) (0.000 s)
    [5] 7,S1,S2,TRUE,FALSE,FALSE,FALSE,FALSE,50 cents credit (test) (0.002 s)
    [6] 8,S2,S1,FALSE,FALSE,TRUE,FALSE,TRUE,25 cents credit (test) (0.014 s)
    [7] 9,S2,S2,FALSE,TRUE,FALSE,FALSE,FALSE,50 cents credit (test) (0.012 s)
    [8] 10,S2,S3,TRUE,FALSE,FALSE,FALSE,FALSE,75 cents credit (test) (0.004 s)
    [9] 11,S3,S2,FALSE,FALSE,TRUE,FALSE,TRUE,50 cents credit (test) (0.005 s)
    [10] 12,S3,S3,TRUE,FALSE,FALSE,FALSE,FALSE,75 cents credit (test) (0.006 s)
    [11] 13,S3,S0,FALSE,TRUE,FALSE,TRUE,FALSE,Welcome (test) (0.005 s)

Failure Trace
```

```java
package Code;

public class Problem2_sodaClass {

    private String m;
    private boolean c,d;
    private Problem2_sodaEnum State;

    public Problem2_sodaClass(String m, boolean c, boolean d, Problem2_sodaEnum state) {
        this.m = m;
        this.c = c;
        this.d = d;
        this.State = state;
    }

    public void processState (boolean q,boolean s,boolean r) {

        switch (State) {

        case S0:if (s || r) {m = "Welcome";State=Problem2_sodaEnum.S0;}
                else
                    {m = "25 cents credit";State=Problem2_sodaEnum.S1;}
                c=d=false;
                break;

        case S1:if (s) {m = "25 cents credit";c=d=false;State=Problem2_sodaEnum.S1;}
                else
                    if (r) {m = "Welcome";d=false; c=true;State=Problem2_sodaEnum.S0;}
                    else
                        {m = "50 cents credit";c=d=false;State=Problem2_sodaEnum.S2;}
                break;

        case S2:if (s) {m = "50 cents credit";c=d=false;State=Problem2_sodaEnum.S2;}
```

```
Problem2_sodaClassTest (Nov 27, 2017 7:36:27 PM)
```



eclipse-workspace - nikhil/src/Code/Problem2_sodaClass.java - Eclipse

```
Finished after 0.142 seconds
Runs: 12/12    Errors: 0    Failures: 0

Code.Problem2_sodaClassTest [Runner: JUnit 4] (0.048 s)
  test (0.048 s)
    [0] 2,S0,S0,FALSE,TRUE,FALSE,FALSE,FALSE,Welcome (test) (0.000 s)
    [1] 3,S0,S0,FALSE,FALSE,TRUE,FALSE,FALSE,Welcome (test) (0.000 s)
    [2] 4,S0,S1,TRUE,FALSE,FALSE,FALSE,FALSE,25 cents credit (test) (0.000 s)
    [3] 5,S1,S0,FALSE,FALSE,TRUE,FALSE,TRUE,Welcome (test) (0.000 s)
    [4] 6,S1,S1,FALSE,TRUE,FALSE,FALSE,FALSE,25 cents credit (test) (0.000 s)
    [5] 7,S1,S2,TRUE,FALSE,FALSE,FALSE,FALSE,50 cents credit (test) (0.002 s)
    [6] 8,S2,S1,FALSE,FALSE,TRUE,FALSE,TRUE,25 cents credit (test) (0.014 s)
    [7] 9,S2,S2,FALSE,TRUE,FALSE,FALSE,FALSE,50 cents credit (test) (0.012 s)
    [8] 10,S2,S3,TRUE,FALSE,FALSE,FALSE,FALSE,75 cents credit (test) (0.004 s)
    [9] 11,S3,S2,FALSE,FALSE,TRUE,FALSE,TRUE,50 cents credit (test) (0.005 s)
    [10] 12,S3,S3,TRUE,FALSE,FALSE,FALSE,FALSE,75 cents credit (test) (0.006 s)
    [11] 13,S3,S0,FALSE,TRUE,FALSE,TRUE,FALSE,Welcome (test) (0.005 s)

Failure Trace
```

```java
                else
                    if (r) {m = "25 cents credit";d=false; c=true;State=Problem2_sodaEnum.S1;}
                    else
                        {m = "75 cents credit";c=d=false;State=Problem2_sodaEnum.S3;}
                break;

            case S3:if (s) {m = "Welcome";d=true; c=false;State=Problem2_sodaEnum.S0;}
                else
                    if (r) {m = "50 cents credit";d=false; c=true;State=Problem2_sodaEnum.S2;}
                    else
                        {m = "75 cents credit";c=d=false;State=Problem2_sodaEnum.S3;}
                break;
            case Start:
            default: //this is for the Start constructor - we ignore this
            }
    }

    public String getM() {
        return m;
    }

    public boolean isC() {
        return c;
    }

    public boolean isD() {
        return d;
    }

    public Problem2_sodaEnum getState() {
        return State;
    }
```

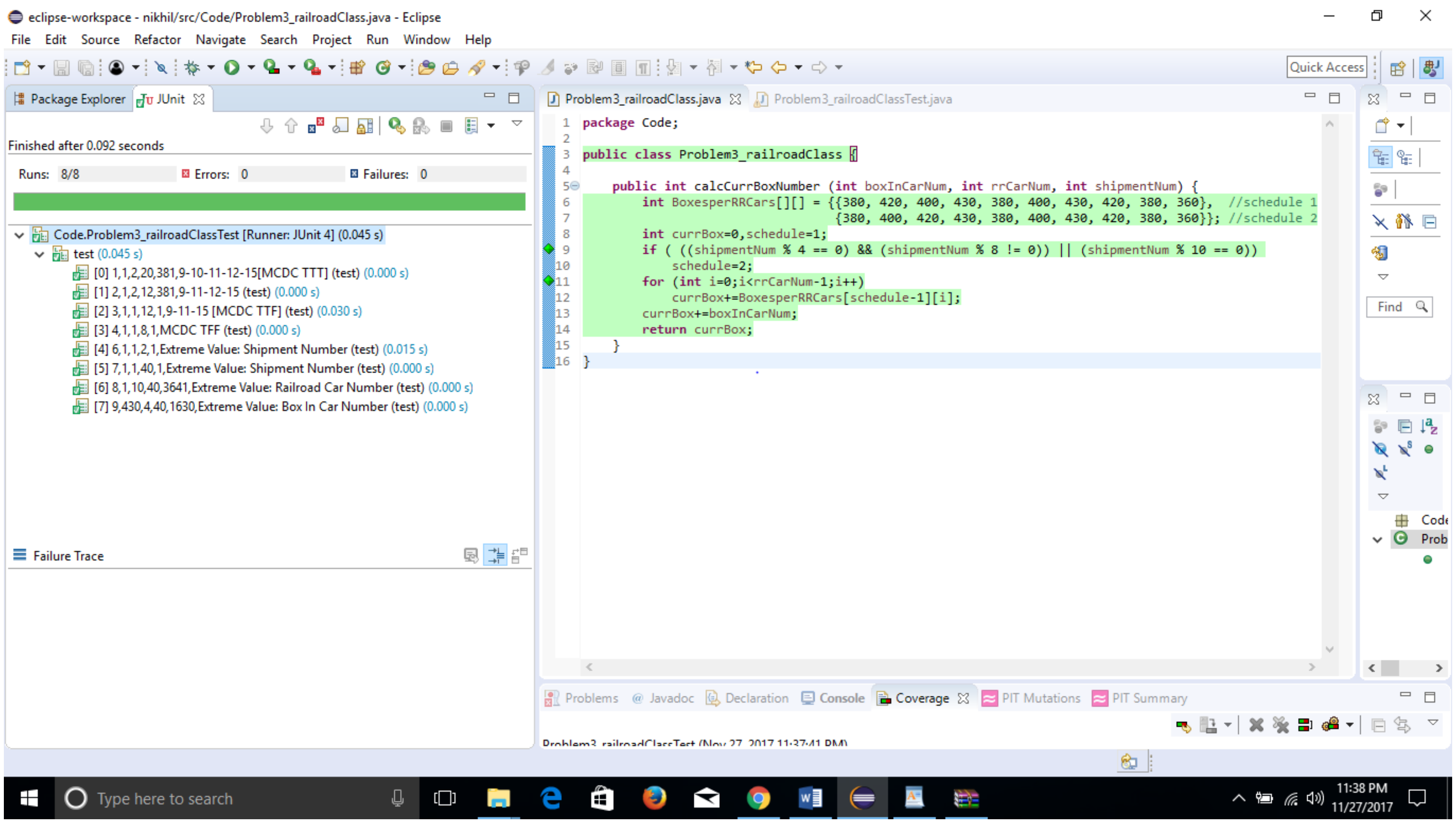
Problem2_sodaClassTest (Nov 27, 2017 7:36:27 PM)

Problem 3:

Test Case Table:

Test case number	Inputs			Exp Out	Basis Path
	Box In Car Number	Railroad Car Number	Shipment Number	Current Box Number	
1	1	2	20	381	9-10-11-12-15 (MCDC TTT)
2	1	2	12	381	9-11-12-15
3	1	1	12	1	9-11-15 (MCDC TTF)
4	1	1	8	1	MCDC TFF
5	1	1	2	1	Extreme Value: Shipment Number
6	1	1	40	1	Extreme Value: Shipment Number
7	1	10	40	3641	Extreme Value: Railroad Car Number
8	430	4	40	1630	Extreme Value: Box In Car Number

JUnit pass indicator (green bar expanded) and JaCoCo statement green source line annotations:


eclipse-workspace - nikhil/src/Code/Problem3_railroadClass.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
JUnit
Problem3_railroadClass.java
Problem3_railroadClassTest.java
Failure Trace
Find
Problems
Javadoc
Declaration
Console
Coverage
PIT Mutations
PIT Summary
Type here to search
11:38 PM
11/27/2017

Finished after 0.092 seconds

Runs: 8/8 Errors: 0 Failures: 0

```
Code.Problem3_railroadClassTest [Runner: JUnit 4] (0.045 s)
  test (0.045 s)
    [0] 1,1,2,20,381,9-10-11-12-15[MCDC TTT] (test) (0.000 s)
    [1] 2,1,2,12,381,9-11-12-15 (test) (0.000 s)
    [2] 3,1,1,12,1,9-11-15 [MCDC TTF] (test) (0.030 s)
    [3] 4,1,1,8,1,MCDC TFF (test) (0.000 s)
    [4] 6,1,1,2,1,Extreme Value: Shipment Number (test) (0.015 s)
    [5] 7,1,1,40,1,Extreme Value: Shipment Number (test) (0.000 s)
    [6] 8,1,10,40,3641,Extreme Value: Railroad Car Number (test) (0.000 s)
    [7] 9,430,4,40,1630,Extreme Value: Box In Car Number (test) (0.000 s)
```

```java
package Code;

public class Problem3_railroadClass {

    public int calcCurrBoxNumber (int boxInCarNum, int rrCarNum, int shipmentNum) {
        int BoxesperRRCars[][] = {{380, 420, 400, 430, 380, 400, 430, 420, 380, 360},  //schedule 1
                                  {380, 400, 420, 430, 380, 400, 430, 420, 380, 360}}; //schedule 2
        int currBox=0,schedule=1;
        if ( ((shipmentNum % 4 == 0) && (shipmentNum % 8 != 0)) || (shipmentNum % 10 == 0))
            schedule=2;
        for (int i=0;i<rrCarNum-1;i++)
            currBox+=BoxesperRRCars[schedule-1][i];
        currBox+=boxInCarNum;
        return currBox;
    }
}
```

Problem3_railroadClassTest (Nov 27, 2017 11:37:41 PM)

Expression for statement 9:

ab’ + c

COI a: XFF – TFF, FFF

COI b: TXF – TTF, TFF

COI c: TTX, FTX, FFX

Base Set: (TFF, FFF, TTF)

UC1: (TFF, FFF, TTF, TTT)

UC2: (TFF, FFF, TTF, FFT)

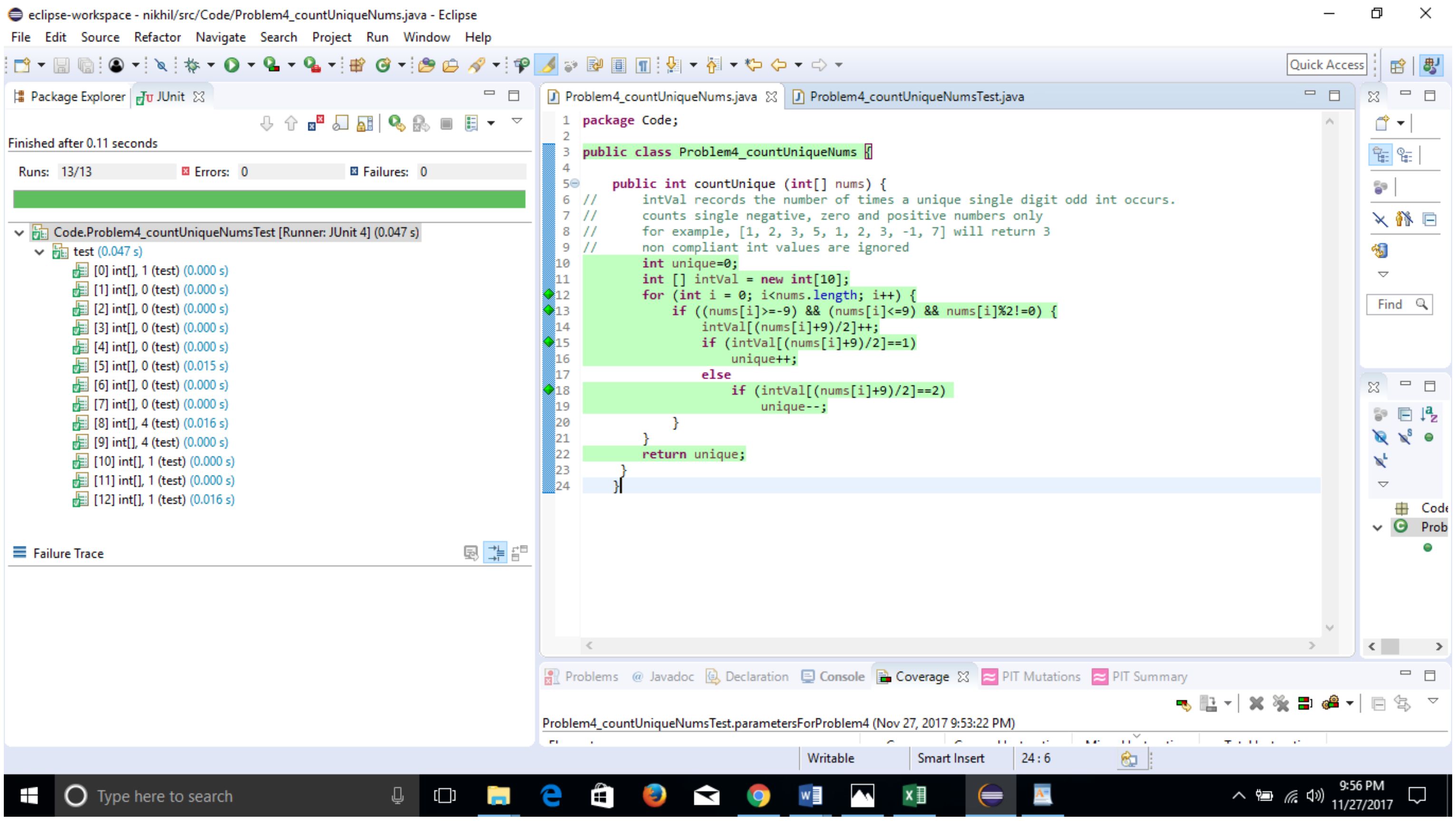
MCDC: TFF, FFF, TTF, TTT, FFT

Infeasible MCDC: FFF, FFT

These MCDC are infeasible because a number which is divisible by 8 would also be divisible by 4 and hence we cannot get both these as False values.

Problem 4:

JUnit pass indicator (green bar expanded) and JaCoCo statement green source line annotations:


eclipse-workspace - nikhil/src/Code/Problem4_countUniqueNums.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
JUnit
Problem4_countUniqueNums.java
Problem4_countUniqueNumsTest.java
Find
Problems
Javadoc
Declaration
Console
Coverage
PIT Mutations
PIT Summary
Writable
Smart Insert
24 : 6
Type here to search
9:56 PM
11/27/2017

```
Finished after 0.11 seconds
Runs: 13/13    Errors: 0    Failures: 0

Code.Problem4_countUniqueNumsTest [Runner: JUnit 4] (0.047 s)
  test (0.047 s)
    [0] int[], 1 (test) (0.000 s)
    [1] int[], 0 (test) (0.000 s)
    [2] int[], 0 (test) (0.000 s)
    [3] int[], 0 (test) (0.000 s)
    [4] int[], 0 (test) (0.000 s)
    [5] int[], 0 (test) (0.015 s)
    [6] int[], 0 (test) (0.000 s)
    [7] int[], 0 (test) (0.000 s)
    [8] int[], 4 (test) (0.016 s)
    [9] int[], 4 (test) (0.000 s)
    [10] int[], 1 (test) (0.000 s)
    [11] int[], 1 (test) (0.000 s)
    [12] int[], 1 (test) (0.016 s)

Failure Trace
```

```java
package Code;

public class Problem4_countUniqueNums {

    public int countUnique (int[] nums) {
//      intVal records the number of times a unique single digit odd int occurs.
//      counts single negative, zero and positive numbers only
//      for example, [1, 2, 3, 5, 1, 2, 3, -1, 7] will return 3
//      non compliant int values are ignored
        int unique=0;
        int [] intVal = new int[10];
        for (int i = 0; i<nums.length; i++) {
            if ((nums[i]>=-9) && (nums[i]<=9) && nums[i]%2!=0) {
                intVal[(nums[i]+9)/2]++;
                if (intVal[(nums[i]+9)/2]==1)
                    unique++;
                else
                    if (intVal[(nums[i]+9)/2]==2)
                        unique--;
            }
        }
        return unique;
    }
}
```

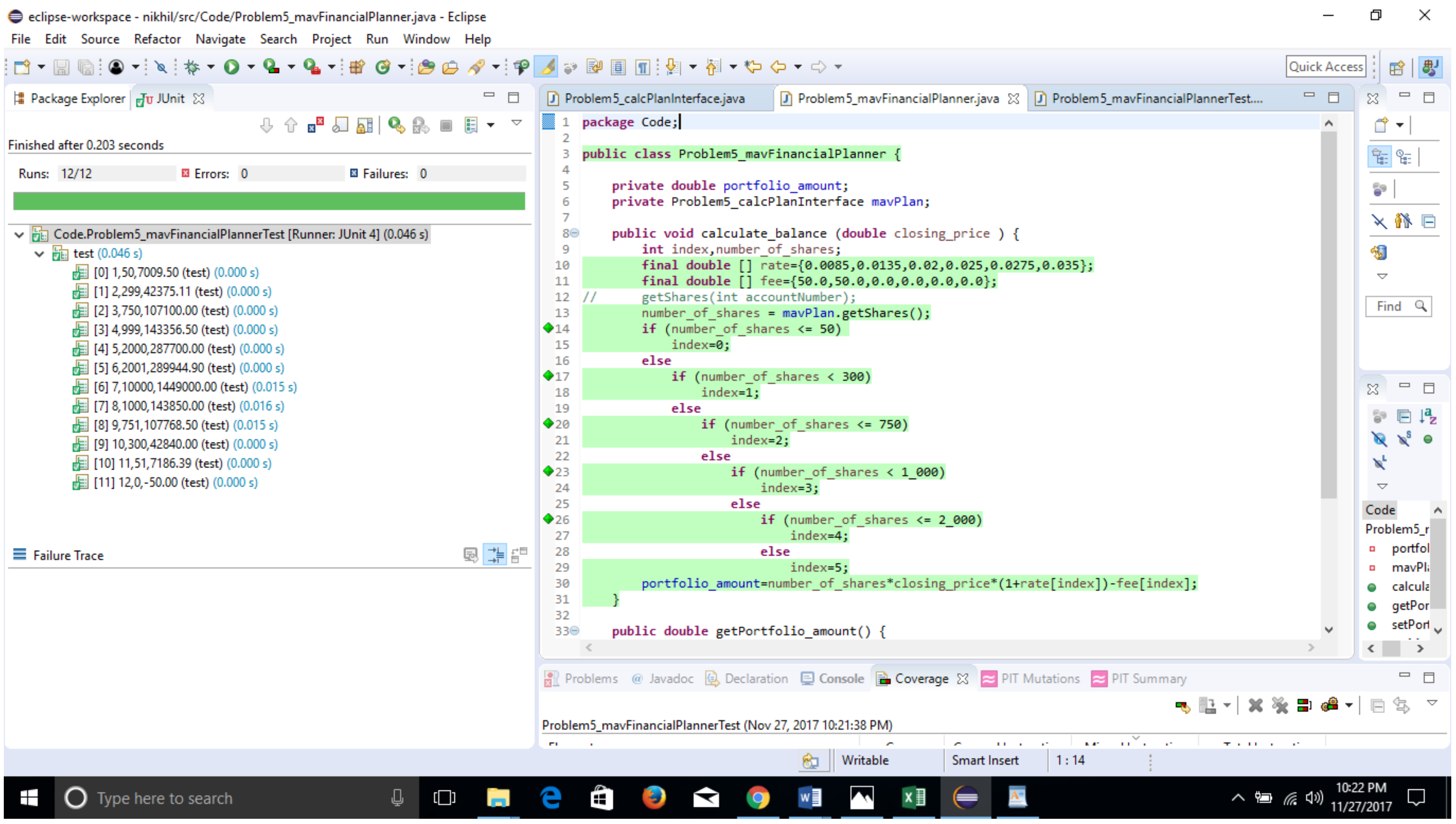
```
Problem4_countUniqueNumsTest.parametersForProblem4 (Nov 27, 2017 9:53:22 PM)
```

Problem 5:

Test Case Table:

	Inputs	**Expected**	
Test Case	**Number of shares**	**Portfolio Amount**	**Basis Path**
1	50	$7,009.50	5-6-21
2	299	$42,375.11	5-8-9-21
3	750	$107,100.00	5-8-11-12-21
4	999	$143,356.50	5-8-11-14-15-21
5	2,000	$287,700.00	5-8-11-14-17-18-21
6	2,001	$289,944.90	5-8-11-14-17-20-21
7	10,000	$1,449,000.00	Boundary Value
8	1,000	$143,850.00	Boundary Value
9	751	$107,768.50	Boundary Value
10	300	$42,840.00	Boundary Value
11	51	$7,186.39	Boundary Value
12	0	-$50.00	Boundary Value

JUnit pass indicator (green bar expanded) and JaCoCo statement green source line annotations:



eclipse-workspace - nikhil/src/Code/Problem5_mavFinancialPlanner.java - Eclipse

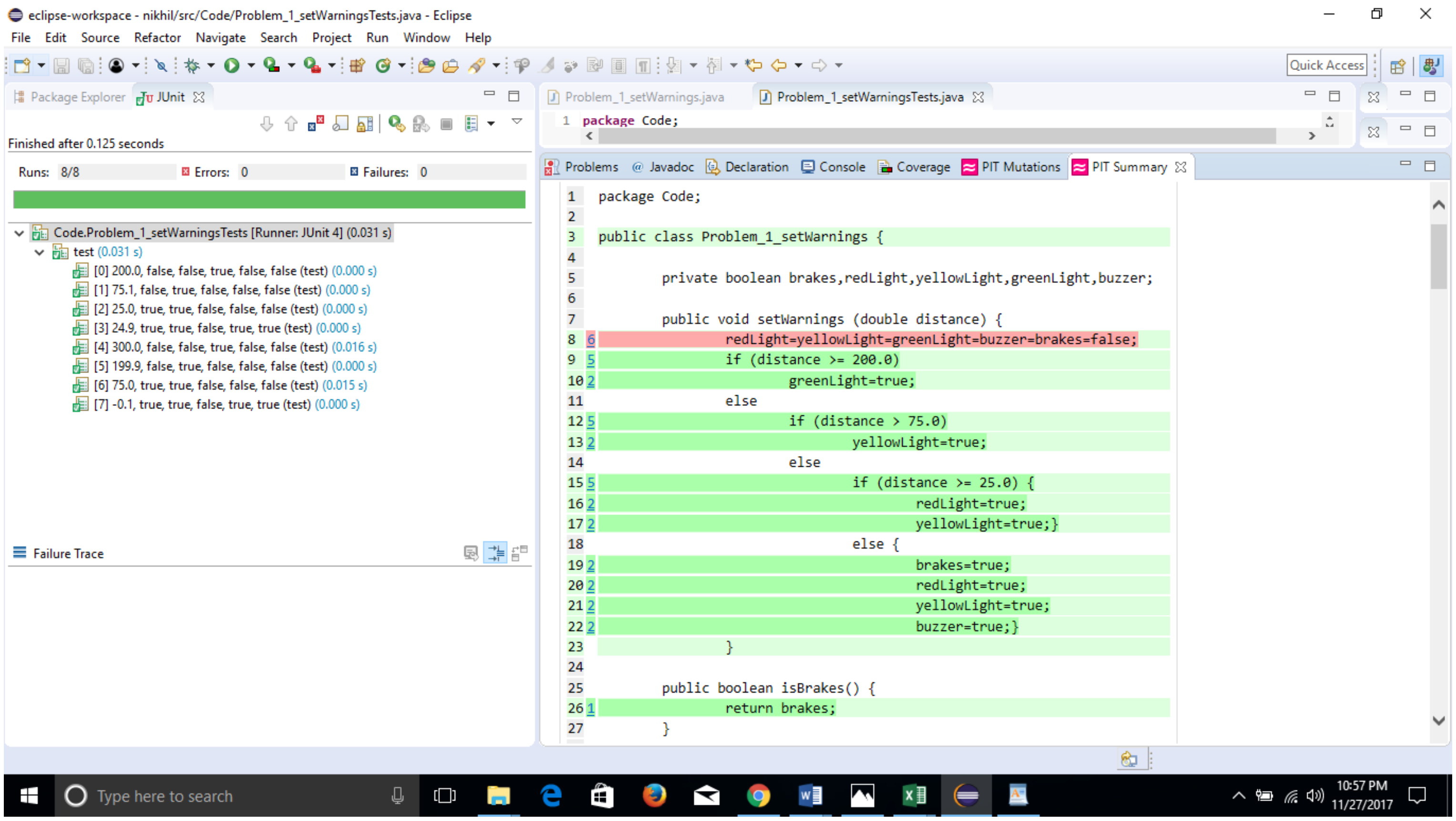
```
Finished after 0.203 seconds
Runs: 12/12    Errors: 0    Failures: 0

Code.Problem5_mavFinancialPlannerTest [Runner: JUnit 4] (0.046 s)
  test (0.046 s)
    [0] 1,50,7009.50 (test) (0.000 s)
    [1] 2,299,42375.11 (test) (0.000 s)
    [2] 3,750,107100.00 (test) (0.000 s)
    [3] 4,999,143356.50 (test) (0.000 s)
    [4] 5,2000,287700.00 (test) (0.000 s)
    [5] 6,2001,289944.90 (test) (0.000 s)
    [6] 7,10000,1449000.00 (test) (0.015 s)
    [7] 8,1000,143850.00 (test) (0.016 s)
    [8] 9,751,107768.50 (test) (0.015 s)
    [9] 10,300,42840.00 (test) (0.000 s)
    [10] 11,51,7186.39 (test) (0.000 s)
    [11] 12,0,-50.00 (test) (0.000 s)
```

```java
package Code;

public class Problem5_mavFinancialPlanner {

    private double portfolio_amount;
    private Problem5_calcPlanInterface mavPlan;

    public void calculate_balance (double closing_price ) {
        int index,number_of_shares;
        final double [] rate={0.0085,0.0135,0.02,0.025,0.0275,0.035};
        final double [] fee={50.0,50.0,0.0,0.0,0.0,0.0};
//      getShares(int accountNumber);
        number_of_shares = mavPlan.getShares();
        if (number_of_shares <= 50)
            index=0;
        else
            if (number_of_shares < 300)
                index=1;
            else
                if (number_of_shares <= 750)
                    index=2;
                else
                    if (number_of_shares < 1_000)
                        index=3;
                    else
                        if (number_of_shares <= 2_000)
                            index=4;
                        else
                            index=5;
        portfolio_amount=number_of_shares*closing_price*(1+rate[index])-fee[index];
    }

    public double getPortfolio_amount() {
```

Problem5_mavFinancialPlannerTest (Nov 27, 2017 10:21:38 PM)

Problem 6:

PIT screen snapshot of the method source code:



Finished after 0.125 seconds

Runs: 8/8 Errors: 0 Failures: 0

```
Code.Problem_1_setWarningsTests [Runner: JUnit 4] (0.031 s)
  test (0.031 s)
    [0] 200.0, false, false, true, false, false (test) (0.000 s)
    [1] 75.1, false, true, false, false, false (test) (0.000 s)
    [2] 25.0, true, true, false, false, false (test) (0.000 s)
    [3] 24.9, true, true, false, true, true (test) (0.000 s)
    [4] 300.0, false, false, true, false, false (test) (0.016 s)
    [5] 199.9, false, true, false, false, false (test) (0.000 s)
    [6] 75.0, true, true, false, false, false (test) (0.015 s)
    [7] -0.1, true, true, false, true, true (test) (0.000 s)
```

```java
package Code;

public class Problem_1_setWarnings {

        private boolean brakes,redLight,yellowLight,greenLight,buzzer;

        public void setWarnings (double distance) {
                redLight=yellowLight=greenLight=buzzer=brakes=false;
                if (distance >= 200.0)
                        greenLight=true;
                else
                        if (distance > 75.0)
                                yellowLight=true;
                        else
                                if (distance >= 25.0) {
                                        redLight=true;
                                        yellowLight=true;}
                                else {
                                        brakes=true;
                                        redLight=true;
                                        yellowLight=true;
                                        buzzer=true;}
                }

        public boolean isBrakes() {
                return brakes;
        }
```

eclipse-workspace - nikhil/src/Code/Problem_1_setWarningsTests.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
JUnit
Problem_1_setWarnings.java
Problem_1_setWarningsTests.java
Problems
Javadoc
Declaration
Console
Coverage
PIT Mutations
PIT Summary
Failure Trace
Type here to search
10:57 PM
11/27/2017