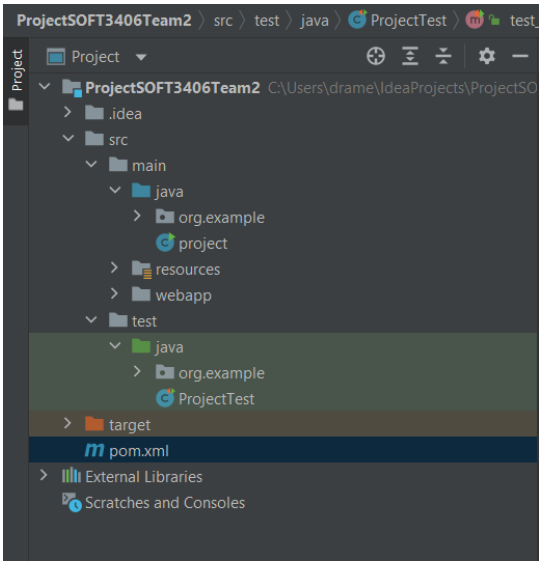
PROJECT TESTING (TEAM2)

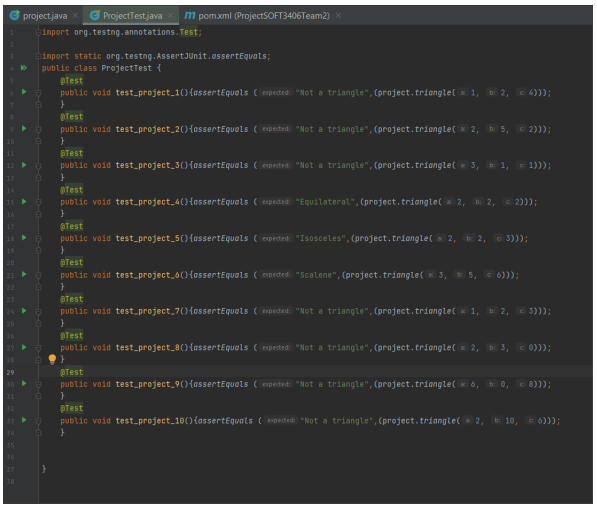
-SOFTWARE VERFICATION AND VALIDATION PROJECT-

- The goal is to test whether the given measures of three sides can form a triangle or not using MC/DC Testing Design.

- This is the overview of the project files. SInce we used the testNG framework, some files got automatically created.



- Here are the test cases we added. those test cases are meant to be adequate for the MC/DC testing design.



```
import org.testng.annotations.Test;

import static org.testng.AssertJUnit.assertEquals;
public class ProjectTest {
    @Test
    public void test_project_1(){assertEquals ( expected: "Not a triangle",(project.triangle( a: 1,  b: 2,  c: 4)));
    }
    @Test
    public void test_project_2(){assertEquals ( expected: "Not a triangle",(project.triangle( a: 2,  b: 5,  c: 2)));
    }
    @Test
    public void test_project_3(){assertEquals ( expected: "Not a triangle",(project.triangle( a: 3,  b: 1,  c: 1)));
    }
    @Test
    public void test_project_4(){assertEquals ( expected: "Equilateral",(project.triangle( a: 2,  b: 2,  c: 2)));
    }
    @Test
    public void test_project_5(){assertEquals ( expected: "Isosceles",(project.triangle( a: 2,  b: 2,  c: 3)));
    }
    @Test
    public void test_project_6(){assertEquals ( expected: "Scalene",(project.triangle( a: 3,  b: 5,  c: 6)));
    }
    @Test
    public void test_project_7(){assertEquals ( expected: "Not a triangle",(project.triangle( a: 1,  b: 2,  c: 3)));
    }
    @Test
    public void test_project_8(){assertEquals ( expected: "Not a triangle",(project.triangle( a: 2,  b: 3,  c: 0)));
    }
    @Test
    public void test_project_9(){assertEquals ( expected: "Not a triangle",(project.triangle( a: 0,  b: 0,  c: 8)));
    }
    @Test
    public void test_project_10(){assertEquals ( expected: "Not a triangle",(project.triangle( a: 2,  b: 10,  c: 6)));
    }


}
```

- the previous test cases will test the program upon the following code:

```
public static String triangle (double a, double b, double c) {
    if ((a + b > c) &&     // should be a + b > c
            (a + c > b) && // should be a + c > b
            (b + c > a)) { // should be b + c > a
        if(a==b && b==c) //check whether all sides are equal (a==b && b==c)
        {
            System.out.println("This is an equilateral triangle.\n");
            return "Equilateral";
        }
        else if(a==b || a==c || b==c) //check whether two sides are equal (a==b || a==c || b==c)
        {
            System.out.println("This is an isosceles triangle.\n");
            return "Isosceles";
        }
        else //check whether no sides are equal
        {
            System.out.println("This is a scalene triangle.\n");
            return "Scalene";// return "Scalene";
        }
    }
    return "Not a triangle";
}
```

PROJECT TESTING (TEAM2)

-SOFTWARE VERFICATION AND VALIDATION PROJECT-

Possible test cases

TC	x	y	z	OutCome
1	T	T	T	T
2	T	T	F	F
3	T	F	T	F
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC (adequate test cases)

TC	x	y	z	OutCome
1	T	T	T	T
2	T	T	F	F
3	T	F	T	F
5	F	T	T	F

Test Cases

TC	x	y	z	a	b	c	Output
1	F	T	T	1	2	4	Not a triangle
This test case tests if a, b, and c are valid measure for building a triangle. since it doesn't satisfy the condition `(a + b > c)`, this doesn't form a triangle, and an output `"Not a triangle"` is returned.							
2	T	F	T	2	5	2	Not a triangle
This test case tests if a, b, and c are valid measure for building a triangle. since it doesn't satisfy the condition `(a + c > b)`, this doesn't form a triangle, and an output `"Not a triangle"` is returned.							
3	T	T	F	3	1	1	Not a triangle
This test case tests if a, b, and c are valid measure for building a triangle. since it doesn't satisfy the condition `(b + c > a)`, this doesn't form a triangle, and an output `"Not a triangle"` is returned.							

PROJECT TESTING (TEAM2)

-SOFTWARE VERFICATION AND VALIDATION PROJECT-

4	T	T	T	2	2	2	Equilateral
This test case tests if a, b, and c are valid measure for building a triangle. since it passes all conditions, it checks what type of triangle those measure can form by checking other conditions. since the condition `(a==b && b==c` meets the given measures, an output `"Equilateral"` is returned.							
5	T	T	T	2	2	3	Isosceles
This test case tests if a, b, and c are valid measure for building a triangle. since it passes all conditions, it checks what type of triangle those measure can form by checking other conditions. since the condition `(a==b || a==c || b==c)` meets the given measures, an output `"Isosceles"` is returned.							
6	T	T	T	3	5	6	Scalene
This test case tests if a, b, and c are valid measure for building a triangle. since it passes all conditions, it checks what type of triangle those measure can form by checking other conditions. since neither the condition (a==b && b==c) nor `(a==b || a==c || b==c)` meet the given measures, this triangle's sides have different lengths and output `"Scalene"` is returned.							
7	F	T	T	1	2	3	Not a triangle
This test case tests if a, b, and c are valid measure for building a triangle. since it doesn't satisfy the condition `(a + b > c)`, this doesn't form a triangle, and an output `"Not a triangle"` is returned.							
8	T	F	T	2	3	0	Not a triangle
This test case tests if a, b, and c are valid measure for building a triangle. since it doesn't satisfy the condition `(a + c > b)`, this doesn't form a triangle, and an output `"Not a triangle"` is returned.							
9	F	T	T	6	0	8	Not a triangle
This test case tests if a, b, and c are valid measure for building a triangle. since it doesn't satisfy the condition `(a + b > c)`, this doesn't form a triangle, and an output `"Not a triangle"` is returned.							
10	T	F	T	2	10	6	Not a triangle
This test case tests if a, b, and c are valid measure for building a triangle. since it doesn't satisfy the condition `(a + c > b)`, this doesn't form a triangle, and an output `"Not a triangle"` is returned.							

Project TEAM (2):

- MOHANAD ALTARAH
- RAED (MOHD FAIQ)SAADEDDIN AL-SHARIF
- BALKISU SHEHU DANMUSA