5/15/2020

Software Testing

Assignment 2

Nabiya Fatima BSE173011
Iqra Ishtiaq BSE173043

Table of Contents

Case Study 2

Introduction 2

Brief Description 2

Identified Functions 3

Function 1 3

Function 2 3

Function 3 3

Black Box Testing 3

Worst Case BVA 3

Function 1 3

Function 2 5

Function 3 6

Strong robust equivalence class Testing: 8

Function 1 8

Function 2 9

Function 3 9

Comparing Strong Robust Equivalence vs Robust Worst BVA 10

Function 1 10

Function 2 10

Function 3 10

Case Study

Introduction

An **Equation** in one unknown quantity (let it be x) in the form $ax^2 + bx + c = 0$ is known as a **Quadratic equation**, where a, b, c are constants and $a \neq 0$ while b and c can be zero. Here "a" is called the coefficient of x^2, "b" is the coefficient of x and "c" is a constant term. The word "Quadratic" comes from "***Quadratum***", the Latin word for square. Hence, a quadratic equation is an equation where the variable is of the second degree. Therefore, a Quadratic equation is also called an "Equation of Degree 2". Many real-world problems can be studied and solved using Quadratic Equations. **Quadratic equations** are used in everyday life too, as when calculating areas, determining a product's profit or formulating the speed of an object, projectile motion etc.

Brief Description

In a programming competition the students are required to design a program that takes three numbers (a, b, c) as inputs and determine whether the equation is Quadratic or not. The standard form of Quadratic Equation is $ax^2+bx+c=0$, where a, b, c are constants and "a" cannot be zero. The program should have a method that calculates the nature of the roots of the Quadratic equation weather the roots of the equation are Real, Equal or Imaginary using the discriminant b^2-4ac. Following are the conditions that should be meet:

- If b^2-4ac>0 the roots are Real and Unequal
- If b^2-4ac=0 the roots are Real and Equal
- If b^2-4ac<0 the roots are Imaginary

The program should also have a method to calculate the roots of the equation using the formula $\boldsymbol{x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}}$. Each function should first test whether the inputs will form the Quadratic equation, and then perform the desired functionality; otherwise, the program should display a message "**Not a Quadratic Equation**". The problem with the Quadratic Formula is that when using Floating point Arithmetic, it may be subjected to loss of significance in calculating rules of the equation, more similar the two numbers are the precision decreases, so the constants a, b, c should be integer rather than Floating-point numbers or Decimal numbers. The inputs for the constants a, b, c should be within the range from [0,200]. The program should display the following menu

1) Check the nature of the roots.
2) Calculate the roots of Quadratic equation.

The users can select the above-mentioned options by pressing the number. The program should throw an exception if the user tries to select the invalid option. The program should also handle all the necessary exceptions.

Identified Functions

Function 1 (isQuadratic (a, b, c))

The function isQuadratic (a, b, c) take three numbers as input and checks weather the numbers form the Quadratic Equation or not, if the coefficient a ≠ 0 the program returns "Quadratic Equation" otherwise the program returns "Not a Quadratic Equation".

Function 2 (checkRoots (a, b, c))

The checkRoots (a, b, c) takes the three input and using the discriminant formula b^2 -4ac checks weather the roots of the Quadratic Equation are "Real", "Equal" or "Imaginary" and returns the nature of roots.

Function 3 (calculateRoots (a, b, c))

The calculateRoots (a, b, c) takes the three input and using the formula $\boldsymbol{x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}}$ calculates the roots of the Quadratic Equation.

Black Box Testing

Worst Case BVA

Function 1 (isQuadratic (a, b, c))

Test Cases	a	b	c	Expected output
1	0	0	0	Not a Quadratic Equation
2	0	0	1	Not a Quadratic Equation
3	0	0	100	Not a Quadratic Equation
4	0	0	199	Not a Quadratic Equation
5	0	0	200	Not a Quadratic Equation
6	0	1	0	Not a Quadratic Equation
7	0	1	1	Not a Quadratic Equation
8	0	1	100	Not a Quadratic Equation
9	0	1	199	Not a Quadratic Equation
10	0	1	200	Not a Quadratic Equation
11	0	100	0	Not a Quadratic Equation
12	0	100	1	Not a Quadratic Equation
13	0	100	100	Not a Quadratic Equation
14	0	100	199	Not a Quadratic Equation
15	0	100	200	Not a Quadratic Equation
16	0	199	0	Not a Quadratic Equation
17	0	199	1	Not a Quadratic Equation
18	0	199	100	Not a Quadratic Equation
19	0	199	199	Not a Quadratic Equation
20	0	199	200	Not a Quadratic Equation
21	0	200	0	Not a Quadratic Equation

22	0	200	1	Not a Quadratic Equation
23	0	200	100	Not a Quadratic Equation
24	0	200	199	Not a Quadratic Equation
25	0	200	200	Not a Quadratic Equation
26	1	0	0	Quadratic Equation
27	1	0	1	Quadratic Equation
28	1	0	100	Quadratic Equation
29	1	0	199	Quadratic Equation
30	1	0	200	Quadratic Equation
31	1	1	0	Quadratic Equation
32	1	1	1	Quadratic Equation
33	1	1	100	Quadratic Equation
34	1	1	199	Quadratic Equation
35	1	1	200	Quadratic Equation
36	1	100	0	Quadratic Equation
37	1	100	1	Quadratic Equation
38	1	100	100	Quadratic Equation
39	1	100	199	Quadratic Equation
40	1	100	200	Quadratic Equation
41	1	199	0	Quadratic Equation
42	1	199	1	Quadratic Equation
43	1	199	100	Quadratic Equation
44	1	199	199	Quadratic Equation
45	1	199	200	Quadratic Equation
46	1	200	0	Quadratic Equation
47	1	200	1	Quadratic Equation
48	1	200	100	Quadratic Equation
49	1	200	199	Quadratic Equation
50	1	200	200	Quadratic Equation
51	100	0	0	Quadratic Equation
52	100	0	1	Quadratic Equation
53	100	0	100	Quadratic Equation
54	100	0	199	Quadratic Equation
55	100	0	200	Quadratic Equation
56	100	1	0	Quadratic Equation
57	100	1	1	Quadratic Equation
58	100	1	100	Quadratic Equation
59	100	1	199	Quadratic Equation
60	100	1	200	Quadratic Equation
61	100	100	0	Quadratic Equation
62	100	100	1	Quadratic Equation
63	100	100	100	Quadratic Equation

64	100	100	199	Quadratic Equation
65	100	100	200	Quadratic Equation

Function 2 (checkRoots (a, b, c))

Test Cases	a	b	c	Expected output
1	0	0	0	Not a Quadratic Equation
2	0	0	1	Not a Quadratic Equation
3	0	0	100	Not a Quadratic Equation
4	0	0	199	Not a Quadratic Equation
5	0	0	200	Not a Quadratic Equation
6	0	1	0	Not a Quadratic Equation
7	0	1	1	Not a Quadratic Equation
8	0	1	100	Not a Quadratic Equation
9	0	1	199	Not a Quadratic Equation
10	0	1	200	Not a Quadratic Equation
11	0	100	0	Not a Quadratic Equation
12	0	100	1	Not a Quadratic Equation
13	0	100	100	Not a Quadratic Equation
14	0	100	199	Not a Quadratic Equation
15	0	100	200	Not a Quadratic Equation
16	0	199	0	Not a Quadratic Equation
17	0	199	1	Not a Quadratic Equation
18	0	199	100	Not a Quadratic Equation
19	0	199	199	Not a Quadratic Equation
20	0	199	200	Not a Quadratic Equation
21	0	200	0	Not a Quadratic Equation
22	0	200	1	Not a Quadratic Equation
23	0	200	100	Not a Quadratic Equation
24	0	200	199	Not a Quadratic Equation
25	0	200	200	Not a Quadratic Equation
26	1	0	0	Equal Roots
27	1	0	1	Imaginary Roots
28	1	0	100	Imaginary Roots
29	1	0	199	Imaginary Roots
30	1	0	200	Imaginary Roots
31	1	1	0	Real Roots
32	1	1	1	Imaginary Roots
33	1	1	100	Imaginary Roots
34	1	1	199	Imaginary Roots
35	1	1	200	Imaginary Roots
36	1	100	0	Real Roots
37	1	100	1	Real Root

38	1	100	100	Real roots
39	1	100	199	Real Roots
40	1	100	200	Real Roots
41	1	199	0	Real Roots
42	1	199	1	Real Roots
43	1	199	100	Real Roots
44	1	199	199	Real Roots
45	1	199	200	Real Roots
46	1	200	0	Real Roots
47	1	200	1	Real Roots
48	1	200	100	Real Roots
49	1	200	199	Real Roots
50	1	200	200	Real Roots
51	100	0	0	Equal Roots
52	100	0	1	Imaginary Roots
53	100	0	100	Imaginary Roots
54	100	0	199	Imaginary Roots
55	100	0	200	Imaginary Roots
56	100	1	0	Real Roots
57	100	1	1	Imaginary Roots
58	100	1	100	Imaginary Roots
59	100	1	199	Imaginary Roots
60	100	1	200	Imaginary Roots
61	100	100	0	Real Roots
62	100	100	1	Real Roots
63	100	100	100	Imaginary Roots
64	100	100	199	Imaginary Roots
65	100	100	200	Imaginary Roots

Function 3 (calculateRoots (a, b, c))

Test Cases	a	b	c	Expected output
1	0	0	0	Not a Quadratic Equation
2	0	0	1	Not a Quadratic Equation
3	0	0	100	Not a Quadratic Equation
4	0	0	199	Not a Quadratic Equation
5	0	0	200	Not a Quadratic Equation
6	0	1	0	Not a Quadratic Equation
7	0	1	1	Not a Quadratic Equation
8	0	1	100	Not a Quadratic Equation
9	0	1	199	Not a Quadratic Equation
10	0	1	200	Not a Quadratic Equation
11	0	100	0	Not a Quadratic Equation

12	0	100	1	Not a Quadratic Equation
13	0	100	100	Not a Quadratic Equation
14	0	100	199	Not a Quadratic Equation
15	0	100	200	Not a Quadratic Equation
16	0	199	0	Not a Quadratic Equation
17	0	199	1	Not a Quadratic Equation
18	0	199	100	Not a Quadratic Equation
19	0	199	199	Not a Quadratic Equation
20	0	199	200	Not a Quadratic Equation
21	0	200	0	Not a Quadratic Equation
22	0	200	1	Not a Quadratic Equation
23	0	200	100	Not a Quadratic Equation
24	0	200	199	Not a Quadratic Equation
25	0	200	200	Not a Quadratic Equation
26	1	0	0	X=0
27	1	0	1	X1=+1*i*, X2=-1*i*
28	1	0	100	X1=+10*i*, X2=-10*i*
29	1	0	199	X1=+14.106*i*, X2=-14.106*i*
30	1	0	200	X1=+14.142*i*, X2=-14.142*i*
31	1	1	0	X1=0, X2=-1
32	1	1	1	X1=-0.5+0.866*i*, X2=-0.5-0.866*i*
33	1	1	100	X1=-0.5+0.9.987*i*, X2=-0.5-9.987*i*
34	1	1	199	X1=-0.5+14.097*i*, X2=-0.5-14.097*i*
35	1	1	200	X1=-0.5+14.133*i*, X2=-0.5-14.133*i*
36	1	100	0	X1=0, X2=-100
37	1	100	1	X1=-0.0100, X2=-99.99
38	1	100	100	X1=-1.01, X2=-98.98
39	1	100	199	X1=-2.03, X2=-97.96
40	1	100	200	X1=-2.04, X2=-97.95
41	1	199	0	X1=0, X2=-199
42	1	199	1	X1=-0.005, X2=-198.99
43	1	199	100	X1=-0.50, X2=-198.49
44	1	199	199	X1=-1.005, X2=-197.99
45	1	199	200	X1=-1.01, X2=-197.99
46	1	200	0	X1=0, X2=-200
47	1	200	1	X1=-0.005, X2=-199.995
48	1	200	100	X1=-0.501, X2=-199.49
49	1	200	199	X1=-1, X2=-199
50	1	200	200	X1=-1.005, X2=-198.99
51	100	0	0	X=0
52	100	0	1	X1=+0.1*i*, X2=-0.1*i*
53	100	0	100	X1=+1*i*, X2=-1*i*

54	100	0	199	X1=+1.410*i*, X2=-1.410*i*
55	100	0	200	X1=+1.4142*i*, X2=-1.4142*i*
56	100	1	0	X1=0, X2=-0.01
57	100	1	1	X1=-0.005+0.099*i*, X2=-0.005-0.099*i*
58	100	1	100	X1=-0.005+0.099*i*, X2=-0.005-0.099*i*
59	100	1	199	X1=-0.005+1.41*i*, X2=-0.005-1.41*i*
60	100	1	200	X1=-0.005+1.414*i*, X2=-0.005-1.414*i*
61	100	100	0	X1=0,X2=-1
62	100	100	1	X1=-0.010,X2=-0.9898
63	100	100	100	X1=-0.5+0.866*i*, X2=-0.5-0.8686*i*
64	100	100	199	X1=-0.5+1.31*i*, X2=-0.5-1.31*i*
65	100	100	200	X1=-0.5+1.322*i*, X2=-0.5-1.322*i*

Strong robust equivalence class Testing:

Function 1 (isQuadratic (a, b, c))
Total test cases: 16 test cases
Test Data: Enter the 3 Integer Value (a, b, c)
Pre-condition: 0 ≤ a ≤ 200, 0 ≤ b ≤ 200 and 0 ≤ c ≤ 200

Test Cases	a	b	c	Expected Output
1	o	100	100	Not a Quadratic Equation
2	100	100	100	Quadratic Equation
3	-1	100	100	value of “a” out of Range
4	100	-1	100	value of “b” out of Range
5	100	100	-1	value of “c” out of Range
6	-1	-1	100	value of “a” & “b” out of Range
7	100	-1	-1	value of “b” & “c” out of Range
8	-1	100	-1	value of “a” & “c” out of Range
9	-1	-1	-1	value of “a”, “b” & “c” out of Range
10	201	100	100	value of “a” out of Range
11	100	201	100	value of “b” out of Range
12	100	100	201	value of “c” out of Range
13	201	201	100	value of “a” & “b” out of Range
14	100	201	201	value of “b” & “c” out of Range
15	201	100	201	value of “a” & “c” out of Range
16	201	201	201	value of “a”, “b” & “c” out of Range

Function 2 (checkRoots (a, b, c))
Total test cases: 18 test cases
Test Data: Enter the 3 Integer Value (a, b, c)
Pre-condition: 0 ≤ a ≤ 200, 0 ≤ b ≤ 200 and 0 ≤ c ≤ 200

Test Cases	a	b	c	Expected Output
1	0	1	1	Not a Quadratic Equation
2	1	1	0	Real Roots
3	1	0	0	Equal Roots
4	1	0	1	Imaginary Roots
5	-1	1	1	value of “a” out of Range
6	1	-1	1	value of “b” out of Range
7	1	1	-1	value of “c” out of Range
8	-1	-1	1	value of “a” & “b” out of Range
9	1	-1	-1	value of “b” & “c” out of Range
10	-1	1	-1	value of “a” & “c” out of Range
11	-1	-1	-1	value of “a”, “b” & “c” out of Range
12	201	1	1	value of “a” out of Range
13	1	201	1	value of “b” out of Range
14	1	1	201	value of “c” out of Range
15	201	201	1	value of “a” & “b” out of Range
16	1	201	201	value of “b” & “c” out of Range
17	201	1	201	value of “a” & “c” out of Range
18	201	201	201	value of “a”, “b” & “c” out of Range

Function 3 (calculateRoots (a, b, c))
Total test cases: 18 test cases
Test Data: Enter the 3 Integer Value (a, b, c)
Pre-condition: 0 ≤ a ≤ 200, 0 ≤ b ≤ 200 and 0 ≤ c ≤ 200

Test Cases	a	b	c	Expected Output
1	o	1	1	Not a Quadratic Equation
2	1	1	0	X1=0, X2=-1
3	1	0	0	X=0
4	1	0	1	X1=+1*i*, X2=-1*i*
5	-1	1	1	value of “a” out of Range
6	1	-1	1	value of “b” out of Range
7	1	1	-1	value of “c” out of Range
8	-1	-1	1	value of “a” & “b” out of Range
9	1	-1	-1	value of “b” & “c” out of Range

10	-1	1	-1	value of “a” & “c” out of Range
11	-1	-1	-1	value of “a”, “b” & “c” out of Range
12	201	1	1	value of “a” out of Range
13	1	201	1	value of “b” out of Range
14	1	1	201	value of “c” out of Range
15	201	201	1	value of “a” & “b” out of Range
16	1	201	201	value of “b” & “c” out of Range
17	201	1	201	value of “a” & “c” out of Range
18	201	201	201	value of “a”, “b” & “c” out of Range

Comparing Strong Robust Equivalence vs Robust Worst BVA

Function 1 (IsQuadratic (a, b, c))

The number of test cases generated using **Strong Robust Equivalence class testing = 16** and the number of test cases generated using robust worst BVA = **49 test case.**

Function 2 (checkRoots (a, b, c))

The number of test cases generated using **Strong Robust Equivalence class testing = 18** and the number of test cases generated using robust worst BVA = **343 test case**

Function 3 (calculateRoots (a, b, c))

The number of test cases generated using **Strong Robust Equivalence class testing = 18** and the number of test cases generated using robust worst BVA = **343 test case**

The Strong Robust Equivalence class testing method includes in the test suite a test case from each element of Cartesian product of all equivalence classes of valid and invalid values of all parameters. The number of test cases generated using **Strong Robust Equivalence class testing are less as compared to** the Robust Worst boundary value testing method which tests boundary values, inner and outer off points, and nominal point and for all parameters comparing the two methods clearly shows the number of reduced test case in **Strong Robust Equivalence class testing**