

PROG 1350 – Software Engineering Fundamentals
Assignment #3 – **Bazinga Unit Test Specification**
Due date: March 18, 2019
Student: Lev Potomkin

Table of Contents

Introduction.....	3
Requirements.....	3
Summary.....	3
Test Cases.....	4
Functional Tests.....	4
Exception Tests.....	6
Boundary Tests.....	8
References.....	9

Introduction

The purpose of this document is to specify requirements and unit test cases necessary for proper testing of the function Bazinga(). The following is extracted from the SRS.

Requirements

Description	Requirement ID
Takes 2 parameters – bazValue which is a float and bazModulus which is an integer	[Requirement : A10]
bazValue is defined as being a value greater than or equal to 0.00 and less than or equal to 100.00	[Requirement : A17]
If Bazinga() is called with a bazValue outside of this range, then the function returns a status value of -1.0 (meaning bazValue out of range)	[Requirement : A17.1]
bazModulus is defined as being a positive value greater than 0 and less than or equal to 15	[Requirement : A16]
If Bazinga() is called with a bazModulus outside of this range, then the function returns a status value of -2.0 (meaning bazModulus out of range)	[Requirement : A16.1]
Bazinga() returns the floating point modulus value (i.e. the remainder) of bazValue % bazModulus	[Requirement: A34]

Summary

- function takes 2 numeric parameters
- each parameter must be within a certain range, otherwise an error codes is returned
- function must return the remainder of the division of the first number by the second

Test Cases

This section provides complete set of unit test cases for the function Bazinga().

Functional Tests

Test ID	1
Description	Type: Functional Requirements Tested: A17, A16, A34 Summary: testing main success scenario
Method of Execution	automated
Input Data	bazValue = 18.0; bazModulus = 5;
Expected Output	3.0
Observed Output	

Test ID	2
Description	Type: Functional Requirements Tested: A17, A16, A34 Summary: testing main success scenario
Method of Execution	automated
Input Data	bazValue = 67.123; bazModulus = 12;
Expected Output	7.123
Observed Output	

Test ID	3
Description	Type: Functional Requirements Tested: A17, A16, A34 Summary: testing main success scenario
Method of Execution	automated
Input Data	bazValue = 3.5; bazModulus = 3;
Expected Output	0.5
Observed Output	

Test ID	4
Description	Type: Functional Requirements Tested: A17, A16, A34 Summary: testing main success scenario
Method of Execution	automated
Input Data	bazValue = 42.0; bazModulus = 14;
Expected Output	0.0
Observed Output	

Exception Tests

Test ID	5
Description	Type: Exception Requirements Tested: A17, A17.1 Summary: testing scenario where bazValue is out of range
Method of Execution	automated
Input Data	bazValue = 1234.5678; bazModulus = 7;
Expected Output	-1.0
Observed Output	

Test ID	6
Description	Type: Exception Requirements Tested: A17, A17.1 Summary: testing scenario where bazValue is out of range
Method of Execution	automated
Input Data	bazValue = -1.45; bazModulus = 10;
Expected Output	-1.0
Observed Output	

Test ID	7
Description	Type: Exception Requirements Tested: A16, A16.1 Summary: testing scenario where bazModulus is out of range
Method of Execution	automated
Input Data	bazValue = 4.98; bazModulus = 123;
Expected Output	-2.0
Observed Output	

Test ID	8
Description	Type: Exception Requirements Tested: A16, A16.1 Summary: testing scenario where bazModulus is out of range
Method of Execution	automated
Input Data	bazValue = 12.5; bazModulus = -18;
Expected Output	-2.0
Observed Output	

Boundary Tests

Test ID	9
Description	Type: Boundary Requirements Tested: A16, A17, A34 Summary: testing scenario where bazValue is on the range boundary
Method of Execution	automated
Input Data	bazValue = 0.001; bazModulus = 10;
Expected Output	0.001
Observed Output	

Test ID	10
Description	Type: Boundary Requirements Tested: A16, A17.1 Summary: testing scenario where bazValue is on the range boundary
Method of Execution	automated
Input Data	bazValue = 100.001; bazModulus = 8;
Expected Output	-1.0
Observed Output	

References

[1] Clarke, S. (2019). *Module 07: Testing*. Retrieved March 12, 2018 from eConestoga.

[2] McConnel, S. (2004). *Code Complete* (2nd ed.). Retrieved March 12, 2018 from <http://aroma.vn/web/wp-content/uploads/2016/11/code-complete-2nd-edition-v413hav.pdf>