Nguyen Trung – e1500953

**Exercise 4**

Software testing

Part 1 – challenges in software testing

* What is the unit testing, and what is the purpose of unit testing?

Unit tests are [automated tests](https://builtin.com/learn/tech-dictionary/automation-testing). In other words, unit testing is performed by software (such as a unit testing framework or unit testing tool) and not manually by a developer. This means unit tests allow automated, repeatable, continuous testing.

* Identify how many units do we have in our calculator program?

2 Units. Add and Subtract.

**Unit Test Plan 1.**

**Application info:**

|  |  |
| --- | --- |
| Module #: Add | Application #: Calc |
| Tester: Student | Test Manager: Teacher. |

**Test information:**

|  |
| --- |
| **Module Overview** |
| *Describe the purpose of the module/class/method/unit.*  The module counts together two integers and returns their sum.  The module also converts the inputs to integer if the input are string values |
| **Module Inputs** |
| *Describe the inputs to the module/class/method/unit.*  The method takes as inputs two integers *(NB!* *No actual values here)* |
| **Module Outputs** |
| *Describe the outputs from the module/class/method/unit.*  The method returns the sum of the two integers *(NB!* *No actual values here)* |
| **Logic Flow** |
| *Describe the logic flow of the module/method/unit.*   * Program assigns a value for a * Program assigns a value for b * Program calls the sum function and returns the sum of a and b |
| **Test Cases** |
| *List all test cases to be executed, ex:*   * *ID number each test case.* * *Positive and negative test cases, valid and invalid test cases (as in exercise 3)*   ***\*Yellow cells should be filled as part of Part 2***   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Test Case ID | Test Case Description | InputA | InputB | Expected Output | Actual Output | Pass/Fail | | 001 | Test Adding Two Positive Numbers | 5 | 10 | 15 |  |  | | 002 | Test Adding Two Negative Numbers | -5 | -10 | -15 |  |  | | 003 | Test Adding Positive Number and Zero | 0 | 10 | 10 |  |  | | 004 | Test Adding Negative Number and Zero | 0 | -10 | -10 |  |  | | 005 | Test Adding Negative Number and Positive Number | 5 | -10 | -5 |  |  | | 006 | Test Adding Negative Number and Positive Number | -5 | 10 | 5 |  |  | | 007 | Test Adding Max Value and Positive Number | 2147483647 | 10 | -2147483639 |  |  | | 008 | Test Adding Max Value and Negative Number | 2147483647 | -10 | 2147483637 |  |  | | 009 | Test Adding Min Value and Posi | -2147483648 | 10 | -2147483638 |  |  | | 010 | Test Adding Min Value and Negative Number | -2147483647 | -10 | 2147483638 |  |  | | 011 | Test Adding a string and a Positive Number | “not a number” | 5 | NumberFormatException |  |  | | 012 | Test Adding Two null values | null | null | NullPointerException |  |  |   Note: The Actual Output and Pass/Fail columns will be filled in during the testing process. |
| **Test Tools** |
| *Identify software used for unit testing.*  Eclipse IDE and Junit framework |
| ***Tester notes:***  After the unit test is perform, identify the bugs |

**Unit Test Plan 2.**

**Application info:**

|  |  |
| --- | --- |
| Module #: Subtract | Application #: Calc |
| Tester: Student | Test Manager: Teacher. |

**Test information:**

|  |
| --- |
| **Module Overview** |
| *Describe the purpose of the module/class/method/unit.*  The module subtracts one integer to another integer and return their difference.  The module also converts the inputs to integer if the input are string values |
| **Module Inputs** |
| *Describe the inputs to the module/class/method/unit.*  The method takes as inputs two integers *(NB!* *No actual values here)* |
| **Module Outputs** |
| *Describe the outputs from the module/class/method/unit.*  The method returns the different of the two integers *(NB!* *No actual values here)* |
| **Logic Flow** |
| *Describe the logic flow of the module/method/unit.*   * Program assigns a value for a * Program assigns a value for b * Program calls the sub function and returns the different of a and b |
| **Test Cases** |
| *List all test cases to be executed, ex:*   * *ID number each test case.* * *Positive and negative test cases, valid and invalid test cases (as in exercise 3)*   ***\*Yellow cells should be filled as part of Part 2***   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Test Case ID | Test Case Description | InputA | InputB | Expected Output | Actual Output | Pass/Fail | | 001 | Test Sub Two Positive Numbers | 5 | 2 | 3 |  |  | | 002 | Test Sub Positive Number to Negative Number | 5 | -10 | 15 |  |  | | 003 | Test Sub Two Negative Numbers | -5 | -2 | -3 |  |  | | 004 | Test Sub Negative Number to Negative Number | -5 | 10 | -15 |  |  | | 005 | Test Sub Large Number | 2147483646 | 2 | 2147483644 |  |  | | 006 | Test Sub Zero and Positive Number | 0 | 10 | -10 |  |  | | 007 | Test Same Input | 5 | 5 | -0 |  |  | | 008 | Test Sub Max Value and Negative Number | 2147483647 | -2 | ArithmeticException |  |  | | 009 | Test Sub Min Value and Positive Number | -2147483648 | 1 | ArithmeticException |  |  | | 010 | Test Sub Null Values | null | Null | NullPointerException |  |  | | 011 | Test Sub Invalid String | “not a number” | 10 | NumberFormatException |  |  |   Note: The Actual Output and Pass/Fail columns will be filled in during the testing process. |
| **Test Tools** |
| *Identify software used for unit testing.*  Eclipse IDE and Junit framework |
| ***Tester notes:***  After the unit test is perform, identify the bugs |