A picture containing icon

Description automatically generated

CS265FZ Software Testing

Lab 2 – Boundary Value Testing

**There are THREE exercises to complete.**

**Two pieces of work need to be submitted:**

1. Fill in this lab sheet and submit it to Moodle. You don’t need to attach your source code in this form. You need to upload your source separately.
2. Submit all the required source code to Moodle. Make sure your source code is tested in Eclipse and is executable.
3. Make sure you provide detailed comments in the source code explaining:
   1. What are the faults in the source code?
   2. How the faults are fixed?

**Problem 1**

(Source Code: ***Lab2\_Program1.java***) A store in a city offers different discounts depending on the purchases made by the individual. To test the software that calculates the discounts, it is possible to identify the ranges of purchase values that earn the different discounts. For example, a purchase in the range of €1 to €50 inclusive has no discount, a purchase over €50 to €200 has a 5% discount, and purchases of over €201 inclusive to €500 have a 10% discounts, and purchases of €501 and above have a 15% discount. Invalid input will have a return value of 0 as discount.

**Note:** discounts are presented as follows: no discount is presented with 0, discount of 5% with 0.05, discount of 10% with 0.1, and 15% discount is presented as 1.5.

1. ***Task 1***Identify the input boundary value classes and output boundary classes.

|  |  |  |
| --- | --- | --- |
| **Input Boundary Values** | | |
| Parameter | Test Class | Boundary Value |
| **purchase** | EC-1\*  EC-2\*  EC-3  EC-4  EC-5  EC-6  EC-7  EC-8  EC-9  EC-10 | Integer.MIN\_VALUE  0  1  50  51  200  201  500  501  Integer.MAX\_VALUE |

|  |  |  |
| --- | --- | --- |
| **Output Boundary Values** | | |
| Parameter | Test Case | Boundary Value |
| Return Value  (**discount**) | EC-19  EC-20  EC-21  EC-22 | 0  0.05  0.1  0.15 |

1. ***Task 2***

Fill in the test data table (The following is one possible solution to the problem. Students may have equally valid solutions that are different from the example solution below.)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Cases/Boundaries Covered** | **Input** | **Expected Outputs** |
| *purchase* | *Discount* |
| T1.1 | EC-3, 19 | 1 | 0 |
| T1.2 | EC-4,[19] | 50 | 0 |
| T1.3 | EC-5,20 | 51 | 0.05 |
| T1.4 | EC-6, [20] | 200 | 0.05 |
| T1.5 | EC-7, 21 | 201 | 0.1 |
| T1.6 | EC-8, [21] | 500 | 0.1 |
| T1.7 | EC-9, 22 | 501 | 1.5 |
| T1.8 | EC-10, [22] | Integer.MAX\_VALUE | 1.5 |
| T1.9\* | EC- 1\* | Integer.MIN\_VALUE | 0 |
| T1.10\* | EC-2\* | 0 | 0 |

1. ***Task 3***

Based on the specification given above, write your testing code in JUnit 5 to test the source code of the program provided on Moodle (“***Lab2\_Program1.java***”). Make sure your test code is named as “***Lab1\_Task1.java***”.

1. ***Task 4***

Based on the test results, provide the correct version of the “***Lab2\_Program1.java***”, and rename it to “***Lab2\_Program1\_Fix.java***”.

**Problem 2**

The program (Source Code: ***Lab2\_Program2.java***) Grade combines an exam and coursework mark into a single grade. The values for exam and coursework are integers. If the exam or coursework mark is less than 50% then the grade returned is a ‘Fail’. To pass the course with a ‘Pass, C’, the student must score between 50% and 60% in the exam, and at least 50% in the coursework. They will pass the course with ‘Pass, B’, if they score at least 60% in the exam and 50% in the coursework. In addition to this, if the average of the exam and the coursework is at least 70%, then they are awarded a ‘Pass, A’. Input values that are less than 0 or greater than 100 for either the exam or coursework are invalid, and the program will return a message to say ‘Marks out of range’.

1. ***Task 1***

Identify the input boundary value classes and output boundary classes.

|  |  |  |
| --- | --- | --- |
| **Input Boundary Values** | | |
| Parameter | Test Class | Boundary Value |
| **exam** | EC-1\*  EC-2  EC-3  EC-4  EC-5  EC-6  EC-7  EC-8  EC-9\*  EC-10\* | Integer.MIN\_VALUE  -1  0  49  50  59  60  100  101  Integer.MAX\_VALUE |
| **course** | EC-11\*  EC-12\*  EC-13  EC-14  EC-15  EC-16  EC-17\*  EC-18\* | Integer.MIN\_VALUE  -1  0  49  50  100  101  Integer.MAX\_VALUE |

|  |  |  |
| --- | --- | --- |
| **Output Boundary Values** | | |
| Parameter | Test Case | Boundary Value |
| Return Value  (**grade**) | EC-19\*  EC-20  EC-21  EC-22  EC-23 | “Marks out of range”  “Fail”  “Pass, C”  “Pass, B”  “Pass, A” |

1. ***Task 2***

Fill in the test data table (The following is one possible solution to the problem. Students may have equally valid solutions that are different from the example solution below.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Cases/Boundaries Covered** | **Inputs** | | **Expected Outputs** |
| *exam* | *course* | *Result* |
| T1.1 | EC-3, 15, 20 | 0 | 50 | Fail |
| T1.2 | EC-4, [15, 20] | 49 | 50 | Fail |
| T1.3 | EC-5, [15], 21 | 50 | 50 | Pass, C |
| T1.4 | EC-6, [15, 21] | 59 | 50 | Pass, C |
| T1.5 | EC-7, [15], 22 | 60 | 50 | Pass, B |
| T1.6 | EC-8, [15], 23 | 100 | 50 | Pass, A |
| T1.7 | EC-[5], 13, [20] | 50 | 0 | Fail |
| T1.8 | EC-[5],14, [20] | 50 | 49 | Fail |
| T1.9 | EC-[5, 15, 21] | 50 | 50 | Pass, C |
| T1.10 | EC-[7], 16, [23] | 60 | 100 | Pass, A |
| T1.11\* | EC-1\*, [15], 19 | INT\_MIN | 50 | Marks out of Range |
| T1.12\* | EC-2\*, [15, 19] | -1 | 50 | Marks out of Range |
| T1.13\* | EC-[9\*, 15, 19] | 101 | 50 | Marks out of range |
| T1.14\* | EC-10\*, [15, 19] | INT\_MAX | 50 | Marks out of range |
| T1.15\* | EC-[5], 11\*, [19] | 50 | INT\_MIN | Marks out of Range |
| T1.16\* | EC-[5], 12\*, [19] | 50 | -1 | Marks out of range |
| T1.17\* | EC-[5], 17\*, [19] | 50 | 101 | Marks out of range |
| T1.18\* | EC-[5], 18\*, [19] | 50 | INT\_MAX | Marks out of Range |

1. ***Task 3***

Based on the specification given above, write your testing code in JUnit 5 to test the source code of the program provided on Moodle (“***Lab2\_Program2.java***”). Make sure your test code is named as “***Lab2\_Task2.java***”.

1. ***Task 4***

Based on the test results, provide the correct version of the “***Lab2\_Program2.java***”, and rename it to “***Lab2\_Program2\_Fix.java***”.

**Problem 3**

A program (Source Code: ***Lab2\_Program3.java***) is used to assess the cost of a car insurance policy. It takes three inputs of age, gender and marital status. If the age entered is less than 16 or greater than 65, the program returns a premium of zero. The input for gender takes the form of ‘M’ for male and ‘F’ for female. If an incorrect value for the gender is entered the program returns a premium of zero. In general a premium is €500. However, if a person is male, under 25 and not married then an extra €1500 is added to the premium. If the person is female or married the premium falls by €200, and if the person is aged between 45 and 65 inclusive the premium falls by €100.

1. ***Task 1***

Identify the input boundary value classes and output boundary classes.

|  |  |  |
| --- | --- | --- |
| **Input Boundary Values** | | |
| Parameter | Test Class | Boundary Value |
| **age** | EC-1\*  EC-2\*  EC-3  EC-4  EC-5  EC-6  EC-7  EC-8  EC-9\*  EC-10\* | Integer.MIN\_VALUE  15  16  24  25  44  45  65  66  Integer.MAX\_VALUE |
| **gender** | EC-11  EC-12  EC-13\* | M  F  Invalid input |
| **married** | EC-14  EC-15 | true  false |

|  |  |  |
| --- | --- | --- |
| **Output Boundary Values** | | |
| Parameter | Test Case | Boundary Value |
| Return Value  (premium) | EC-16  EC-17  EC-18  EC-19  EC-20  EC-21 | 0  200  300  400  500  2000 |

1. ***Task 2***

Fill in the test data table (The following is one possible solution to the problem. Students may have equally valid solutions that are different from the example solution below.)

**Test Data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Cases Covered** | **Inputs** | | | **Expected Outputs** |
| *age* | *gender* | *married* | *premium* |
| T1.1 | EC-7, 12, 14, 17 | 45 | F | True | 200 |
| T1.2 | EC-5, [12, 14], 18 | 25 | F | True | 300 |
| T1.3 | EC-8, 11, 15, 19 | 65 | M | False | 400 |
| T1.4 | EC-6, [11, 15], 20 | 44 | M | False | 500 |
| T1.5 | EC-4, [11, 15], 21 | 24 | M | False | 2000 |
| T1.6 | EC-3, [11, 15], 21 | 16 | M | False | 2000 |
| T1.7\* | EC-1\* | Integer.MIN\_VALUE | M | False | 0 |
| T1.8\* | EC-2\* | 15 | M | False | 0 |
| T1.9\* | EC-9\* | 66 | M | False | 0 |
| T1.10\* | EC-10\* | Integer.MAX\_VALUE | M | False | 0 |
| T1.11\* | EC-13\* | 25 | G | False | 0 |

1. ***Task 3***

Based on the specification given above, write your testing code in JUnit 5 to test the source code of the program provided on Moodle (“***Lab2\_Program3.java***”). Make sure your test code is named as “***Lab2\_Task3.java***”.

1. ***Task 4***

Based on the test results, provide the correct version of the “***Lab2\_Program3.java***”, and rename it to “***Lab2\_Program3\_Fix.java***”.