**HW-T2**

***Category-Partition Method***

|  |  |
| --- | --- |
| Group Member Names |  |
| |  |  | | --- | --- | | 1. | Carson Davis | | 2. | Johnnie Oldfield | | |  |  | | --- | --- | | 3. | Sean Northcutt | | 4. | Andrew Russel | |

You can work with your group, or by yourself or with someone else. For this homework, you will use the CPM to develop the appropriate unit tests for a method.

**Problem to be tested**

1. Consider this problem for which you have been provided the code:

You will write a method, *getSum* that returns the sum of the integers read from a file. However, some numbers are skipped. When you encounter a line whose first character is ‘s’, then an integer will follow it. The integer tells you how many of the next consecutive numbers should be skipped. For Example 1 below, the yellow numbers are added to produce 60, skipping the values as indicated. As shown in Example 3, a skip of zero doesn’t skip anything, effectively ignoring the skip. Example 4 shows a skip beyond the length of the file which should just skip to the end of the file, ignoring the invalid skips. Finally, Example 5 shows overlapping (or embedded) skips. The embedded skip, s5, should be ignored, meaning it is treated as any other skipped value.

|  |  |  |  |
| --- | --- | --- | --- |
| Example 1 – Sum=60 | Example 2 – Sum=90 | Example 3 – Sum=10 | Example 5 – Sum=10 |
| E:\Data-Classes\CS 1302 - Programming 2\homework\Fall, 2015\hw7\a.jpg | s5  2  43  3  6  9  12  4  56  s2  10  4  8  s2  1  5  s2  4  2  10 | 2  3  s0  4  1 | 2  3  s3  4  s5  2  4  1 |
| Example 4 – Sum=5 |
| 2  3  s7  4  1 |

**Test Derivation for *getSum():int***

Use the template below to use the category-partition method to derive a set of test cases for this method. You can assume the file exists and is in the proper format as described above. Nothing else is assumed.

1. **Identify independently testable feature (use case, method)**

We will test the *getSum* method in the *NumAdder* class.

1. **Identify inputs/parameters for each feature under test.**
2. **Identify the categories/characteristics for each input/parameter.**
3. **Partition categories into choices**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category/Characteristic** | **Characteristic**  **Identifier** | **Choices** | **Choice**  **Identifier** | **Comment** |
| # of skips | NS | 0 | 0 | No skips |
|  |  | 1 | 1 | Exactly one skip |
|  |  | >=2 | More than 1 | More than skip |
| # of skipped lines | NSL | 0 | 0 | No lines are skipped |
|  |  | >=1 | More than 0 | Number of total skipped lines |
| # of lines in file | NLF | 0 | 0 | No lines in file |
|  |  | 1 | Exactly 1 | Exactly 1 line in file |
|  |  | >=2 | More than 1 | More than 1 line in file |
| # of imbedded Skips | NIS | True | yes | A skip is inside a skip |
|  |  | False | no | There are no skips inside a skip |

1. **TSL Input – Preliminary**

TSL input file

#Number of Lines in File

NLF:

0.

1.

>=2.

#Number of Skips

NS:

0.

1.

>=2.

#Number of Skipped Lines

NSL:

0.

>=1.

#Number of Imbedded Skips

NIS:

True.

False.

1. **TSL Output – Worst Case**

*----------------------------------------*

*TSLgenerator*

*(C) University of California Irvine,*

*and Oregon State University, 2001*

*----------------------------------------*

*36 test frames generated and written to getSum.txt.tsl*

1. **TSL Input & Justification for Constraints**
2. TSL input file

#Number of Lines in File

NLF:

0. [error]

1. [property oneLine]

>=2. [property lines]

#Number of Skips

NS:

0. [property noSkip]

1.

>=2. [property skips]

#Number of Skipped Lines

NSL:

0. [if noSkip] [if oneLine]

>=1. [if lines]

#Number of Imbedded Skips

NIS:

True. [if skips] [single]

False.

1. Justification for the constraints:

|  |  |
| --- | --- |
| **Constraint** | **Justification** |
| noSkip on NLBS | If there are no skips then the number of lines skipped is 0. |
| noSkip on NIS | If there are no skips then there cannot be imbedded skips. |
| oneLine on NSL | If there is only one line then no lines can be skipped |
| lines on NSL | Lines can only be skipped if there are more than 1 line in the file. |
| error on NLF | If there are no lines in a file then only do one test for an error. |
| single on NIS | Only do one test for imbedded skips. |

1. **TSL Output – Test Frames**

|  |  |  |
| --- | --- | --- |
| Test Case 1 <error>  NLF : 0  Test Case 2 <single> (follows [if])  NIS : True  Test Case 3 (Key = 2.1.1.2.)  NLF : 1  NS : 0  NSL : 0  NIS : False  Test Case 4 (Key = 2.2.1.2.)  NLF : 1  NS : 1  NSL : 0  NIS : False |  | Test Case 5 (Key = 2.3.1.2.)  NLF : 1  NS : >=2  NSL : 0  NIS : False  Test Case 6 (Key = 3.1.2.2.)  NLF : >=2  NS : 0  NSL : >=1  NIS : False  Test Case 7 (Key = 3.2.2.2.)  NLF : >=2  NS : 1  NSL : >=1  NIS : False  Test Case 8 (Key = 3.3.2.2.)  NLF : >=2  NS : >=2  NSL : >=1  NIS : False |

1. **Test Cases**

*[Specify the exact test cases here, erase this comment. Note that the file contents can be specified as a string, space delimited. I’ve provide you a method to write the string to a file for testing purposes. For example:*

String vals = "1 2 3 s3 1 2 3 4";

// Writes vals to a file, one token per line.

*writeFile*(fileName,vals);

*Erase this comment.]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test: | 1 | Expected Result: | | error |
| File Contents: | | | "" | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test: | 2 | Expected Result: | | 9 |
| File Contents: | | | "1 2 3 s3 1 s2 3 4" | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test: | 3 | Expected Result: | | 1 |
| File Contents: | | | "1" | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test: | 4 | Expected Result: | | 3 |
| File Contents: | | | "s2" | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test: | 5 | Expected Result: | | 0 |
| File Contents: | | | "s1 s1" | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test: | 6 | Expected Result: | | 0 |
| File Contents: | | | "s1 3" | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test: | 7 | Expected Result: | | 3 |
| File Contents: | | | "1 2 s1 4" | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test: | 8 | Expected Result: | | 15 |
| File Contents: | | | "1 4 5 s2 2 3 s1 3 5" | | |

1. **Write & Run JUnit Tests**

*2 test cases do not work. 5 & 6. 5 wants 2 skips but no skipped lines which is not possible. 6 wants more than one skipped line without skips*

1. **Submission**

Place this completed document in the *prob3* folder along with the code and JUnit tests. Zip the *prob3* folder and submit on Blazeview. Only one submission for each group.