Psuedocode for InputMismatch.java

* Non-constant variables:
  + sum to hold the sum of two ints
  + intsFound to hold how many ints were passed
* Open a do while block that does what’s below until intsFound is 2
  + Ask user to enter an integer
  + Try to record the integer into a temporary placeholder
    - If what was passed wasn’t an integer, don’t do what’s below, but do this because it’s an error, go back to the beginning of the do loop:
      * Tell user to put an integer
      * Reset the input
  + Add the temporary variable into the sum
  + Increase intsFound by 1
* Print the sum of the two ints found

Test cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cases** | **Input** | **Expected Result** | **Actual Result** | **Did it pass?** |
| Case 1 | 1.3  H  3  Ro  %  6 | Sum: 9 | Sum: 9 | Y |
| Case 2 | 63.453  100.000001  Hello  324  382h  1 | Sum: 325 | Sum: 325 | Y |
| Case 3 | 1  21 | Sum: 22 | Sum: 22 | Y |

Screenshots: A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

UML Class Diagram:

A screenshot of a cell phone

Description automatically generated

Lessons learned:

For this assignment, I learned how to work with Exceptions and how to catch any errors my programs are making. I used various exception types to catch user errors and attempted to current them/let them know of their errors. Exception catching is very important when making programs for the real world because you can never trust the end user with a simple task, so you must make it as stupidly easy and problematic as possible, as shown in InputMismatch.

Checklist

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Source java files** | **Y** |  |
|  | **Compressed files:** | **Y** |  |
|  | FirstInitialLastName\_Project8\_Moss.zip | **Y** |  |
|  | FirstInitialLastName\_Project8\_doc.zip | **Y** |  |
|  | **Program compiles** | **Y** |  |
|  | **Program runs** | **Y** |  |
|  | **Checklist is completed and included in the Documentation** | **Y** |  |
|  | **Documentation file:** | **Y** |  |
|  | **Comprehensive Test Plan** | **Y** |  |
|  | **Screenshots based on Test Plan** | **Y** |  |
|  | **UML Diagram** | **Y** |  |
|  | **Algorithms/Pseudocode** | **Y** |  |
|  | **Flowchart** | **Y** |  |
|  | **Lessons Learned** | **Y** |  |