Assessment: Assignments 02

Student Name: Lin Chen (student num:041053912)

Lab Professor Name: Gustavo Adami

Lab Section Number: 314

Due Date: Friday, July 29

Part 1 Understand the Problem

The program needs to input the size and weight of the potato chip bags and code a method by using Boolean to compare if within 0.01 oz or not to get the good bag or bad bag, using math method to calculate the difference(absolute value). Use loop(do,while) methods to verify the quality of the bags and calculate the total count of the bags.

Part 2 UML, Pseudocode and flow-charts 1) UML

PotatoChipBag

- +REGULAR: int = 1
- +LARGE: int = 2
- -REGULAR_WEIGHT: double = 9.25
- -LARGE_WEIGHT: double = 15.75
- +EPSILON:double= 0.010
- -size:int
- -weight:double
- +PotatoChipBag()
- +PotatoChipBag(size: int, weight:double)
- +getSize():int
- +setSize(size:int):void
- +getWeight():double
- +setWeight(weight:double):void
- +isBagCorrectWeight():boolean

UML Class Diagram by Lin Chen Assignment 02 (22S) UML Class Diagrams

method main is static so it is underlined in UML.

Assignment02

+main(args:String[]):void

User

- +inputInteger():int
- +inputInteger(String message):int
- +inputDouble():double
- +inputDouble(String message):double

```
2) Pseudocode for the method of Boolean isBagCorrectWeight()
```

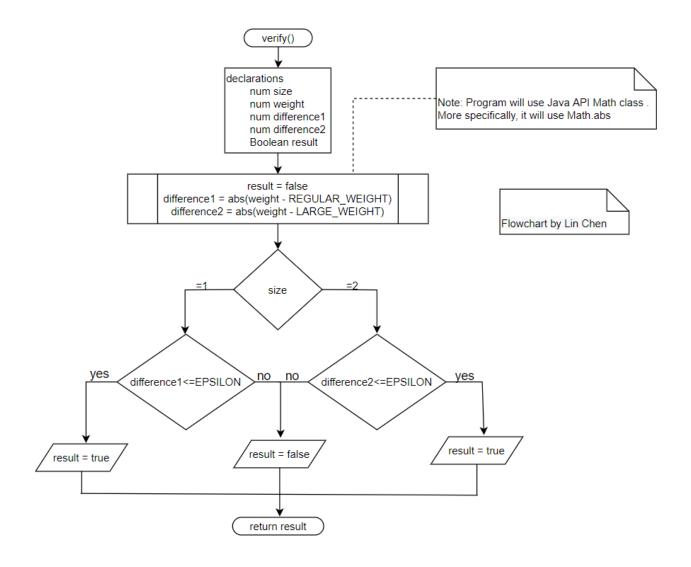
```
//verify()
           Declarations
                num size
                num weight
                num difference1
                num difference2
                Boolean result
                result = false
                difference1 = absolute value of (weight-REGULAR WEIGHT)
                difference2 = absolute value of (weight-LARGE_WEIGHT)
                     if size = 1
                           if difference1 <= EPSILON
                                 result = true;
                     else if size = 2
                           if difference2 <= EPSILON
                                 result = true;
                     endif
                     return result
3) Pseucode for the mian method
         //main(String[] args)
         Start
                 Declarations
                        PotatoChipBag bag
                        User user
                        num size
                        num weight
                        num goodBags=0
                        num badBags=0
                        num totalBags=0
                        String value
             do
                    output"Enter bag size: "
                    output"1 for regular size"
                    output"2 for large size"
                    keyboard input size
                    bag.setSize(size)
```

if size = 1 or size = 2

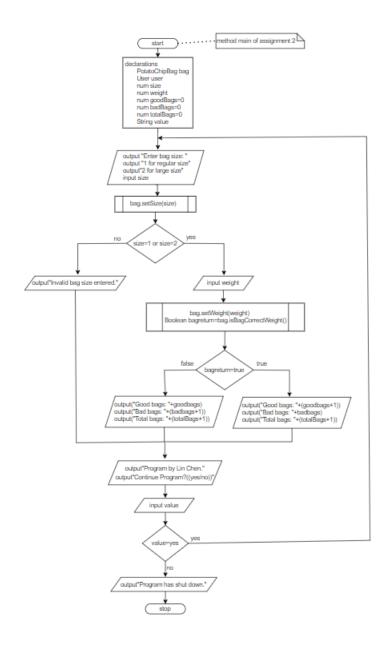
keyboard input weight

```
bag.setWeight(weight)
          Boolean bagreturn = bag.isBagCorrectWeight()
               If bagreturn = true
                     output"Good bags: " + (goodBags + 1)
                     output"Bad bags: " + badBags
                     output"Total bags: " + (totalBags + 1)
               else output"Good bags: " + goodBags
                     output"Bad bags: " + (badBags + 1)
                     output"Total bags: " + (totalBags + 1)
               endif
      else output"Invalid bag size entered"
      endif
     output"Program by Lin Chen"
     output"Continue Program?(yes/no)"
     keyboard input value
 while value = "yes"
      output"Program has shut down"
Stop
```

4) Flowchart for the method of Boolean isBagCorrectWeight()



5) Flowchart for the mian method



Part 3 Test Algorithm with Simple Inputs

Table 1: Test Plan for method main for the repetition structure logic (looping tests)

Input	Expected Output	Actual Output	Description
1	Good Bag: 1	Good Bag: 1	test passes, and the program continues.
9.25	Bad Bag: 0	Bad Bag: 0	
yes	Total Bags: 1	Total Bags: 1	
	Continue? (yes/no)	Continue? (yes/no)	
	Enter Bag Size:	Enter Bag Size:	
	1 for regular size	1 for regular size	
	2 for large size	2 for large size	

2	Good Bag: 1	Good Bag: 1	test passes, and the program continues.
10.00	Bad Bag: 1	Bad Bag: 1	
YES	Total Bags: 2	Total Bags: 2	
	Continue? (yes/no)	Continue? (yes/no)	
	Enter Bag Size:	Enter Bag Size:	
	1 for regular size	1 for regular size	
	2 for large size	2 for large size	
1	Good Bag: 2	Good Bag: 2	test passes, and the program shuts down.
9.24	Bad Bag: 1	Bad Bag: 1	
no	Total Bags: 3	Total Bags: 3	
	Continue? (yes/no)	Continue? (yes/no)	
	Program has shut down	Program has shut down	
1	Good Bag: 1	Good Bag: 1	The program shuts down.
9.25	Bad Bag: 0	Bad Bag: 0	
carrot	Total Bags: 1	Total Bags: 1	
	Continue? (yes/no)	Continue? (yes/no)	
	Program has shut down	Program has shut down	

Table 2: Test Plan for method main for good chip bag, bad chip bag, total bags

Input	Expected output	Actual output	Description
1	Good Bag: 1	Good Bag: 1	Matches, one good bag, correct report,
9.25	Bad Bag: 0	Bad Bag: 0	program exits.
No	Total Bags: 1	Total Bags: 1	
1	Good Bag: 1	Good Bag: 1	Matches, one good bag, one bad bag,
9.25	Bad Bag: 0	Bad Bag: 0	correct report, program exits
yes	Total Bags: 1	Total Bags: 1	
2			
16.75	Good Bag: 1	Good Bag: 1	
NO	Bad Bag: 1	Bad Bag: 1	
	Total Bags: 2	Total Bags: 2	
3	Invalid bag size entered	Invalid bag size entered	Matches, invalid bag size entered, one bad
yes			bag input, program exits
1	Good Bag: 0	Good Bag: 0	
10.00	Bad Bag: 1	Bad Bag: 1	
	Total Bags: 1	Total Bags: 1	

Table 3: Test Plan for method is BagCorrectWeight()

Field values	Expected return value	Actual return value	Description
Size=1 Weight=9.24	true	true	Matches
Size=1 Weight=9.25	true	true	Matches
Size=1 Weight=9.26	true	true	Matches
Size=2 Weight=15.74	true	true	Matches
Size=2 Weight=15.75	true	true	Matches
Size=2 Weight=15.76	true	true	Matches
Size=1 Weight=9.2399	false	false	Not matches
Size=1 Weight=9.2601	false	false	Not matches
Size=2 Weight=15.7399	false	false	Not matches
Size=2 Weight=15.7601	false	false	Not matches

Part 4 Translate the Algorithm into Java

DONE.

Part 5 Compile and Test Your Program

```
ories r < terminated > Assignment02 [Java Application] C:\AmazonCorretto\jdk11.0.15_9\bin\javaw.exe
   Enter bag size:
    1 for regular size
    2 for Large size
   Enter weight
    9.25
   Good bags: 1
   Bad bags: 0
   Total bags: 1
   Program by Lin Chen
   Continue Program? (yes/no)
   yes
   Enter bag size:
    1 for regular size
    2 for Large size
   Enter weight
   15.75
   Good bags: 2
   Bad bags: 0
   Total bags: 2
   Program by Lin Chen
   Continue Program? (yes/no)
   yes
```

```
Continue Program? (yes/no)
yes
Enter bag size:
1 for regular size
2 for Large size
42
Invalid bag size entered
Program by Lin Chen
Continue Program?(yes/no)
yes
Enter bag size:
1 for regular size
2 for Large size
Enter weight
15.74
Good bags: 2
Bad bags: 1
Total bags: 3
Program by Lin Chen
Continue Program?(yes/no)
Program has shut down
```

```
Console × History

<terminated > AssignmentO2 [Java Application] C:\AmazonCorretto

Enter bag size:

1 for regular size

2 for Large size

2

Enter weight

15.74

Good bags: 1

Bad bags: 0

Total bags: 1

Program by Lin Chen

Continue Program? (yes/no)

no

Program has shut down
```

Input	Expected Output	Actual Output	Description
1	Good Bag: 1	Good Bag: 1	Java program output matches the
9.25	Bad Bag: 0	Bad Bag: 0	expected output.
yes	Total Bags: 1	Total Bags: 1	
	Continue? (yes/no)	Continue? (yes/no)	
	Enter Bag Size:	Enter Bag Size:	
	1 for regular size	1 for regular size	
	2 for large size	2 for large size	
2	Good Bag: 1	Good Bag: 1	Java program output matches the
10.00	Bad Bag: 1	Bad Bag: 1	expected output.
YES	Total Bags: 2	Total Bags: 2	
	Continue? (yes/no)	Continue? (yes/no)	
	Enter Bag Size:	Enter Bag Size:	
	1 for regular size	1 for regular size	
	2 for large size	2 for large size	
1	Good Bag: 2	Good Bag: 2	Java program output matches the
9.24	Bad Bag: 1	Bad Bag: 1	expected output.
no	Total Bags: 3	Total Bags: 3	
	Continue? (yes/no)	Continue? (yes/no)	
	Program has shut down	Program has shut down	
1	Good Bag: 1	Good Bag: 1	The program shuts down.
9.25	Bad Bag: 0	Bad Bag: 0	
carrot	Total Bags: 1	Total Bags: 1	
	Continue? (yes/no)	Continue? (yes/no)	
	Program has shut down	Program has shut down	

References

- 1. [1] Cay Horstmann. (2019). Big Java Early Objects. 7th Ed. Wiley.
- 3. Slides from week 10 and 11.