Assessment: Assignment 02

Student Name: Vedant Goswami

Section Number: 321

Due Date: 29/7/2022

# Understand the problem.

* I will be needed to make a program that can count the number of good and bad chips and their total. In addition to that the program will be needed to hold the count till user enter that the input is no longer needed and close the program. The input is 4 decimal point long so I will have to format the input to only take 4 decimal points.
* In addition to that I will have to work using pre-made methods from another programmer and

Alteration of that code is prohibited. And therefore, I will have to write my part of code in main method and isBagCorrectWeight() method.

* For calculation there will be few such as in isBagCorrectWeight() where I will be needed to use tolerance = weight - REGULAR\_WEIGHT or tolerance = weight - LARGE\_WEIGHT.
* And will be needed to use increment function (++) to increase the count of good bag and bad bag according to user input.
* For the isBagCorrectWeight() method I will use nested if and else statements. However, for main method I will use while and if-else loop.

# Pseucode.

# Pseucode for main method.

deceleration of variables and calling out methods

Scanner input = new Scanner

potatochipbag bag = new potatochipbag

user = new user

num Good\_Bag

num Bad\_Bag

num Sum

string ToContinue = “yes”

// creating a loop

//not cash sensitive

While (ToContinue = “yes”){

//calling out beg size method from PotatoChipBag() and calling out inputinteger method from user class

Bag.setSize(user.inputInteger("Enter Bag size \n" + "1 for Regular size \n"+ "2 for Large size"))

//creating nested if to evaluate the user input and decide the size of bag

If(beg size() ==1 or bag size == 2 ){

Print enter weight

Bag.Getweight

// creating if statement in side if to evaluate the user input and decide whether the bag is good, bad or invalid

If (Bag.isBagCorrectWeight()==true) {

//increasing the count of good bag.

Good\_Bag++; } }

//increasing the count of good bag.

Else { Bad\_Bag ++}

//increasing the sum of total value.

Sum = good\_bag + bad\_bag

//printing the report

Print ("Good Bag: "+Good\_Bag)

Print ("Bad Bag: "+Bad\_Bag)

Print ("Total: "+Sum)

Print (" Continue Program? (yes/no)")

//caling out ToContinue tocheak the user’s Boolean (yes/no) input in next line

ToContinue = input.nextLine(); }

else {

print ("The input is Invalid \n" +" Do you want to Continue program?(yes/no)");

/caling out ToContinue tocheak the user’s Boolean(yes/no) input in next line

ToContinue = input.nextLine(); }

Stop.

# Pseucode for isBagCorrect() method.

Declaration

Boolean result = false;

// creating two if statements to check the size of bag.

if(this.size==REGULAR) {

//always positive

if((weight-REGULAR\_WEIGHT)>=EPSILON) {

return result;

}

else { result = true; return result }

}

//checking the size for large bag

If (this.size = LARGE) {

//always positive use Manth.abs.

if( (weight-LARGE\_WEIGHT)>=EPSILON) { return result;}

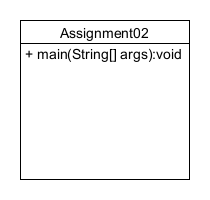
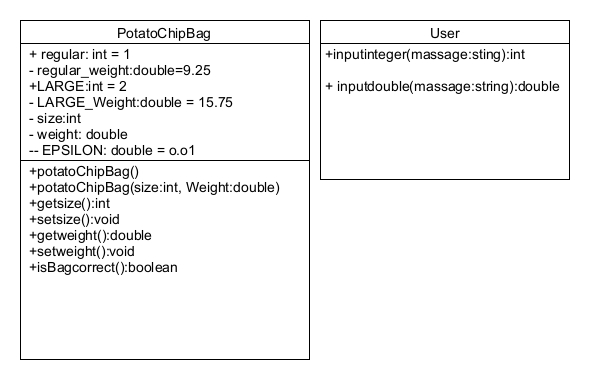
else { result = true;

return result;} }

return result;

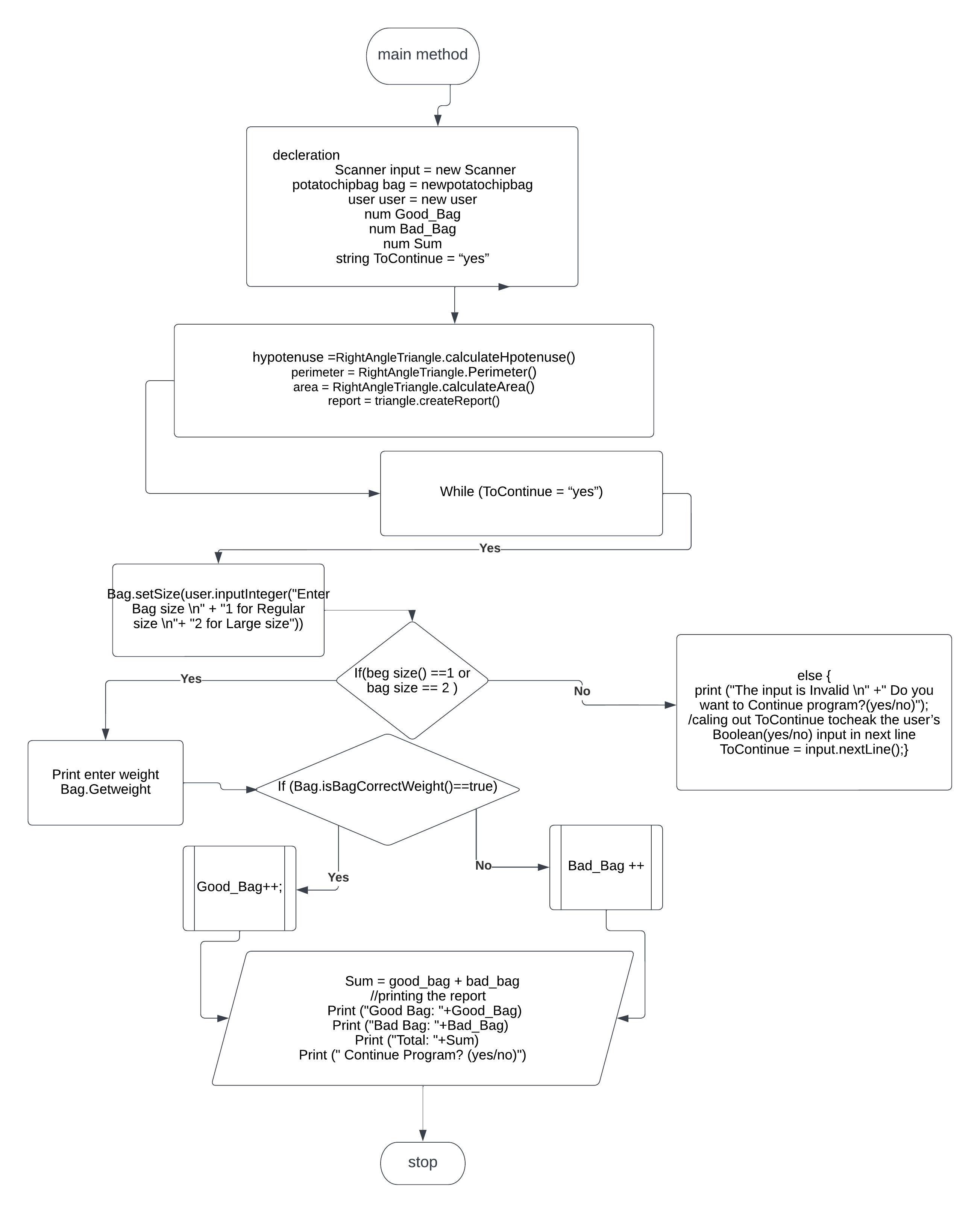
}

# UMl



# Flowchart

## Flowchart for main method.



yes

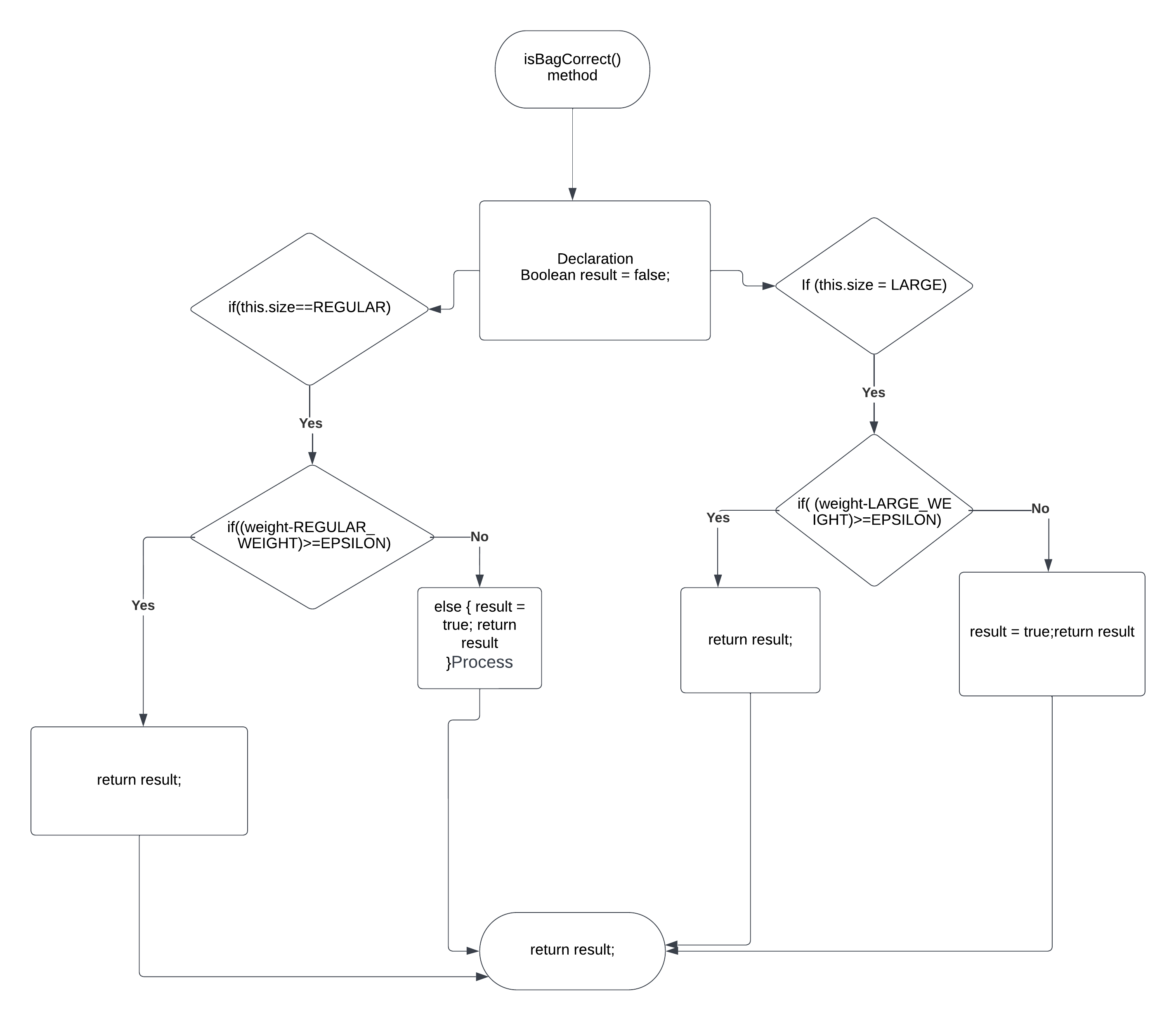
If yes/no?



no

stop

## Flowchart for isBagCorrect() method.



Test plan for algorithm

## Test plan for algorithm looping.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Description |
| 1  9.25  yes | Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  yes  Enter Bag size  1 for Regular size  2 for Large size | Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  yes  Enter Bag size  1 for Regular size  2 for Large size | **Algorithm test plan Expected outcome of program matches the actual come. And program continues.** |
| 2  15.75  no | Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  no | Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  no | **Algorithm test plan Expected outcome of program matches the actual come. And program stops** |
| 1  9.26  veant | Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  veant | Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  veant | **Algorithm test plan Expected outcome of program matches the actual come. And program stops** |
| 2  15.76  YES | Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  YES  Enter Bag size  1 for Regular size  2 for Large size | Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  YES  Enter Bag size  1 for Regular size | **Algorithm test plan Expected outcome of program matches the actual come. And program stops** |

## Test plan for method main for good chip bag, bad chip bag, total bags.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Description |
| 1  9.25  No | Good Bag: 1  Bad Bag: 0  Total Bags: 1  Continue Program? | Good Bag: 1  Bad Bag: 0  Total Bags: 1  Continue Program? | **Algorithm test plan Expected outcome of program matches the actual come. And program stops after one bag.** |
| 1  9.25  Yes  2  15.75  No | Good Bag: 1  Bad Bag: 0  Total Bag: 1  Continue Program?  Good Bag: 2  Bad Bag: 0  Total Bags: 2 | Good Bag: 1  Bad Bag: 0  Total Bag: 1  Continue Program?  Good Bag: 2  Bad Bag: 0  Total Bags: 2 | **Algorithm test plan Expected outcome of program matches the actual come. And program stops after two good bags.** |
| 0  No | Invalid bag size entered | Invalid bag size entered | **Algorithm test plan Expected outcome of program matches the actual come. And program stops and show invalid input and ask user if he wants to continue.** |
| 1  9.22  no | Good Bag: 0  Bad Bag: 1  Total Bag: 1  Continue Program? | Good Bag: 0  Bad Bag: 1  Total Bag: 1  Continue Program? | **Algorithm test plan Expected outcome of program matches the actual come. And program stops after one bag.** |
| 1  9.26  Yes  2  14.66  no | Good Bag: 1  Bad Bag: 0  Total Bag: 1  Continue Program?  : yes  Good Bag: 1  Bad Bag: 1  Total Bag: 2  Continue Program? | Good Bag: 1  Bad Bag: 0  Total Bag: 1  Continue Program?  : yes  Good Bag: 1  Bad Bag: 1  Total Bag: 2  Continue Program? | **Algorithm test plan Expected outcome of program matches the actual come. And program stops after one good bag and one bad bag .** |
| 3  no | The input is Invalid | The input is Invalid | **Algorithm test plan Expected outcome of program matches the actual out come. program shows invalid input and stops when continue input is no.** |
| 1  9.25  Yes  2  15  Yes  3  Yes  1  9.25  no | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.25  Good Bag: 1  Bad Bag: 0  Total: 1  yes  Enter Bag size  1 for Regular size  2 for Large size  2  Enter Weight:  15  Good Bag: 1  Bad Bag: 1  Total: 2  yes  Enter Bag size  1 for Regular size  2 for Large size  3  The input is Invalid    yes  Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.25  Good Bag: 2  Bad Bag: 1  Total: 3 | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.25  Good Bag: 1  Bad Bag: 0  Total: 1  yes  Enter Bag size  1 for Regular size  2 for Large size  2  Enter Weight:  15  Good Bag: 1  Bad Bag: 1  Total: 2  yes  Enter Bag size  1 for Regular size  2 for Large size  3  The input is Invalid    yes  Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.25  Good Bag: 2  Bad Bag: 1  Total: 3 | **Algorithm test plan Expected outcome of program matches the actual out come. And the end output shows**  **2 good bag one bad bag and one invalid entry and the total is 3.** |

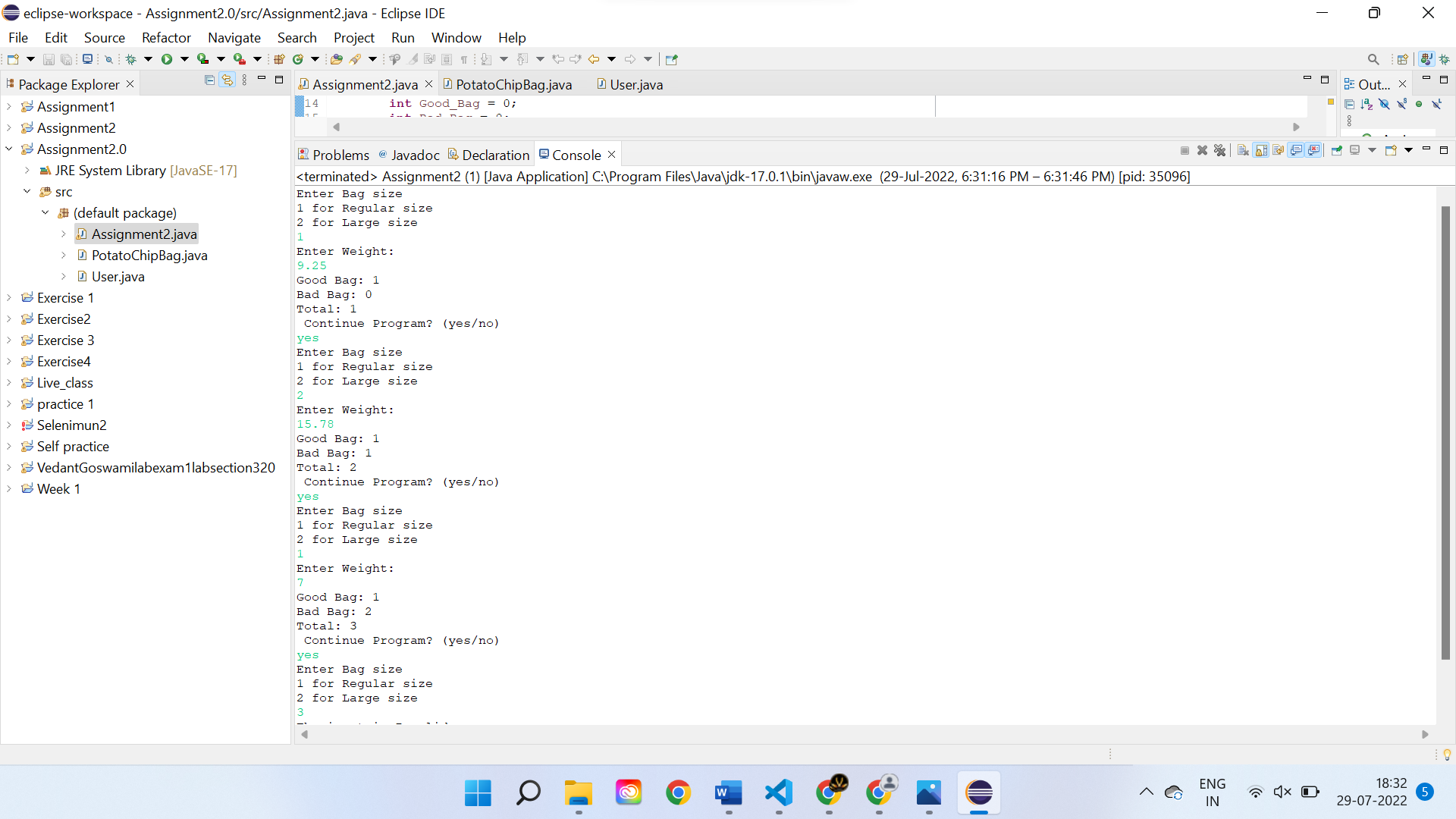
# Test Plan for method isBagCorrectWeight().

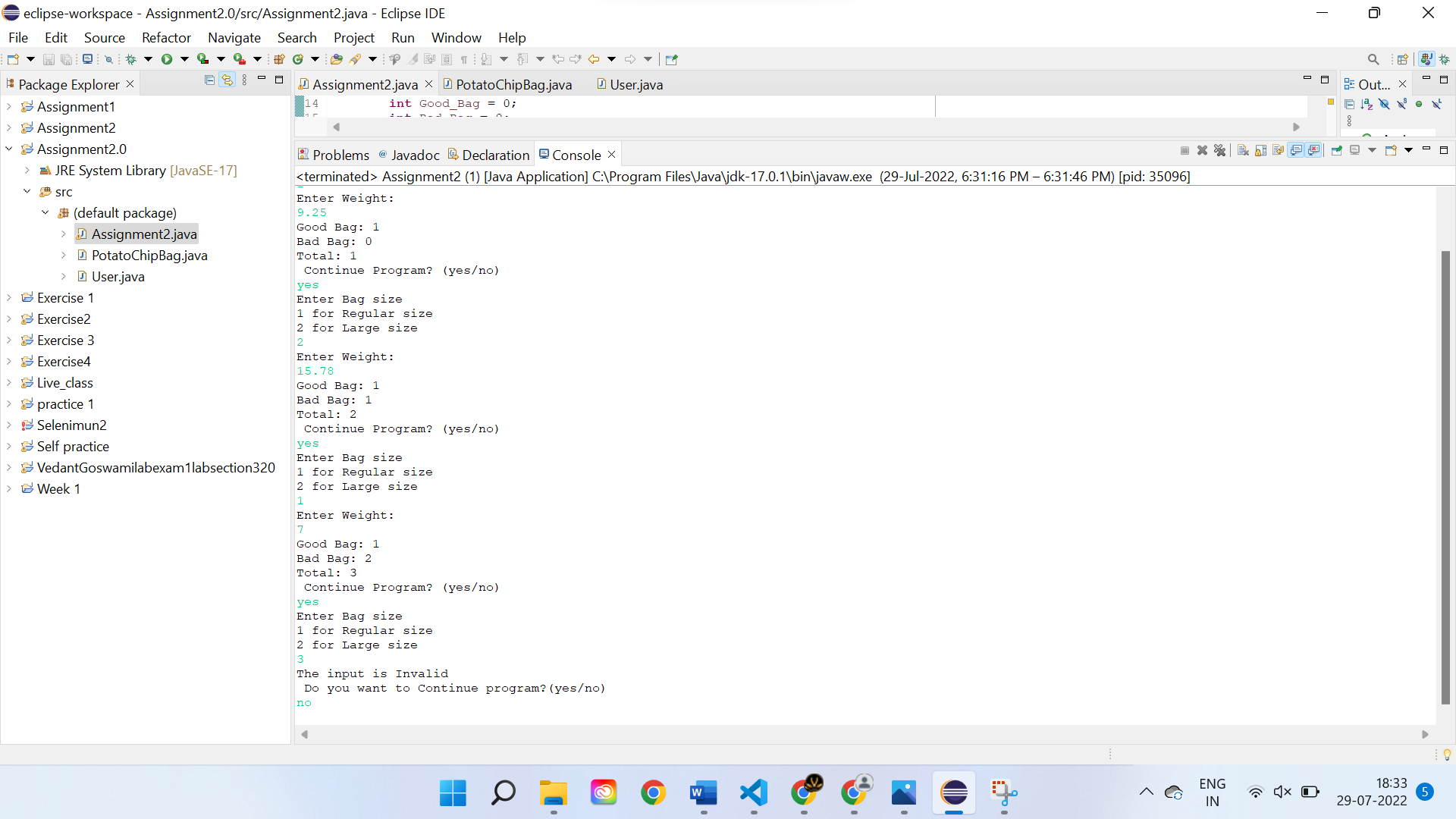
|  |  |  |  |
| --- | --- | --- | --- |
| Field values | Expected return value | Actual return value | Description |
| size = 1  weight = 9.24 | true | true | Matches |
| size = 1  weight = 9.25 | true | true | Matches |
| size = 1  weight = 9.26 | true | true | Matches |
| size = 2  weight = 15.75 | true | true | Matches |
| size = 2  weight = 15.74 | true | true | Matches |
| size = 2  weight = 15.76 | true | true | Matches |
| size = 1  weight = 9.2399 | false | false | Matches |
| size = 1  weight = 9.22601 | false | false | Matches |
| size = 1  weight = 15.7399 | false | false | Matches |
| size = 1  weight = 15.7601 | false | false | Matches |
|  |  |  |  |

## Test plan for Java code.

|  |  |  |  |
| --- | --- | --- | --- |
| 1  9.25  No | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.25  Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  no  ? | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.25  Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  no | Expected out-put of program matches the actual programs output. |
| 1  9.25  Yes  2  15.75  No | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.25  Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  yes  Enter Bag size  1 for Regular size  2 for Large size  2  Enter Weight:  15.75  Good Bag: 2  Bad Bag: 0  Total: 2  Continue Program? (yes/no)  no | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.25  Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  yes  Enter Bag size  1 for Regular size  2 for Large size  2  Enter Weight:  15.75  Good Bag: 2  Bad Bag: 0  Total: 2  Continue Program? (yes/no)  no | Expected out-put of program matches the actual programs output. |
| 0  No | Enter Bag size  1 for Regular size  2 for Large size  0  The input is Invalid  Do you want to Continue program?(yes/no)  no | Enter Bag size  1 for Regular size  2 for Large size  0  The input is Invalid  Do you want to Continue program?(yes/no)  no | Expected out-put of program matches the actual programs output. |
| 1  9.22  no | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.22  Good Bag: 0  Bad Bag: 1  Total: 1  Continue Program? (yes/no)  no | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.22  Good Bag: 0  Bad Bag: 1  Total: 1  Continue Program? (yes/no)  no | Expected out-put of program matches the actual programs output. |
| 1  9.26  Yes  2  14.66  no | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.26  Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  yes  Enter Bag size  1 for Regular size  2 for Large size  2  Enter Weight:  14.66  Good Bag: 1  Bad Bag: 1  Total: 2  Continue Program? (yes/no)  no | Enter Bag size  1 for Regular size  2 for Large size  1  Enter Weight:  9.26  Good Bag: 1  Bad Bag: 0  Total: 1  Continue Program? (yes/no)  yes  Enter Bag size  1 for Regular size  2 for Large size  2  Enter Weight:  14.66  Good Bag: 1  Bad Bag: 1  Total: 2  Continue Program? (yes/no)  no | Expected out-put of program matches the actual programs output. |
| 3  no | Enter Bag size  1 for Regular size  2 for Large size  3  The input is Invalid  Do you want to Continue program?(yes/no)  no | Enter Bag size  1 for Regular size  2 for Large size  3  The input is Invalid  Do you want to Continue program?(yes/no)  no | Expected out-put of program matches the actual programs output. |
| 1  vedant | Error | 1 for Regular size  2 for Large size  1  Enter Weight:  ve  Exception in thread "main" java.util.InputMismatchException  at java.base/java.util.Scanner.throwFor(Scanner.java:939)  at java.base/java.util.Scanner.next(Scanner.java:1594)  at java.base/java.util.Scanner.nextDouble(Scanner.java:2564)  at User.inputDouble(User.java:26)  at User.inputDouble(User.java:34)  at Assignment2.main(Assignment2.java:28) | As expected the values are expected to be in numerical value so the strings give error |

## Screen shot of Running program





# JAVA code

# Main method :

/\*Assessment: Assignment 02

Student Name: Vedant Goswami

Section Number: 321

Due Date: 29/7/2022

description: A program that takes size and weight as input and tess user if it is good bag or bad bag and also sum up the total

bag count; it runs till user asks to run again.

\*

\*/

//importing scanner util

**import** java.util.Scanner;

**public** **class** Assignment2{

**public** **static** **void** main(String[] args) {

//deceleration of variables and calling out methods

PotatoChipBag Bag = **new** PotatoChipBag();

Scanner input = **new** Scanner(System.***in***);

User = **new** User();

**int** Good\_Bag = 0;

**int** Bad\_Bag = 0;

**int** Sum = 0;

String ToContinue = "yes";

// creating a loop so program does not stop till perticular input.

**while**(ToContinue.equalsIgnoreCase("yes")) {

//calling out beg size method from PotatoChipBag() and calling out inputinteger method from user class

Bag.setSize(user.inputInteger("Enter Bag size \n" + "1 for Regular size \n"+ "2 for Large size"));

//creating nested if to evaluate the user input and decide the size of bag.

**if**(Bag.getSize()==1 || Bag.getSize()==2) {

Bag.setWeight(user.inputDouble("Enter Weight: "));

// creating if statement in side if to evaluate the user input and decide whether the bag is good, bad or invalid

**if**(Bag.isBagCorrectWeight()==**true**) {

//increasing the count of good bag.

Good\_Bag++;

}

**else** {

//increasing the count of bad bag.

Bad\_Bag++;

}

Sum = Good\_Bag + Bad\_Bag;

System.***out***.println("Good Bag: "+Good\_Bag);

System.***out***.println("Bad Bag: "+Bad\_Bag);

System.***out***.println("Total: "+Sum);

System.***out***.println(" Continue Program? (yes/no)");

ToContinue = input.nextLine();

}

**else** {

System.***out***.println("The input is Invalid \n" +" Do you want to Continue program?(yes/no)");

//caling out ToContinue tocheak the user’s Boolean (yes/no) input in next line.

ToContinue = input.nextLine();

}

};

}

}

# User:

//importing scanner util

**import** java.util.Scanner;

//creating user class fot tahking input from main method.

**public** **class** User {

**private** Scanner keyboard = **new** Scanner(System.***in***);

//decalration of inputInteger method, which will be used for size of chip bag.

**public** **int** inputInteger() {

**int** value = keyboard.nextInt();

keyboard.nextLine();

**return** value;

}

//decleraation of method to get sting output for size of bag.

**public** **int** inputInteger(String message) {

System.***out***.println(message);

**int** value = inputInteger();

**return** value;

}

//decalration of inputDouble method, which will be used for weight of chip bag.)

**public** **double** inputDouble() {

**double** value = keyboard.nextDouble();

keyboard.nextLine();

**return** value;

}

//decleraation of method to get sting output for weight of bag.

**public** **double** inputDouble(String message) {

System.***out***.println(message);

**double** value = inputDouble();

**return** value;

}

}

# PotatoChipBag:

/\*creating PotatoChipBag() class to get the size and weight and it will be used to check if the bag is good or bad in main method.\*/

**public** **class** PotatoChipBag {

//decleration of constants and variables.

**private** **int** size; // Regular or Large (use constants below)

**private** **double** weight;// oz

**public** **static** **final** **int** ***REGULAR*** = 1;

**private** **static** **final** **double** ***REGULAR\_WEIGHT*** = 9.25; // oz

**public** **static** **final** **int** ***LARGE*** = 2;

**private** **static** **final** **double** ***LARGE\_WEIGHT*** = 15.75; // oz

**private** **static** **final** **double** ***EPSILON*** = 0.01;

//creating method for regular size bag and its weight

**public** PotatoChipBag() {

**this**(***REGULAR***, ***REGULAR\_WEIGHT***);

}

//creating mutator for potatochipbag and setting values of size and weight.

**public** PotatoChipBag(**int** size, **double** weight) {

**this**.size = size;

**this**.weight = weight;

}

//getters for size

**public** **int** getSize() {

**return** size;

}

//mutators for size

**public** **void** setSize(**int** size) {

**this**.size = size;

}

//getters for weight

**public** **double** getWeight() {

**return** weight;

}

//mutators for size

**public** **void** setWeight(**double** weight) {

**this**.weight = weight;

}

// creating isBagCOrrectWeight() method to check if the weight input belongs to good or bad bag

**public** **boolean** isBagCorrectWeight() {

**boolean** result = **false**;

//checking if user have input regular or large size.

**if**(**this**.size == ***REGULAR***) {

//checking good or bad bag for regular size chips.

**if**(Math.*abs*(weight-***REGULAR\_WEIGHT***)>=***EPSILON***) {

**return** result;

}

**else** {

result = **true**;

**return** result;

}

}

**if**(**this**.size == ***LARGE***) {

//checking good or bad bag for large size chips.

**if**(Math.*abs*(weight-***LARGE\_WEIGHT***)>=***EPSILON***) {

**return** result;

}

**else** {

result = **true**;

**return** result;

}

}

**return** result;

}

}