Assessment: Exercise 04

Student Name: Vedant Goswami

Section Number: 321

Due Date: 22/7/2022

Understand the problem,

For solution I will be needed to create code that will ask user to input the length of meterstick in meter. Then after I will have to create a menu that will have four options verify(which will check if the length of meter stick is within the tolerance of 0.0001 meters),covert to centimetre(which will convert the length of meterstick into centimetre), convert to millimetre(which will convert the length of meterstick into millimetre), and report(which will show the length of stick). And last but not least output will need to be formatted to 5 decimal points.

Calculation needed:-

Difference = always positive (Actual length - Expected length).

## Pseudocode

# 1] Pseudocode for verify() method.

Verify(){

Declaration

Num Epsilon = o.ooo1

Num expected\_length = 1

Num length

Num Tolerance = Math.abs(length – expected\_length)

If (Tolrance <= Epsilone ) { Print “The length of stick is within the tolerance 0.0001” }

Else Print “The length of stick is not within the tolerance 0.0001”

Return Tolerance

}

# 2] Pseudocode for toString() method.

toString(){

Declaration

Num length

String report = “The length of meter stick is” + length

return report

}

# 3] Pseudocode for main() method.

Main(){

declaration

Import scanner keyboard = new scanner

MeterStick stick = new MeterStick();

Num userLength

Num NumberOfOpration

Constant Num CheckTolrance = 1

Constant Num Convertintocentimeter = 2

Constant Num Convertintomillimeter = 3

Constant Num PrintReport = 4

//creating menu

Print ("1. Check Tolerance"

+ "2. Convert into centimetre"

+ "3. Convert into millimetre"

+ "4. Print report")

//taking the input from user

NumberOfOpration = //take input from user in integer using keyboard.nextInt()

// creating if / else statements for the selected menu items

If (NumberOfOpration = Convertintocentimeter){

print( stick.toCentimeters()}

else if(NumberOfOpration = Convertintomillimeter){

print("The length of meter stick in millimeter is"+" "+ stick.toMillimeters()) }

else if (NumberOfOpration = Printreport){

print("The length of meter stick is"+" "+ stick.toString())}

else if(NumberOfOpration = CheckTolerance){ print (stick.verify())}

//exemption statement if the given integer is out of menu select options.

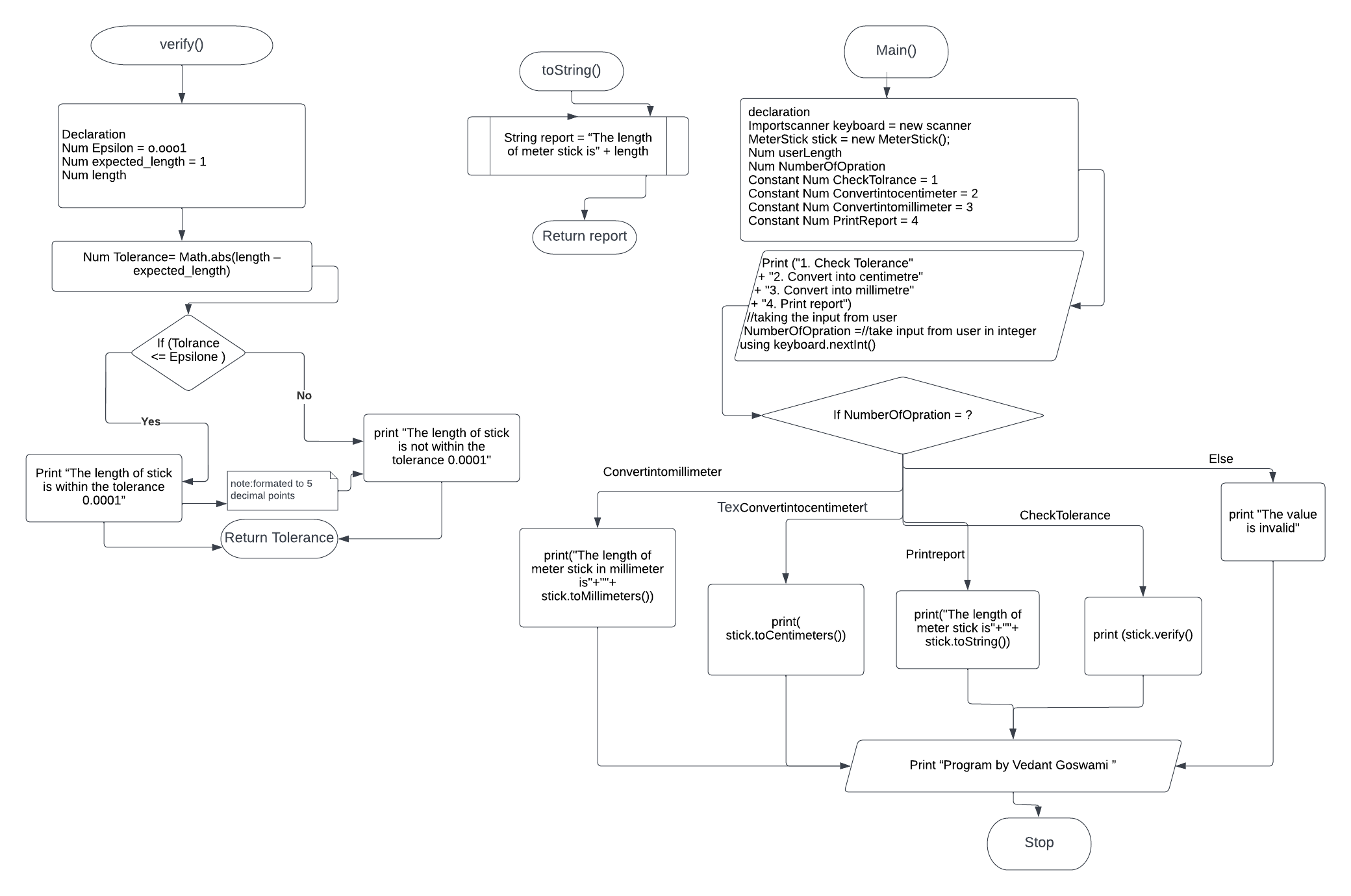
Else print ("The value is invalid")

// printing the name of writer

Print “Program by Vedant Goswami ”

Stop }

## Flowchart



# UML Class

# 

# Algorithm Test plane for Verify()

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Expected outcome** | **Actual outcome** | **Description** |
| **1** | **The length of stick is within the tolerance 0.00010** | **The length of stick is within the tolerance 0.0001** | **Algorithm test plan Expected outcome of program matches the actual come.** |
| **1.00003** | **The length of stick is within the tolerance 0.00010** | **The length of stick is within the tolerance 0.00010** | **Algorithm test plan Expected outcome of program matches the actual come.(**  **Note the values are calculated using a calculator)** |
| **0.888** | **The length of stick is not within the Tolerance 0.00010** | **The length of stick is not within the Tolerance 0.00010** | **Algorithm test plan Expected outcome of program matches the actual come.(**  **Note the values are calculated using a calculator)** |
| **0.9999** | **The length of stick is within the tolerance 0.00010** | **The length of stick is within the tolerance 0.00010** | **Algorithm test plan Expected outcome of program matches the actual come.(**  **Note the values are calculated using a calculator)** |
| **0.100001** | **The length of stick is not within the Tolerance 0.00010** | **The length of stick is not within the Tolerance 0.00010** | **Algorithm test plan Expected outcome of program matches the actual come.(**  **Note the values are calculated using a calculator)** |

# Algorithm Test plane for Mainmethod()

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected outcome | Actual outcome | Description |
| 1  1 | The length of stick is within the tolerance 0.00010 | The length of stick is within the tolerance 0.0001 | Algorithm test plan Expected outcome of program matches the actual come. |
| 1  2 | The length of meter stick in centimetre is 100.00000 | The length of meter stick in centimetre is 100.00000 | Algorithm test plan Expected outcome of program matches the actual come.(  Note the values are calculated using a calculator) |
| 1  3 | The length of meter stick in millimetre is 1000.00000 | The length of meter stick in millimetre is 1000.00000 | Algorithm test plan Expected outcome of program matches the actual come.(  Note the values are calculated using a calculator) |
| 1  4 | The length of meter stick is 1.00000 | The length of meter stick is 1.00000 | Algorithm test plan Expected outcome of program matches the actual come.(  Note the values are calculated using a calculator) |
| 1  11 | The value is invalid | The value is invalid | Algorithm test plan Expected outcome of program matches the actual come.(  Note the values are calculated using a calculator) |

# Screenshot Of Running Program

# Program Test plane for Verify()

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected outcome | Actual outcome | Description |
| 1 | The length of stick is within the tolerance 0.00010 | The length of stick is within the tolerance 0.0001 | Expected out-put of program matches the actual programs output |
| 1.00003 | The length of stick is within the tolerance 0.00010 | The length of stick is within the tolerance 0.00010 | Expected out-put of program matches the actual programs output |
| 0.888 | The length of stick is not within the Tolerance 0.00010 | The length of stick is not within the Tolerance 0.00010 | Expected out-put of program matches the actual programs output |
| 0.9999 | The length of stick is within the tolerance 0.00010 | The length of stick is within the tolerance 0.00010 | Expected out-put of program matches the actual programs output |
| 0.100001 | The length of stick is not within the Tolerance 0.00010 | The length of stick is not within the Tolerance 0.00010 | Expected out-put of program matches the actual programs output |

# Program Test plane for Main method()

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected outcome | Actual outcome | Description |
| 1  1 | The length of stick is within the tolerance 0.00010 | The length of stick is within the tolerance 0.0001 | Expected out-put of program matches the actual programs output. |
| 1  2 | The length of meter stick in centimetre is 100.00000 | The length of meter stick in centimetre is 100.00000 | Expected out-put of program matches the actual programs output. |
| 1  3 | The length of meter stick in millimetre is 1000.00000 | The length of meter stick in millimetre is 1000.00000 | Expected out-put of program matches the actual programs output. |
| 1  4 | The length of meter stick is 1.00000 | The length of meter stick is 1.00000 | Expected out-put of program matches the actual programs output. |
| 1  11 | The value is invalid | The value is invalid | Expected out-put of program matches the actual programs output. |
| 1  vedant | error | 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.nextInt(Scanner.java:2258)  at java.base/java.util.Scanner.nextInt(Scanner.java:2212)  at main.main(main.java:92) | As expected the values are expected to be in numerical value so the strings give error |
| vedant | error | 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 main.main(main.java:76) | As expected the values are expected to be in numerical value so the strings give error |

# Java code for main method

import java.util.Scanner;

/\*public class main {

public static void main(String[] args) {

// FIRST

double lengthOfStick, ExpactedLength = 1.00000;

Scanner input = new Scanner(System.in);

System.out.print("What is a lenght of meter stick");

lengthOfStick = input.nextDouble();

double Tolerance = Math.abs(lengthOfStick - ExpactedLength);

if(Tolerance <= 0.0001) { System.out.print("The length of stick is within the Tolerance" + " "+Tolerance +"\n"); }

else System.out.print("The length of stick is not within the Tolerance" + " "+Tolerance +"\n");

// SECOND

int CheckTolerance = 1,Convertintomillimeter = 3,Printreport = 4 ;

int Convertintocentimeter =2;

Scanner input2 = new Scanner(System.in);

System.out.print("1. Cheak Tolerance\n"

+ "2. Convert into centimeter\n"

+ "3. Convert into millimeter\n"

+ "4. Print report");

System.out.print("\n Enter the number of process you want.");

int NumberOfOpration;

NumberOfOpration = input.nextInt();

if(NumberOfOpration == Convertintocentimeter ){

System.out.print("The length of meter stick in centimeter is"+" "+ lengthOfStick\*100 +"\n"); }

else if(NumberOfOpration == Convertintomillimeter){

System.out.print("The length of meter stick in millimeter is"+" "+ lengthOfStick\*1000); }

else if(NumberOfOpration == Printreport){

System.out.print("The length of meter stick is"+" "+ lengthOfStick); }

//else if(NumberOfOpration == CheckTolerance){

// System.out.print("The length of meter stick is"+" "+ lengthOfStick); };;

else System.out.print("The value is invalid") ;

}\*/

/\*

\* File level comment goes here

\*/

import java.util.Scanner;

/\*

Assessment: Exercise 04

Student Name: Vedant Goswami

Section Number: 321

Due Date: 22/7/2022

\*/

public class main {

/\*

\* main method

\*/

public static void main(String[] args) {

//declaration and import of scanner + MeterStick

Scanner keyboard = new Scanner(System.in);

MeterStick stick = new MeterStick();

double userLength;

//asking user to give the length of meter stick in meters

System.out.println("Meter stick checker program.");

System.out.print("Enter measured length in meters: \n");

userLength = keyboard.nextDouble();

keyboard.nextLine();

stick.setLength(userLength);

//print command for showing Menu

System.out.print("1. Check Tolerance\n"

+ "2. Convert into centimetre\n"

+ "3. Convert into millimetre\n"

+ "4. Print report\n");

System.out.print("\n Enter the number of process you want.");

//taking input from the user

int NumberOfOpration;

NumberOfOpration = keyboard.nextInt();

// create constants for the menu

final int CheckTolerance = 1,Convertintomillimeter = 3,Printreport = 4 ;

final int Convertintocentimeter =2;

// creating if / else statements for the selected menu items

if(NumberOfOpration == Convertintocentimeter ){

System.out.printf( "%.5f",stick.toCentimeters()); }

else if(NumberOfOpration == Convertintomillimeter){

System.out.printf("The length of meter stick in millimetre is %.5f ",stick.toMillimeters()); }

else if(NumberOfOpration == Printreport){

System.out.printf(stick.toString(),stick.getLength()); }

else if(NumberOfOpration == CheckTolerance){

System.out.printf("%.5s",stick.verify());

}

//exemption statement if the given integer is out of menu select options.

else System.out.printf("The value is invalid") ;

System.out.println("\n Program by Vedant Goswami"); // replace with your name

}

}

# Java code for Meterstick

**public** **class** MeterStick {

//deceration

**public** **static** **final** **double** ***EXPECTED\_LENGTH*** = 1.0; // meters

**public** **static** **final** **double** ***EPSILON*** = 0.0001;

**private** **double** length;

**double** Tolerance;

//Constructor for Meterstick : Expacted\_Length

**public** MeterStick() {

**this**(***EXPECTED\_LENGTH***);

}

/\*

\* Constructor for MeterstickLength

\*/

**public** MeterStick(**double** length) {

**this**.length = length;

}

/\*

\* mutator of length

\*/

**public** **double** getLength() {

**return** length;

}

/\*

\* mutator of length

\*/

**public** **void** setLength(**double** length) {

**this**.length = length;

}

/\*

\* creating method to convert meter into centimeters

\*/

**public** **double** toCentimeters() {

**double** centimeters;

centimeters = length \* 100;

System.***out***.print("The length of meter stick in centimetre is"+" ");

**return** centimeters;

}

/\*

\* creating method to convert meter into millimetres

\*/

**public** **double** toMillimeters() {

**double** millimeters;

millimeters = length \* 1000;

**return** millimeters;

}

/\*

\* method for report that gives length of stick

\*/

**public** String toString() {

String report;

report = "The length of meter stick is %.5f ";

**return** report;

}

/\*

\* creating method to check whether the length of meter stick is withinthe Tolerance or not

\*/

**public** String verify() {

String report="0";

**double** Tolerance = Math.*abs*(length - ***EXPECTED\_LENGTH***);

**if**(Tolerance <= 0.0001) { System.***out***.print("The length of stick is within the tolerance 0.0001"); }

**else** System.***out***.print("The length of stick is not within the Tolerance 0.0001");

**return** report;}

}