Assessment: Assignment 02

Student Name: Christopher Frosst

Lab Professor Name: David Houtman

Lab Section Number: 301

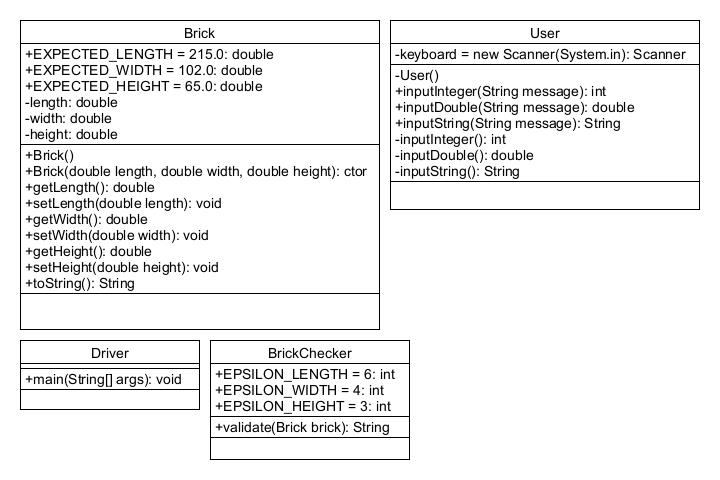
Due Date: March 29 2024

# Understand the problem

In the BrickChecker Class, I will create 3 static final for WIDTH (4mm), LENGTH(6mm) and HEIGHT(3mm) which will be used to check the sizes between epsilon size and max size. Additionally, I will create local variables such as doubles and strings to check the tolerances then implement them into an if & else statement that determines if the user input is within range or not, and then the he output will be returned.

For the Driver class. I will make outputs that will both inform the user of information and also take in user input to do checks and prompt the user to loop back with a do & while statement afterwards.

# Detailed UML Class Diagrams



# Pseudocode

Class brickChecker

String validate(Brick brick)

Declarations

String report

num lengthDiffer

num widthDiffer

num heightDiffer

if (lengthDiffer <= brick && widthDiffer <= brick && heightDiffer <= brick)

return “Brick meets all standards”

else if(brick.getLength <= 0 && brick.getHeight <= 0 && brick.getWidth <= 0)

report = “Negative dimension detected, can not verify brick”

else

report = “brick does not meet all standards”

endif

if (lengthDiffer > EPSILON\_LENGTH)

report = “Length exceeds tolerance of “ + EPSILON\_LENGTH+ “ millimeters

if (widthDiffer > EPSILON\_WIDTH)

report = “Width exceeds tolerance of “ + EPSILON\_LENGTH+ “ millimeters

if (heightDiffer > EPSILON\_HEIGHT)

report = “Height exceeds tolerance of “ + EPSILON\_LENGTH+ “ millimeters

endif

return report

Class driver

Start

Declarations

String answer

num length

num width

num height

do

length = User.inputDouble(“Enter measured length (millimeters): ”)

brick.setlength(length)

width = User.inputDouble(“Enter measured width (millimeters): ”)

brick.setWidth(width)

height = User.inputDouble(“Enter measured height (millimeters): ”)

brick.setHeight(height)

output brick.toString()

output brickChecker.validate(brick)

output “Program by Christopher Frosst”

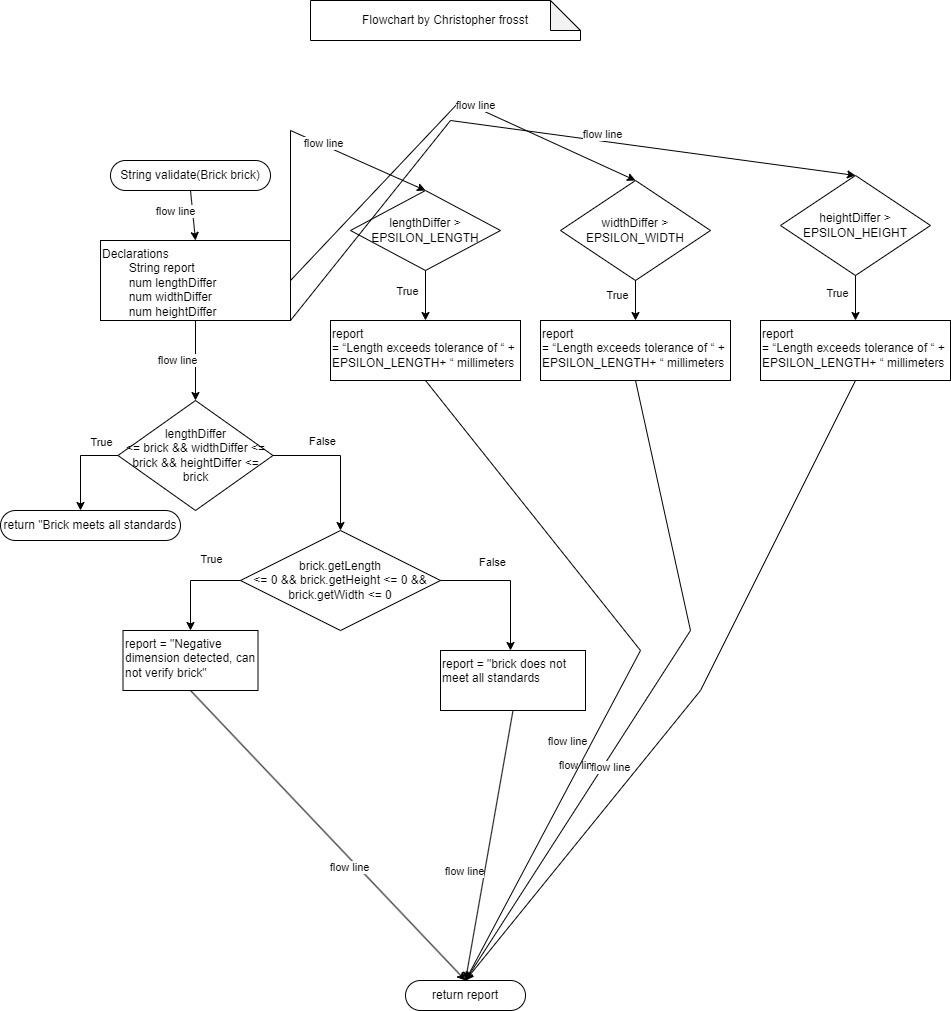
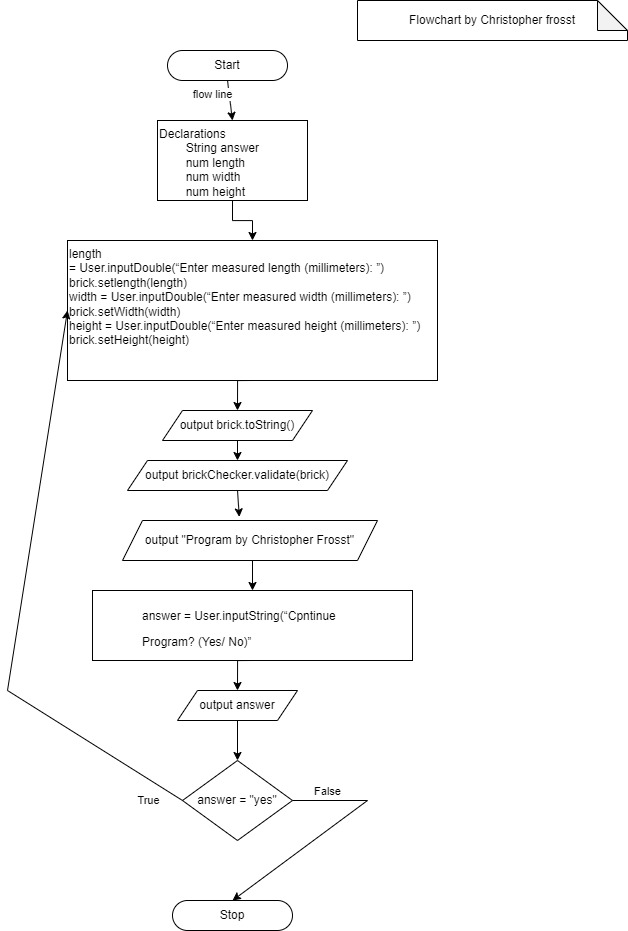
answer = User.inputString(“Cpntinue Program? (Yes/ No)”

output answer

while (answer = “yes”)

Stop

# Flowcharts



# 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 |
| 215  102  65  YES | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres):  Enter measured width (millimetres):  Enter measured height (millimetres):  Brick: length 215.0000, width 102.0000, height 65.0000  Brick dimensions meet all tolerance standards.  Program by Christopher Frosst  Continue Program? (yes / no)  Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres): | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres):  Enter measured width (millimetres):  Enter measured height (millimetres):  Brick: length 215.0000, width 102.0000, height 65.0000  Brick dimensions meet all tolerance standards.  Program by Christopher Frosst  Continue Program? (yes / no)  Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres): | The logic of the program is sequenced correctly and is in the correct order, additionally, when the user inputs yes, the program continues. |
| 215  102  65  NO | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres):  Enter measured width (millimetres):  Enter measured height (millimetres):  Brick: length 215.0000, width 102.0000, height 65.0000  Brick dimensions meet all tolerance standards.  Program by Christopher Frosst  Continue Program? (yes / no)  Exiting Program  Program shuts down | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres):  Enter measured width (millimetres):  Enter measured height (millimetres):  Brick: length 215.0000, width 102.0000, height 65.0000  Brick dimensions meet all tolerance standards.  Program by Christopher Frosst  Continue Program? (yes / no)  Exiting Program  Program shuts down | The logic of the program is sequenced correctly and is in the correct order, additionally, when the user inputs no, the program shuts down. |
| 215  102  65  maybe | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres):  Enter measured width (millimetres):  Enter measured height (millimetres):  Brick: length 215.0000, width 102.0000, height 65.0000  Brick dimensions meet all tolerance standards.  Program by Christopher Frosst  Continue Program? (yes / no)  Exiting Program  Program shuts down | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres):  Enter measured width (millimetres):  Enter measured height (millimetres):  Brick: length 215.0000, width 102.0000, height 65.0000  Brick dimensions meet all tolerance standards.  Program by Christopher Frosst  Continue Program? (yes / no)  Program shuts down | The logic of the program is sequenced correctly and is in the correct order, additionally, when the user inputs maybe, the program does not display the text “Exiting Program” because “no” was not inputted but the program shuts down |

## Table 2: Test Plan for method main for verifying the brick against any Declared Size.

# Program Test Plan (complete with at least 5 screen shots)

* You may re-use your algorithm test plans and document testing your Java program, e.g. **update the description column to document the program tests to indicate if the correct report was produced for the given output.** (There is no requirement to re-write all of the tests).
* Include the following extra test input in your table and explain what class and what method produced the results.

| Input | Actual Output | Code that produces highlighted output |
| --- | --- | --- |
| 1.467  5.5  Sammy | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured thickness (inches):  Enter measured width (inches):  Enter measured length (inches):  Invalid input. Enter a number: | The inputDouble method in the User class detects an invalid input with outputs to the user to re-enter a valid number |
| -215  102  65 | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured thickness (inches):  Negative dimension detected, can not verify brick. | The validate method in BrickChecker class notices a negative number and outputs the following text. |
| 295  132  7 | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres):  Enter measured width (millimetres):  Enter measured height (millimetres):  Brick: length 295.0000, width 132.0000, height 7.0000  Brick does not meet all standards  Length exceeds tolerance of 6 millimeters  Width exceeds tolerance of 4 millimeters  Height exceeds tolerance of 3 millimeters | The validate method reads the “If” statement in the BrickChecker classregarding tolerance levels and outputs according to user input and calculations done by the code |
|  | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres):  Enter measured width (millimetres):  Enter measured height (millimetres):  Brick: length 295.0000, width 132.0000, height 7.0000  Brick does not meet all standards  Length exceeds tolerance of 6 millimeters  Width exceeds tolerance of 4 millimeters  Height exceeds tolerance of 3 millimeters | The validate method in the BrickChecker class checks the “if” statements to confirm which unit of measure exceeded the tolerance and outputs accordingly |
|  | Enter brick dimensions noting that Declared Sizes are:  215.000 for length, 102.000 for width, 65.000 for height  Enter measured length (millimetres):  Enter measured width (millimetres):  Enter measured height (millimetres):  Brick: length 295.0000, width 132.0000, height 7.0000  Brick does not meet all standards  Length exceeds tolerance of 6 millimeters  Width exceeds tolerance of 4 millimeters  Height exceeds tolerance of 3 millimeters | The toString method in the Brick class formats the user inputted numbers and prints it out in the Driver class |

# References

[1] Cay Horstmann. (2019). Big Java Early Objects. 7th Ed. Wiley.

A black and white sign with white text

Description automatically generated

[2] <https://www.imperialbricks.co.uk/guidance/everything-you-need-to-know-about-imperial-brick-sizes>

[3] <https://www.imperialbricks.co.uk/guidance/dimensions-size-tolerance-guidance>