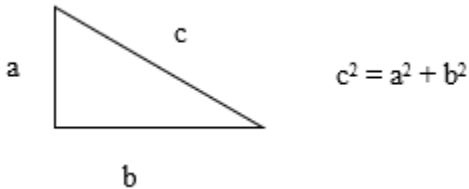


Assignment



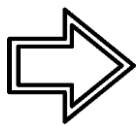
Overview

In math, the Pythagorean theorem is a fundamental relation in Euclidean geometry between the three sides of a right triangle. The formula is $a^2 + b^2 = c^2$ where a and b are the sides and c is the hypotenuse(long side of the triangle)



Of course we can easily make a console based application that gets inputs and calculates it for us, but it is way better to do it GUI based.

You are tasked in creating a program that gets A and B then calculates the C value. Your program shouldn't allow the user to enter in anything on the C text field and should display error messages if either the value for A and B are blank or not numbers.



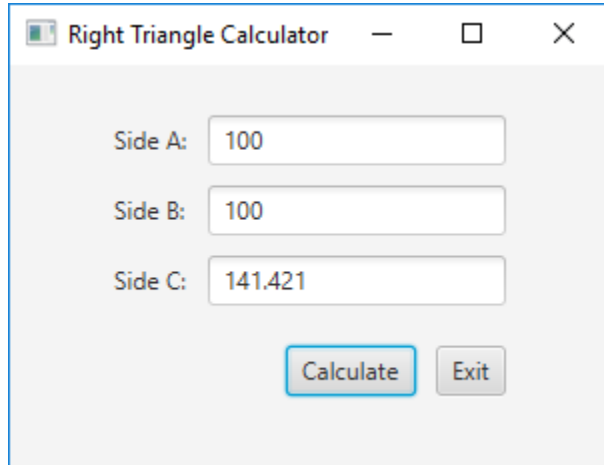
Directions

To Get Started:

1. Create a new maven JavaFX project in IntelliJ(or your preferred IDE) called YOUR_MCC_Name-Assignment3.
2. Under the java->projectname folder create a new java class called **PythagGUI.java** this will hold the code for that assignment. You can delete the other files.

Part 1: Create the GUI

Create a GUI that has 3 labels and 3 matching text fields. Add two buttons at the bottom, one that calculates and another that will exit. Use all the different properties for padding and spacing adjusted to your liking. The text field for Side C shouldn't allow the user to enter in a value but is only for displaying the calculated results. Make sure that there is a title as well.



Part 2: Perform the Calculations

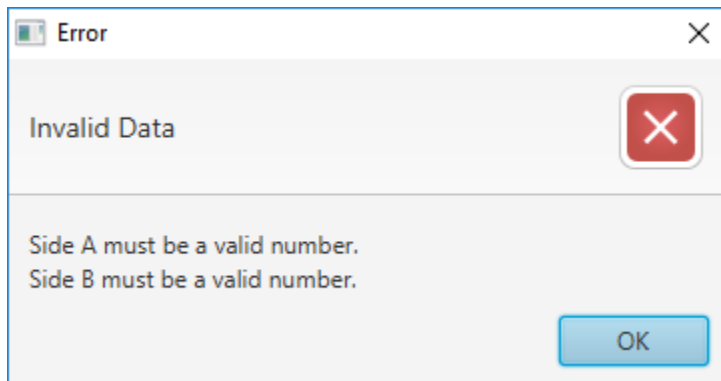
Now code event handlers that call methods for when the calculate and exit button are clicked. When the calculate button is pressed you should get the text values from field A and B, convert to numbers and calculate for C. (Remember the Math class has our math operations like `.pow()` and `.sqrt()`). Then put the answer into the text field. Make sure to also round to 2 decimal places. (Part of the grade!!) Also note that the `setText()` method only takes String value so you can Use a `NumberFormat` object that returns a String value already or `Integer.toString(intValue)`

For the exit button, it should simply exit with code 0.

Check your calculations work correctly, that is part of the grade!! Simple check would be use $a = 3$ and $b = 4$ and you should get 5 for c.

Part 3: Validate the Data

Now validate the data that was entered. We want to first make sure something is entered but also make sure it is number values. For this you can use the Validation class that is shown in Chapter 12 to validate the input. Or you can hardcode some methods on your own if you would like. You should though show one or more messages depending of text field A and/or B are blank/no numbers.



Submission

1. Compress the IntelliJ project folder and submit it to this assignment.

to compress: On Windows, right click -> send to -> compressed .zip file

on Mac, right-click -> Compress

Hints/Tips (Before Submitting):

- Check that your calculations are correct, yes that is part of the grade
- Round the answer to 2 decimals, use a `NumberFormatter` or `Math.round()`
- Make note of the `setText()` in a text field only takes `String` values - `NumberFormat` does return a `String` OR use can use `Integer.toString(intValue)`
- Don't forget to have a header at the top of your file and include a **Resource statement**.
- Use comments throughout for full points.
- Follow all Java Styling Guides as covered