

Project 3

Objectives

The objective of this project is to learn how create a class on your own and how to test code with the unit testing framework `JUnit`. You will also learn how to use `if-else` statements, object composition, and methods in `Math` class

Background

The project contains one class – `Project3Test`.

The `Project3Test` class represents a test suite for a class `Triangle.java` that you will be writing.

The `Project3Test` class is a test suite that is used to test some aspects of the `Triangle` class. It does this by creating 3 Triangle objects and printing the results of calling some of their methods on these 3 objects.

Objectives

This project provides you with the opportunity to put together nearly all of the material you have learned so far to create a single program. Most of what you are asked to do is very similar to what you did in the labs and in-class work, so look back at those labs and class examples for ideas.

You will be writing code for one class - `Triangle`.

Procedure

In this programming project, you will write a simple program to do the following.

1. Implement a class `Triangle` that represents a triangle in two-dimensional space. Triangle is represented by its 3 corner points. The class has the following instance variables
 - `p1, p2, p3` - these are Point objects. To use these, you need to import `java.awt.Point`
 - `angle1, angle2, angle3` - these represent the three corner angles - these are of type `double`

In class we discussed that it is a bad practice to keep derived attributes as instance variables. (such as *volume* in the `Balloon` class, and *age* in the `Student` class). Since the angle calculations of a triangle are tedious, we are keeping the 3 angles as instance variables even though these are derived attributes (3 corner angles can be computed given the 3 corner points). We are trading off memory here for time. Programming always involves trade-offs - do you want to use more memory or do you want to spend more time? In this case, we decided to use more memory (24 bytes for 3 doubles - for the three angles of the `Triangle` object, instead of time - computing the 3 angles each time takes time). Trade-offs could also be complexity of code versus and the possible error of not updating the angles that storing the angles every time the `Triangle` object is mutated. Keeping derived attributes as instance variables always introduces the possibility of having objects in an inconsistent state.

The class has the following methods (14 `public` methods and one `private` method):

1. The constructor takes 3 corner points (each of them type `Point`).
2. `getSide1_Length` method takes no parameter; computes and returns the distance between points p1 and p2
3. `getSide2_Length` method takes no parameter; computes and returns the distance between points p2 and p3
4. `getSide3_Length` method takes no parameter; computes and returns the distance between points p1 and p3
5. 3 mutator methods `setPoint1`, `setPoint2`, `setPoint3` - these takes a parameter of type `Point` and sets the appropriate corner point
6. 3 accessor methods for each of the 3 corner points:
`getPoint1`, `getPoint2`, `getPoint3`
7. 3 accessor methods for each of the 3 angles: `getAngle1`, `getAngle2`, `getAngle3`
8. a *private* method `computeAngles` that computes the three corner angles and assigns them to `angle1`, `angle2` and `angle3` respectively. Please visit [Computing corner angles](#) to learn how to compute the corner angles. You will be using

`Math.toRadians()`, `Math.toDegrees()`, `Math.sin()`, `Math.asin()`, `Math.acos()` methods from the `Math` class. You must read and understand the API documentation for the `Math` class to do this project. Figure out what `Math.acos` and `Math.asin` methods take as input and what they return. Similarly, figure out what `Math.sin` method takes and what it returns. This method should be called to compute and set the three angles - `angle1`, `angle2` and `angle3`

9. `toString` method that prints the `Triangle` object on the screen. Please visit [String formatter](#) to learn how to format a decimal value to a `String`. Study the test suite to find out how the `toString` method should work.

2. A program `Project3Test` is already provided for you. To use this code in your Bluej package, do the following:

1. Create a folder `Project3Code` within `Project3` folder
2. Create empty `Triangle` class.
3. Create empty `TriangleTest` class by doing the following steps. To turn on the testing tools in BlueJ:
 1. Open `Preferences...`
 2. Click on the `Interfaces` tab
 3. Check the `Show Unit Testing Tools` check-box
 4. Right-click on the box for the `Triangle` class you just created, and choose `Create Test Class`. A new box will appear (colored green) called `TriangleTest`, and with the annotation `<<unit test>>`.
 5. Double-click the `TriangleTest` class to view the source code.
 6. The `TriangleTest` class imports some code from `org.junit`, and sets up three methods – a constructor, a `setUp` method and a `tearDown` method.
 7. Replace the entire `TriangleTest.java` with the contents of `Project3Test.java` file.
4. Compile your code and run the tests. This should pass because you have an empty class and an empty test class.
5. You should get in the habit of writing and testing a single method before moving on to the next.
6. Write your constructor for the `Triangle` class, and then test it by uncommenting the `setup` method in `TriangleTest`, compiling both classes, and running the tests. If your test passes, move on to the next method. If your test fails, make corrections to your constructor until the test passes.
7. Continue this process for each of the methods in the `Triangle` and their associated tests:

Specifications

You need to submit your `Project3` folder with the subfolder `Project3Code` with `Triangle.java` and `TriangleTest.java` files. A log file which must address **what you learned in this project and the difficulties you faced**, time you spent on the project, web sources you referenced etc. **The log file must be a plain text file, be named `DiscussionLog.txt`, and must be at the top level of your repository.**

This is an Individual Assignment - No Partners

As this is a Project (and not a Lab) you will be working on your own, not with a partner. You should not be sharing your code with anyone else, other than the instructor.

You will need to fork your own private `Project3` repository on GitLab for this project. The only person who should have any access to your repository is your instructor.

You can ask questions on Piazza about setting up your repository on GitLab, about using Git to send code to the instructor, and general questions about how to write your code. However you should not be posting sections of code and asking others to find your errors.

You can cut down your coding time by a significant percentage by:

- understanding the project specifications mentioned in this file
- tracing the `Project3Test.java` to understand each of the test methods
- going over the Triangle.html given in the `doc` folder
- figuring out what needs to be done before you start typing the code
- going over the APIs for `java.awt.Point` and the `Math` class
- having a clear understanding of how to find the 3 corner angles given the 3 corner points

Deliverables

Be sure that you have your name and an explanation of what your program does in the Javadoc comments. Be sure that you have indented consistently.

You have to submit a discussion log along with the project. Please refer to DiscussionLogGuidelines repository or Lab2 for more information.

The instructor will pull your Project3 from your GitLab repository to grade it. Make sure:

1. You have pushed all changes to your shared repository. (I can't access local changes on your computer.)
2. You have added your instructor as Master to your shared GitLab repository.

Due Date/Time

Your project deadline will be determined by your instructor

Copyright and License

©2017 CS 140 Instructors, Worcester State University



This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.