Challenge Details

Task Requirements

As part of a bigger project, you are required to implement a basic data model that will represent geometric shapes.
The team lead in charge of this project has defined interfaces and base classes that should be used and prepared a simple test code that will help you test your work.

Your task is to finish this implementation by adding support for circle, rectangle, and square shapes as well as a factory method for creating objects of these classes.

Technical Requirements

You need to add classes to support the required shapes.

Supported shapes are defined as:

- Circle - A circle is defined by a single parameter representing radius.
- Rectangle - A rectangle is defined by two parameters representing the size of each edge.
- Square - A square is defined by a single parameter representing the size of edges.

Besides implementing the classes, you also need to finish the implementation of a factory class called *ShapeFactory* by implementing the *CreateShape* method.

Technical Considerations

While implementing the solution it is important to pay attention to the technical considerations defined in this section.

- Use defined constant *PI* from *Constants* for calculating the area and the perimeter of a circle.
- The factory method (i.e. *CreateShape*) should throw the following exceptions:
    - *WrongParamCountException* - Thrown if a wrong number of parameters is sent to the *CreateShape* method for a given shape. Each shape should only accept exactly the parameter count it needs.
    - *UsupportedShapeException* - Thrown if an unsupported shape is requested from the *CreateShape* factory method.
- The above exceptions should not only be thrown but also defined.
- You should define your classes in their respective files.

IMPORTANT NOTES:

1. You must continue building on top of the preexisting source code and you should **NOT DELETE** any of the existing source code, files and folders **NOR** move or rename any of the existing files, classes, methods or variables. However, you’re free to add new source files to the project and extend existing classes with your own methods and variables.
2. Please follow the instructions in the existing source code. Not following the instructions and not using the preexisting source code as a base might result in automatic test failures.
3. Before you submit the challenge, please make sure the project was successfully built.

Prerequisites/Instructions

In order to work on this task, you need to follow the following prerequisites/instructions:

- openjdk 8 (Java 1.8) needs to be installed.
- maven 3.8.1 needs to be installed.
- You need to install git if you don't have it on your machine already.
- To build and run the project please refer to the README file inside the repository root.
- Once you are ready, please click on the "Begin Challenge" button, and you will get instructions on how to clone the repository and submit the solution.