Assignment 1c

COMP 2526 Object-Oriented Programming with Java

Due on Sunday October 1st at 11:59 PM

1 Purpose

In assignment 1c, you will practice basic programming and Java skills and explore the WHY and the HOW in Object Oriented Programming by designing a multi-class solution.

2 Description

You are to develop a program that prints out shapes instead of arithmetic tables using the parameters <type of shape><width><height>. Your program design should be a modification of A1b. For A1c, you must take advantage of Inheritance to improve the design. See Assignments 1a and 1b for further details.

3 Requirements

1. The base class or super class must be called Shape:
   (a) It must contain all the common data members and methods found in the classes that will extend it. Remember in an inheritance hierarchy we like to push common elements to the highest logical point in the hierarchy. A good side benefit is reduced code duplication.
   (b) The Shape class must have a single constructor with the following signature: protected Shape(final int width, final int height, final String type) where width is the width of the shape and height is the height of the shape and type is one of "r", "t", or "d".
2. Each subclass that extends the Shape class should be modified to reflect an OOP design. In particular, note that a Shape can be constructed and can be drawn. All Shapes are drawn from the grid they fill out, so all work on specific Shapes is done in the constructor.
3. Prepare an enumerated type (separate file) for the Shape types.
4. Your program must accommodate the following Shapes:
   (a) Rectangle, whose constructor accepts a width and a height, and which is displayed as a asterisk (*).
   (b) Triangle, whose constructor accepts a width and a height, but the height entered is ignored and instead the height is calculated as (width / 2 + 1). A Triangle is is displayed as an at sign (@).
   (c) Diamond, whose constructor accepts a width and a height, but the height is ignored and instead the height stored is equal to the width. A Diamond is displayed as a number sign (#).
5. Triangle and Diamond need width to be odd in value, i.e., 3, 5, 7, 9, etc. If a Triangle or Diamond constructor is passed a width argument that is not odd, throw the BadWidthException provided. Main will catch the exception and call the usage() method.
6. Complete and use the provided driver class, Main.java. In particular, in function main() you need to get the Shape type and then display it.
7. Validate your code with Checkstyle. We will continue to use the Eclipse Checkstyle plugin. Checkstyle is a great tool that helps us to write code that's consistent and easy to read.

4 Marking Guidelines

Submit your assignment to the correct Dropbox folder in D2L at or before 11:59 PM on Sunday October 1st 2017. Late assignments will not be accepted. Grades will be assigned as follows:

30%	Base (super) class
40%	Subclasses
10%	Main class including code completing in the main() method
20%	Comments and style (remember to use the Checkstyle plugin!)

Good luck, and have fun!