Lab 10 Extra Credit (**+10 points**)

There are several learning objectives to this assignment

- Creating Classes that use Inheritance
- Creating Interfaces and Abstract Classes
- Creating Classes that implement Interfaces and Abstract Classes
- Creating advanced algorithms

In this lab, you will draw arrow shapes with a tail and head that are potentially drawn several lines from the top line and offset from the left edge. This simulates a low end display that does not have any graphics capabilities and only allows text output.

You will need to do the following:

(1) Create BasicInterface
(2) Create ArrowInterface
(3) Create BasicShape class
(4) Create RightArrow and LeftArrow classes
(5) Create appropriate constructors, getters and setters
(6) Create ArrowDemo driver that incorporates the above reqts in the driver section

ArrowDemo is based on the following convention (xAdj, yAdj, tail, width)
The default constructor should be xAdj=0, yAdj=0, tail=5, width=5.

Basic Interface

1) Create an interface called BasicInterface based on the following UML. Ensure that each method has an appropriate description (javadoc)

BasicInterface
setXAdj (newXAdj:int): void getXAdj (): int setYAdj (newYAdj:int): void getYAdj (): int drawShapeOn (): void drawShape (): void

ArrowInterface

1) Create an interface called ArrowInterface that is based on the following UML. Ensure that each method has an appropriate description (javadoc)

ArrowInterface
+ setArrowDim(newTail:int, newWidth:int): void

BasicShape

1) Create an ***abstract class BasicShape*** that implements BasicInterface with two private instance integer variables called xAdj and yAdj. **Note: After dropping down to the appropriate line based on yAdj, drawShapeOn () invokes drawShape() based on xAdj**. ***drawShape is an abstract method***. Ensure that each method has an appropriate description (javadocs)

RightArrow (8pts)

1) Create a RightArrow class that extends BasicShape and implements ArrowInterface. RightArrow has two private instance integer variables tail and width. If width is an even number, you should increase by one so that it is odd. While not mandated, you should consider private methods such as drawTail() to draw the tail and drawSpaces() to fill in the necessary spaces due to xAdj, tail, or arrow head spaces. yAdj is accounted for in drawShapeOn().

LeftArrow (10pts)

1) Create a LeftArrow class that extends BasicShape and implements ArrowInterface. LeftArrow has two private instance integer variables tail and width. If width is an even number, you should increase by one so that it is odd. While not mandated, you should consider private methods such as drawTail() to draw the tail and drawSpaces() to fill in the necessary spaces due to xAdj, tail, or arrow head spaces. yAdj is accounted for in drawShapeOn().

ArrowDemo Driver (2pts)

1) ArrowDemo is based on the following convention (xAdj, yAdj, tail, width). NOTE: RightArrow (and LeftArrow) classes should have a default constructor and a second constructor that passes xAdj, yAdj, tail, and width.
    a. Create a default RightArrow object, right1
    b. Draw right1 object
    c. Adjust object in (a) to xAdj of 10
    d. Draw right1 object
    e. Adjust object in (a) to yAdj of 5
    f. Draw right1 object
    g. Simultaneously adjust object in (a) tail to 10 and width to 15
    h. Draw right1 object
    i. Create a 2nd RightArrow object (right2) with xAdj =5, yAdj =10, tail=15 and width=**14**
    j. Draw right2 object
    k. Create a LeftArrow object (left1) with xAdj =13, yAdj =0, tail=15 and width=15
    l. Draw left1 object
    m. Simultaneously adjust object in (k) tail to 10 and width to **10**
    n. Draw left1 object
    o. Adjust object in (k) to xAdj of 5
    p. Draw left1 object
    q. Adjust object in (k) to yAdj of 5
    r. Draw left1 object
    s. Create a default LeftArrow object, left2
    t. Draw left2 object

Submitting your work

For all labs you will need to provide a copy of all .java files. **No need to provide .class files. I cannot read these.** ***NOTE – For Replit, please update Main.java to another name such as TempProb.java, ProChall3.java, etc.*** In addition to your .java files, you will need to provide output files of your console. The name of the output file should match the class name and have the .txt extension such as TempProbOut.txt, ProChall3Output.txt. For GUIs such as JOptionPane, you will instead need to create screenshots. For Windows users, Snipping Tool is a great way to do this. Chromebook - Shift+Ctrl+Show

Windows. Mac OS users, you can see how to take screenshots using the following url - https://support.apple.com/en-us/HT201361.