Assignment - Circle CSIS-1400

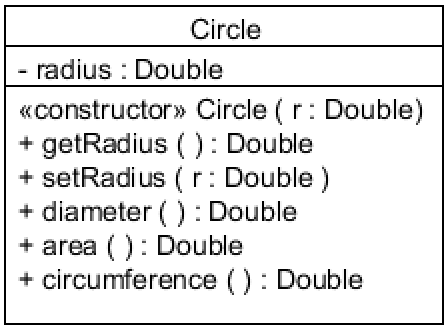
Learning Objectives:

- Initialize the newly created object with a constructor
- Use a setter (a.k.a. set method) to provide data validation
- Access the constant PI from the Java API
- Access class members (methods, fields, and constants) from another class

Description:

Create a project called **ACircle** with two java files called **Circle.java** and **CircleTest.java**
On top of each source code file include a comment with your name and the assignment number.

Here is the UML class diagram of class Circle. This specification is binding. That means your class members need to match the name, type, parameter list, and return type specified in the UML diagram, and no class members should be added or removed.



In Circle.java:

- create a public class called **Circle**.
- Include one private field of type double; its name is **radius**.
- Include a public constructor that takes one parameter.
  The purpose of the constructor is to initialize the field of the new Circle object.
  Call the setRadius method to assign the parameter value of the constructor to the field radius.
  By calling the setRadius method you can write the code for the input validation in one place (the set method) but use it both for the first initialization in the constructor as well as for later updates.
- Include five public methods: getRadius, setRadius, diameter, area, and circumference
  - **getRadius** .. the get method has the purpose of exposing the private field radius. It makes its value accessible outside of the class.
  - **setRadius** .. the set method provides a way to update (modify) the value of the private field radius.
    Here some input validation should be performed: only positive numbers should be permitted as radius.
    Check whether the parameter passed is a positive number. If that is the case assign the parameter value to the private field radius. Otherwise do not update the field (leave the field as it was)
  - **diameter**.. calculates and returns the value of the circle's diameter; it takes no parameter and returns a double

- **area** .. calculates and returns the area of the circle.
  In order to calculate the area of a circle you need the constant pi. Java provided this constant for you in the Math class. You can access the value of pi by calling Math.PI (notice that PI is spelled in uppercase letters. That is because pi is a constant and it is a naming convention in Java to spell constants with uppercase letters)
  The method area takes no parameter and returns a double
- **circumference**.. calculates and returns the circumference of the circle. It takes no parameter and returns a double.
  Use the method diameter to calculate the circumference.

Output:

```
Parameter: -2

radius = 0.0
diameter = 0.0
area = 0.0
circumference = 0.0
radius = 0.0
diameter = 0.0
area = 0.0
circumference = 0.0

Parameter: 2

radius = 2.0
diameter = 4.0
area = 12.6
circumference = 12.6
radius = 4.0
diameter = 8.0
area = 50.3
circumference = 25.1

Parameter: 6

radius = 6.0
diameter = 12.0
area = 113.1
circumference = 37.7
radius = 12.0
diameter = 24.0
area = 452.4
circumference = 75.4
```

In CircleTest.java

Create 2 methods: `circleTest` and `main`

Ad circleTest:

CircleTest is a private static method that returns no value and has one parameter of type int. The method header looks like this:

```
private static void circleTest (int r)
```

It includes code to test the Circle class for a given radius.

- Create a Circle based on the parameter value (i.e. the radius has the same value as the parameter **r**)
- Use 4 printf statements to print out the radius, diameter, area ,and circumference of the circle. Each print statement should be labeled and displayed in a separate line. (see output)
  Caveat: Make sure to access the field radius rather than printing the value of parameter **r**
  Display the results rounded to one decimal place (see output).
  Hint : if a method returns a value (i.e. return type is not void) you can print that value by calling the method directly as an argument of print, println, or printf e.g. System.out.printf("area: %.1f", myCircle.area());
- Modify the value of the field radius by doubling its size
  Once again, make sure to double the current radius value rather than re-assigning the double of the parameter
- Once again use 4 printf statements to print out the radius, diameter, area ,and circumference of the updated (modified) circle as described above.

Ad main:

In the main method call circleTest three times.
Hard-code the following parameter values: -2, 2, and 6.
Each time display the parameter value passed to make the output more meaningful to the use. Use empty lines to group the output. (see output)

Turning in:

Zip up your project and name the file **ACircle.zip** Turn it in via Canvas.