COMP2396B Object-oriented Programming and Java

Assignment 1: Class and Object

Due date: 7th March 2021 23:59

Notes about submitting this assignment

- Please submit your solution to Moodle.
- Please double check your submission. Please check the assignment page again after submission to ensure all essential files are submitted.
- Program comment is not necessary for this assignment.
- The main programs shown on this sheet are only for your reference to develop the classes. We will use hidden test cases to test your program outputs against standard outputs.
- Please contact us as soon as possible if you encounter any problem regarding this assignment.

Question 1 [60%]

You are required to build a Circle class to define the blueprint of circles. The primitive properties about a certain circle include x, y, and r, which define the x, y position (coordinate) of the center of the circle and r defines its radius.

A Circle class is defined as follows

```
public class Circle {
	private double x;
	private double y;
	private double r;

	public Circle(double x, double y, double r) {
	}

	public void setPosition(double x, double y) {
	}

	public void setRadius(double r) {
	}

	public double getPositionX() {
	}

	public double getPositionY() {
	}

	public double getRadius() {
	}

	public double getArea() {
	}

	public boolean isCollidedWith(Circle c) {
	}
}
```

You need to implement 8 methods for the Circle class, namely Circle(), setPosition(), setRadius(), getPositionX(), getPositionY(), getRadius(), getArea() and isCollidedWith(). You are NOT allowed to make change to the parameters list and return type.

Part 1: Implement the constructor and get methods

1. Circle() is the constructor of the class. This function is called when creating a Circle object.
   - Parameters:
     - x – (double): define the position x of the new circle object.
     - y – (double): define the position y of the new circle object.
     - r – (double): define the radius of the new circle object.

2. getPositionX() returns the x position of this circle object.
   - Return (double): return the x position of this circle.

3. getPositionY() returns the y position of this circle object.
   - Return (double): return the y position of this circle.

4. getRadius() returns the radius of this circle object.
   - Return (double): return the radius of this circle.

5. getArea() returns the area of this circle. Area (A) is defined as $A = \pi \times r^2$ where r is the radius. You may make use of `Math.PI` in Java to get the value of π.
   - Return (double): return the area of this circle.

Sample main()	

```
public static void main(String[] args) {
	Circle c1 = new Circle(10, 10, 3);

	System.out.println("Reporting c1:");
	System.out.println("x: " + c1.getPositionX());
	System.out.println("y: " + c1.getPositionY());
	System.out.println("r: " + c1.getRadius());
	System.out.println("Area: " + c1.getArea());
}
```

Sample output	

```
Reporting c1:
x: 10.0
y: 10.0
r: 3.0
Area: 28.274333882308138
```

Part 2: Implement set methods

6. setPosition() sets the new position for this circle. Both the x and y position should be specified when setting a new position.
   - Parameters:
     - x – (double): define the new position x for this circle object.
     - y – (double): define the new position y for this circle object.

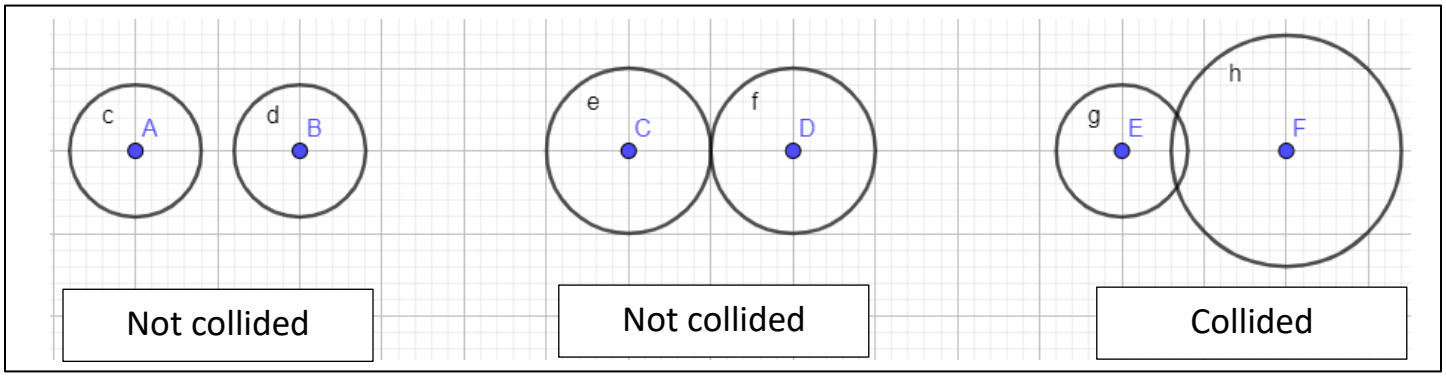
7. setRadius() sets the new radius for this circle.
   - Parameter
     - r – (double): define the new value of radius for this circle object

Sample main()	`public static void main(String[] args) {`
	`  Circle c1 = new Circle(10, 10, 3);`
	
	`  System.out.println("Setting new position and radius for c1...\n");`
	`  c1.setPosition(20, 20);`
	`  c1.setRadius(10);`
	
	`  System.out.println("Reporting c1:");`
	`  System.out.println("x: " + c1.getPositionX());`
	`  System.out.println("y: " + c1.getPositionY());`
	`  System.out.println("r: " + c1.getRadius());`
	`  System.out.println("Area: " + c1.getArea());`
	`}`

Sample output	`Setting new position and radius for c1...`
	
	`Reporting c1:`
	`x: 20.0`
	`y: 20.0`
	`r: 10.0`
	`Area: 314.1592653589793`

Part 3: Implement the advanced function

8. isCollidedWith() checks whether this circle object is collided/overlapped with another circle object
   - Parameter
     - c – (Circle): define the circle object we want to check whether it collides with this circle.
   - Return (boolean): return true if it is collided, false otherwise.



Sample main()	`public static void main(String[] args) {`
	`  System.out.println("Creating c1 and c2...\n");`
	
	`  // Creates Circle objects`
	`  Circle c1 = new Circle(20, 20, 10);`
	`  Circle c2 = new Circle(30, 10, 3);`
	
	`  // Test the advanced function - isCollidedWith()`
	`  if (c1.isCollidedWith(c2) == true) {`
	`  System.out.println("c1 and c2 collide");`
	`  } else {`
	`  System.out.println("c1 and c2 do not collide");`
	`  }`
	`}`

Sample output	`Creating c1 and c2...`
	
	`c1 and c2 do not collide`

Submit the Circle class to Moodle and then evaluate.

Question 2 [40%]

Before proceeding to Question 2, you are advised to back up your program code for Question 1 or create a new project for Question 2. In this question, you are asked to wrap the position related properties (x and y) of the Circle class into a new class called Position.

The Position class is defined as follows

```
public class Position {
	private double x;
	private double y;

	public Position(double x, double y) {
	}

	public double getX() {
	}

	public double getY() {
	}
}
```

Part 1: Implement the constructor and get methods for the Position class

1. Position() is the constructor of the Position class.
   - Parameters:
     - x – (double): define the x value of the new position object.
     - y – (double): define the y value of the new position object.

2. getX() returns the x value of this position object.
   - Return (double): return the x value of this position.

3. getY() returns the y value of this position object.
   - Return (double): return the y value of this position.

Sample main()	`public static void main(String[] args) {`
	` System.out.println("Creating p...");`
	` Position p = new Position(50,60);`
	
	` System.out.println("Reporting p:");`
	` System.out.println(p.getX());`
	` System.out.println(p.getY());`
	`}`

Sample output	`Creating p...`
	`Reporting p:`
	`50.0`
	`60.0`

Part 2:

The Circle class is now no longer storing the primitive values of position x and y but a reference to a Position object

The new Circle class is defined as follows

```
public class Circle {
	private Position p;
	private double r;

	public Circle(Position p, double r) {
	}

	public void setPosition(Position p) {
	}

	public void setRadius(double r) {
	}

	public Position getPosition() {
	}

	public double getRadius() {
	}

	public double getArea() {
	}

	public boolean isCollidedWith(Circle c) {
	}
}
```

Remove getPositionX(), getPositionY() and modify setPosition(), getPosition() in the Circle class in Question 1 so as to work with the Position class in this Question 2. In addition, you may need to make change to other methods, includes getRadius(), getArea() and isCollidedWith(), so they can perform in the same way as mentioned in Question 1.

Sample main()

```
public static void main(String[] args) {
	Circle c1 = new Circle(new Position(10, 10), 5);
	Circle c2 = new Circle(new Position(20, 10), 5);

	System.out.println("Reporting c1:");
	System.out.println("r: " + c1.getRadius());
	System.out.println("Area:" + c1.getArea());

	c1.setPosition(new Position(18, 10));

	Position c1_p = c1.getPosition();
	System.out.println("c1 is now at " + c1_p.getX() + ", " + c1_p.getY());

	if (c1.isCollidedWith(c2) == true) {
		System.out.println("c1 and c2 collide");
	} else {
		System.out.println("c1 and c2 do not collide");
	}

}
```

Sample output	Reporting c1: r: 5.0 Area:78.53981633974483 c1 is now at 18.0, 10.0 c1 and c2 collide

Submit the Position class and the new Circle class to Moodle and then evaluate.

-END-