Lab/Tutorial 6: Polymorphism

Determine abstract classes and methods in the above class diagram. Then use the above diagram and the information below to implement classes: Point, Shape, Circle, Rectangle, ShapeArray.

**Point:**
- *public Point(int x, int y):* Constructor with two parameters that initializes a Point object.
- Getters
- *public String toString():* overrides method toString() to return a point object as a string. Example: it returns strings “(1,2)”, “(2,4)”, etc.

**Shape:**
- *public Shape(Point center):* constructor with one parameter to initialize a Shape object.
- *public double area():* returns area of a Shape object.

**Circle:**
- *public Circle(Point center, int radius):* Constructor to initialize a Circle object.
- Getter
- *public String toString():* overrides inherited toString() method to include the Circle Object’s radius in the returned string.

**Rectangle**
- *public Rectangle(Point center, int length, int width):* Constructor to initialize a Rectangle object.
- Getters
- *public String toString():* overrides inherited toString() method to include the Rectangle Object’s length, width in the returned string.
ShapeArray

- **public ShapeArray(int size):** Constructor that initializes a ShapeArray object.
- **public boolean addShape(Shape shape):** adds a Circle or Rectangle object to ShapeArray object, returns true if added or false otherwise.
- **public double averageArea():** returns the average area of Rectangle and Circle objects in a ShapeArray object.
- **public int circleCount():** returns the number of Circle objects in a ShapeArray object.
- **public int rectangleCount():** returns the number of Rectangle objects in ShapeArray object.

Write a class called ShapeArrayProgram with main method to test all the methods of ShapeArray