Write a Java program to create a base class Shape with methods draw() and calculateArea(). Create two subclasses Circle and Cylinder. Override the draw() method in each subclass to draw the respective shape. In addition, override the calculateArea() method in the Cylinder subclass to calculate and return the total surface area of the cylinder.

Sample Solution:

Java Code:

```
// Shape.java
// Define an abstract class named Shape
abstract class Shape {
  // Declare an abstract method draw that must be implemented by subclasses
  public abstract void draw();

  // Declare an abstract method calculateArea that must be implemented by s
  public abstract double calculateArea();
}
```

```
// Circle.java
// Define a class named Circle that extends Shape
class Circle extends Shape {
  // Declare a private double variable radius
  private double radius;

  // Constructor for Circle that takes a double radius as a parameter
  public Circle(double radius) {
    // Assign the parameter radius to the instance variable radius
    this.radius = radius;
  }

  // Override the draw method from Shape class
  @Override
  public void draw() {
    // Print "Drawing a circle" to the console
    System.out.println("Drawing a circle");
  }

  // Override the calculateArea method from Shape class
  @Override
```

```
  public double calculateArea() {
    // Calculate and return the area of the circle
    return Math.PI * radius * radius;
  }

  // Protected method to get the radius of the circle
  protected double getRadius() {
    // Return the radius
    return radius;
  }
}
```

```
// Cylinder.java
// Define a class named Cylinder that extends Circle
class Cylinder extends Circle {
  // Declare a private double variable height
  private double height;

  // Constructor for Cylinder that takes a double radius and a double heigh
  public Cylinder(double radius, double height) {
    // Call the superclass (Circle) constructor with the radius parameter
    super(radius);
    // Assign the parameter height to the instance variable height
    this.height = height;
  }

  // Override the draw method from Circle class
  @Override
  public void draw() {
    // Print "Drawing a cylinder" to the console
    System.out.println("Drawing a cylinder");
  }

  // Override the calculateArea method from Circle class
  @Override
  public double calculateArea() {
    // Calculate the area of the circular base using the superclass method
    double circleArea = super.calculateArea();
    // Calculate the side area of the cylinder
    double sideArea = 2 * Math.PI * getRadius() * height;
```

```
        // Return the total surface area of the cylinder (2 circles + side area
        return 2 * circleArea + sideArea;
    }
}
```

```
// Main.java
// Define the Main class
public class Main {
    // Main method, entry point of the application
    public static void main(String[] args) {
        // Create a Shape reference pointing to a Circle object with radius 7.0
        Shape circle = new Circle(7.0);
        // Create a Shape reference pointing to a Cylinder object with radius 4
        Shape cylinder = new Cylinder(4.0, 9.0);

        // Call the drawShapeAndCalculateArea method with the circle object
        drawShapeAndCalculateArea(circle);
        // Call the drawShapeAndCalculateArea method with the cylinder object
        drawShapeAndCalculateArea(cylinder);
    }

    // Static method to draw the shape and calculate its area
    public static void drawShapeAndCalculateArea(Shape shape) {
        // Call the draw method of the shape object
        shape.draw();
        // Call the calculateArea method of the shape object and store the resu
        double area = shape.calculateArea();
        // Print the area of the shape to the console
        System.out.println("Area: " + area);
    }
}
```

Output:

```
Drawing a circle
Area: 153.93804002589985
Drawing a cylinder
Area: 326.7256359733385
```

Explanation: