```java
/* Реализовать 3 класса геометрических фигур: круг, квадрат и прямоугольник.
Круг определяется радиусом, квадрат и прямоугольник своими сторонами.
В каждом классе должен быть метод расчета площади фигуры.
    1. В main создать List из нескольких разных фигур и посчитать суммарную площадь всех фигур в List
    2. Написать метод, который находит фигуру с самой большой площадью
    3. Написать метод, который формирует List из фигур, с площадью больше, чем заданное значение
*/

import java.util.ArrayList;
import java.util.List;

public class Main {
    public static void main(String[] args) {

        List<Shape> shapes = new ArrayList<>();

        shapes.add(new Circle(10));
        shapes.add(new Circle(7.5));
        shapes.add(new Square(10));
        shapes.add(new Square(1.5));
        shapes.add(new Rectangle(5, 7));
        shapes.add(new Rectangle(8, 3));

       /* for (int i = 0; i < shapes.size(); i++) {
            System.out.println(shapes.get(i));
        }*/

        for (Shape shape: shapes) {  // частный случай обычного цикла для перебора объектов
                                     // если не нужен индекс
            System.out.println(shape);
            // System.out.println(shape.calcArea());
        }

        Shape lergestShape = getLergestShape(shapes);
        System.out.println("The largest shape is " + lergestShape + "with area: " + lergestShape.calcArea(
        System.out.println("----------------------------");
        System.out.println(getShapesLargeThat(shapes, 90));

    }

    public static Shape getLergestShape(List<Shape> shapes) {
        if (shapes == null || shapes.isEmpty()) {
            return null;
        }
```

```
45              Shape largestShape = shapes.get(0);
46              for (Shape shape: shapes) {
47                  if (shape.calcArea() > largestShape.calcArea()){
48                      largestShape = shape;
49                  }
50
51              }
52              return largestShape;
53          }
54
55          public static List<Shape> getShapesLargeThat ( List<Shape> shapes, double area) {
56              List<Shape> result = new ArrayList<>();
57              for (Shape shape: shapes) {
58                  if (shape.calcArea() > area) {
59                      result.add(shape);
60                  }
61              }
62              return result;
63          }
64      }
```