

```
public class Calculation implements Runnable {
    public double radius;
    public double side;
    public double area;
    public double phi = 3.14;

    @Override
    public void run() {
        System.out.println();
        System.out.println("=====Program will start in =====");
        for (int i = 3; i > 0; i--) {
            try {
                System.out.println("Starting with thread " + i);
                Thread.sleep(1000);
            } catch (Exception e) {
                System.out.println(e.getMessage());
            }
        }
    }

    public void setSquare(double side) throws IllegalArgumentException {
        if (side < 1) {
            throw new IllegalArgumentException("illegal exception: side value must be greater than or equal to");
        }
        this.side = side;
        this.area = side * side;
    }

    public double getSquare() {
        return this.area;
    }
}
```

```
public class Calculation implements Runnable {
    public double radius;
    public double side;
    public double area;
    public double phi = 3.14;

    @Override
    public void run() {
        System.out.println();
        System.out.println("=====Program will start in =====");
        for (int i = 3; i > 0; i--) {
            try {
                System.out.println("Starting with thread " + i);
                Thread.sleep(1000);
            } catch (Exception e) {
                System.out.println(e.getMessage());
            }
        }
    }

    public void setSquare(double side) throws IllegalArgumentException {
        if (side < 1) {
            throw new IllegalArgumentException("illegal exception: side value must be greater than or equal to");
        }
        this.side = side;
        this.area = side * side;
    }

    public double getSquare() {
        return this.area;
    }

    public void setCircle(double radius) throws IllegalArgumentException {
        if (radius < 1) {
            throw new IllegalArgumentException("illegal exception: radius value must be greater than or equal to");
        }
        this.radius = radius;
        this.area = phi * radius * radius;
    }

    public double getCircle() {
        return this.area;
    }

    public void setTrapezoid(double a, double b, double t) throws IllegalArgumentException {
        if (a < 1 || b < 1 || t < 1) {
            throw new IllegalArgumentException("illegal exception: side and height values must be greater than");
        }
        this.area = (a + b) * t / 2;
    }

    public double getTrapezoid() {
        return this.area;
    }
}
```

