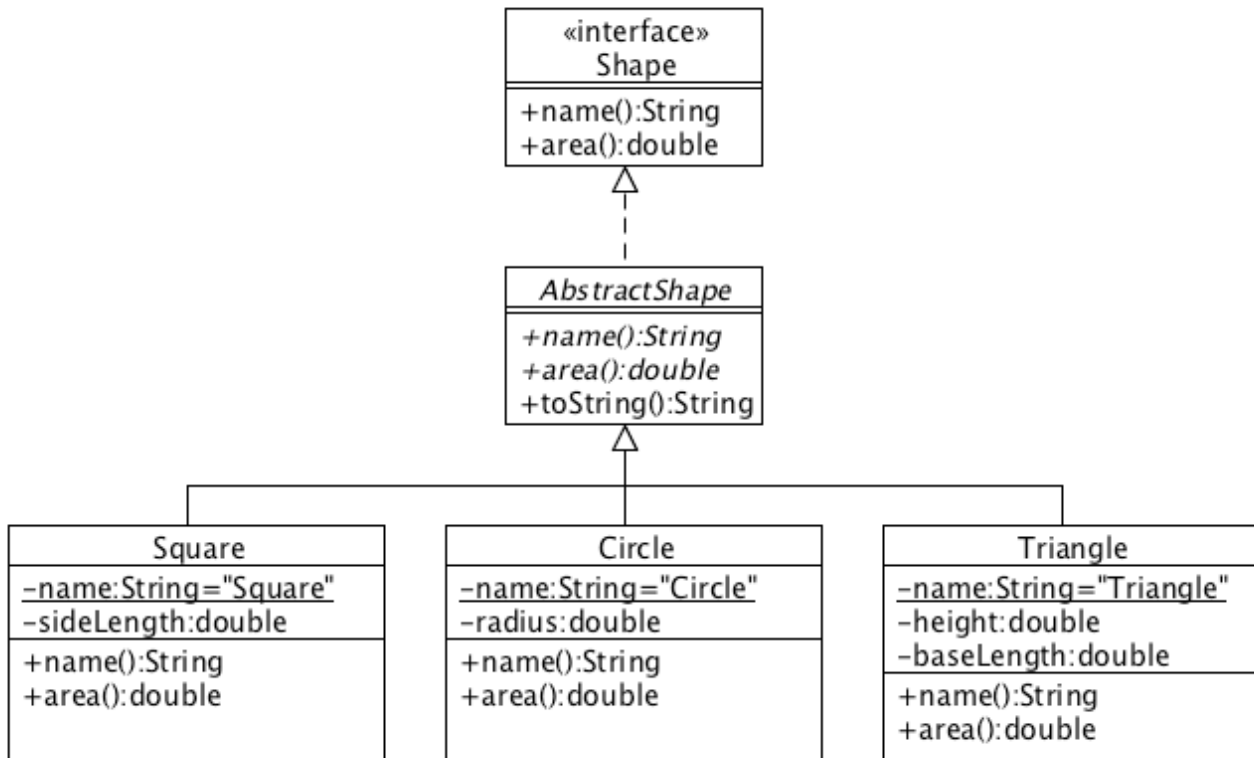


CIS 3022 – Lecture #40 – 201011.29



Translated to Java

```
interface Shape {
    public String name();
    public double area();
}
```

```
abstract class AbstractShape implements Shape {
    abstract public String name();
    abstract public double area();

    public String toString() {
        return "A " + name() + " with an area of " + area();
    }
}
```

<p>all methods declared, but <i>not</i> implemented <i>must</i> be declared abstract</p>

<p>because class AbstractShape implements Shape, for each Shape operation, the abstract class must either declare or implement it.</p>

```
class Square implements Shape extends AbstractShape {
    private static final String name = "Square";
    private double sideLength;

    public Square( double sideLength ) {
        this.sideLength = sideLength;
    }

    public String name() {
        return Square.name;
    }

    public double area() {
        return sideLength * sideLength;
    }

    public String toString() {
        return "A " + name() + " with an area of " + area();
    }
} // end class Square
```

```
class Circle implements Shape extends AbstractShape {
    private static final String name = "Circle";
    private double radius;

    public Circle( double radius ) {
        this.radius = radius;
    }

    public String name() {
        return Circle.name;
    }
}
```

shape

```
public double area() {
    return Math.PI * radius * radius;
}

public String toString() {
    return "A " + name() + " with an area of " + area();
}
} // end class Circle

class Triangle implements Shape extends AbstractShape {
    private static final String name = "Triangle";
    private double base;
    private double height;

    public Triangle( double base, double height )
        this.base = base;
        this.height = height;
    }

    public String name() {
        return Triangle.name;
    }

    public double area() {
        return 0.5 * base * height;
    }

public String toString() {
    return "A " + name() + " with an area of " + area();
}
} // end class Circle
```