Class in Java & Inheritance

Goals

- Practice about Java class with an example
- Begin coding to learn about composite classes
  - Objects inside other class
  - Methods with object arguments
  - Methods return object
- Learn about inheritance in Java with an example

Review

Inheritance

- **Definition:**
  - A class (called *subclass, derived class, extended class*, or *child class*) that is derived from another class (called *superclass, base class* or a *parent class*).

```
public class ClassName extends SuperClass {
    ...
}
```

- **What You Can Do in a Subclass**
  - A subclass inherits all of the *public* and *protected* members of its parent. (NOT vice versa)
  - You can declare a field in the subclass with the same name as the one in the superclass, thus *hiding* it (**Not recommended**)
  - The inherited methods can be used directly as they are.
  - You can write a new instance method in the subclass that has the same signature as the one in the superclass, thus overriding it.
  - You can write a subclass constructor that invokes the superclass's constructor, either implicitly or by using the keyword *super*.

```
super(parameter);  //call parent class 's constructor
super.parentMethodName(parameter);
super.parentFieldName;
```

- Java does **not support multiple inheritance,** but you can use Interface for implementing multiple inheritance. (next class)

Exercise

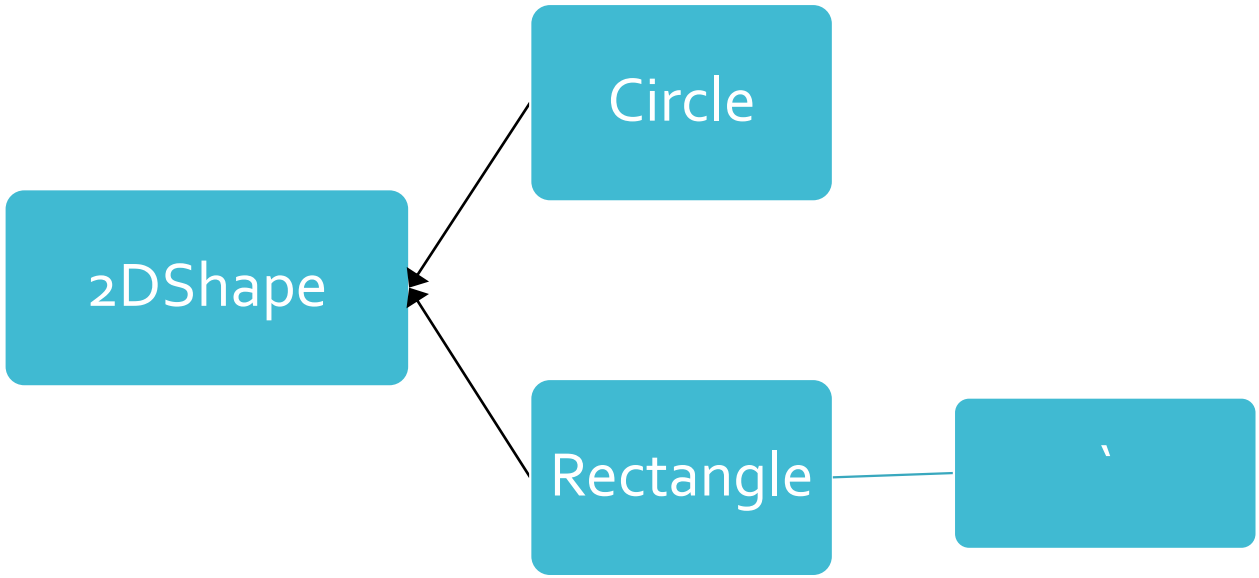
Exercise

Introduction

- Objective:
  - Use inheritance to create hierarchies of related classes
  - Extend behavior and override existing behavior
- To Do:
  - Create the hierarchies with classes: Circle, Rectangle, Square, TwoDimensionalShape.
  - Implement constructor for all class.
  - Implement method to return the String of the current class and direct super class.
  - Implement method to return the area of object.

Class hierarchy

- Design superclasses to store common characteristics
- Design the subclasses to store specialized characteristics

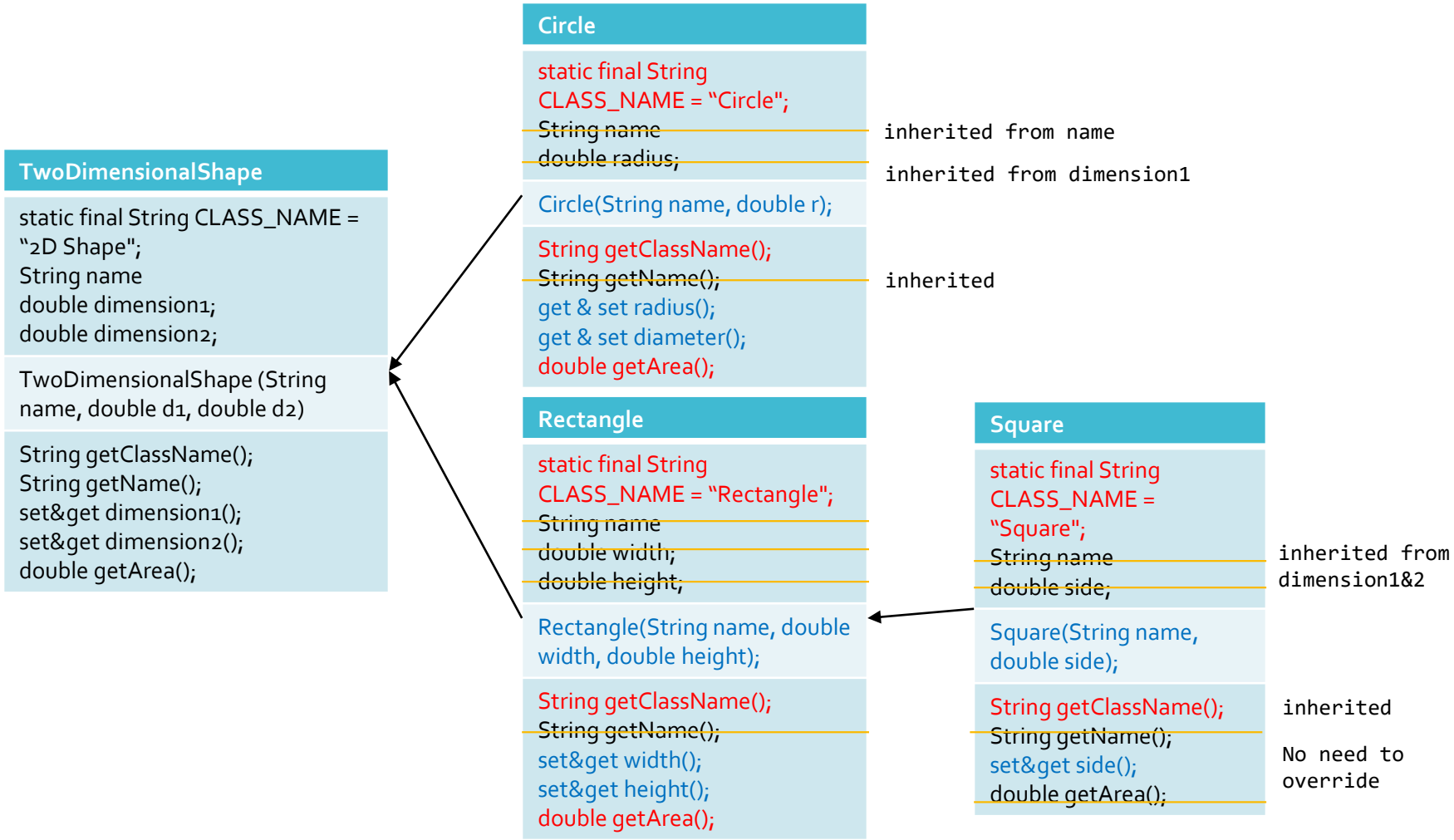


Exercise output

- Create objects and print the result like this:

```
Four shapes have been created:
1.Circle [Radius()=3.0]
Cir One's area is 28.27, radius is 3.00
2.Rectangle [Width()=3.0, Height()=4.0]
Rec One's area is 12.00, width is 3.00, height is is 4.00
3.Square [Side()=4.0]
Square Two's area is 16.00, side is 4.00
4.Square [Side()=4.0]
Square Two's area is 16.00

Is Rec One a TwoDimensionalShape? true
```

- Design superclasses to store common characteristics
- Design the subclasses to store specialized characteristics



TwoDimensionalShape Class

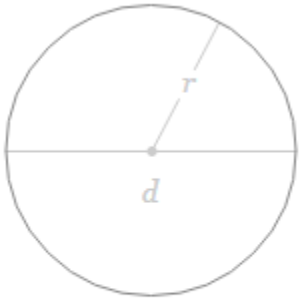
TwoDimensionalShape	Description
static final String CLASS_NAME = "2D Shape"; String name double dimension1; double dimension2;	*class fields*
TwoDimensionalShape (String name, double d1, double d2)	*initialize all class fields*
String getClassName() String getName() Get & Set dimension1() Get & Set dimension2(); double getArea();	*return the CLASS_NAME* *return the name of the class* *return 0, will be overrided by child class*

```java
/**
 * Create class of 2Dshape and can be subclassed.
 */
public class TwoDimensionalShape {
  private static String CLASS_NAME = "2D Shape";
  private String name;
  private double dimension1;
  private double dimension2;

  // constructor
  public TwoDimensionalShape(String name, double d1, double d2)
{
      this.name = name;
      this.dimension1 = d1;
      this.dimension2 = d2;
  }

  // get name and class name
  public String getClassName() {
      return CLASS_NAME;
  }

  public String getName() {
      return name;
  }

  // get & set methods for dimension 1
  public double getDimension1() {return this.dimension1; }
  public void setDimension1(double d) {this.dimension1=d;}

  // get & set methods for dimension 2
  public double getDimension2() {return this.dimension2;}
  public void setDimension2(double d) {}

  // don't know the kind of current shape
  // so return 0 only
  // must be implement in subclass
  public double getArea() {
      return 0;
  }
}
```

Circle Class

Circle	Description
static final String CLASS_NAME = "Circle";	*override the CLASS_NAME of super class*
Circle(String name, double radius);	*initialize all class fields* *The name is inherited from name of super class* *The radius is inherited from dimension1 of super class*

String getClassName();
get & set radius();
get & set diameter();
double getArea();
	return the CLASS_NAME fields of this class
	Call the super method to update the dimension1
	Calculation and call the super method to update the dimension1
	return the area of the circle



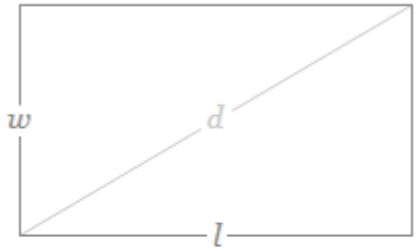
$$A = \pi r^2$$

```java
public class Circle extends TwoDimensionalShape {

   private static String CLASS_NAME = "Circle";


   public Circle(String name, double r) {

        super(name, r, r);

   }


   @Override

   // return current class name

   public String getClassName() {

        return CLASS_NAME;

   }


   public double getRadius() {

        return super.getDimension1();

   }


   public void setRadius(double r) {

        super.setDimension1(r);

        super.setDimension2(r);

   }


   @Override

   public double getArea() {

        return Math.PI * super.getDimension1() *
super.getDimension1();

   }


   @Override

   public String toString() {

        return String.format("%s is a [%s], and is a [%s]",

   super.getName(), getClassName(), super.getClassName());

   }

}
```

Rectangle Class

Rectangle	Description
static final String CLASS_NAME = "Rectangle";	*override the CLASS_NAME of super class*
Rectangle (String name, double width, double height);	*initialize all class fields* *The name is inherited from name of super class* *The width is inherited from dimension1 of super class* *The height is inherited from dimension2 of super class*

String getClassName();
double getWidth();
double getHeight()
setSize(double w, double h);
double getArea();
	return the CLASS_NAME fields of this class
	Call the super method to get the dimension1
	Call the super method to get the dimension2
	Call the super method to set the dimension1&dimension2
	return the area of the Rectangle



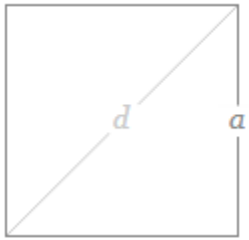
$$A = w\,l$$

- Similar with Circle class

```
public class Rectangle extends TwoDimensionalShape {
  private static String CLASS_NAME = "Rectangle";
  public Rectangle(String name, double width, double height) {
      // store width in field demension1, height in field
demension2
      super(name, width, height);
  }

  public String getClassName(){
      return CLASS_NAME;
  }

  public double getWidth() {
      return super.getDimension1();
  }

  public double getHeight() {
      return super.getDimension2();
  }

  public void setSize(double w, double h){
      super.setDimension1(w);
      super.setDimension2(h);
  }

  public double getArea() {
      return super.getDimension1() * super.getDimension2();
  }

  public String toString() {
      return String.format("%s is a [%s], and is a [%s]",
          super.getName(), getClassName(), super.getClassName());
  }
}
```

Square Class

Square	Description
static final String CLASS_NAME = "Square";	*override the CLASS_NAME of super class*
Square(String name, double side);	*initialize all class fields* *The name is inherited from name of (indirect) super class* *The side is inherited from dimension1 and dimension2*
String getClassName(); Get & Set side();	*return the CLASS_NAME fields of this class* *Call the super method to get & set both the dimension1 and the dimension2*



$$A = a^2$$

- Inheritance from Rectangle Class

```
public class Square extends Rectangle {
    private static String CLASS_NAME = "Square";
    public Square(String name, double side) {
        // this will call the constructor of Rectangle
        super(name, side, side);
    }

    public String getClassName(){
        return CLASS_NAME;
    }

    public double getSide() {
        return getWidth();
    }

    public void setSide(int side) {
        super.setSize(side, side);
    }

    public String toString() {
        return String.format("%s is a [%s], and is a [%s]",
    super.getName(), getClassName(), super.getClassName());    }
    }
}
```

Test Program

- Create these objects:
  - Circle cir1 = new Circle("Cir One", 3.0);
  - Rectangle rec1 = new Rectangle("Rec One", 3.0, 4.0);
  - Square sq1 = new Square("Square One", 6.0);
- Print the fields of the class
  - Circle: area, radius, diameter
  - Rectangle: area, width, height
  - Square : area, side

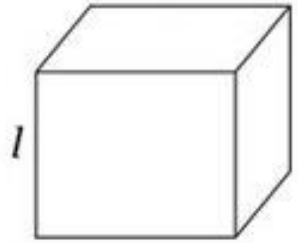
```
public class ShapeTester {

  public static void main(String[] args) {


      // create an object

      Circle cir1 = new Circle("Cir One", 3.0);

      Rectangle rec1 = new Rectangle("Rec One", 3.0, 4.0);

      Square sq1 = new Square("Square One", 6.0);


      TwoDimensionalShape sq2 = new Square("Square Two", 4.0);

      System.out.println("Four shapes have been created:");


      // print the object properties

      System.out.println("1." + cir1);

      System.out.printf( "%s's area is %.2f, radius is %.2f\n",

       cir1.getName(),cir1.getArea(), cir1.getRadius());


      System.out.println("2." + rec1);

      System.out.printf( "%s's area is %.2f, width is %.2f,
height is %.2f\n",

          rec1.getName(),rec1.getArea(), rec1.getWidth(),
rec1.getHeight());


      System.out.println("3." + sq1);

      System.out.printf( "%s's area is %.2f, side is %.2f\n",

          sq1.getName(), sq1.getArea(), sq1.getSide());


      // print all circle shape

      System.out.printf("Is %s a TwoDimensionalShape?
%s\n",sq1.getName(), sq1 instanceof TwoDimensionalShape);

      System.out.printf("Is %s a Rectangle? %s\n",sq1.getName(),
sq1 instanceof Rectangle);

      System.out.printf("Is %s a Square? %s\n", sq1.getName(),
sq1 instanceof Square);

  }

}
```

Self practice Exercise

CUBE

$A = 6l^2$

$V = l^3$



- Design a class **Cube**, **Cone** and **Sphere**. Then create the objects of each class and print to screen this information:
  - Object name, the class name and the parent class name
  - Surface Area value and Volume value (The formulas are below)

SPHERE

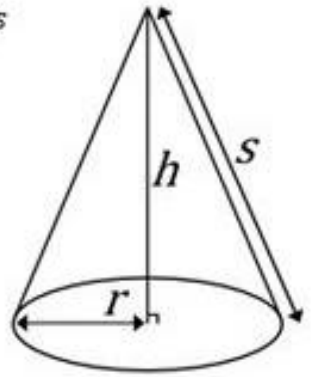
$S = 4\pi r^2$

$V = \frac{4\pi r^3}{3}$

RIGHT CIRCULAR CONE

$A = \pi r^2 + \pi rs$

$V = \frac{1}{3}\pi r^2 h$



Self practice Exercise

- Implement **Cube**, **Cone**, **Sphere** class as inherited class of the classes created today (2DShape, Circle, Rectangle, Square) to reduce the code and complexity.
- You can decide which class is the superclass (parentclass).
- You will override the **getArea()** method and implement new method called **getVolume()** for these classes **(Cube, Cone, Sphere)**

Possible classes designs