

Ex. 1. Super class Figure, subclass Circle

- a) Create the new **Lab5** project (*Java application*) and add new file to the project (*File/New file -> Java class*) entitled **Figure**.
- b) Complete the Figure class definition (**Listing 1**)
- c) Create the new Java class file entitled Circle and complete the code (Listing 2)
- d) In the main project file (Lab5.java) create two Figure and two Circle objects and show information about them. For example create figures named "Figure1", "Figure2" and circle 1 named "Circle1" with radius=10, and circle 2 named "Circle 2" with radius=123. Check what fields and methods you can use for figures and circles.
- e) Create the array of Figures and display information about figures from array, for example:

```
Figure f1=new Figure("Figure1");
Figure f2=new Figure("Figure2");
Circle c1=new Circle("Circle1",10);
Circle c2=new Circle("Circle2",123);
Figure fig[]={f1,f2,c1,c2}; //array of 4 figures
for(int i=0;i<fig.length;i++)
{
    fig[i].show();
}
```

Listing 1.

```
//The super class Figure definition:
public class Figure {
    //protected field - can be accessed also in the subclass:
    protected String name="Figure";
    //constructor with no parameters:
    public Figure() { //name will be still equal "Figure"
    }
    //constructor with one parameter:
    public Figure(String n) { //name will be equal n:
        this.name=n;
    }
    //get/set methods:
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
    //public show method
    public void show(){
        System.out.println("Figure entitled:"+name);
    }
}
```

Listing 2.

```
//The subclass Circle definition:
public class Circle extends Figure{
    //the new field - just for Circle:
    protected double radius=1;

    //constructors - just for Circle:
    public Circle() { //constructor with no parameter
    }
    public Circle(String n) { //constructor with one String parameter
        super(n); //You can use the super class Figure constructor
    }
}
```

```

    public Circle(String n, double r) { //constructor with two parameters
        super(n); //You can use the super class Figure constructor
        this.radius=r;
    }

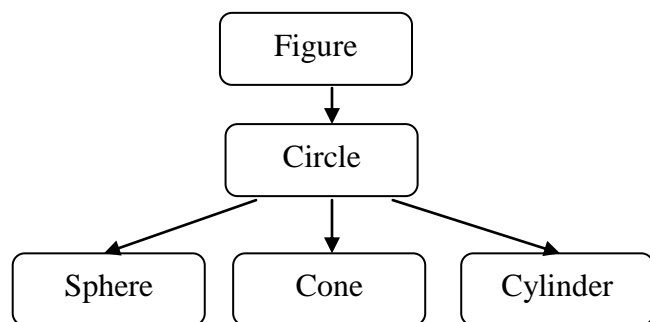
    //the new methods - just for Circle:
    public double getRadius() {
        return radius;
    }
    public void setRadius(double radius) {
        this.radius = radius;
    }
    public double calculateArea(){
        return Math.PI*radius*radius;
    }

    //overridden method show
    @override
    public void show(){
        super.show(); //You can use the super class method in the subclass method
        System.out.println(" radius:"+radius);
    }
}

```

Ex. 2. Subclasses Sphere, Cone and Cylinder value of

- Create the new java class named Sphere, extended the Circle class. In the class define 2 constructors, overridden show and calculateArea methods and new calculateVolume method.
- Create two object of Sphere class and test them in the main method (Lab5.java file).
- Create the Cone and Cylinder classes with the additional height field, constructors and analogical methods as in Sphere class.
- Test the Cone and Cylinder classes.
- Create the array of all figures and solids, display information about them.
- Choose the solids with the maximum volume.



Forms for the surface area of the solid can be found at:

<https://www.mathopenref.com/surface-area.html>

Forms for the volume of the solid can be found at::

<https://www.mathopenref.com/volume.html>