

## EJERCICIO 3.- Solución

- Clase **Cuerpo** (ABSTRACTA)

```
1 public abstract class Cuerpo{
2     protected float superficie, volumen;
3     protected Color col;
4
5
6     public String toString(){
7         return ("Superficie: "+superficie+"\nVolumen: "+volumen+ " - Color: "+ col);
8     }
9
10    //métodos abstractos.
11    public abstract void calcularSuperficie();
12    public abstract void calcularVolumen();
13
14 }
```

### Enumeración **Color**

```
1 public enum Color{
2     ROJO, VERDE, AZUL, ROSA, BLANCO, AMARILLO;
3 }
```

- Clase **Esfera** (hereda de **Cuerpo**)

```
1 public class Esfera extends Cuerpo {
2     private float radio;
3
4     Esfera (float radio, Color col){
5         this.radio=radio;
6         this.col=col;
7     }
8
9     public float getRadio() {
10         return radio;
11     }
12     public void setRadio(float radio) {
13         this.radio = radio;
14     }
15
16     public void calcularSuperficie(){
17         //almaceno superficie en la variable de la superclase
18         superficie = (float) (Math.PI * (4*radio * radio));
19     }
20
21     public void calcularVolumen(){
22         //almaceno volumen en la variable de la superclase
23         volumen = (float) ((4 * Math.PI) * radio*radio*radio)/3; // Math.pow (radio,3)
24     }
25
26     public String toString(){
27         return ("Radio: "+radio+"\n"+super.toString());
28     }
29 }
```

- Clase **Prisma** (hereda de **Cuerpo**)

```
1 public class Prisma extends Cuerpo {
2     private float ladoBase, altura;
3
4     Prisma(float ladoBase, float altura, Color col){
5         this.ladoBase=ladoBase;
6         this.altura=altura;
7         this.col=col;
8     }
9
10    public float getAltura() {
11        return altura;
12    }
13    public void setAltura(float altura) {
14        this.altura = altura;
15    }
16    public float getLadoBase(){
17        return ladoBase;
18    }
19    public void setLadoBase(float ladoBase){
20        this.ladoBase=ladoBase;
21    }
22
23    public void calcularSuperficie(){
24        superficie = 4*ladoBase*altura+ 2*ladoBase*ladoBase;
25    }
26
27    public void calcularVolumen(){
28        volumen = ladoBase*ladoBase*altura;
29    }
30
31    public String toString(){
32        return ("Lado: "+ladoBase+" - Altura: "+altura+"\n"+super.toString());
33    }
34 }
```

- Clase **Cilindro** (hereda de **Cuerpo**)

```
1 public class Cilindro extends Cuerpo {
2     private float radioBase, altura;
3
4     Cilindro(float radioBase, float altura, Color col){
5         this.radioBase=radioBase;
6         this.altura=altura;
7         this.col=col;
8     }
9
10    public float getAltura() {
11        return altura;
12    }
13
14    public void calcularSuperficie(){
15        superficie = (float) Math.PI* 2*radioBase* (radioBase + altura);
16    }
17
18    public void calcularVolumen(){
19        volumen = (float) Math.PI* radioBase*radioBase*altura;
20    }
21
22    public String toString(){
23        return ("RadioBase: "+radioBase+ " - Altura: " + altura+ "\n"+super.toString());
24    }
25 }
```

- Clase con función principal donde se declaren varios objetos de las diferentes clases, calculando luego para ellos su volumen y superficie, para posteriormente ver todos los datos de los objetos creados a través del método sobrescrito toString.

```
1 public class Ejercicio3{
2     public static void main(String[] args) {
3         Cilindro cill1 = new Cilindro(2.5f,3.3f, Color.ROJO);
4         cill1.calcularSuperficie();
5         cill1.calcularVolumen();
6         System.out.println("CILINDRO");
7         System.out.println(cill1.toString());
8
9         System.out.println();
10        System.out.println("ESFERA");
11        Esfera esf1 = new Esfera(5, Color.AMARILLO);
12        esf1.calcularSuperficie();
13        esf1.calcularVolumen();
14        System.out.println(esf1.toString());
15
16        System.out.println();
17        System.out.println("PRISMA");
18        Prisma pris1 = new Prisma(5,2, Color.AZUL);
19        pris1.calcularSuperficie();
20        pris1.calcularVolumen();
21        System.out.println(pris1.toString());
22    }
23 }
```

