

Super Classe **Object** –Sotto Classe **Cerchio**

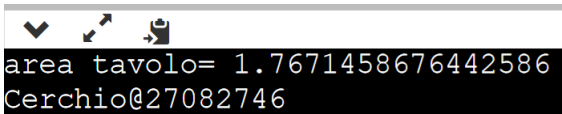
Esempio

```
class ProgCerchio {  
    public static void main(String[] args) {  
        Cerchio tavolo;  
        tavolo = new Cerchio();  
        tavolo.setRaggio (0.75);  
        System.out.println("area tavolo= " + tavolo.area() );  
        System.out.println(tavolo);  
    }  
}
```

ProgCerchio.java

```
class Cerchio {  
    private double raggio;  
  
    public void setRaggio (double r) {  
        raggio = r;  
    }  
    public double area () {  
        return (raggio * raggio * Math.PI);  
    }  
}
```

Cerchio.java



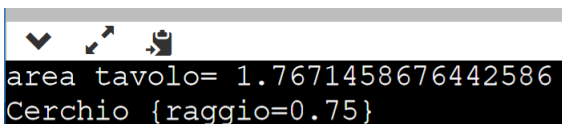
```
area tavolo= 1.7671458676442586  
Cerchio@27082746
```

Il metodo `println(tavolo)` espone quanto restituito dal metodo `toString()` della Super Classe **Object** , ovvero la stringa di rappresentazione dell' **oggetto tavolo** di Sotto Classe **Cerchio** (Cerchio@27082746).

Utilizzando invece il **polimorfismo** di tipo **overriding** (la classe *derivata* –Cerchio– può avere un metodo identico per numero e tipo dei parametri ad un metodo già presente nella classe *base* -Object) si può **riscrivere** il metodo `toString()` per la Sotto Classe **Cerchio**:

```
class Cerchio {  
    private double raggio;  
  
    public void setRaggio (double r) {  
        raggio = r;  
    }  
    public double area () {  
        return (raggio * raggio * Math.PI);  
    }  
    @Override //annotazione  
    public String toString() {  
        return getClass() + " {" +  
            "raggio=" + raggio + '}';  
    }  
}
```

Cerchio.java



```
area tavolo= 1.7671458676442586  
Cerchio {raggio=0.75}
```

← metodi
← della super classe Object