

```

class ProgCilindro_Cerchi {
    public static void main(String[] args) {
        Cilindro cil = new Cilindro(3.0, 10.5);
        System.out.print ("Primo cilindro ---\n");
        System.out.println("area base = " + cil.area("B")); //-(overloading)
        System.out.println("volume = " + cil.volume());
        System.out.println("circonfer. = " + cil.circonferenza ());
        //-- metodo ereditato
        System.out.println("superficie = " + cil.area());
        //-- metodo ridefinito (overriding)
        cil.setRaggio (1); //-- metodo ereditato
        cil.setAltezza(20.7);
        System.out.print ("\nSecondo cilindro ---\n");
        System.out.println("area base = " + cil.area("B")); //-(overloading)
        System.out.println("volume = " + cil.volume());
        System.out.println("circonfer. = " + cil.circonferenza ());
        System.out.println("superficie = " + cil.area());
    }
}

Cerchio cerchio[] = new Cerchio [3];
for (int i=0; i<3; i++) {
    cerchio [ i ] = new Cerchio ( i + 1 ); //-- di raggio i + 1
}
for (int i=0; i<3; i++) {
    System.out.println ("\n--- cerchio " + (i+1) + " ---");
    System.out.println ("area =" + cerchio [ i ]. area ());
    System.out.println ("perim.=" + cerchio [ i ]. circonferenza ());
}
}
}

```

Il programma alloca un oggetto di classe **Cilindro** (sottoclasse della **sopraclasse Cerchio**), ne espone l'area della base, il volume, la circonferenza del cerchio di base e la superficie totale del solido (somma delle area di tutte le facce).

Successivamente alloca un array di 3 oggetti di classe **Cerchio** esponendone le misure dell'area e della circonferenza.

La classe **Cilindro** eredita attributi e metodi della classe **Cerchio** :

- l'attributo **raggio** per la base (perchè definito **protected**)
- i metodi **circonferenza()** e **setRaggio(...)** definiti **public**

metodi che può richiamare come se fossero metodi della classe **Cilindro**.

Invece **ridefinisce (overriding)** il metodo **area()** e **arrotonda(..)**;

il metodo **arrotonda(..)** viene anche sovraccaricato (**overloading**) con un secondo parametro di input; anche il metodo **area()** è sovraccaricato (**overloading**) con un parametro in **area(String b)**.

I metodi **volume()** e **setAltezza(..)** sono specifici della **sottoclasse Cilindro**.

```

Primo cilindro ---
area base = 28.27
volume = 296.835
circonfer. = 18.85
superficie = 254.47

Secondo cilindro ---
area base = 3.14
volume = 64.998
circonfer. = 6.28
superficie = 136.28

--- cerchio 1 ---
area =3.14
perim.=6.28

--- cerchio 2 ---
area =12.57
perim.=12.57

--- cerchio 3 ---
area =28.27
perim.=18.85

```

**NOTA:** Solo la misura del volume viene arrotondata con 3 decimali

classe base <b>Cerchio</b> (superclasse)	classe derivata <b>Cilindro</b> (sottoclasse)
<pre> class <b>Cerchio</b> {     <b>protected</b> double <b>raggio</b>;      <b>public Cerchio</b> (double r) {         <b>raggio</b> = r;     }      <b>public void setRaggio (double r)</b> {         <b>raggio</b> = r;     }      <b>public double area ()</b> {         return <b>arrotonda</b> (raggio * raggio * <b>Math.PI</b>);     }      <b>public double circonferenza ()</b> {         return <b>arrotonda</b> (2 * raggio * <b>Math.PI</b>);     }      <b>protected</b> double <b>arrotonda (double x)</b> {         double temp;         temp = <b>x</b> * 100;         temp = <b>Math.round</b> (temp); <b>//--- intero</b>         temp = temp / 100;          <b>//--- double a 2 decimali</b>         return temp;     } } </pre>	<pre> class <b>Cilindro</b> <b>extends Cerchio</b> {     <b>protected</b> double <b>altezza</b>;      <b>public Cilindro</b> (double raggio, double altezza) {         <b>super (raggio)</b>; <b>//-- richiamo del costruttore Cerchio</b> (double r)         <b>this.altezza</b> = altezza;     }      <b>public void setAltezza (double altezza)</b> {         <b>this.altezza</b> = altezza;     }      <b>public double volume ()</b> {                                     <b>// pag 99</b>         double vol = <b>super.area()</b> * <b>altezza</b>;         return <b>arrotonda(vol, 3)</b>; <b>//---arrotondamento a 3 decimali</b>     }      <b>public double area ()</b> {                                       <b>// pag 102</b>         double supBase, supLaterale;         supBase = <b>super.area()</b> * 2 ;         supLaterale = <b>circonferenza()</b> * <b>altezza</b>;         return <b>arrotonda</b>(supBase + supLaterale);     }      <b>public double area (String b)</b> {         return <b>super.area()</b>; <b>//--- richiamo metodo area() ereditato</b>     }      <b>protected</b> double <b>arrotonda (double x, double p)</b> {         double temp;         double pot10 = <b>Math.pow</b> (10, <b>p</b>); <b>//potenza p di 10, es 10<sup>3</sup> = 1000</b>         temp = <b>x</b> * pot10;         temp = <b>Math.round</b> (temp); <b>//--- intero</b>         temp = temp / pot10;          <b>//--- double con p decimali</b>         return temp;     } } </pre>