

```
public class Main {
    public static void main(String Args[])
    {
        Counter c = new Counter();
        Counter c1 = new Counter2();
    }
}
```

```
public class Counter2
extends Counter {
    public void dec() { val--; }
}
```

```
public class Counter {
    protected int val;
    public Counter()
    { System.out.println("Counter : costruttore di default");
      val = 1; }
    public Counter(int v)
    {System.out.println("Counter : costruttore");
     val = v; }
    public void reset() { val = 0; }
    public void inc() { val++; }
    public int getValue() { return val; }
}
```

Ereditarietà - 1

Ereditarietà - 2

```
public class Main {
    public static void main(String Args[])
    {
        Counter c = new Counter();
        Counter c1 = new Counter2();
    }
}
```

```
public class Counter2 extends Counter {
    public Counter2()
    { System.out.println("Counter2 :
    costruttore di default!");
      val = 1; }
    public void dec() { val--; }
}
```

Nel costruttore di Counter2 ho una chiamata implicita a super

```
public class Counter {
    protected int val;
    public Counter()
    { System.out.println("Counter : costruttore di default");
      val = 1; }
    public Counter(int v)
    {System.out.println("Counter : costruttore");
     val = v; }
    public void reset() { val = 0; }
    public void inc() { val++; }
    public int getValue() { return val; }
}
```

Ereditarietà - 3

Ereditarietà - 4

```
public class Main {
    public static void main(String Args[])
    {
        Counter c = new Counter();
        Counter c1 = new Counter2();
    }
}
```

```
public class Counter2
extends Counter {
    public void dec() { val--; }
}
```

```
public class Counter {
    protected int val;

    ELIMINO
    COSTRUTTORE DI
    DEFAULT

    public Counter(int v)
    {System.out.println("Counter : costruttore");
     val = v; }
    public void reset() { val = 0; }
    public void inc() { val++; }
    public int getValue() { return val; }
}
```

Ereditarietà - 5

```
C:\work\costruttori\Main.java:6: cannot resolve symbol
symbol : constructor Counter ()
location: class Counter
    Counter c = new Counter();
    ^
C:\work\costruttori\Counter2.java:1: cannot resolve symbol
symbol : constructor Counter ()
location: class Counter
public class Counter2 extends Counter {
    ^
2 errors
Process completed.
```

**NON VIENE GENERATO  
AUTOMATICAMENTE UN  
COSTRUTTORE DI  
DEFAULT per Counter**

Ereditarietà - 6

```
public class Main {
    public static void main(String Args[])
    {
        Counter c = new Counter(1);
        Counter c1 = new Counter2();
    }
}
```

Counter2 cerca il costruttore di default di Counter...

```
C:\work\costruttori\Counter2.java:1: cannot resolve symbol
symbol : constructor Counter ()
location: class Counter
public class Counter2 extends Counter {
    ^
1 error
```

Ereditarietà - 7

```
public class Counter2 extends Counter {
    public Counter2()
    { System.out.println("Costruttore di default di Counter2"); }
    public void dec() { val--; }
}
```

Counter2 cerca il costruttore di default di Counter...

```
symbol : constructor Counter ()
location: class Counter
{
    ^
1 error
```

Ereditarietà - 8

```
public class Counter2 extends Counter {
    public Counter2()
    { System.out.println("Costruttore di default di Counter2");
      super(0); }
    public void dec() { val--; }
}
```

```
C:\work\costruttori\Counter2.java:3: cannot resolve symbol
symbol : constructor Counter ()
location: class Counter
{
    ^
```

Super deve essere la prima istruzione...

```
C:\work\costruttori\Counter2.java:5: call to super must be first statement in constructor
super(0);
    ^
2 errors
```

Ereditarietà - 9

```
public class Counter2 extends Counter {
    public Counter2()
    {super(0);}
    System.out.println("Costruttore di default di Counter2"); }
    public void dec() { val--; }
}
```

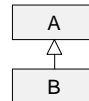
```
public class Main {
    public static void main(String Args[])
    {
        Counter c = new Counter();
        Counter c1 = new Counter2();
    }
}
```

```
public class Counter {
    protected int val;
    public Counter(int v)
    {System.out.println("Counter : costruttore");
     val = v; }
    public void reset() { val = 0; }
    public void inc() { val++; }
    public int getValue() { return val; }
}
```

Ereditarietà - 10

```
C:\Program Files\Xinox Software\JCreator LE\GE2001.exe
Counter : costruttore
Counter : costruttore
Costruttore di default di Counter2
Press any key to continue...
```

Ereditarietà - 11



La chiamata al costruttore del parent deve essere la prima istruzione dei costruttori delle classi che ereditano  
Se non è specificato super nei costruttori delle classi che ereditano viene invocato automaticamente il costruttore di default della classe da cui si eredita

Il costruttore di default può essere generato automaticamente se nessun altro costruttore della classe è definito

Se non è presente un costruttore di default (definito dal programmatore o generato in automatico) di una classe, le classi che ereditano da questa devono esplicitamente usare la super ed invocare il corretto costruttore

Ereditarietà - 12