Esempio:
Costruttori ed ereditarietà

Introduzione

- Java prevede due automatismi legati ai costruttori:
  - Se una classe non ha costruttori viene creato automaticamente il costruttore di default (quello senza parametri)
  - Se in un costruttore di un classe derivata non viene invocato il costruttore delle classe base viene automaticamente inserita la chiamata a super()
- Questi due meccanismi sono molto comodi ma bisogna fare attenzione perché si possono creare situazioni di errore non facili da comprendere
- Di seguito viene riportato un esempio che ha lo scopo di chiarire questi aspetti

Prima versione

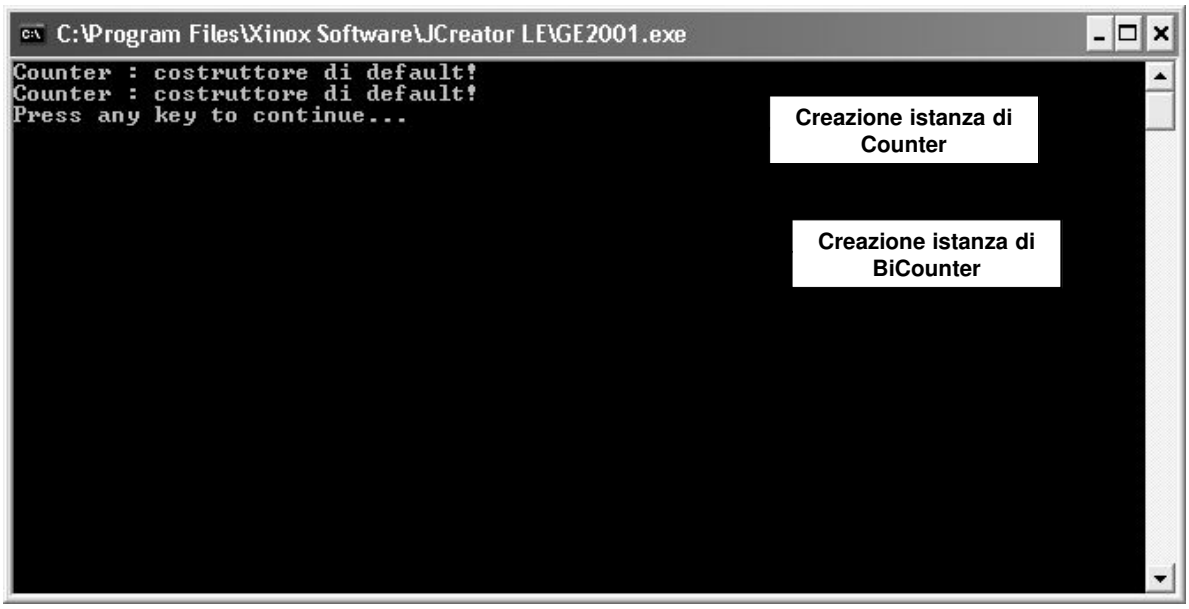
```
public class Counter
{
  protected int val;
  public Counter()
  {
    System.out.println("Counter:
     costruttore default!");
    val = 1;
  }
  public Counter(int v)
  {
    System.out.println(
     "Counter:costruttore");
    val = v;
  }
  public void reset()
  { val = 0; }
  public void inc()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

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

Non definiamo un costruttore per BiCounter

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


Risultato



- **Il compilatore Java ha aggiunto un costruttore di defaul in BiCounter e ha inserito in esso una chiamata a super()**

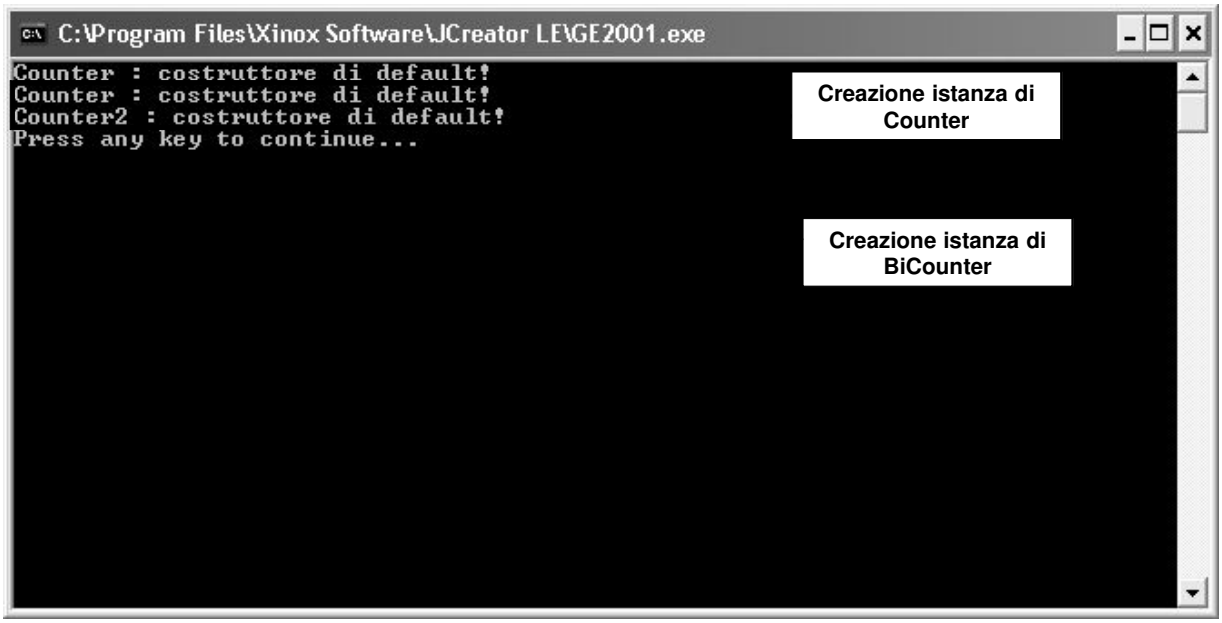
Seconda versione

```
public class Counter
{
  protected int val;
  public Counter()
  {
    System.out.println("Counter:
     costruttore 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;}
}
```

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

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


Risultato



- **Il compilatore ha aggiunto una chiamata a super() nel costruttore di BiCounter**

Terza versione

```
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;}
}
```

Eliminiamo il costruttore di default

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

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


Risultato

```
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 BiCounter extends Counter {
       ^
2 errors
Process completed.
```

- **Il costruttore di default non è stato inserito perché Counter ha almeno un costruttore**
- **Abbiamo quindi 2 errori di compilazione**

Una prima correzione

- **Il primo errore è dovuto al fatto che in main() creiamo un'istanza di Counter invocando il costruttore di default che non esiste più**
- **Modifichiamo quindi main() in modo che l'istanza venga creata invocando il costruttore non di default**

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


Risultato

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

- **Ci resta ancora un errore dovuto al fatto che nel costruttore di BiCounter il compilatore ha inserito una chiamata a super() (costruttore di default della classe base)**

Seconda correzione

- **Inseriamo quindi nel costruttore di BiCounter una chiamata esplicita al costruttore effettivamente esistente di Counter**

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


Risultato

```
C:\work\costruttori\Counter2.java:3: cannot resolve symbol
symbol  : constructor Counter  ()
location: class Counter
 {
 ^
C:\work\costruttori\Counter2.java:5:
call to super must be first statement in constructor
  super(0);
       ^
2 errors
```

- **Otteniamo ancora degli errori di compilazione**
- **Infatti la chiamata a super(0) deve essere la prima istruzione del costruttore**

Terza correzione

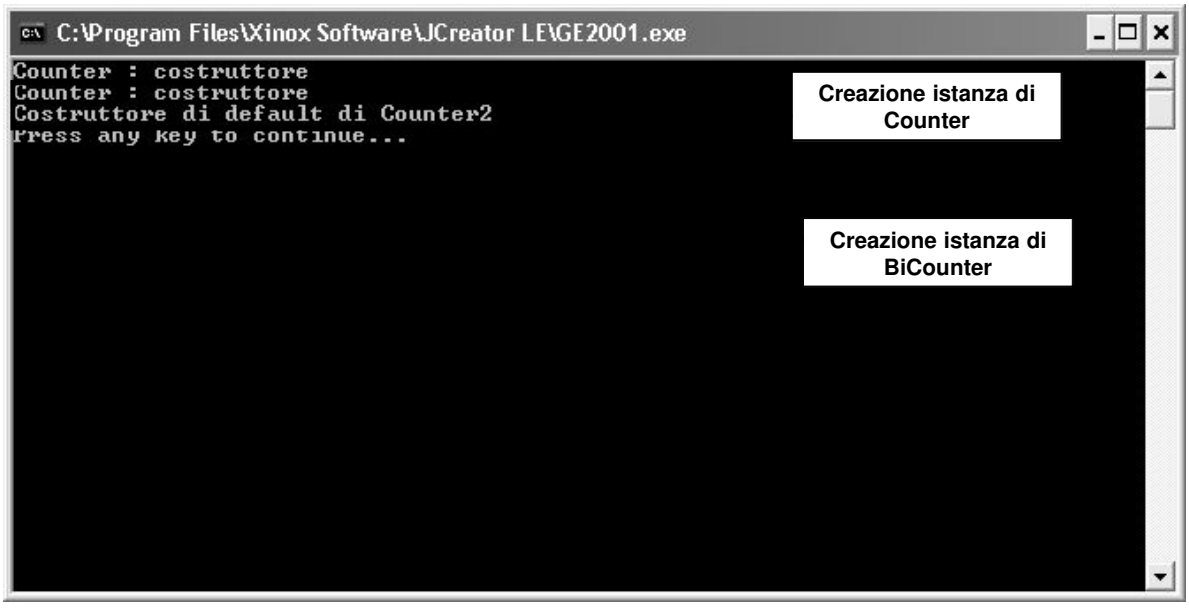
- **Mettiamo la chiamata a super in prima posizione:**

```
public class BiCounter
  extends Counter
{
  public BiCounter2()
  {
    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(1);
    Counter c1 =
     new BiCounter();
   }
}
```


Risultato finale

- **Adesso funziona tutto correttamente**



Ricapitolando…

- La chiamata al costruttore del parent (super) 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 solo 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