

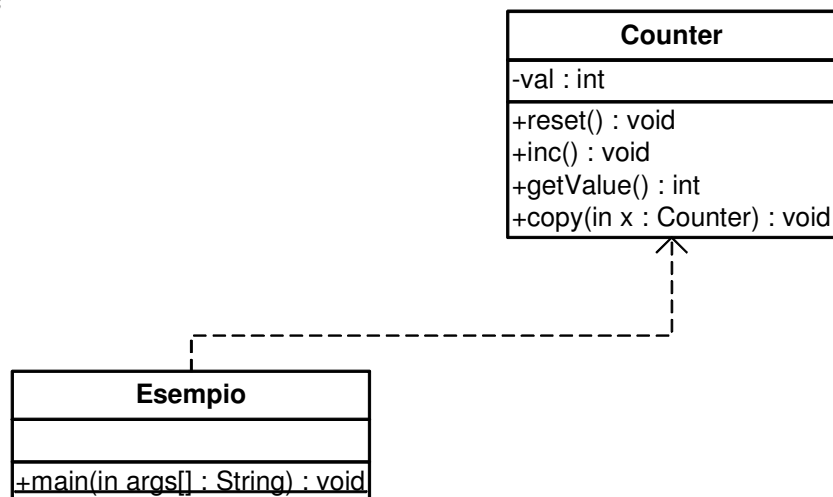


## Esempio: Classi e istanze

Esempio classi-istanze

### Un semplice programma Java

- Proviamo ad analizzare un semplice programma Java, composto da due classi, per cercare di capirne a fondo il funzionamento
- Lo seguiremo passo passo esaminando il funzionamento del programma, delle classi e delle istanze



Esempio classi-istanze

## Il programma

```
public class Counter
{
    private int val;
    public void reset()
    {
        val = 0;
    }
    public void inc()
    {
        val++;
    }
    public int getValue()
    {
        return val;
    }
    public void copy(Counter x)
    {
        val = x.val;
    }
}
```

```
public class Esempio
{
    public static void
        main(String Args[])
    {
        Counter c1, c2;
        c1 = new Counter();
        c1.inc();
        c1.inc();
        c2 = new Counter();
        c1.copy(c2);
    }
}
```

Esempio classi-istanze

## Lanciamo il programma

- Sulla riga di comandi scriviamo:  
**C:\>java Esempio**
- La Java Virtual Machine (JVM) carica la classe **Esempio** e verifica se contiene il metodo **main()**
- Il metodo esiste e quindi il programma può partire
- Esempio utilizza la classe **Counter** e quindi la JVM la carica
- Pertanto **in memoria abbiamo due classi**: Esempio e Counter
- Ricordiamo infatti che **le classi sono statiche**: esistono per tutto il tempo di vita del programma

①

Esempio
+main(in args[] : String) : void

②

Counter
-val : int
+reset() : void
+inc() : void
+getValue() : int
+copy(in x : Counter) : void

Esempio classi-istanze

## Esecuzione di main

---

- **main** è **static**: può essere invocato sulla classe  
Esempio anche se non esistono ancora istanze
- La JVM invoca quindi il metodo **main()** e vengono eseguite in sequenza le seguenti istruzioni:

```
Counter c1, c2;  
c1 = new Counter();  
c1.inc(); // primo incremento  
c1.inc(); // secondo incremento  
c2 = new Counter();  
c1.copy(c2);
```

- Seguiamole passo-passo

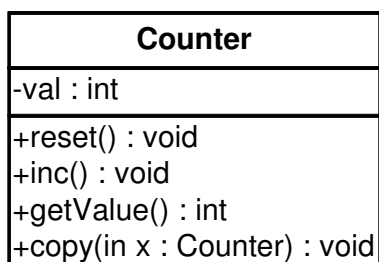
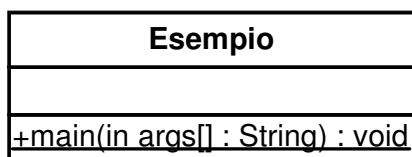
---

Esempio classi-istanze

## Counter c1, c2;

---

- Vengono creati due riferimenti a istanze di classe Counter
- La situazione in memoria è quindi:

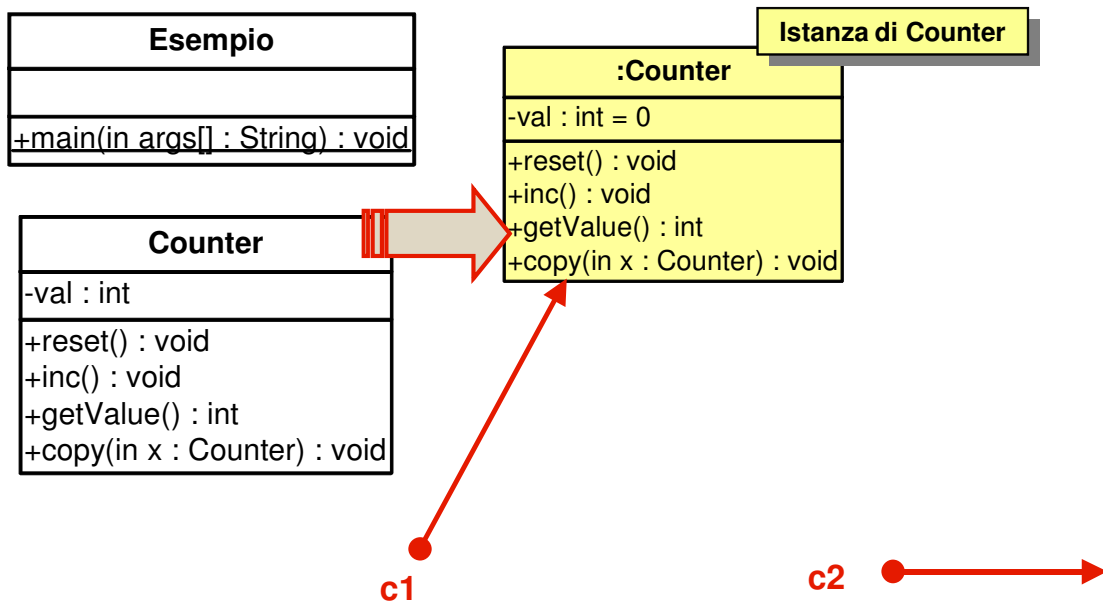


---

Esempio classi-istanze

**c1 = new Counter();**

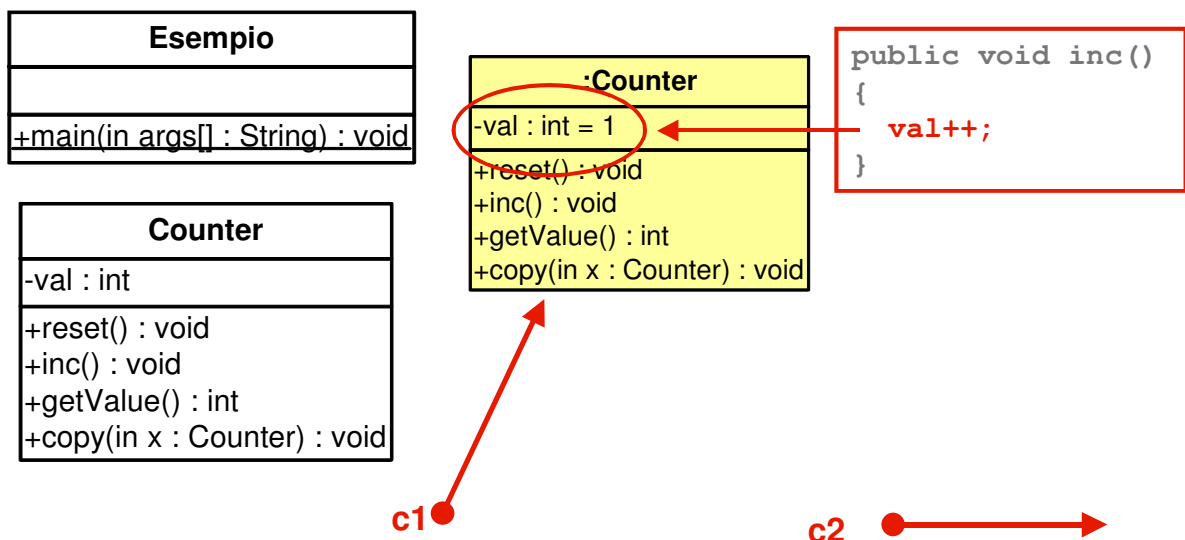
- Viene creata la prima istanza di Counter e c1 punta a questa istanza.



Esempio classi-istanze

**c1.inc(); //primo incremento**

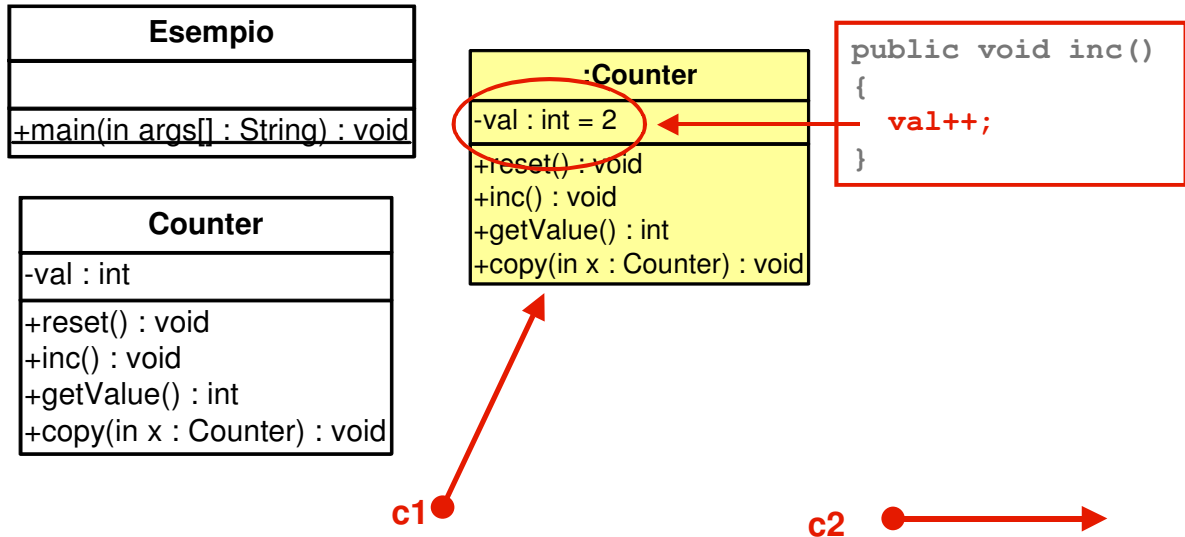
- Il metodo `inc()` viene eseguito sull'istanza puntata da `c1`
- Viene quindi incrementato il valore della variabile `val` contenuto nell'istanza puntata da `c1`: era 0, diventa 1



Esempio classi-istanze

## c1.inc(); //secondo incremento

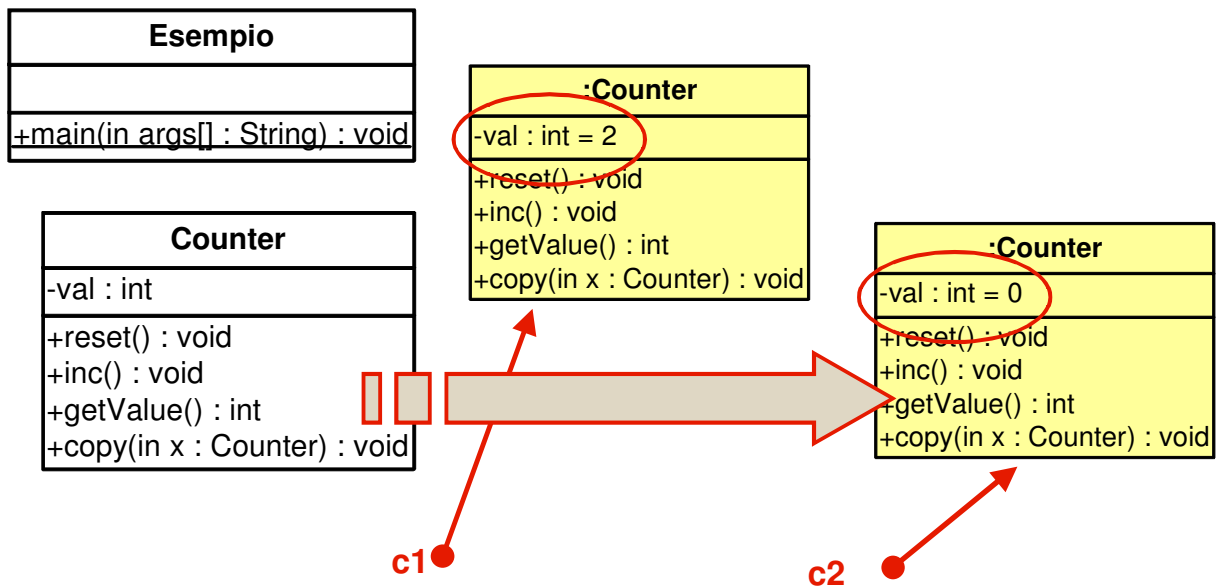
- Il metodo inc() viene eseguito una seconda volta sull'istanza puntata da c1
- Viene quindi incrementato nuovamente il valore val nell'istanza puntata da c1: era 1, diventa 2



Esempio classi-istanze

## c2 = new Counter();

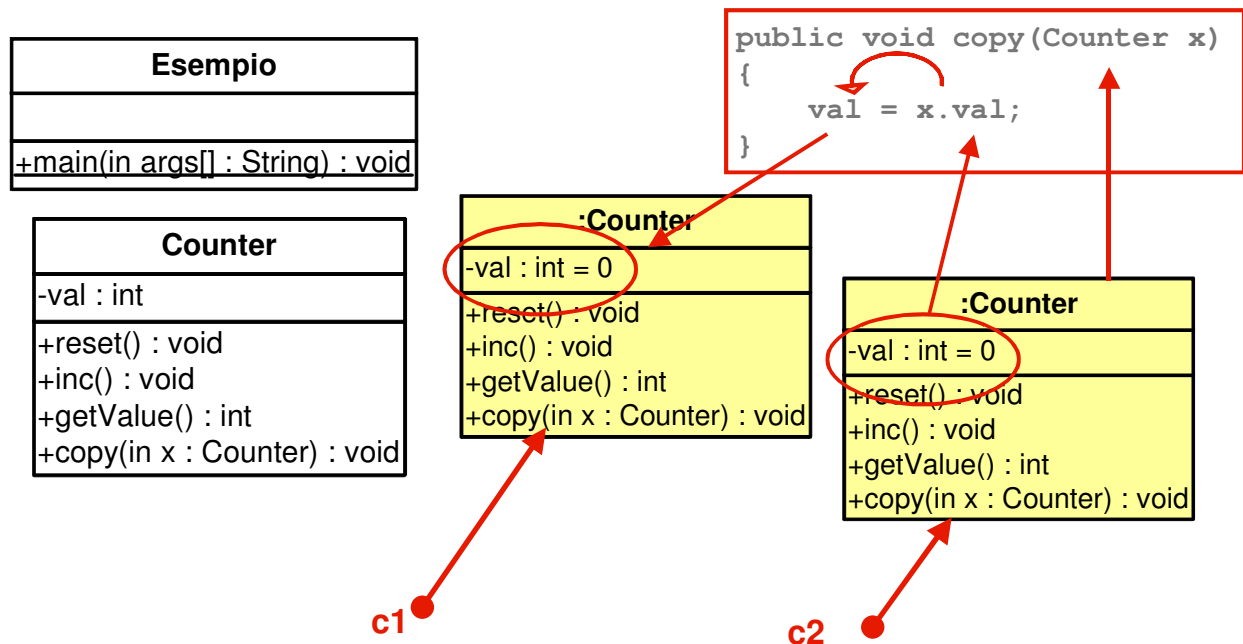
- Viene creata una seconda istanza di Counter e c2 punta a questa istanza
- Le due istanze hanno **identità e stato diversi**



Esempio classi-istanze

## c1.copy(c2);

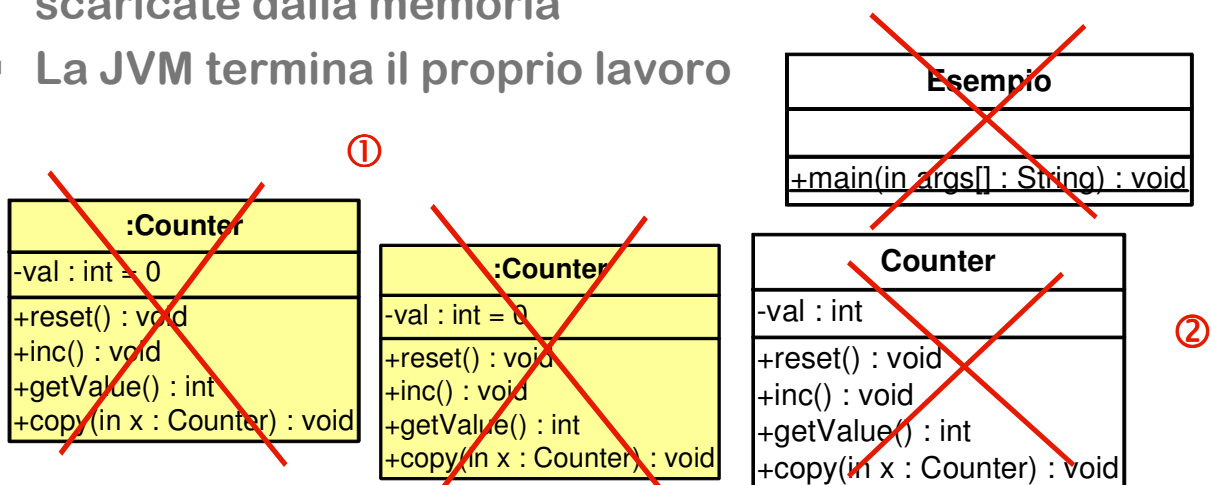
- Viene invocato sull'istanza puntata da c1 il metodo copy passando c2 come parametro
- Lo stato di c2 (val) viene copiato in c1



Esempio classi-istanze

## Fine esecuzione

- Quando il metodo main() termina le variabili c1 e c2 (variabili locali) cessano di esistere
- Le due istanze di Counter non sono più accessibili e vengono distrutte dal garbage collector
- L'applicazione termina e anche le due classi vengono scaricate dalla memoria
- La JVM termina il proprio lavoro



Esempio classi-istanze