

Il Design Pattern "Singleton"

di Marcello Pirazzoli

Il pattern singleton ci permette di avere una sola istanza di una classe, ossia la classe viene istanziata la prima volta che viene creato l'oggetto, nelle successive invocazioni da parte del programma viene restituita sempre la stessa istanza della classe.

Per realizzare un singleton basta costruire una classe con un solo costruttore con visibilità **private** in modo che non possa essere istanziata direttamente e un metodo **static** (non privato) che restituisca l'istanza della classe stessa. Generalmente tale metodo viene chiamato **getInstance()**.

In pratica abbiamo:

```
public class MioSingleton {  
  
    private static MioSingleton instance = null;  
  
    // costruttore con visibilità privata  
  
    private MioSingleton () {  
  
    }  
  
    public static MioSingleton getInstance() {  
  
        if(instance == null) {  
  
            instance = new MioSingleton();  
  
        }  
  
        return instance;  
  
    }  
  
    .....  
  
    // altri metodi  
  
}
```

Come si vede l'invocazione del metodo `getInstance()` restituisce sempre la stessa istanza della classe `MioSingleton`

perché il **new** per creare una nuova istanza è sotto il controllo dell'**if**.

Un esempio di utilizzo del singleton potrebbe essere la gestione di stampa con una stampante sola.

In questo caso si vuole evitare che più oggetti concorrenti accedano alla risorsa.

Note: Non è possibile utilizzare il Singleton in ambiente Cluster.