

# INDE: Document Orientation

[Stampa](#)  
(10) » [Errori INDE RD3](#) » [INDE: Creazione di servizi C#](#) » **[INDE: Document Orientation](#)**

**Indice dei Contenuti** [[Nascondi/Mostra](#)]

[Scenario](#)  
[Overview](#)

## Scenario

La **Document Orientation** permette di creare, tramite "Instant Developer", Classi e Form ad esse collegate con le logiche della programmazione ad oggetti.  
E' possibile quindi utilizzare i concetti come **ereditarietà** e **polimorfismo**, estendere classi e documenti, definire ed implementare **interfacce**,...

Tramite i riferimenti riportati in fondo la pagina, è possibile visualizzare un documento (ancora non ultimato) contenente un elenco di appunti e considerazioni relativi alla Document Orientation (da qui in avanti verrà chiamata **DO**) e riportando principalmente i riassunti dei video corsi disponibili in rete.

Per apprendere le logiche della DO è **consigliabile** seguire i relativi **video corsi** (Vedi link nei Riferimenti).

## Overview

Fino ad ora chi ha sviluppato applicazioni con Inde, ha utilizzato la normale metodologia definita **Table Orientation** che crea un legame molto stretto tra la struttura del database e l'interfaccia utente.  
Se per qualche ragione la struttura del database deve essere modificata, anche l'interfaccia dovrà essere adattata ed è facile che le stesse elaborazioni vengano riscritte più volte all'interno del codice, causando un aumento del numero di righe di codice e quindi la complessità dell'applicazione.

Inoltre un limite della Table Orientation è che è un modello che non è in grado di rappresentare compiutamente la realtà fisica che si deve modellare, se prendiamo come esempio la gestione di un ordine, nella descrizione del sistema manca un metodo per descrivere l'oggetto Ordine come un tutt'uno e non separatamente i pannelli per la testata, riga,...

### **Alcune Differenze tra Table Orientation e Document Orientation:**

- Le master query (MQ) definite nei pannelli Document Orientation non possono contenere le clausole di where e le subquery (si possono però fare tramite una MQ dentro la classe).
- Nei pannelli DO Master-Detail non c'è bisogno di specificare nella clausola di where la join tra pannelli, lo fa automaticamente se la classe master ha una collection sulla classe detail.  
Il **vantaggio** è che quindi non c'è bisogno di gestire nell'evento before insert la join in ogni pannello che si deve realizzare.
- In una Form DO con pannelli Master-Detail è possibile salvare e inserire righe anche se non ho ancora salvato la testata. Inoltre se cancello una riga e poi annullo l'operazione prima di non aver salvato la testata, la riga verrà ripresa e quindi non cancellata fisicamente.  
Quindi una form viene vista come un oggetto unico, salvando il tutto in una singola transazione.

=== Riferimenti ===

- Segui i Video Corsi: <http://www.progamma.com/doc> seguire nell'indice la voce **Video Corso->Document Orientation**.

- [Scarica gli \*\*Appunti\*\* disponibili sulla Document Orientation](#)
- [Scarica il \*\*Progetto Inde\*\* di prova sulla Document Orientation](#)
- **Daniele Nucci**