

DomandaRisposta
correttaPunteggio
ottenuto 2,00
su 2,00Contrassegna
domanda

Si selezionino fra le seguenti le risposte **corrette**.

- ☐ a. Per applicazioni di **modeste dimensioni**, non è necessario applicare alcun *pattern* architetturale, che risulterebbe in ogni caso in una sovra-ingegnerizzazione del sistema.
- ☒ b. In applicazione, le componenti dovrebbero essere raggruppate in **diversi insiemi**, sulla base della loro responsabilità. Ad esempio, si individuano insiemi di componenti che contengono l'*application logic*, la *business logic* e la *persistence logic*. ✓
- ☐ c. Per facilitare la **composizione dei test di unità** di una componente, è opportuno utilizzare **metodi statici** di altre classi.
- ☒ d. Esistono *pattern* architetturali universalmente riconosciuti nei quali la *business logic* dipende dalla *persistence logic*. ✓
- ☐ e. Concentrare tutti i metodi contenenti la *business logic* in un'unica componente facilita il suo riuso, anche se questi metodi vengono utilizzati da **insiemi disgiunti** di *client*.
- ☒ f. Per implementare correttamente una strategia di test di unità, le **dipendenze di una componente** devono poter essere facilmente sostituite da *mock* o *stub*. ✓

Risposta corretta.

Le risposte corrette sono:

In applicazione, le componenti dovrebbero essere raggruppate in **diversi insiemi**, sulla base della loro responsabilità. Ad esempio, si individuano insiemi di componenti che contengono l'*application logic*, la *business logic* e la *persistence logic*.

Per implementare correttamente una strategia di test di unità, le **dipendenze di una componente** devono poter essere facilmente sostituite da *mock* o *stub*.

Esistono *pattern* architetturali universalmente riconosciuti nei quali la *business logic* dipende dalla *persistence logic*.