# Modulo 6

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Elena |
|  |  | Cognome | Zazzetti |
|  |  | Data | 05/03/2021 |

Leggete attentamente ogni domanda e argomentare quanto più possibile **fornendo anche degli esempi**.  
ATTENZIONE: Le domande a risposta multipla possono contenere più risposte corrette.

1. *Quale/i delle seguenti affermazioni è esatta riguardo la connection string?*

La connection string è un oggetto che gestisce la connessione a un database

La connection string è una stringa che fornisce i parametri di inizializzazione per creare una connessione

La connection string è composta da valori in formato chiave-valore

Se un utente accede al db da diverse postazioni, la connection string cambia

1. *Quale/i degli oggetti seguenti sono forniti da un provider per gestire l’accesso ai dati?*

Connection

DataSet

Command

DataTable

1. *Quale/i delle seguenti affermazioni è vera riguardo ai DataSet?*

Il Dataset dipende da un’unica sorgente di dati

Il Dataset contiene righe e colonne di dati

Il Dataset può gestire sorgenti di dati multiple

Il DataSet è composto da una gerarchia ad “albero”

1. *Spiegare come funziona una connection pool*

*La connection pool è l’insieme delle connessioni attive di cui predisponde il database che permettono a più client di connettersi al database. Quando il client richiede una connessione, non se ne crea una nuova ma utilizza una delle connessioni della connection pool. Quando al client non è più necessaria, si staccherà dalla connessione ma questa non verrà chiusa. Il database quindi manterrà delle connessioni attive senza ricrearne ad ogni richiesta. Nel caso in cui tutte le connessioni siano occupate, una nuova richiesta rimarrà in attesa fino alla fine del timeout (specificato nella connection string) o fino a che non si liberi una connessione.*

1. *Spiegare le differenze tra le modalità di connessione al database*

*Con ADO.NET è possibile accedere ai dati in due modalità: la modalità connessa e disconnessa. La modalità connessa è utilizzata per la sola lettura e legge i record delle tabelle in maniera consecutiva. Le classi utilizzate sono Connection, che contiene le informazioni per la connessione alla sorgente di dati e contiene i metodi per aprire e chiudere la connessione, la classe Command che fa eseguire gli statement SQL (query, stored procedure..), Parameter per passare i parametri necessari e SqlDataReader che fa effettivamente l’accesso al dato, dato un comando e una connessione, è quello che esegue il comando.*

*La modalità disconnessa invece permette di manipolare i dati recuperati dalla sorgente dati in locale e riconciliarli con questa successivamente. Anche in questo caso c’è la connessione al database ma una volta recuperati i dati, lavoro con i dati disconnessi indipendentemente dalla sorgente dei dati. Le classi utilizzate in questa modalità sono DataSet, DataTable, DataColumn, DataRow, Constraint e DataRelation. In questo caso gli oggetti di tipo DataSet per esempio sono indipendenti dal tipo di database che ci ha fornito i dati. Può essere quindi utilizzato con diverse origini dati. In modalità disconnessa inoltre c’è bisogno di un Adapter che fa da ponte tra il DataSet e il database.*

1. *Spiegare cosa è un DataAdapter e specificare in quale modalità si usa*

*Il DataAdapter si usa in modalità disconnessa e fa da ponte tra il DataSet locale e il database, a cui è connesso.*

*L’Adapter utilizza Command per caricare i dati dal database al DataSet e per riportare le modifiche dal DataSet locale al database. L’Adapter è quindi connesso al database e da questo prende i dati e li riconcilia dopo le modifiche apportate in locale. Ovviamente l’Adapter deve essere istruito per poter eseguire i comandi CRUD.*

1. *Spiegare cosa è un’interfaccia ed eventuali differenze con una classe astratta. Descriverne il funzionamento con un esempio.*

*Un’interfaccia è utilizzata per contenere diverse funzionalità che devono essere implementate da classi non astratte e struct. Questa contiene solo metodi e proprietà astratte, quindi che devono essere implementate. È molto simile a una classe astratta pura, cioè con contenente soli membri astratti, che devono quindi obbligatoriamente essere implementati nelle classi derivate. Una differenza sta nel fatto che una classe non può derivare da più classi mentre può derivare da più interfacce. Infatti l’interfaccia viene usata per simulare un’ereditarietà multipla che in .NET non è ammessa.*

*Esempio:*

*interface IMyInterface {*

*double f(); // dichiaro I metodi ma non li definisco.*

*int g();*

*}*

*Public class MyClass: IMyInterface { // MyClass è l’implementazione concreta di IMyInterface*

*public double f(){ // qui definisco metodi con corpo*

*…return;}*

*public int g(){....}*

*}*

1. *Dare una definizione di Extension Method fornendo un esempio di utilizzo.*

*L’Extension method viene utilizzato per estendere le funzionalità a tipi esistenti. Per esempio posso utilizzarlo per convertire il SqlDataReader che legge dei generici oggetti in una classe da me strutturata. Inoltre questi metodi devono essere definiti statici.*

*Static public MyClass ConvertToMyClass(this SqlDataReader reader){*

*reader.Read();*

*MyClass myobj=new MyClass();*

*Myobj.ID=reader[“ID”];*

*Myobj.Name=reader[“Name”];*

*….*

*return myobj;*

*}*

1. *Definire una funzione virtuale e spiegare eventuali differenze con una funzione “normale” e una funzione astratta.*

*public abstract class Padre{*

*public virtual void f(){*

*Console.WriteLine(“Padre”);}*

*}*

*public class Figlio: Padre{*

*public override void f(){*

*Console.WriteLine(“Figlio”);}*

*}*

*La funzione virtuale è una funzione che definisco in una classe base e di cui posso fare un override nella classe derivata. Se non è segnata come virtual non posso fare un override e se la ridefinisco nella classe derivata risulteranno due implementazioni dello stesso metodo. Se marcata virtual posso sovrascrivere la funzione nella classe derivata o semplicemente ereditarla senza sovrascriverla, mentre se è segnata come astratta, questa non ha un’implementazione nella classe base che quindi deve essere fornita nella derivata.*

*.*

*Esercitazione pratica*

*Fare un’applicazione Console che implementi le seguenti funzionalità:*

*-Mostrare tutti gli agenti*

*-Mostrare gli agenti assegnati ad una determinata area data da input dell’utente.*

*-Mostrare gli agenti con anni di servizio maggiori o uguali rispetto ad un input dato dall’utente*

*-Dare la possibilità all’utente di inserire un nuovo record di agente. (Modalità disconnessa)*

*L’agente deve essere inteso come una derivazione di un’astrazione di Persona.*

*Persona ha le seguenti caratteristiche:*

* *Nome*
* *Cognome*
* *Codice Fiscale*

*Due Persone sono uguali se hanno lo stesso codice fiscale.*

*I dati relativi ad un agente devono essere mostrati a schermo tramite il seguente formato:*

*Codice Fiscale – Nome Cognome – AnnidiServizio anni di servizio*

*Opzionale:*

*Gestire l’input dell’utente tramite un menu iniziale*

*Mettere la prova pratica e teorica su Github.*