# Esercitazione di Fine Settimana – Week 5

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Roberta |
|  |  | Cognome | Beretta |
|  |  | Data | 30/07/2021 |

Leggete attentamente ogni domanda e argomentare quanto più possibile fornendo anche degli esempi.

1. **Descrivere le modalità di utilizzo di ADO.NET**

ADO.NET è un modo che abbiamo visto per collegare Visual Studio a SQL SMS con il fine di gestire tramite codice C# un DB creato con SQL SMS.

La gestione che abbiamo visto consiste nelle operazioni CRUD, ovvero di Create, Read Update e Delete.

Abbiamo visto che queste operazioni si possono fare in due modalità: una modalità connessa ed una modalità disconnessa.

In connected mode dalla console app su cui stiamo lavorando si crea una connessione ad un certo database tramite una stringa di connessione che contiene delle informazioni sensibili, quali nome del database di riferimento e dove è posizionato, credenziali di accesso come userID e password.

Noi ci siamo connessi ad un db locale, cioè che risiede sulla nostra macchina e abbiamo usato una trusted connection, ovvero le credenziali dell’account Windows.

Nella modalità connessa, una volta ricevuto il permesso di accedere al db, si possono apportare modifiche allo stesso in tempo reale.

Anche nella modalità disconnessa abbiamo bisogno di una connessione al db.

La prima volta che ci connettiamo è per scaricare le informazioni delle tabelle e di questo se ne occupa il DataSet che è una sorta di copia locale del db. Dopodichè possiamo chiudere la connessione e apportare tutte le modifiche che verranno effettuate solo sulle tabelle del DataSet e non nelle tabelle del database effettivo. Infine per allineare i dati è necessario riconnettersi al database e fare l’Update.

1. **Quali sono i metodi di esecuzione della classe DbCommand disponibili e in quali casi vanno utilizzati?**

La classe SQLCommand in C# ci permette di compiere modifiche al database, infatti istanziando un nuovo oggetto della classe, possiamo scrivere una query che poi andrà ad interagire con il db.

Ad esempio:

Sqlcommand leggi = new();

leggi.Connection = conn; (conn è la connessione al db tramite ConnectionString)

leggi.CommandType = System.Data.CommandType.Text;

leggi.CommandText = “ select \* from Autori”;

Questo pezzo di codice legge dal database tutte le colonne della tabella autori.

Oltre a select possiamo utilizzare anche insert into, delete from e update)

1. **Descrivere l’utilizzo della classe DataAdapter**

La classe DataAdapter è la classe che si occupa di gestire le tabelle in locale del DataSet, dunque si utilizza nella modalità disconnected di ADO.NET.

Per ogni tabella è necessario istanziare un nuovo oggetto DataAdapter.

Il DataAdapter contiene ad esempio un metodo Fill che permette di riempire la tabella locale con i dati della corrispondente tabella del db.

Inoltre attraverso il DataAdapter e con SQLCommand possiamo compiere le operazioni di CRUD sulla tabella associata al DataAdapter.

1. **Cos’è una Interfaccia in C#? Come può essere utilizzata?**

Un’interfaccia è un tipo di dato custom che contiene metodi astratti, ovvero metodi che hanno una firma ma non contengono una implementazione.

Ad esempio potrei avere l’interfaccia IOpener che contiene il metodo astratto ToOpen() .

Supponendo di avere oggetti diversi che si possono aprire e che questi oggetti siano già immersi in una gerarchia, possiamo marcare questi oggetti (ad esempio macchina, mela..) con l’interfaccia IOpener per poterli vedere, al bisogno, solo come oggetti che si aprono e non con le altre caratteristiche che li contraddistinguono.

Le interfacce risolvono il “problema” dell’ereditarietà singola di C# infatti utilizzando le interfacce non ci si gioca la carta dell’ereditarietà e inoltre un oggetto può avere più di una interfaccia.

1. **Cos’è un Costruttore? E come si utilizza la keyword this nella definizione di un Costruttore?**

Un costruttore è un membro di una classe, in particolare è uno speciale tipo di metodo.

Un costruttore di una classe si utilizza al momento dell’istanziamento di un nuovo oggetto della classe. Volendo, in una classe ci possono essere più costruttori.

Dal momento che in una classe ci possono essere più costruttori, è anche possibile collegarli tra loro e renderli “a cascata” , facendo di che appunto a cascata vengano chiamati uno sopo l’altro. Questa opzione si realizza tramite la keyword this.

Ad esempio supponiamo di Avere la classe Persona con le proprietà: Nome, Cognome, DataDiNascita, LuogoDiNascita.

Possiamo fare un costruttore che chieda in input tutti i campi come il seguente:

public void Persona(string nome, string cognome, string LuogoDiNascita)

E poi un costruttore che non chieda niente in input, ma che rimanda al costruttore precedente con parametri di default:

public void Persona() : this(“John”, “Doe”, “Unknown”)

**Esercitazione Pratica**

Realizzare un sistema di gestione di un magazzino merci che si basa su:

* Un database **Magazzino** (SQL Server), costituito dalla tabella
  + **Prodotti**
    - *Id* (int, PK, auto-incrementale)
    - *Codice Prodotto* (string, UNIQUE)
    - *Categoria* (nvarchar(20)(Alimentari, Cancelleria, Sanitari, ))
    - *Descrizione* (varchar(500))
    - *Prezzo Unitario* (numeric(10, 2))
    - *Quantità Disponibile* (int)
* Una **Console app** che consenta di:
  + Elencare iIProdotti
  + Inserire / Modificare / Cancellare Prodotti
  + Mostri
    - l'elenco dei prodotti con giacenza limitata (Quantità Disponibile < 10)
    - Il numero di Prodotti per ogni Categoria

VINCOLI TECNICI

* Utilizzare ADO.NET con Connected OPPURE Disconnected Mode (a scelta) per la parte di CRUD