Classi e Oggetti: Introduzione

Il linguaggio di Microsoft Visual Basic utilizza il concetto di classe per identificare o gestire le parti di un'applicazione. Se, per esempio, consideriamo un oggetto come una casa, avrà delle caratteristiche come il tipo (residenziale, condominio, villa, etc.), il numero di camere da letto, il numero di bagni, etc. e tutte queste caratteristiche le utilizziamo per descrivere la casa a qualcuno che vuole acquistarla. In programmazione, per ottenere un tale oggetto, è necessario definire i criteri che lo descrivono.
Ecco un esempio:

House
[indirizzo
tipo_di_casa
numero_di_camere
numero_di_bagni
ha_il_garage]

Queste informazioni vengono utilizzate per descrivere la casa e sulla base di questo, la casa (**House**) si chiama classe. In realtà per descrivere una vera e propria casa, è necessario fornire informazioni più dettagliate per ciascuna delle caratteristiche di cui sopra, come per esempio:

House: Mandela
indirizzo: V. Libertà 123
tipo_di_casa: Monofamiliare
numero_di_camere: 4
numero_di_bagni: 3
ha_il_garage: Sì

In questo caso, la casa di nome **Mandela** non è più una classe, ma è una casa vera e propria esplicitamente descritta, pertanto, Mandela è un oggetto. Sulla base di questo, possiamo definire una classe come una tecnica utilizzata per fornire i criteri per definire un oggetto, che è il risultato di una descrizione basata su una classe.

Le proprietà di un oggetto

Nel nostro esempio della casa, abbiamo usato dei termini per descriverla, come: Indirizzo, Tipo di casa, numero di camere, numero di bagni etc. in programmazione invece le caratteristiche utilizzate per descrivere un oggetto sono indicate come sue **proprietà**. Mentre la maggior parte degli oggetti forniscono solo le caratteristiche per descriverli, altri possono eseguire azioni, ad esempio, una casa può essere utilizzata per proteggere le persone quando fuori piove. In programmazione, invece, un'azione che un oggetto può eseguire è indicato come **metodo**. In precedenza, abbiamo definito una classe **Casa**con le sue proprietà, ma a differenza di una proprietà, un metodo deve visualizzare delle parentesi sul lato destro per distinguerlo da una proprietà. Un esempio potrebbe essere:

House
[indirizzo
tipo_di_casa
numero_di_camere
numero_di_bagni
ha_il_garage
Proteggi_pioggia ()]

Quando un oggetto ha un **metodo**, per accedere a tale metodo, si deve immettere il nome dell'oggetto, seguito da un punto, e dal nome del metodo con le parentesi.

Ad esempio, se si dispone di un oggetto casa di nome Mandela e volete chiedere di proteggere dalla pioggia, si deve digitare:

Mandela.Proteggi_pioggia ()

Questo è indicato anche come *richiamare un metodo*.

Quando è stato chiesto di eseguire un'azione, un metodo può avere bisogno di uno o più valori con cui lavorare, se un metodo necessita di un valore, tale valore è chiamato **argomento**. Mentre un certo metodo può avere bisogno di un argomento, un altro metodo potrebbe averne bisogno più di uno e il numero di argomenti di un metodo dipendono dalla sua portata e sono visualizzati tra parentesi. Supponiamo di avere un oggetto casa e si vuole proteggere il suo contenuto, per svariate ragioni, per cui l'interno deve essere protetto per:

- La pioggia
- La polvere
- Il vento
- Il sole, etc.

Sulla base di questo, potrebbe essere necessario fornire informazioni aggiuntive per indicare perché o come la parte interna deve essere protetta. Per questo motivo, quando tale metodo viene chiamato, queste ulteriori informazioni devono essere fornite, tra le parentesi del metodo. Ecco un esempio:

House
[indirizzo
tipo_di_casa
numero_di_camere
numero_di_bagni
ha_il_garage
Proteggi_pioggia (ragione)]

Come accennato in precedenza, un metodo può essere creato per prendere più di un argomento. In questo caso, gli argomenti sono separati da virgole. Ecco un esempio:

House
[indirizzo
tipo_di_casa
numero_di_camere
numero_di_bagni
ha_il_garage
Proteggi_pioggia (ragione, quando)]

Gli argomenti vengono utilizzati per aiutare l'oggetto ad eseguire l'azione prevista e una volta creato il metodo, può essere utilizzato varie volte e se un metodo necessita di un argomento, quando si richiama, è necessario fornire un valore, altrimenti il metodo non funzionerebbe. Per richiamare un metodo che richiede un argomento, si deve digitare il nome del metodo seguito dalla parentesi aperta "**(**" e seguito dal valore che sarà l'argomento e seguito da una parentesi chiusa "**)**". L'argomento che si passa può essere un valore costante, regolare o può essere il nome di un altro oggetto e se il metodo richiede più di un argomento, per richiamarlo, si devono digitare i valori per gli argomenti, nell'ordine esatto indicato, separati l'uno dall'altro da una virgola.

Abbiamo detto che, quando si richiama un metodo che richiede un argomento, è necessario fornire un valore per l'argomento, ma c'è un'eccezione.

A seconda di come è stato creato il metodo, può essere configurato per utilizzare un valore di default se non si fornisce un valore, ma non tutti i metodi seguono questa regola.
Se un metodo che accetta un argomento ha un valore di default per questo, allora non c'è bisogno di fornire un valore quando si chiama questo metodo e tale argomento è considerato facoltativo, inoltre gli argomenti che hanno valori predefiniti possono essere utilizzati e non c'è bisogno di fornirli.

Tecniche di Accesso ad un oggetto: L'enunciato Me

Finora abbiamo visto che un oggetto dispone di proprietà e metodi e abbiamo anche visto come accedere a una proprietà di un oggetto. Per esempio, immaginate di avere una classe House definita come segue:

House
[indirizzo
tipo_di_casa
numero_di_camere
numero_di_bagni
ha_il_garage
Proteggi_pioggia ()]

Se si dispone di un oggetto denominato **Gino** e che è di tipo **House**, per accedere ad alcune delle sue proprietà, è necessario utilizzare il codice come segue:

```
Gino.indirizzo
Gino.tipodicasa
```

Se si sta lavorando all'interno di un metodo della classe, per esempio, se si lavora nel corpo del metodo Proteggi_pioggia, è anche possibile accedere alle proprietà nello stesso modo, questa volta senza il nome dell'oggetto. Ciò potrebbe essere fatto nel modo seguente:

```
Proteggi_pioggia ()
indirizzo
tipo_di_casa
numero_di_camere
numero_di_bagni
End
```

Quando si accede ad un membro di una classe all'interno di uno dei suoi metodi, è possibile precedere il membro con l'oggetto **Me** preceduto dall'operatore (punto). Ecco un esempio:

```
Proteggi_pioggia ()
Me. indirizzo
Me. Tipo_di_casa
Me. Numero_di_camere
Me. Numero_di_bagni
End
```

Ricordate che l'oggetto **Me** viene utilizzato per accedere ai membri di un oggetto mentre si è all'interno di un altro membro dell'oggetto.

Il ciclo With

Abbiamo visto che è possibile utilizzare il nome di un oggetto per accedere ai suoi membri. Ecco un esempio:

```
Gino. indirizzo
Gino. tipo_di_casa
Gino. numero_di_camere
Gino. Numero_di_bagni
```

Invece di utilizzare il nome dell'oggetto ogni volta, è possibile avviare una sezione con la parola chiave **With** seguita dal nome dell'oggetto con espressione:

```
With Gino

End With
```

Tra le parole chiave **With** e **End With** si inseriscono le linee di codice per accedere al membro della classe preceduto da un punto seguito dal membro desiderato. Ciò dovrebbe essere fatto nel modo seguente:

```
With Gino
.indirizzo
.tipo_di_casa
.numero_di_camere
.numero_di_bagni
.ha_il_garage
End With
```