



Oggetti e Collezioni nei moduli di classe

Una raccolta è una serie di elementi in cui ogni elemento ha le stesse caratteristiche, in altre parole, tutti gli elementi possono essere descritte nello stesso modo. In programmazione, una collezione è una serie di elementi in cui tutti gli elementi condividono le stesse **proprietà** e **metodi**, se presenti. Ad esempio, una raccolta può essere fatta dai dipendenti di una società in cui ogni dipendente può essere descritto con le stesse caratteristiche ad esempio il nome.

■ Creazione di una collezione

Per supportare le collezioni, il linguaggio Visual Basic è dotato di una classe denominata **Collection**, in realtà, la Collezione di classe che ci accingiamo a studiare è quella definita in VBA e può essere usata per creare una collezione. Per fare questo, si deve dichiarare una variabile di tipo Collection. Ecco un esempio:

```
Sub Test()  
    Dim dipendenti As Collection  
End Sub
```

Dopo aver dichiarato la variabile, si deve allocare lo spazio in memoria per la stessa, utilizzando la parola chiave **Set** per assegnare una nuova istanza alla variabile. Ecco un esempio:

```
Sub Test()  
    Dim dipendenti As Collection  
    Set dipendenti = New Collection  
End Sub
```

Invece di creare sempre una nuova collezione, a meno che non si debba, VBA è dotato di molte collezioni standard integrate, in modo da evitare di creare sempre la propria collezione e sono indicate come le collezioni incorporate.

Le classi Collection incorporate sono derivati dalla classe Collection di Visual Basic e hanno tutte le loro funzionalità primarie ereditate dalla collezione di classe. Questo significa anche che tutto ciò che menzioneremo per la raccolta classe si applica a qualsiasi raccolta incorporata. Per utilizzare una collezione precostituita, è possibile dichiarare una variabile. Ecco un esempio:

```
Sub Test()  
    Dim CurrentSheets As Worksheets  
End Sub
```

In realtà, quando Microsoft Excel viene avviato, la maggior parte (se non tutte) delle classi sono già disponibili in modo che non c'è bisogno di dichiarare la loro variabile prima di utilizzarle.

■ Caratteristiche, e operazioni su, una raccolta

L'operazione primaria da effettuare su una collezione consiste nell'aggiungere elementi, a sostegno di questa, la Collezione classe è dotata del metodo **Add**. La sua sintassi è:

```
Public Sub Add( _  
    ByVal Item As Object, _  
    Optional ByVal Key As String, _  
    Optional ByVal { Before | After } As Object = Nothing _  
)
```

Questo metodo richiede tre argomenti, ma solo il primo è necessario e la voce "argomento" specifica l'oggetto da aggiungere alla collezione. Ecco un esempio:



```
Sub Test()  
    Dim dipendenti As Collection  
    Set dipendenti = New Collection  
    dipendenti.Add "Patrizia"  
End Sub
```

Allo stesso modo, è possibile aggiungere il numero di elementi che si desidera:

```
Sub Test()  
    Dim dipendenti As Collection  
    Set dipendenti = New Collection  
    dipendenti.Add "Patrizia"  
    dipendenti.Add "Giacomo"  
    dipendenti.Add "Alice"  
    dipendenti.Add "Franco"  
End Sub
```

Ricordate che se si sta utilizzando una delle classi standard precostituite, non è necessario dichiarare una variabile, si può solo richiamare il metodo **Add** per aggiungere un elemento. Ecco un esempio:

```
Sub Test()  
    Worksheets.Add  
End Sub
```

■ Accesso a un elemento in una raccolta

Le voci di una raccolta sono organizzate in una sequenza in cui ogni elemento contiene un indice specifico. Il primo elemento dell'insieme possiede un indice di 1, il secondo elemento contiene un indice di 2, e così via. Per accedere alle voci di una raccolta, la collezione di classe è dotata di una proprietà denominata **Item**. Ci sono due modi per utilizzare questa proprietà, uno utilizzando formalmente il valore della proprietà, digitando il nome dell'oggetto collezione, seguito dall'operatore periodo e seguito da un articolo e le parentesi opzionali. Dopo la proprietà Item o all'interno delle sue parentesi, si deve digitare l'indice della voce desiderata. Ecco un esempio:

```
Sub Test()  
    Dim dipendenti As Collection  
    Set dipendenti = New Collection  
    dipendenti.Add "Patrizia"  
    dipendenti.Add "Giacomo"  
    dipendenti.Add "Alice"  
    dipendenti.Add "Franco"  
    dipendenti.Item 2  
End Sub
```

Oppure si possono usare anche le parentesi:



```
Sub Test()  
    Dim dipendenti As Collection  
    Set dipendenti = New Collection  
    dipendenti.Add "Patrizia"  
    dipendenti.Add "Giacomo"  
    dipendenti.Add "Alice"  
    dipendenti.Add "Franco"  
    dipendenti.Item (2)  
End Sub
```

Invece di utilizzare la proprietà Item, è possibile applicare l'indice direttamente all'oggetto collezione. Ecco alcuni esempi:

```
Sub Test()  
    Dim dipendenti As Collection  
    Set dipendenti = New Collection  
    dipendenti.Add "Patrizia"  
    dipendenti.Add "Giacomo"  
    dipendenti.Add "Alice"  
    dipendenti.Add "Franco"  
  
    dipendenti.Item 2  
    dipendenti.Item (2)  
    dipendenti 2  
    dipendenti (2)  
End Sub
```

Tutte queste quattro tecniche (quelle in rosso) consentono di accedere alla voce di cui abbiamo specificato il suo indice

■ Rimozione di un elemento da una raccolta

In contrasto con l'aggiunta di un nuovo elemento, è possibile eliminare uno. A sostegno di questa operazione, la classe Collection è dotato di un nome metodo **Remove**. La sua sintassi è:

```
Public Sub Remove(Index As Integer)
```

Questo metodo accetta un argomento e quando si richiama, si deve passare l'indice della voce che si desidera eliminare. Ecco un esempio:

```
Sub Test()  
    Dim dipendenti As Collection  
    Set dipendenti = New Collection  
    dipendenti.Add "Patrizia"  
    dipendenti.Add "Giacomo"  
    dipendenti.Add "Alice"  
    dipendenti.Add "Franco"  
    dipendenti.Remove 2  
End Sub
```



Questo codice elimina il secondo elemento della collezione.

■ Numero di elementi in una Collezione

Quando si inizia una nuova collezione, ovviamente è vuota e il suo numero di elementi è 0. Per tenere traccia del numero di elementi in una collezione, la collezione di classe è dotata di una proprietà denominata **Count** il cui tipo è un numero intero. Si deve ricordare che tutte le classi di Collection incorporate ereditano il loro comportamento dalla classe Collection, ciò significa che le classi di raccolta incorporate sono dotate di una proprietà denominata Count per contenere il numero di elementi.

Abbiamo visto come è possibile aggiungere nuovi elementi a un insieme e ogni volta che si aggiunge un nuovo elemento alla raccolta, la proprietà Count aumenta di 1. Sappiamo anche come rimuovere un elemento da una raccolta e ogni volta che un elemento esistente viene eliminato, il valore della proprietà Count è diminuito di 1.

In qualsiasi momento, possiamo conoscere il numero di elementi di una collezione, ottenendo il valore della sua proprietà Count