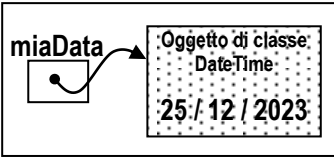


## Creare e Utilizzare Oggetti (generati da Classi già Esistenti)

<p><b>Cos'è una Classe</b></p>	<p>Una <b>Classe</b> è un <b>modello</b> sulla base del quale <b>creare uno o più oggetti</b>, tutti con le stesse caratteristiche e funzionalità.</p> <p>Gli oggetti creati a partire da una Classe si chiamano anche <b>Istanze</b></p>
<p><b>Membri di una Classe</b></p>	<p>Una <b>Classe</b> definisce un insieme di <b>Membri</b> classificabili in:</p> <p><b>Costruttori</b>: Consentono di <b>Creare un Oggetto (Istanza)</b> della Classe.</p> <p><b>Proprietà</b>: Sono le <b>Caratteristiche</b> dell'Oggetto. Come le Variabili, hanno un Nome, un Tipo e modificando il loro Valore, è possibile agire su aspetto e funzionalità dell'oggetto.</p> <p><b>Metodi</b>: Sono le <b>Azioni</b> che l'Oggetto è in grado di realizzare. Spesso hanno bisogno di "parametri", ossia di dati aggiuntivi necessari per realizzare l'operazione richiesta.</p> <p><b>Eventi</b>: Sono <b>Particolari Situazioni</b> (ad esempio, il <i>click</i> su un pulsante) al verificarsi delle quali viene eseguita una porzione di codice chiamata <i>Gestore dell'Evento</i>.</p> <p>Gli Oggetti di una Classe hanno tutti gli stessi Costruttori, Metodi, Proprietà, Eventi.</p>
<p><b>Dichiarare un Oggetto</b></p>	<p>Quando si <b>DICHIARA</b> un oggetto, viene stanziata in RAM la sua <i>variabile riferimento</i>: essa contiene il valore speciale <b>null</b>, per indicare che l'oggetto ancora <i>non esiste</i>:</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 20px;"> <code>&lt;classe&gt; &lt;nome-oggetto&gt;</code> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <p><b>miaData</b></p> <p><i>null</i></p> </div> </div> <div style="margin-top: 10px; background-color: #e0f0e0; padding: 5px; border: 1px solid #ccc;"> <p>Esempio – dichiarare un oggetto "miaData" di classe "DateTime":</p> <p><b>DateTime miaData;</b></p> </div>
<p><b>Creare un Oggetto: il Costruttore</b></p>	<p>Per poter usare l'oggetto è quindi necessario <b>CREARE</b> l'oggetto, utilizzando la parola riservata <b>new</b> seguita dal <b>Costruttore</b> (che ha lo <i>stesso nome della classe</i>) e gli eventuali <b>parametri</b> necessari alla creazione dell'oggetto stesso:</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 20px;"> <code>&lt;nome-oggetto&gt; = new &lt;classe&gt; ( &lt;parametri&gt; )</code> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <p><b>miaData</b></p>  </div> </div> <div style="margin-top: 10px; background-color: #e0f0e0; padding: 5px; border: 1px solid #ccc;"> <p>Esempio – creare l'oggetto "miaData" di classe "DateTime", contenente la data 25/12/2023:</p> <p><b>miaData = new DateTime ( 2023, 12, 25 );</b></p> </div>
<p><b>Leggere o Impostare una Proprietà</b></p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <code>&lt;nome-oggetto&gt; . &lt;proprietà&gt;</code> </div> <p>Le <b>Proprietà</b> hanno un <b>Nome</b>, sono di un certo <b>Tipo</b> e assumono un <b>Valore</b>.</p> <p>Come per le Variabili, il loro valore può essere <i>utilizzato in una espressione (lettura)</i> oppure può essere <i>impostato con una assegnazione (scrittura)</i>.</p> <div style="background-color: #e0f0e0; padding: 5px; border: 1px solid #ccc; margin-top: 10px;"> <p>Esempio – impostare il testo (proprietà Text) di un oggetto di classe Label (lblSaluto):</p> <p><b>lblSaluto.Text = "CIAO A TUTTI";</b></p> <p>Esempio – recuperare il giorno (proprietà Day) da un oggetto di classe DateTime (miaData):</p> <p><b>int Giorno = miaData.Day ;</b></p> </div>

**Richiamare un Metodo**

I **Metodi**, ossia le “azioni” che un oggetto è in grado di realizzare, possono richiedere nessuno, 1 o più **Parametri**, per poter effettuare l’operazione:

```
<nome-oggetto> . <metodo> ( <parametri> )
```

Esempio – calcolare la data (NuovaData) che si ottiene aggiungendo 5 giorni (metodo AddDays) alla data contenuta in un oggetto di classe DateTime (miaData):

```
NuovaData = miaData.AddDays ( 5 );
```

I Metodi possono **non restituire nulla** (metodi void, cioè vuoti) ...

Esempio – selezionare tutto il testo (metodo SelectAll) in un oggetto di classe TextBox (txtDato):

```
txtDato.SelectAll ( );
```

... o possono **restituire un risultato**, che può essere un **semplice valore** o anche un **nuovo oggetto**. In entrambi i casi, per non andare perso, il risultato *deve essere opportunamente gestito* (cioè assegnato, visualizzato o usato in una espressione):

Esempio – visualizzare per esteso (metodo ToLongDateString) in una Label (lblRis) la data contenuta in un oggetto di classe DateTime (miaData):

```
txtRis.Text = miaData.ToLongDateString ( );
```

... qui, il metodo ToLongDateString restituisce un singolo valore (testo) che viene subito “assegnato” alla proprietà Text della Label per essere visualizzato sulla form.

Esempio – calcolare l’intervallo di tempo (oggetto mioIntervallo di classe TimeSpan) che intercorre fra le date contenute in 2 oggetti di classe DateTime (miaData1 e miaData2):

```
mioIntervallo = miaData2.Subtract ( miaData1 );
```

... qui, il metodo Subtract restituisce un OGGETTO di classe TimeSpan: il suo riferimento viene subito memorizzato nell’oggetto mioIntervallo, che conterrà l’intervallo di tempo che intercorre fra miaData1 e miaData2.

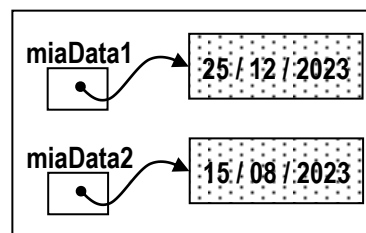
**Assegnazione fra Oggetti**

Quando si “assegna” (con l’istruzione di Assegnazione che usa il simbolo “=”) un oggetto a un altro oggetto, viene **duplicato il RIFERIMENTO** e non il contenuto dell’oggetto.

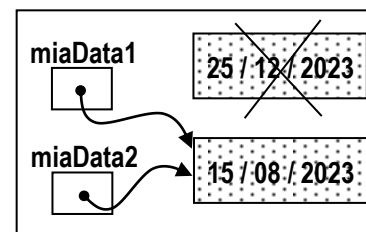
Esempio – se sono stati creati due oggetti di classe DateTime di nome miaData1 e miaData2, l’effetto dell’istruzione di assegnazione

```
miaData1 = miaData2
```

è quella di COPIARE, in miaData1, il RIFERIMENTO presente in miaData2: alla fine anche miaData1 “punta” allo stesso oggetto a cui punta miaData2.



L’effetto di una assegnazione fra due oggetti non è quello di creare un secondo oggetto “copia” di quello assegnato (come per i valori delle variabili), bensì è quello di ottenere **due identificatori che indicano lo stesso oggetto**: a questo punto, modificando il primo, risulta modificato anche il secondo, visto che puntano al medesimo oggetto.



Inoltre, il riferimento presente inizialmente nell’oggetto a cui assegnamo, viene sovrascritto e, quindi cancellato: **l’oggetto a cui puntava NON E’ PIU’ ACCESSIBILE** e viene rimosso dalla RAM.