Polimorfismo

Il termine polimorfismo indica la possibilità di definire **metodi e proprietà con lo stesso nome**, in modo che, ad esempio, una classe derivata possa ridefinire un metodo della classe base con lo stesso nome.
Continuiamo ad usare le nostre classi Persona e Studente realizzate nelle lezione precedente per capire il concetto di polimorfismo. Inizialmente, avevamo definito il metodo pubblico StampaMessaggio() nella classe Persona, ma non ne abbiamo ancora scritto il corpo. Vogliamo fare in modo che questa routine stampi a video le informazioni sulla persona.

È naturale pensare che, analogamente, anche la classe Studente abbia un metodo StampaMessaggio() che mostri, oltre alle informazioni sulla persona, anche quelle proprie di uno studente. Perciò dobbiamo definire i metodi nel modo seguente:

```
public class Persona
{
  //…
  public virtual void StampaMessaggio()
  {
    System.Console.WriteLine("Nome: " + mNome + " " + mCognome);
  }
  //…
}

public class Studente : Persona
{
  //…
  public override void StampaMessaggio()
  {
    System.Console.WriteLine("Nome: " + mNome + " " + mCognome);
    System.Console.WriteLine("Matricola: " + mMatricola);
  }
  //…
}
```

Il metodo StampaMessaggio() è definito in entrambe le classi, ma ha un **comportamento diverso**, dal momento che in Studente viene stampato anche il numero di matricola.
A seconda che si dichiari un oggetto di tipo Persona oppure Studente, verrà richiamata la routine StampaMessaggio() corrispondente. Per fare questo, abbiamo usato il polimorfismo: il metodo

StampaMessaggio() nella classe base è stato dichiarato usando la **parola chiave «virtual»** : con essa si indica che è possibile avere un'altra definizione per la stessa routine in una classe derivata.
In Studente, infatti, StampaMessaggio() è stato definito specificando la **parola chiave «override»** : in questo modo si dice che tale funzione sovrascrive un metodo con lo stesso nome che è stato definito nella classe base.
In alcuni casi può essere necessario, all'interno del metodo ridefinito, **richiamare il metodo che si sta ridefinendo**. Nel nostro esempio, il metodo StampaMessaggio() della classe Studente potrebbe richiamare il metodo StampaMessaggio() di Persona, che si occupa già di mostrare il nome e il cognome, limitandosi quindi a stampare solo le informazioni proprie di uno studente.
Per fare questo, si deve usare la **parola chiave «base»**, che, come già accennato nella lezione precedente, consente di avere accesso alla classe che si sta derivando; alla luce di queste considerazioni, il metodo StampaMessaggio() di Studente diventa:

```
public override void StampaMessaggio()
{
  base.StampaMessaggio();
  System.Console.WriteLine("Matricola: " + mMatricola);
}
```

In pratica, base.StampaMessaggio() richiama il metodo **corrispondente nella classe base**.
C'è anche un altro modo per sfruttare il polimorfismo, ovvero usando la **parola chiave «new»** (equivalente a Shadows in VB .NET), che nasconde una definizione della classe base con lo stesso nome.
L'esempio sopra riportato, usando «new», diventa:

```
public class Persona
{
  //…
  public void StampaMessaggio()
  {
    System.Console.WriteLine("Nome: " + mNome + " " + mCognome);
  }
  //…
}

public class Studente : Persona
{
  //…
  public new void StampaMessaggio()
  {
```

```
    base.StampaMessaggio();
    System.Console.WriteLine("Matricola: " + mMatricola);
  }
  //…
}
```

Come si vede, nella classe base la parola chiave «virtual» è sparita, mentre in Studente il metodo StampaMessaggio() è preceduto da «new».

È importante notare che, usando la **coppia «virtual» e «override»**, il metodo che sovrascrive una routine della classe base deve avere non solo lo stesso nome, ma anche lo stesso numero di argomenti e dello stesso tipo.
Invece, **usando «new»**, è possibile ridefinire un metodo completatmente. Con «new» si può addirittura nascondere una variabile della classe base con un intero metodo nella classe derivata.
Questa rapida panoramica è solo una introduzione al polimorfismo, per approfondire è sempre utile consultare la Guida in linea.