Riepilogo – Uso di delegati

➔I “delegati” consentono di utilizzare funzioni come fossero dati da assegnare ad apposite variabili e quindi rende possibile scrivere metodi che hanno come parametri delle funzioni.

Non si tratta di un concetto nuovo: in vecchi linguaggi si parlava di “puntatori a funzione” e nel paradigma di programmazione funzionale “tutto” è considerato come un “dato”.

Tuttavia, nell’ambito del paradigma di programmazione ad oggetti si può beneficiare di questo concetto, che pur essendo estraneo alla programmazione ad oggetti, consente di semplificare la programmazione, producendo metodi molto più flessibili e versatili (adattabili a diverse esigenze!).

TIPI DI DELEGATI DEL C#

Func<T1, T2>

Si tratta di una funzione che ha un parametro di tipo T1 e che restituisce un risultato di tipo T2

Action<T>

Si tratta di una funzione che ha un parametro di tipo T e che non restituisce nulla, ma produce qualche effetto collaterale (una stampa, una modifica di un oggetto)

Predicate<T>

Si tratta di una funzione booleana con parametro di tipo T

Esempio di dichiarazione di una variabile di tipo Func a cui possiamo assegnare diverse funzioni per ottenere diversi calcoli matematici:

```
class Program
{

   public static void Main()
   {
```

```
Func<int,int> f;    // dichiarazione della variabile f di tipo Func, cioè una funzione di parametro int
                    // che restituisce un risultato di tipo int


        f = Raddoppia;   // assegnazione ad f della funzione Raddoppia

        int risultato = f(7);   // effettuazione del calcolo tramite la variabile f

        Console.WriteLine(risultato);    //  14


        f = Quadrato;   // ora ad f si assegna la funzione Quadrato

        int risultato2 = f(7);

        Console.WriteLine(risultato2);   // 49


        // addirittura potrei creare la funzione al volo      fantastico!!!

        f = (x => x * 3);      // assegnazione di una funzione anonima (espressione lambda)

        int risultato3 = f(7);

        Console.WriteLine(risultato3);   // 21

   }
```

La dichiarazione delle funzioni usate per fare i calcoli matematici può essere fatta nella stessa classe che contiene i metodi che le utilizzano, oppure in un'altra classe che funga da deposito di funzioni.

```
// delegati per fare alcuni calcoli matematici -  sono di tip Func<int, int>

        public static int Raddoppia(int  n)
        {
            return n * 2;
         }


        public static int Quadrato(int  k)
        {
            int appoggio = k * k;
           return appoggio;
         }

} // fine class Program
```

Scrittura di un metodo nella classe Program che riceve come parametro un delegato per fare di volta in volta i calcoli matematici desiderati:

```
public   void  StampaCalcolo(Func<int,int> f, int x)
{
    Console.WriteLine(  f(x)  );
}


// utilizzo di tale metodo:

StampaCalcolo(Raddoppia, 7);     // stamperà  14

StampaCalcolo(Quadrato,  7);     //  stamperà  49

StampaCalcolo((x=>x*3),  7);       // stamperà 21
```

Nota: se le funzioni delegate vengono messe nella classe Operazione come metodi statici, essi verranno chiamati specificando anche il nome della classe che le contiene, come ad esempio:

```
  StampaCalcolo( Operazione.Raddoppia, 7);
```