

## Realizzazione di un DSL (Domain Specific Language)

Autore Bandiera Roberto

Si vuole realizzare un interprete per un semplice linguaggio che consente all'utente di scrivere delle operazioni matematiche, come ad esempio "Moltiplica 50 2" o anche "Dividi" "Somma" "Sottrai"

Abbiamo diversi modi di operare:

1° modo) utilizzo di una struttura condizionale multipla

Data la ridotta gamma di operazioni matematiche previste, risulta facile impostare una struttura condizionale multipla per decidere quale calcolo fare:

```
class Calcolatore
{
public int EffettuaCalcolo(string s) // esempio s = "Moltiplica 50 2";
{
    string[] arr = s.Split(); // "Moltiplica" "50" "2"
    string op = arr[0];
    int x = Convert.ToInt32(arr[1]);
    int y = Convert.ToInt32(arr[2]);

    int ris = 0; // risultato
    if (op == "Moltiplica")
    { ris = x * y; }
    else if (op == "Dividi")
    { ris = x / y; }
    else if (op == "Somma")
    { ris = x + y; }
    else if (op == "Sottrai")
    { ris = x - y; }
    return ris;
}
}
```

Tuttavia si tratta di un metodo di programmazione poco flessibile.

2° modo) utilizzo di un Dictionary per associare il nome di ciascuna operazione con la corrispondente funzione di calcolo (funzione che riceve una coppia di interi e restituisce un intero)

```
class Calcolatore
{
    private Dictionary<string, Func<int, int, int> dict;

    public Calcolatore()
```

```

{
    dict = new Dictionary<string, Func<int,int,int>>();
    // carico nel dictionary le operazioni che prevedo di poter fare
    dict["Moltiplica"] = Moltiplica;
    dict["Dividi"] = Dividi;
    dict["Somma"] = Somma;
    dict["Sottrai"] = Sottrai;
}

// le funzioni di calcolo (delegati)
public static int Moltiplica(int x, int y)
{ return x * y; }
public static int Dividi(int x, int y)
{ return x / y; }
public static int Somma(int x, int y)
{ return x + y; }
public static int Sottrai(int x, int y)
{ return x - y; }

public int EffettuaCalcolo(string s) // esempio s = "Moltiplica 50 2";
{
    string[] arr = s.Split(); // "Moltiplica" "50" "2"
    string op = arr[0];
    int x = Convert.ToInt32(arr[1]);
    int y = Convert.ToInt32(arr[2]);

    int ris = 0;
    if (dict.ContainsKey(op))
    {
        Func<int,int,int> f = dict[op]; // prendo il delegato dal dictionary
        int ris = f(x,y);
    }
    return ris;
}
}

```

Il dictionary consente di predisporre l'elenco dei "servizi di calcolo" messi a disposizione, in modo analogo a come fanno le moderne applicazioni web.

3° modo) uso della tecnica di Reflection

Si tratta di una tecnica che consente di "trasformare" il nome di una funzione nella funzione stessa.

```
using System.Reflection;
```

```
class Calcolatore
```

```

{
    // le funzioni di calcolo (delegati)
    public static int Moltiplica(int x, int y)
    { return x * y; }
    public static int Dividi(int x, int y)
    { return x / y; }
    public static int Somma(int x, int y)
    { return x + y; }
    public static int Sottrai(int x, int y)
    { return x - y; }

public int EffettuaCalcolo(string s) // esempio s = "Moltiplica 50 2";
{
    string[] arr = s.Split(); // "Moltiplica" "50" "2"
    string op = arr[0];
    int x = Convert.ToInt32(arr[1]);
    int y = Convert.ToInt32(arr[2]);

    int ris = 0;

    Type type = typeof(Calcolatore); // la classe che contiene le funzioni
    // dal nome dell'operazione ottengo il corrispondente metodo
    try{
        MethodInfo m = type.GetMethod(op);
        // creo il delegato - è necessario effettuare un casting
        Func<int, int, int> f =
            (Func<int, int, int>) Delegate.CreateDelegate( typeof(Func<int, int, int>), m );
        ris = f(x, y);
    }
    catch (Exception e) { }

    return ris;
}
}

```

Questo metodo e quello basato sul Dictionary sono molto flessibili e consentono di ampliare agevolmente il vocabolario del linguaggio del nostro calcolatore.

## ESERCITAZIONE

Realizzare una calcolatrice programmabile con la seguente interfaccia grafica:

Form - Calcolatrice Programmabile	
Valore iniziale _____	← textbox
Programma	← list box
Somma 5	
Somma 8	
Sottrai 3	
Dividi 2	
Moltiplica 5	
[ESEGUI] Risultato _____	← label
[SALVA PROGRAMMA]	
[CARICA PROGRAMMA]	

La classe Calcolatore ha una memoria dove mantiene il risultato provvisorio dei calcoli effettuati.

Il programma consiste in una lista di stringhe.

Notare come sia possibile scrivere un programma con un qualsiasi editor di testo e poi farlo interpretare ed eseguire alla nostra calcolatrice programmabile.

Se anziché una listbox si utilizza un textbox multilinea, si può scrivere il testo della sequenza dei comandi direttamente nell'interfaccia grafica del programma.

Approfondimento:

Aggiungere un pulsante per l'esecuzione passo-passo.

L'esecuzione passo-passo consente di vedere selezionata l'istruzione corrente e il risultato provvisorio (conviene introdurre un delay ad esempio di 1000 msec tra una istruzione e la successiva).