

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

**PROGETTO E REALIZZAZIONE DI UN
COMPONENTE PER LA
VISUALIZZAZIONE E LA
VALUTAZIONE DI ABSTRACT SYNTAX
TREE**

CANDIDATO:

Andrea Asta

RELATORE:

Chiar.mo Prof. Giuseppe Bellavia

Introduzione

- **Abstract Syntax Tree**

- Albero che modella una porzione di codice

- **Obiettivi del progetto**

- Poter visualizzare la struttura di un AST
- Poter eseguire alcune tipologie di AST

- **Output del progetto**

- Libreria di classi .NET 4.0 in C# per il raggiungimento degli obiettivi preposti

Expression Trees

EXPRESSION TREE

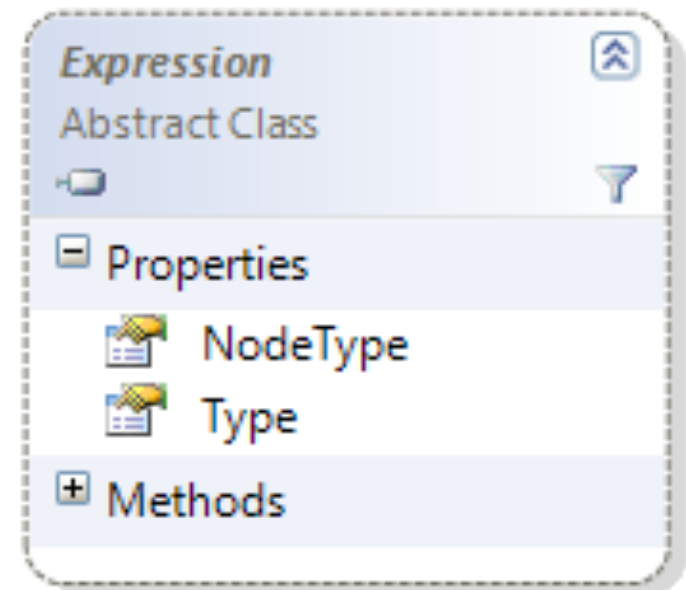
Expression Tree

- Specifica .NET per **Abstract Syntax Tree**
- ET v.2 nel **framework .NET 4.0**
 - Potenza espressiva rispetto alla versione precedente
 - Legame con il *Dynamic Language Runtime*
- Namespace

`System.Linq.Expressions`

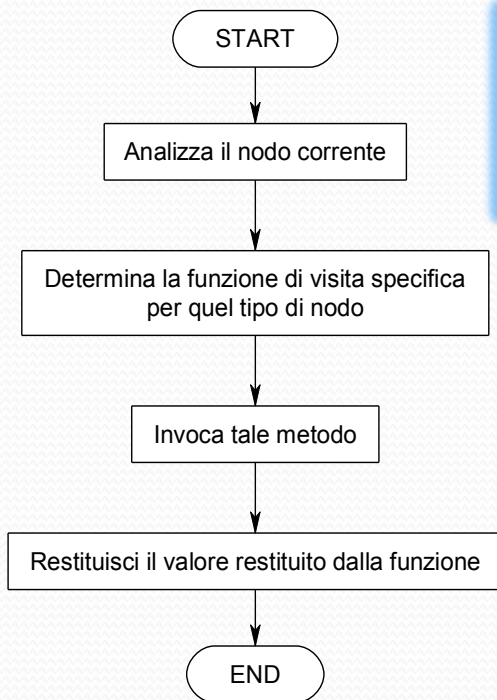
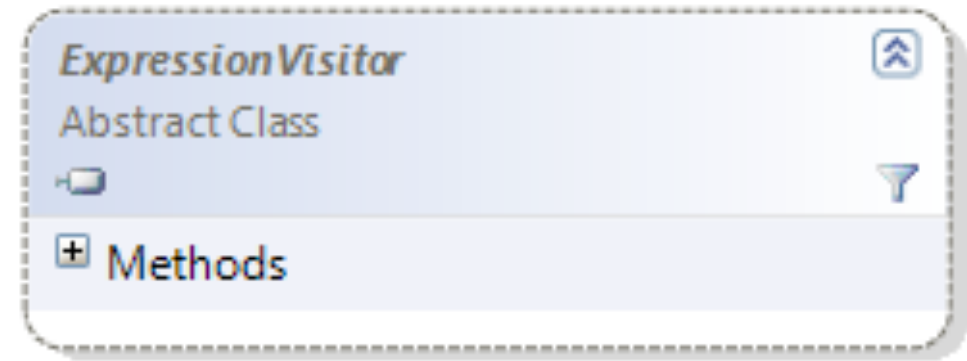
- Famiglia di classi di `Expression`
- Nodi speciali
- Costruzione via lambda expression
- Costruzione nodi mediante factory

Andrea Asta – ET Viewer

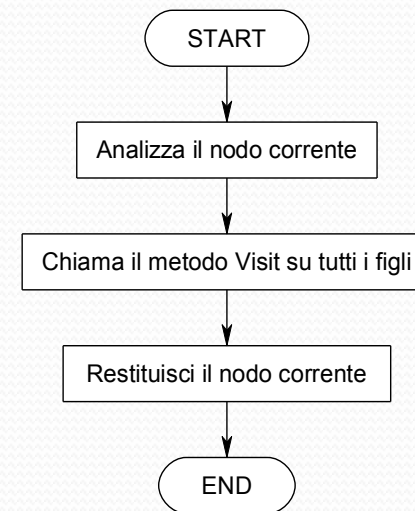


Visita dell'albero

- Classe `ExpressionVisitor`
 - Metodo `Visit()`
 - Metodi di visita specifici per le tipologie di nodo

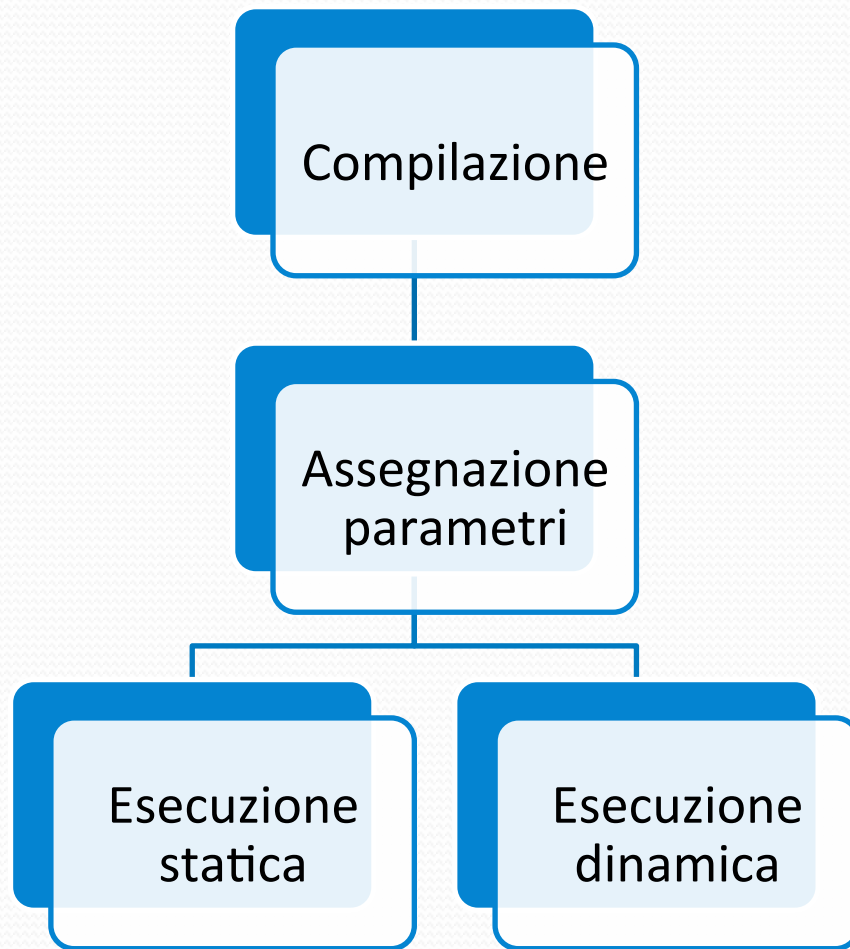


**Visit()
base**



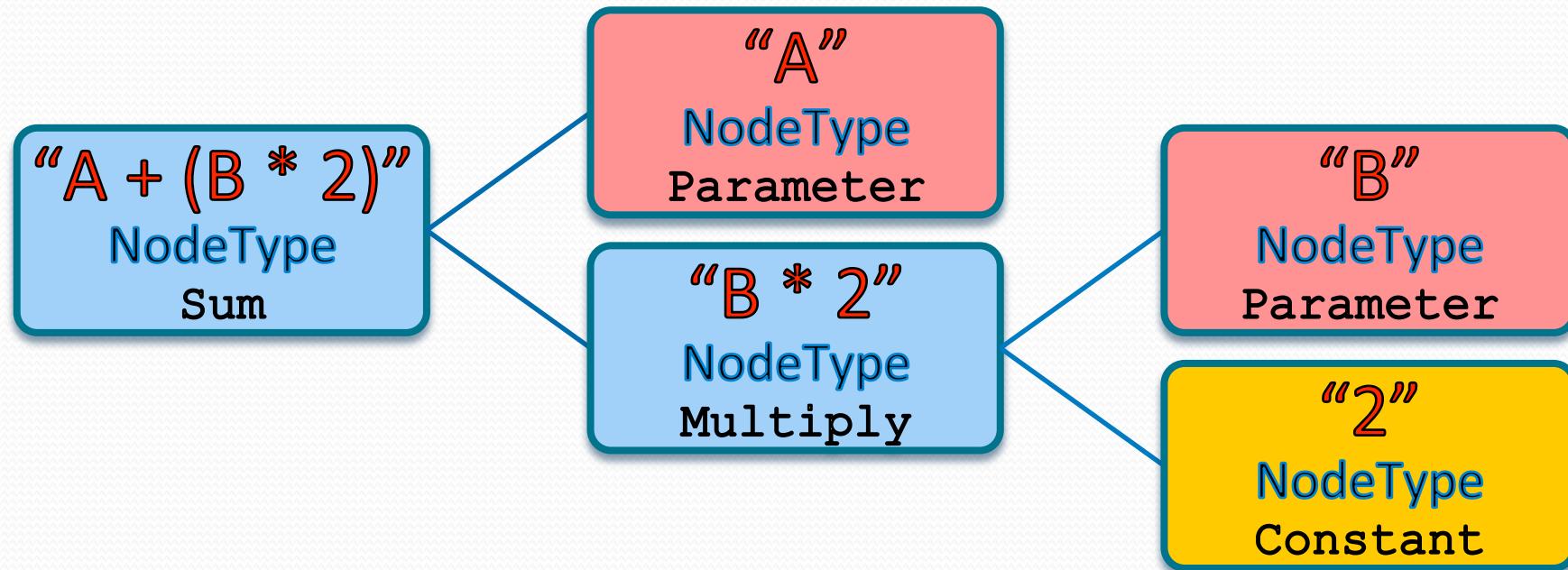
**VisitX()
base**

Compilazione ed esecuzione



- **Compilazione**
 - Metodo `Compile()` di `LambdaExpression`
 - Tipo restituito: `Delegate`
- **Esecuzione statica**
 - Utilizzo del delegato da codice
- **Esecuzione dinamica**
 - Chiamata a `DynamicInvoke()` del delegato

Esempio: $(A + (B * 2))$



Legenda:

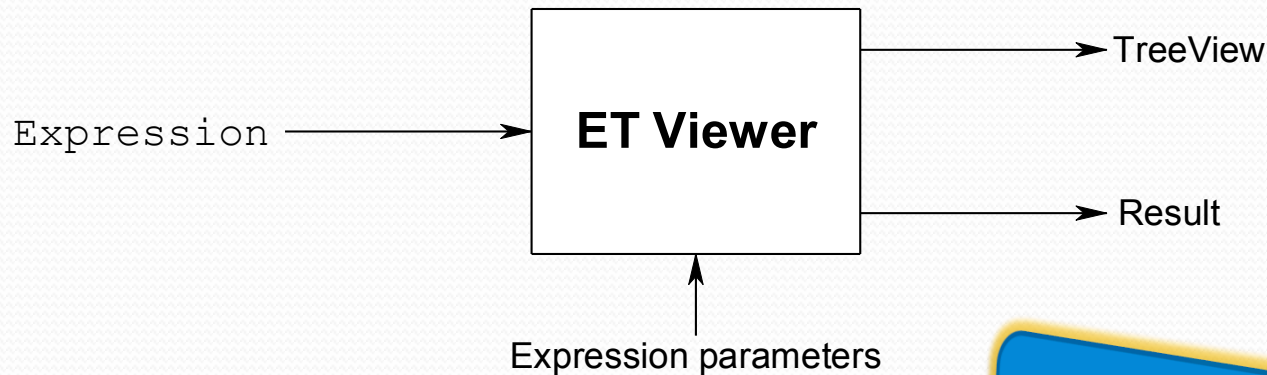
Nodi
BinaryExpression

Nodi
ParameterExpression

Nodi
ConstantExpression

Analisi e Specifica dei Requisiti

Modello del sistema



- **Funzionalità richieste**

- Funzioni di visualizzazione
- Raccolta input
- Esecuzione di espressioni compilabili
- Visualizzazione dettagli del risultato
- Gestione di sotto – espressioni

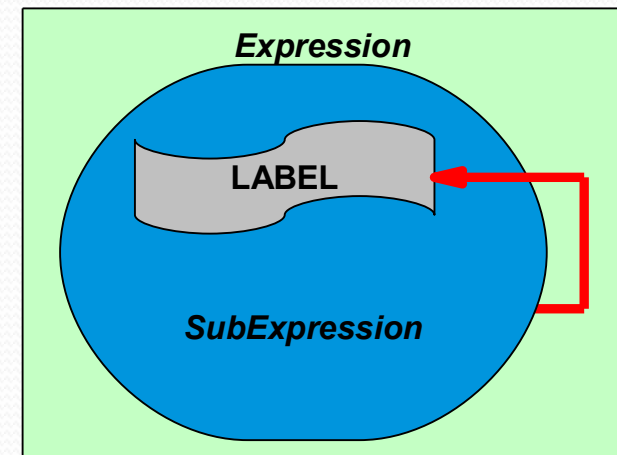
Modulo DLL

**Programma di
test**

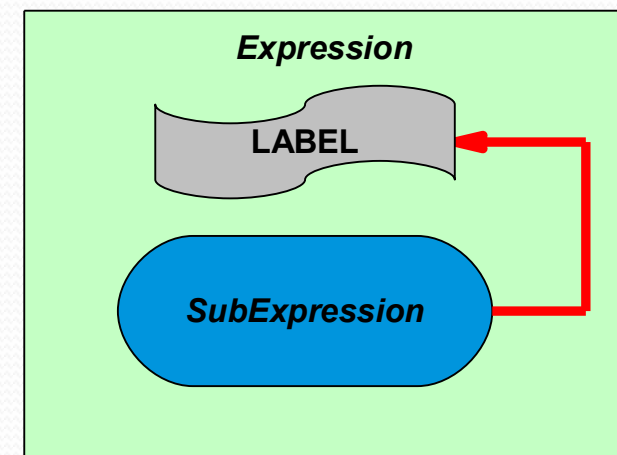
Criteri di compilabilità

- **Analisi dei tipi di input**
 - Ammessi solo tipi semplici
- **Analisi delle label**
 - Attenzione a riferimenti esterni al blocco attuale
- **Analisi del tipo di nodo**
 - Non ha senso eseguire nodi di tipo `ParameterExpression` e `ConstantExpression`

COMPILABILE

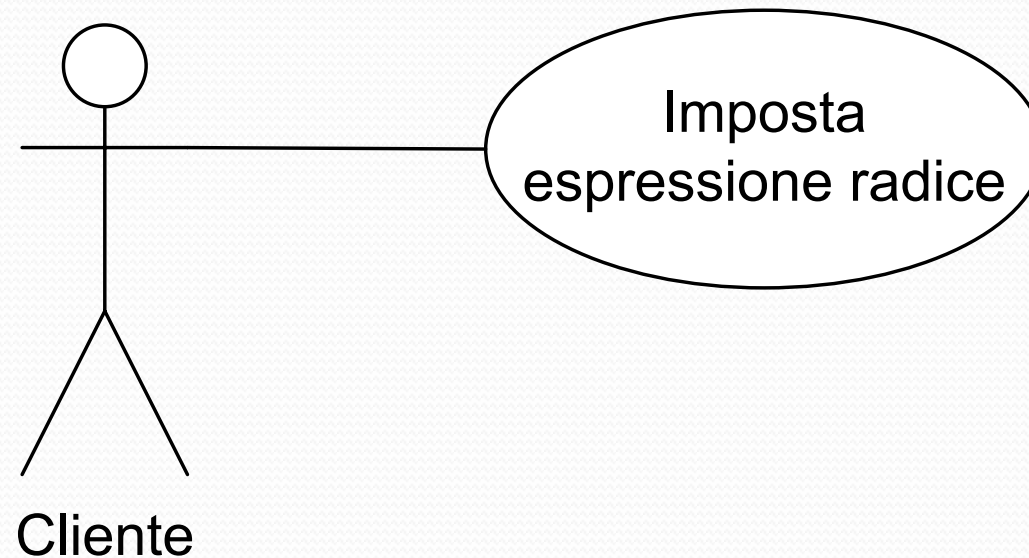


NON COMPILABILE

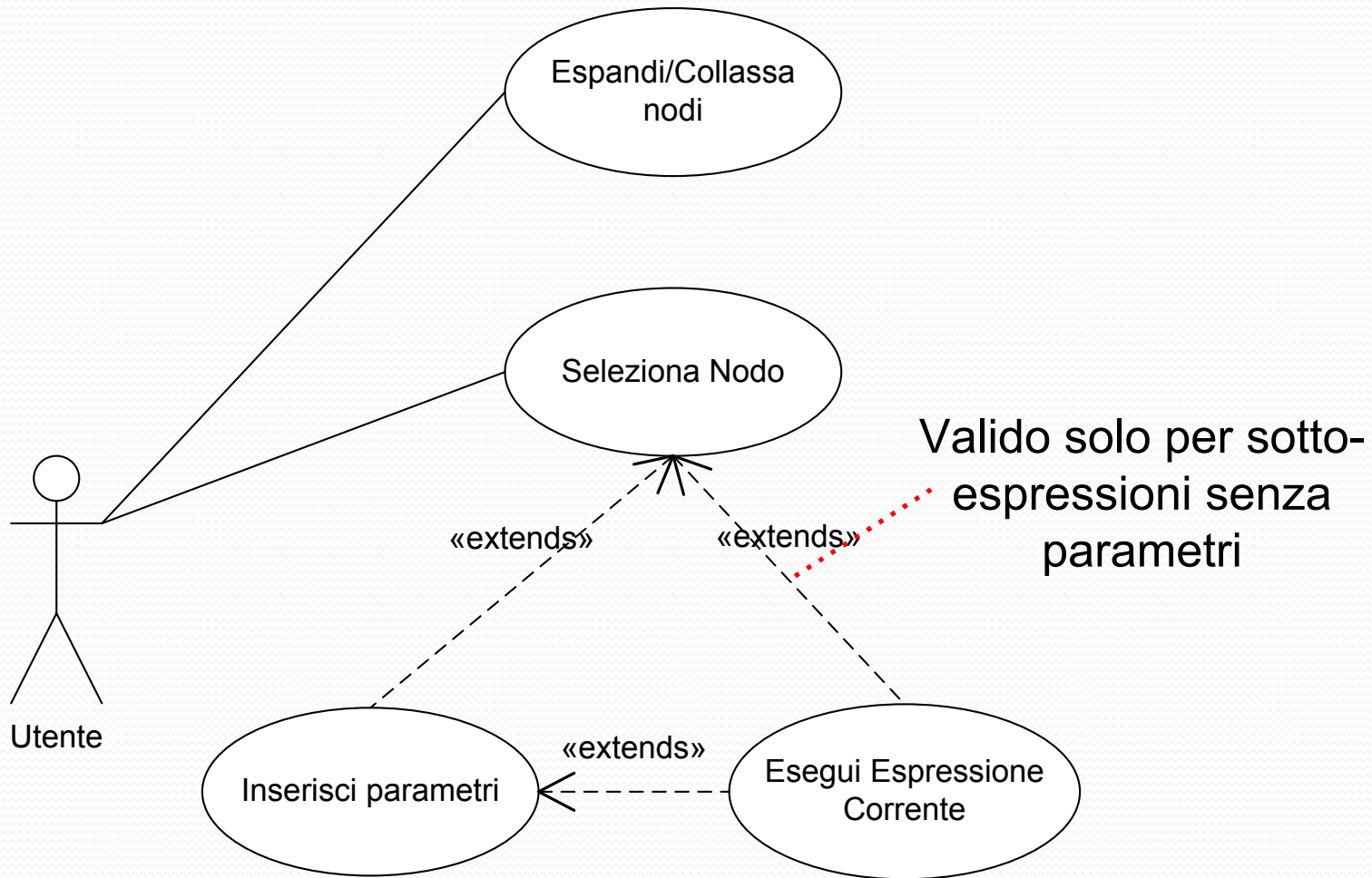


Caso d'uso: Cliente esterno

- Software distribuito come **libreria di classi** (DLL)



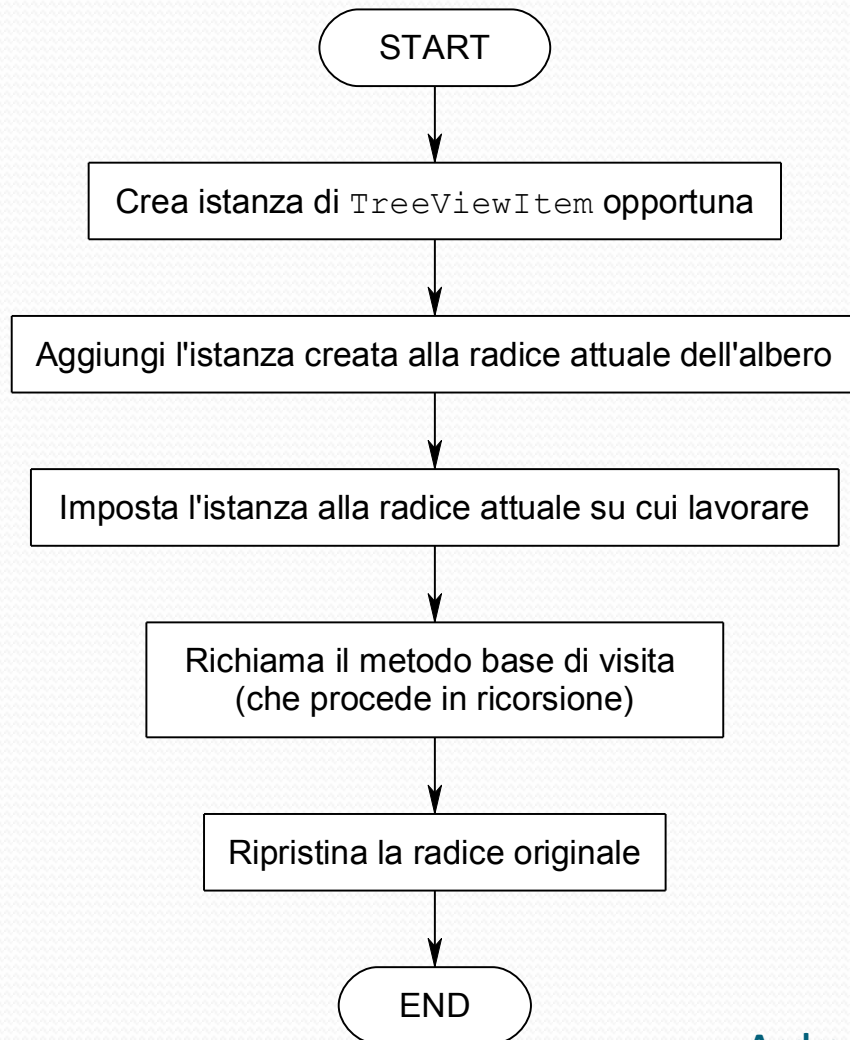
Caso d'uso: Interazione utente



Modello dei dati

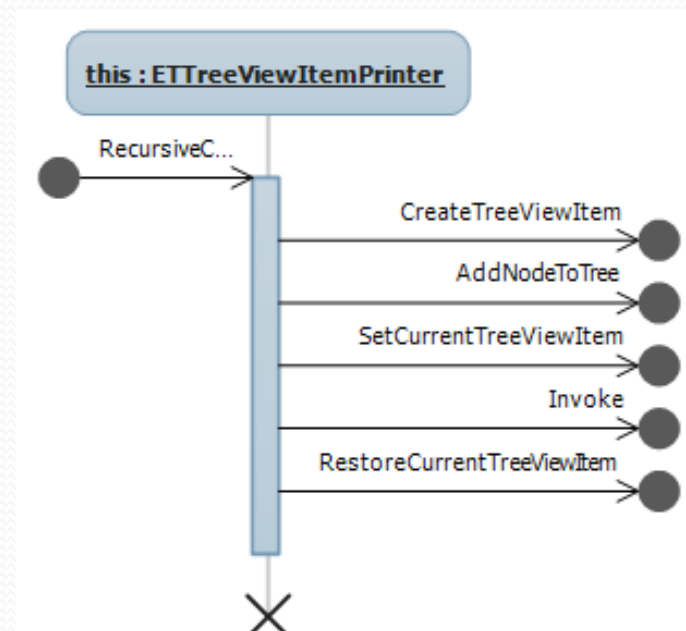
MODELLO DEI DATI

Costruttore dell'albero grafico












- Ridefinizione dei metodi di visita di ciascun nodo

```
public class  
ETTreeViewItemPrinter  
: ExpressionVisitor
```



Parametri e Variabili

ParameterWithValue	
	_parameter:ParameterExpression {readOnly}
	_value:object
	_enabled:bool
	«GetAccessor, SetAccessor, property» Enabled():bool
	«constructor» ParameterWithValue(in parameter:ParameterExpression, in value:object=null)
	«GetAccessor, property» Parameter():ParameterExpression
	«GetAccessor, SetAccessor, property» Value():object
	«event» ValueChanged():Action
	OnValueChanged():void

- Distinzione tra **parametro** e **variabile**
public class

```
ETParameterFinder  
:  
ExpressionVisitor
```

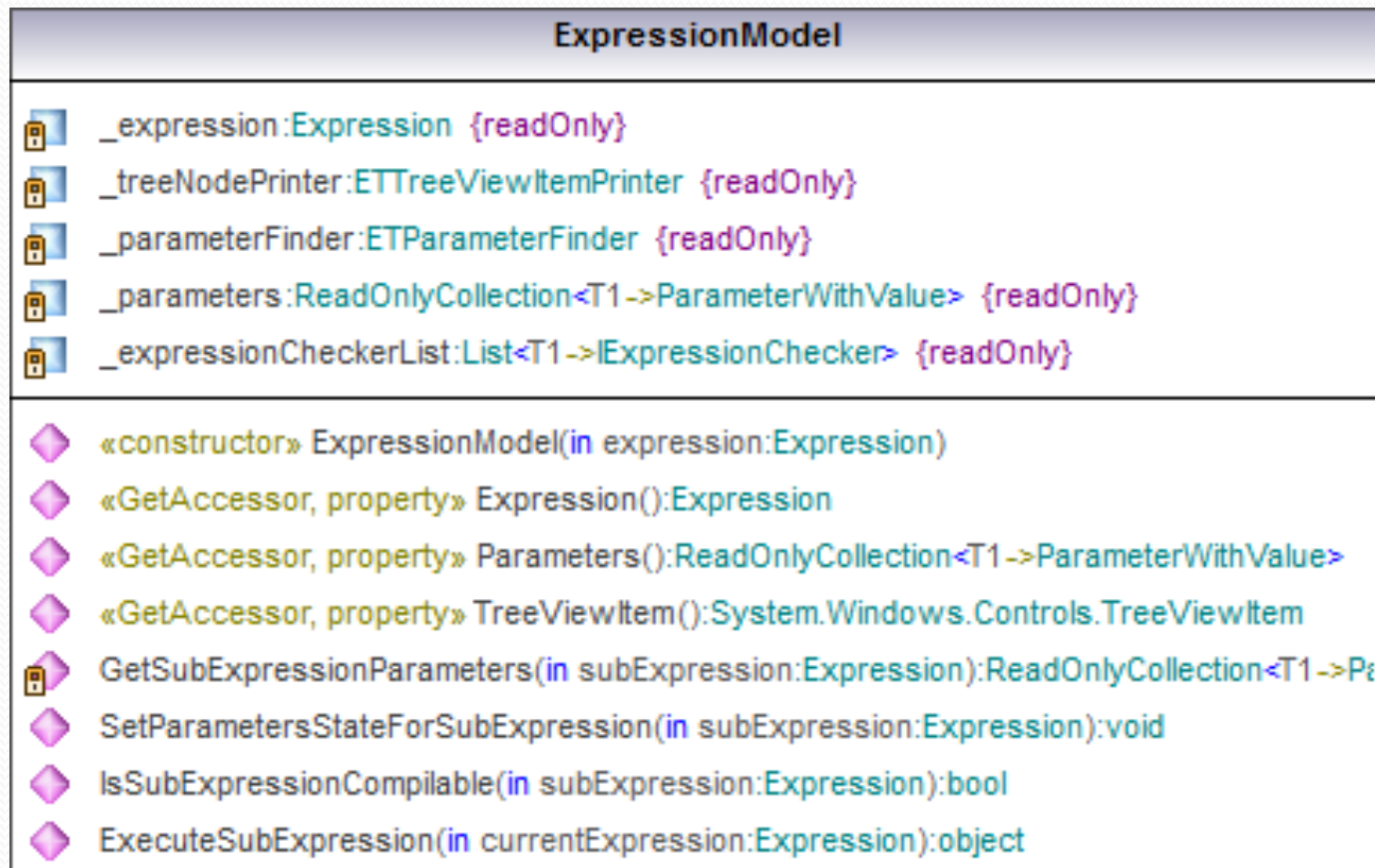
CONTENITORI DI VARIABILI

BlockExpression

CatchBlock

RuntimeVariablesExpression

Modello completo



Progetto grafico

Progetto grafico

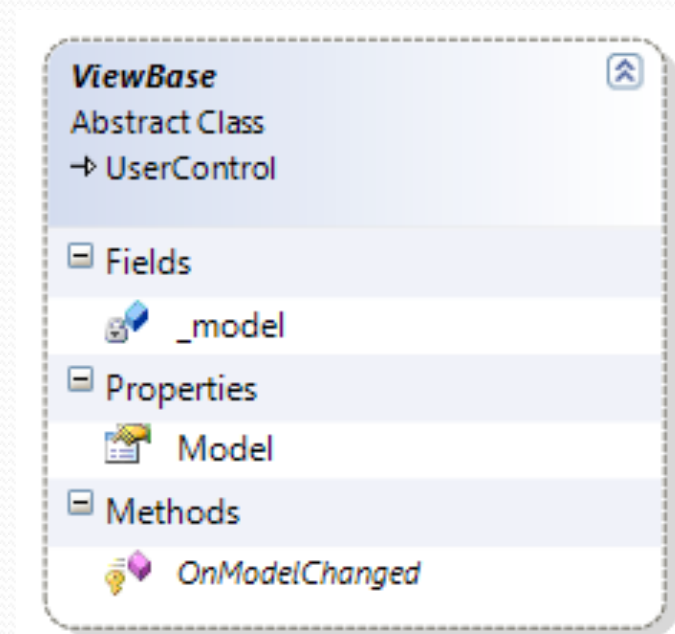
Introduzione

- **Windows Presentation Foundation**

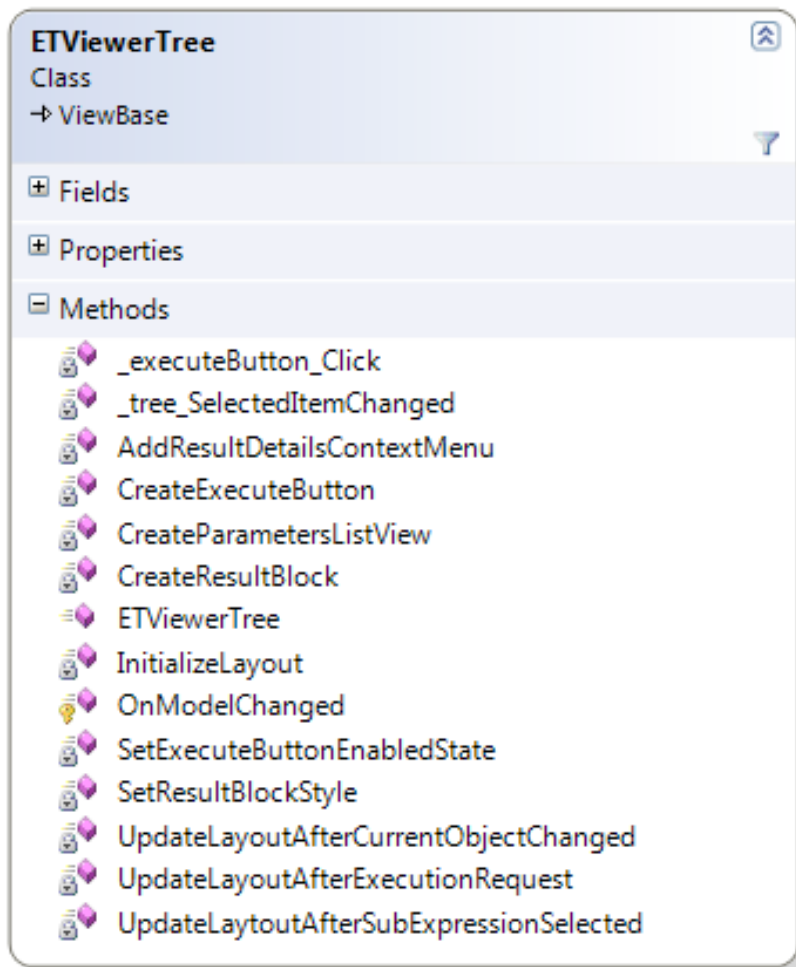
- Libreria .NET per la visualizzazione grafica
- File XAML e code behind
- Stili e trigger
- Data Template
- Data Binding
- Input validation

- **Pattern Document/View**

- Document = Modello dei dati
- View = classe figlia di `ViewBase`



Classe ETViewerTree



- **Componenti**
 - File XAML
 - Code behind
- **Struttura code behind**
 - Stato interno
 - Metodi di modifica dello stile
 - Gestori di eventi
 - Metodi di aggiornamento del layout
 - Metodi factory

Componenti del layout

- **Visualizzazione dell'albero**
 - TreeView con Data Template per diversi tipi di nodo

Parameter (System.Int64)

Constant (System.Int64)

SwitchCase Node

Greater Than Expression

- **Dettagli sui nodi**
 - PropertyGrid di WindowsForm

- **Raccolta e validazione input**

Expression execution

5	a (System.Int32)
d	b (System.Int32)

Input string was not in a correct format.

Node properties

Misc	
Body	(a + (b * 2))
CanReduce	False
Name	
NodeType	Lambda
Parameters	(Collection)
ReturnType	System.Int32
TailCall	False
Type	System.Func`3[System.Int32, Sys

Name

- **Visualizzazione output**

Show result details

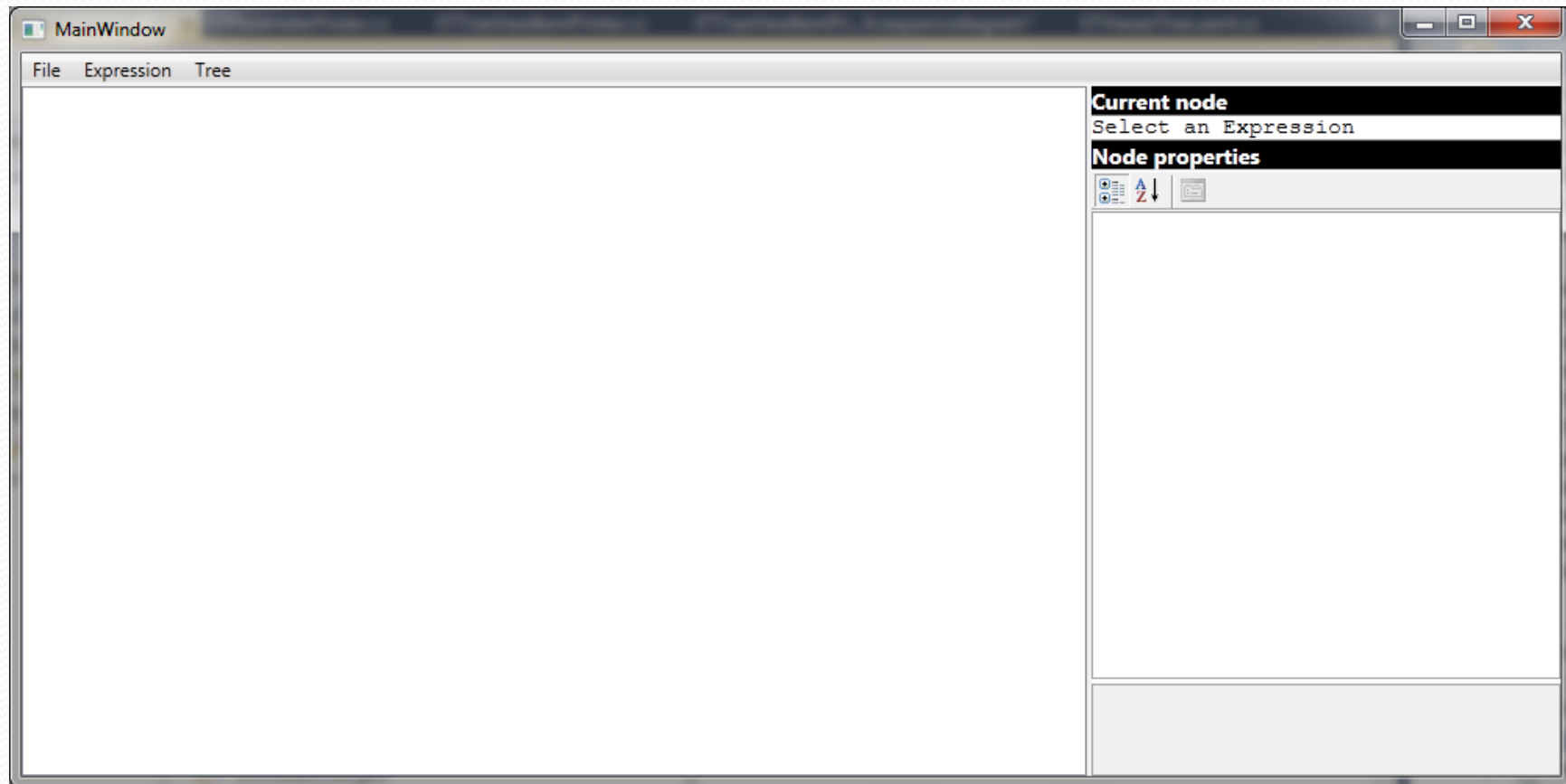
07/12/2010 00:00:00

Execution result
Void method

Execution result
Exception details

Snapshot

Avvio applicazione di prova



Selezione nodo compilabile

The screenshot shows the ET Viewer application window titled "MainWindow". The interface includes a menu bar with "File", "Expression", and "Tree". The main area displays a tree view of expressions:

- Lambda Expression
 - Add Expression
 - Parameter (System.Int32)
 - Multiply Expression
 - Parameter (System.Int32)
 - Constant (System.Int32)
 - Parameter (System.Int32)
 - Parameter (System.Int32)

The right-hand side of the window contains a "Current node" section showing the expression $(a, b) \Rightarrow (a + (b * 2))$. Below it is the "Node properties" section, which includes a "Misc" table:

Misc	
Body	$(a + (b * 2))$
CanReduce	False
Name	
NodeType	Lambda
Parameters	(Collection)
ReturnType	System.Int32
TailCall	False
Type	System.Func`3[System.Int32, Sys

Below the table is a "CanReduce" section with a checkbox. At the bottom right, there is an "Expression execution" section with input fields for "a (System.Int32)" and "b (System.Int32)", and an "Execute" button.

Selezione nodo compilabile – 2

The screenshot shows the ET Viewer application window titled 'MainWindow'. The interface is divided into several sections:

- File Expression Tree**: A menu bar at the top.
- Expression Tree**: A tree view on the left showing a hierarchy of nodes:
 - Lambda Expression** (blue)
 - Add Expression** (blue)
 - Parameter (System.Int32)** (orange)
 - Multiply Expression** (blue)
 - Parameter (System.Int32)** (orange)
 - Constant (System.Int32)** (yellow)
 - Parameter (System.Int32)** (orange)
 - Parameter (System.Int32)** (orange)
- Current node**: A panel on the right showing the selected node: `(b * 2)`.
- Node properties**: A table of properties for the current node:

Misc	
CanReduce	False
Conversion	
IsLifted	False
IsLiftedToNull	False
Left	b
Method	
NodeType	Multiply
Right	2
Type	System.Int32
- CanReduce**: A section below the properties table.
- Expression execution**: A section at the bottom right, highlighted with a red and yellow border, containing a table of variables and an 'Execute' button:

Expression execution	
	a (System.Int32)
	b (System.Int32)

Execute

Selezione nodo non compilabile

The screenshot shows the ET Viewer interface. The tree view on the left contains the following nodes:

- Lambda Expression
 - Add Expression
 - Parameter (System.Int32)
 - Multiply Expression
 - Parameter (System.Int32)
 - Constant (System.Int32)
 - Parameter (System.Int32)
 - Parameter (System.Int32)

The right-hand pane displays the properties of the selected node:

Current node
b

Node properties

Misc	
CanReduce	False
IsByRef	False
Name	b
NodeType	Parameter
Type	System.Int32

CanReduce

Esecuzione espressione

The screenshot displays the ET Viewer application window titled 'MainWindow'. The interface is divided into several sections:

- Tree View:** Shows a hierarchical structure of expressions:
 - Lambda Expression
 - Add Expression
 - Parameter (System.Int32)
 - Multiply Expression
 - Parameter (System.Int32)
 - Constant (System.Int32)
 - Parameter (System.Int32)
 - Parameter (System.Int32)
- Current node:** $(a, b) \Rightarrow (a + (b * 2))$
- Node properties:** A table with the following data:

Misc	
Body	$(a + (b * 2))$
CanReduce	False
Name	
NodeType	Lambda
Parameters	(Collection)
ReturnType	System.Int32
TailCall	False
Type	System.Func`3[System.Int32, ...]
- Expression execution:** A table showing the execution of parameters:

Step	Value
4	a (System.Int32)
3	b (System.Int32)
- Execute:** A button to trigger the execution.
- Execution result:** The result of the execution, which is 10.

Risultato complesso

The screenshot displays the ET Viewer application interface. The main window, titled 'MainWindow', contains a menu bar with 'File', 'Expression', and 'Tree'. The central area shows a tree view of expressions:

- Lambda Expression
 - MemberInit Expression
 - New Expression
 - MemberAssignment Node
 - Parameter (System.Int32)

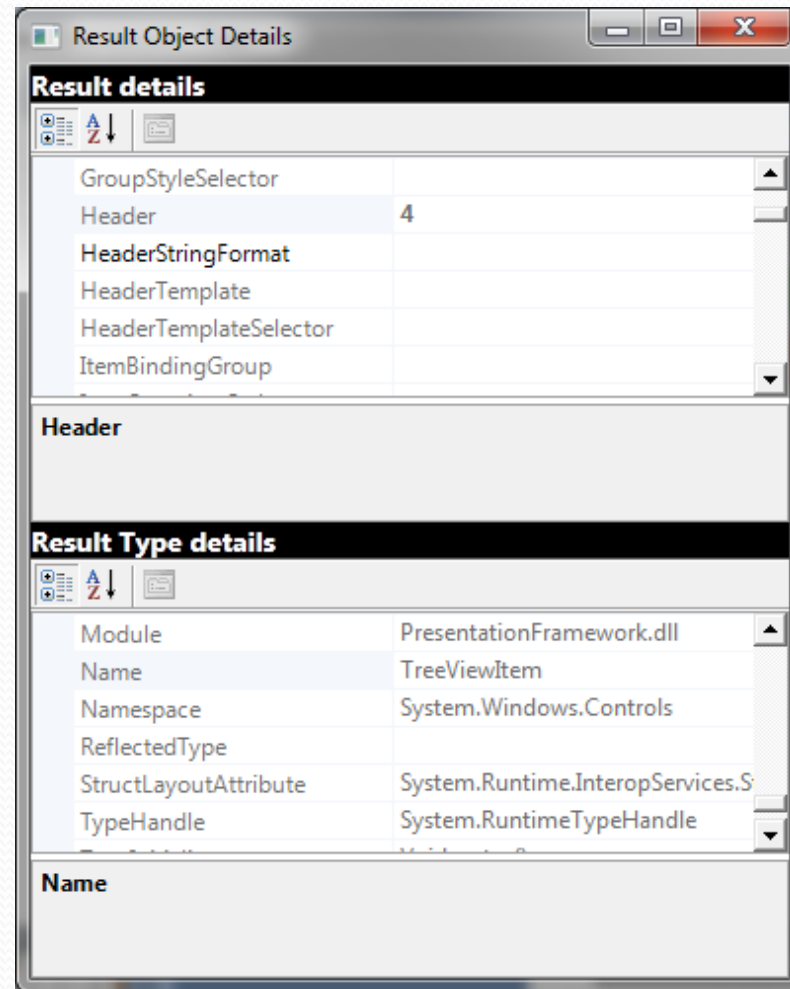
The 'Parameter (System.Int32)' node is highlighted in red. The right-hand side of the window features several panels:

- Current node:** `a => new TreeViewItem() {Header = Co`
- Node properties:** A table listing properties for the selected node:

Misc	
Body	new TreeViewItem() {Header = Co
CanReduce	False
Name	
NodeType	Lambda
Parameters	(Collection)
ReturnType	System.Windows.Controls.T
TailCall	False
- Name:** A text field.
- Expression execution:** A table showing the execution of the expression:

Index	Value
4	a (System.Int32)
- Execution:** A table with a 'Show result details' button highlighted by a red box.

Dettagli risultato complesso



**Grazie
per
l'attenzione**