

# GraphFramework

Erzeugt von Doxygen 1.8.9.1

Die Jun 30 2015 08:34:18



# Inhaltsverzeichnis

<b>1</b>	<b>Verzeichnis der Namensbereiche</b>	<b>1</b>
1.1	Pakete . . . . .	1
<b>2</b>	<b>Hierarchie-Verzeichnis</b>	<b>3</b>
2.1	Klassenhierarchie . . . . .	3
<b>3</b>	<b>Klassen-Verzeichnis</b>	<b>5</b>
3.1	Auflistung der Klassen . . . . .	5
<b>4</b>	<b>Datei-Verzeichnis</b>	<b>7</b>
4.1	Auflistung der Dateien . . . . .	7
<b>5</b>	<b>Dokumentation der Namensbereiche</b>	<b>9</b>
5.1	Paket GraphFramework . . . . .	9
5.2	Paket GraphFramework.Interfaces . . . . .	9
<b>6</b>	<b>Klassen-Dokumentation</b>	<b>11</b>
6.1	GraphFramework.Attribute Klassenreferenz . . . . .	11
6.1.1	Ausführliche Beschreibung . . . . .	12
6.1.2	Beschreibung der Konstruktoren und Destruktoren . . . . .	12
6.1.2.1	Attribute . . . . .	12
6.1.3	Dokumentation der Propertys . . . . .	12
6.1.3.1	Name . . . . .	12
6.1.3.2	Type . . . . .	12
6.1.3.3	Value . . . . .	12
6.2	GraphFramework.Edge Klassenreferenz . . . . .	12
6.2.1	Ausführliche Beschreibung . . . . .	14
6.2.2	Beschreibung der Konstruktoren und Destruktoren . . . . .	14
6.2.2.1	Edge . . . . .	14
6.2.3	Dokumentation der Propertys . . . . .	14
6.2.3.1	Node1 . . . . .	14
6.2.3.2	Node2 . . . . .	14
6.3	GraphFramework.Graph Klassenreferenz . . . . .	14

6.3.1	Ausführliche Beschreibung	16
6.3.2	Beschreibung der Konstruktoren und Destruktoren	16
6.3.2.1	Graph	16
6.3.3	Dokumentation der Elementfunktionen	17
6.3.3.1	AddEdge	17
6.3.3.2	AddEdge	18
6.3.3.3	AddGraph	18
6.3.3.4	AddGraph	19
6.3.3.5	AddNode	19
6.3.3.6	GetEdges	20
6.3.3.7	GetSingleNodes	20
6.3.3.8	MergeGraphs	20
6.3.3.9	MergeGraphs	21
6.3.3.10	RemoveEdge	22
6.3.3.11	RemoveNode	22
6.3.4	Dokumentation der Propertyys	23
6.3.4.1	Edges	23
6.3.4.2	Nodes	23
6.4	GraphFramework.GraphElement Klassenreferenz	23
6.4.1	Ausführliche Beschreibung	24
6.4.2	Beschreibung der Konstruktoren und Destruktoren	24
6.4.2.1	GraphElement	24
6.4.3	Dokumentation der Elementfunktionen	24
6.4.3.1	AddAttribute	24
6.4.3.2	GetAttributeByName	24
6.4.3.3	RemoveAttribute	25
6.4.3.4	RemoveAttribute	25
6.4.4	Dokumentation der Datenelemente	25
6.4.4.1	mAttributes	25
6.4.5	Dokumentation der Propertyys	25
6.4.5.1	Attributes	25
6.5	GraphFramework.Interfaces.IAttribute Schnittstellenreferenz	25
6.5.1	Ausführliche Beschreibung	26
6.5.2	Dokumentation der Propertyys	26
6.5.2.1	Name	26
6.5.2.2	Type	26
6.5.2.3	Value	26
6.6	GraphFramework.Interfaces.IEdge Schnittstellenreferenz	27
6.6.1	Ausführliche Beschreibung	28
6.6.2	Dokumentation der Propertyys	28

6.6.2.1	Node1	28
6.6.2.2	Node2	28
6.7	GraphFramework.Interfaces.IGraph Schnittstellenreferenz	28
6.7.1	Ausführliche Beschreibung	29
6.7.2	Dokumentation der Elementfunktionen	30
6.7.2.1	AddEdge	30
6.7.2.2	AddEdge	31
6.7.2.3	AddGraph	31
6.7.2.4	AddGraph	31
6.7.2.5	AddNode	31
6.7.2.6	GetEdges	32
6.7.2.7	GetSingleNodes	32
6.7.2.8	RemoveEdge	32
6.7.2.9	RemoveNode	32
6.7.3	Dokumentation der Propertyys	32
6.7.3.1	Edges	32
6.7.3.2	Nodes	33
6.8	GraphFramework.Interfaces.IGraphElement Schnittstellenreferenz	33
6.8.1	Ausführliche Beschreibung	33
6.8.2	Dokumentation der Elementfunktionen	34
6.8.2.1	AddAttribute	34
6.8.2.2	GetAttributeByName	35
6.8.2.3	RemoveAttribute	35
6.8.2.4	RemoveAttribute	35
6.8.3	Dokumentation der Propertyys	35
6.8.3.1	Attributes	35
6.9	GraphFramework.Interfaces.INode Schnittstellenreferenz	35
6.9.1	Ausführliche Beschreibung	36
6.10	GraphFramework.Node Klassenreferenz	36
6.10.1	Ausführliche Beschreibung	37
6.10.2	Beschreibung der Konstruktoren und Destruktoren	38
6.10.2.1	Node	38
<b>7</b>	<b>Datei-Dokumentation</b>	<b>39</b>
7.1	Attribute.cs-Dateireferenz	39
7.2	Edge.cs-Dateireferenz	39
7.3	Graph.cs-Dateireferenz	39
7.4	GraphElement.cs-Dateireferenz	40
7.5	Interfaces/IAttribute.cs-Dateireferenz	40
7.6	Interfaces/IEdge.cs-Dateireferenz	40

7.7	<a href="#">Interfaces/IGraph.cs-Dateireferenz</a>	40
7.8	<a href="#">Interfaces/IGraphElement.cs-Dateireferenz</a>	41
7.9	<a href="#">Interfaces/INode.cs-Dateireferenz</a>	41
7.10	<a href="#">Node.cs-Dateireferenz</a>	41
7.11	<a href="#">Properties/AssemblyInfo.cs-Dateireferenz</a>	41
<b>Index</b>		<b>43</b>

# Kapitel 1

## Verzeichnis der Namensbereiche

### 1.1 Pakete

Hier folgen die Pakete mit einer Kurzbeschreibung (wenn verfügbar):

<a href="#">GraphFramework</a> . . . . .	9
<a href="#">GraphFramework.Interfaces</a> . . . . .	9





## Kapitel 2

# Hierarchie-Verzeichnis

### 2.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

GraphFramework.Interfaces.IAttribute . . . . .	25
GraphFramework.Attribute . . . . .	11
GraphFramework.Interfaces.IGraphElement . . . . .	33
GraphFramework.GraphElement . . . . .	23
GraphFramework.Edge . . . . .	12
GraphFramework.Graph . . . . .	14
GraphFramework.Node . . . . .	36
GraphFramework.Interfaces.IEdge . . . . .	27
GraphFramework.Edge . . . . .	12
GraphFramework.Interfaces.IGraph . . . . .	28
GraphFramework.Graph . . . . .	14
GraphFramework.Interfaces.INode . . . . .	35
GraphFramework.Node . . . . .	36



## Kapitel 3

# Klassen-Verzeichnis

### 3.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

<a href="#">GraphFramework.Attribute</a>	
The attribute class. A attribute is a holder of a generic value which has a name . . . . .	11
<a href="#">GraphFramework.Edge</a>	
The edge class . . . . .	12
<a href="#">GraphFramework.Graph</a>	
the graph class: holder of edges and nodes + methods to modify the graph . . . . .	14
<a href="#">GraphFramework.GraphElement</a>	
the abstract graphelement class nearly every framework class inherits from. The class generalize the relation between classes that can have attributes and the attribute class . . . . .	23
<a href="#">GraphFramework.Interfaces.IAttribute</a>	
An interface for classes that contain a value that has a name. Objects of classes that implement this interface can be added to objects of classes that implement the interface <a href="#">IGraphElement</a> . .	25
<a href="#">GraphFramework.Interfaces.IEdge</a>	
An interface for classes that are able to connect two objects of a class that implements the interface <a href="#">INode</a> . Objects of classes that implement this interface can be added to objects of classes that implement the interface <a href="#">IGraph</a> . . . . .	27
<a href="#">GraphFramework.Interfaces.IGraph</a>	
Classes that implement this interface are a representation of a set of objects that implement the interfaces <a href="#">INode</a> and <a href="#">IEdge</a> and the relations between them. It is able to perform actions on this objects. . . . .	28
<a href="#">GraphFramework.Interfaces.IGraphElement</a>	
An interface for classes that are able to hold objects of classes that implement the interface <a href="#">IAttribute</a> . The interfaces <a href="#">INode</a> , <a href="#">IEdge</a> and <a href="#">IGraph</a> inherit from this interface. . . . .	33
<a href="#">GraphFramework.Interfaces.INode</a>	
An interface for a class that is able to hold objects of a class that implements the interface <a href="#">IAttribute</a> . Objects of classes that implement this interface can be added to objects of classes that implement the interface <a href="#">IGraph</a> . . . . .	35
<a href="#">GraphFramework.Node</a>	
<a href="#">Node</a> class . . . . .	36



# Kapitel 4

## Datei-Verzeichnis

### 4.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

<a href="#">Attribute.cs</a>	39
<a href="#">Edge.cs</a>	39
<a href="#">Graph.cs</a>	39
<a href="#">GraphElement.cs</a>	40
<a href="#">Node.cs</a>	41
Interfaces/ <a href="#">IAttribute.cs</a>	40
Interfaces/ <a href="#">IEdge.cs</a>	40
Interfaces/ <a href="#">IGraph.cs</a>	40
Interfaces/ <a href="#">IGraphElement.cs</a>	41
Interfaces/ <a href="#">INode.cs</a>	41
Properties/ <a href="#">AssemblyInfo.cs</a>	41



## Kapitel 5

# Dokumentation der Namensbereiche

### 5.1 Paket GraphFramework

#### Namensbereiche

- package [Interfaces](#)

#### Klassen

- class [Attribute](#)  
*The attribute class. A attribute is a holder of a generic value which has a name*
- class [Edge](#)  
*The edge class*
- class [Graph](#)  
*the graph class: holder of edges and nodes + methods to modify the graph*
- class [GraphElement](#)  
*the abstract graphelement class nearly every framework class inherits from. The class generalize the relation between classes that can have attributes and the attribute class*
- class [Node](#)  
*Node class*

### 5.2 Paket GraphFramework.Interfaces

#### Klassen

- interface [IAttribute](#)  
*An interface for classes that contain a value that has a name. Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraphElement](#).*
- interface [IEdge](#)  
*An interface for classes that are able to connect two objects of a class that implements the interface [INode](#). Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraph](#).*
- interface [IGraph](#)  
*Classes that implement this interface are a representation of a set of objects that implement the interfaces [INode](#) and [IEdge](#) and the relations between them. It is able to perform actions on this objects.*
- interface [IGraphElement](#)  
*An interface for classes that are able to hold objects of classes that implement the interface [IAttribute](#). The interfaces [INode](#), [IEdge](#) and [IGraph](#) inherit from this interface.*

- interface [INode](#)

*An interface for a class that is able to hold objects of a class that implements the interface [IAttribute](#). Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraph](#).*

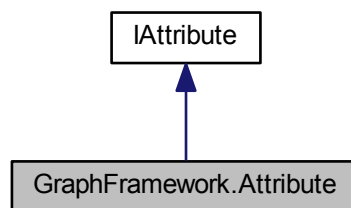


## Kapitel 6

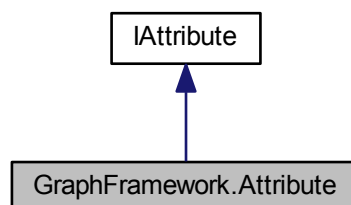
# Klassen-Dokumentation

### 6.1 GraphFramework.Attribute Klassenreferenz

The attribute class. A attribute is a holder of a generic value which has a name  
Klassendiagramm für GraphFramework.Attribute:



Zusammengehörigkeiten von GraphFramework.Attribute:



### Öffentliche Methoden

- [Attribute](#) (string name, object val)

*attribute constructor*

## Propertys

- object [Value](#) [get, set]  
*Gets and sets the value of the attribute*
- Type [Type](#) [get]  
*Gets the type of the attribute*
- string [Name](#) [get]  
*Gets the name of the attribute*

### 6.1.1 Ausführliche Beschreibung

The attribute class. A attribute is a holder of a generic value which has a name

### 6.1.2 Beschreibung der Konstruktoren und Destruktoren

#### 6.1.2.1 GraphFramework.Attribute.Attribute ( string name, object val )

attribute constructor

Parameter

<i>name</i>	the name
<i>val</i>	the generic value

### 6.1.3 Dokumentation der Propertys

#### 6.1.3.1 string GraphFramework.Attribute.Name [get]

Gets the name of the attribute

#### 6.1.3.2 Type GraphFramework.Attribute.Type [get]

Gets the type of the attribute

#### 6.1.3.3 object GraphFramework.Attribute.Value [get], [set]

Gets and sets the value of the attribute

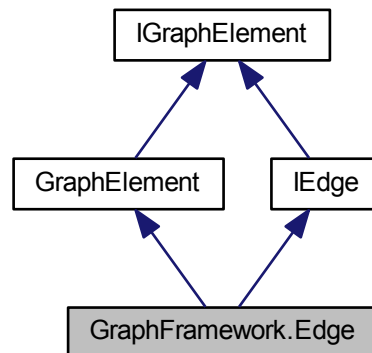
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Attribute.cs](#)

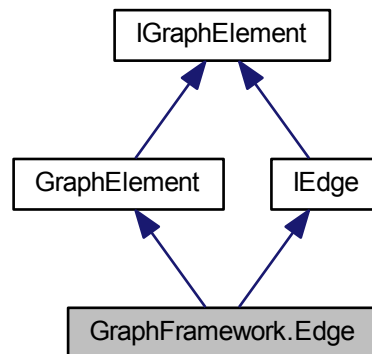
## 6.2 GraphFramework.Edge Klassenreferenz

The edge class

Klassendiagramm für GraphFramework.Edge:



Zusammengehörigkeiten von GraphFramework.Edge:



### Paketfunktionen

- `Edge (INode node1, INode node2, params IAttribute[] attributes)`  
*Add also attributes to the edge*

### Propertys

- `INode Node1` [get]  
*The first node the edge connects*
- `INode Node2` [get]  
*The second node the edge connects*

## Weitere Geerbte Elemente

### 6.2.1 Ausführliche Beschreibung

The edge class

### 6.2.2 Beschreibung der Konstruktoren und Destruktoren

#### 6.2.2.1 GraphFramework.Edge.Edge ( INode *node1*, INode *node2*, params IAttribute[] *attributes* ) [package]

Add also attributes to the edge

Parameter

<i>node1</i>	first node
<i>node2</i>	second node
<i>attributes</i>	

### 6.2.3 Dokumentation der Property's

#### 6.2.3.1 INode GraphFramework.Edge.Node1 [get]

The first node the edge connects

#### 6.2.3.2 INode GraphFramework.Edge.Node2 [get]

The second node the edge connects

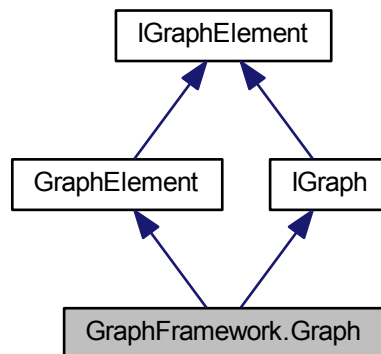
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Edge.cs](#)

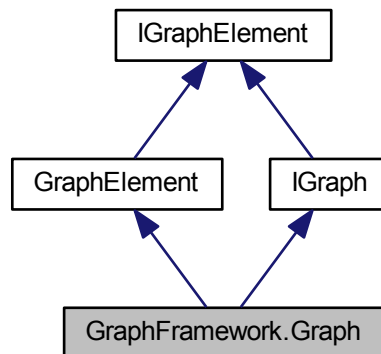
## 6.3 GraphFramework.Graph Klassenreferenz

the graph class: holder of edges and nodes + methods to modify the graph

Klassendiagramm für GraphFramework.Graph:



Zusammengehörigkeiten von GraphFramework.Graph:



## Öffentliche Methoden

- `Graph` (params `IAttribute[]` attributes)  
*creates a new graph with the given parameters*
- void `AddNode` (`INode` node)  
*adds the node to the node collection of the graph*
- void `RemoveNode` (params `INode[]` nodesToRemove)  
*removes the nodes identified by its reference from the graph*
- void `AddEdge` (`INode` node1, `INode` node2, params `IAttribute[]` attributes)  
*adds the edge to the edge collection of the graph*
- void `AddEdge` (`IEdge` edge)  
*adds the edge to the edge collection of the graph*

- void **RemoveEdge** (params **IEdge**[] edgesToRemove)  
*removes the edge identified by its reference from the graph*
- IEnumerable< **INode** > **GetSingleNodes** ()  
*returns all Nodes that are not connected by Edges*
- void **AddGraph** (**IGraph** graphToAdd)  
*adds one **Graph** object to this graph*
- void **AddGraph** (**IGraph** graphToAdd, **INode** thisGraphConnectionNode, **INode** graphToAddConnectionNode, params **IAttribute**[] attributes)  
*adds one **Graph** object to this graph and connects it with an edge*
- IEnumerable< **IEdge** > **GetEdges** (**INode** node1, **INode** node2)  
*gets the edges which are connected to the nodes node1 and node2*

## Öffentliche, statische Methoden

- static **IGraph** **MergeGraphs** (**IGraph** graph1, **IGraph** graph2)  
*merges the two graphs to one new*
- static **IGraph** **MergeGraphs** (**IGraph** graph1, **IGraph** graph2, **INode** graphOneConnectionNode, **INode** graphTwoConnectionNode, params **IAttribute**[] attributes)  
*merges the two graphs to one new and connects the two given nodes*

## Property

- IEnumerable< **INode** > **Nodes** [get]  
*contains all nodes*
- IEnumerable< **IEdge** > **Edges** [get]  
*contains all edges*

## Weitere Geerbte Elemente

### 6.3.1 Ausführliche Beschreibung

the graph class: holder of edges and nodes + methods to modify the graph

### 6.3.2 Beschreibung der Konstruktoren und Destruktoren

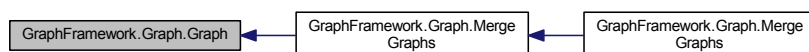
#### 6.3.2.1 GraphFramework.Graph.Graph ( params **IAttribute**[] attributes )

creates a new graph with the given parameters

Parameter

<i>attributes</i>	attributes to add to the graph
-------------------	--------------------------------

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



### 6.3.3 Dokumentation der Elementfunktionen

6.3.3.1 void GraphFramework.Graph.AddEdge ( INode *node1*, INode *node2*, params IAttribute[] *attributes* )

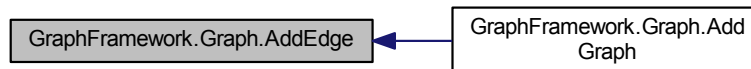
adds the edge to the edge collection of the graph

## Parameter

<i>node1</i>	first node to which the edge is connected
<i>node2</i>	second node to which the edge is connected
<i>attributes</i>	attributes of the edge

Implementiert [GraphFramework.Interfaces.IGraph](#).

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



### 6.3.3.2 void GraphFramework.Graph.AddEdge ( IEdge edge )

adds the edge to the edge collection of the graph

## Parameter

<i>edge</i>	edge to add to graph
-------------	----------------------

Implementiert [GraphFramework.Interfaces.IGraph](#).

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



### 6.3.3.3 void GraphFramework.Graph.AddGraph ( IGraph graphToAdd )

adds one [Graph](#) object to this graph

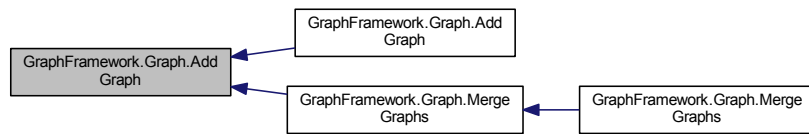
## Parameter

<i>graphToAdd</i>	<a href="#">Graph</a> object that will be added to this
-------------------	---

Implementiert [GraphFramework.Interfaces.IGraph](#).



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



**6.3.3.4** `void GraphFramework.Graph.AddGraph ( IGraph graphToAdd, INode thisGraphConnectionNode, INode graphToAddConnectionNode, params IAttribute[] attributes )`

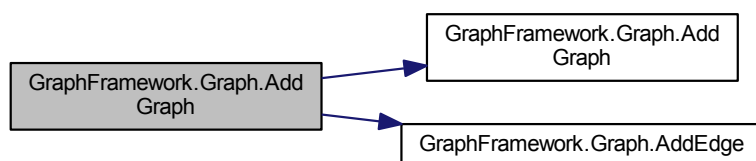
adds one [Graph](#) object to this graph and connects it with an edge

Parameter

<i>graphToAdd</i>	<a href="#">Graph</a> object that will be added to this
<i>thisGraphConnectionNode</i>	first <a href="#">Node</a> for new <a href="#">Edge</a> . Contained in graph 1.
<i>graphToAddConnectionNode</i>	second <a href="#">Node</a> for new <a href="#">Edge</a> . Contained in graph 2.
<i>attributes</i>	Attributes of new <a href="#">Edge</a>

Implementiert [GraphFramework.Interfaces.IGraph](#).

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



**6.3.3.5** `void GraphFramework.Graph.AddNode ( INode node )`

adds the node to the node collection of the graph

Parameter

<i>node</i>	reference of the node to add
-------------	------------------------------

Implementiert [GraphFramework.Interfaces.IGraph](#).

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



#### 6.3.3.6 IEnumerable<IEdge> GraphFramework.Graph.GetEdges ( INode *node1*, INode *node2* )

gets the edges which are connected to the nodes *node1* and *node2*

Parameter

<i>node1</i>	first node to which the edge has to be connected
<i>node2</i>	second node to which the edge has to be connected

Rückgabe

all the edges which connect the nodes *node1* and *node2*

Implementiert [GraphFramework.Interfaces.IGraph](#).

#### 6.3.3.7 IEnumerable<INode> GraphFramework.Graph.GetSingleNodes ( )

returns all Nodes that are not connected by Edges

Rückgabe

returns all single Nodes

Implementiert [GraphFramework.Interfaces.IGraph](#).

#### 6.3.3.8 static IGraph GraphFramework.Graph.MergeGraphs ( IGraph *graph1*, IGraph *graph2* ) [static]

merges the two graphs to one new

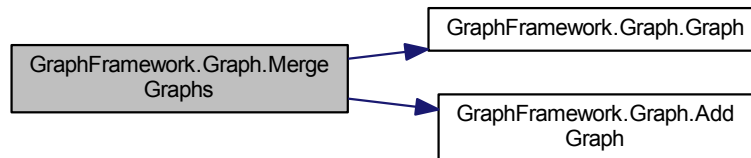
Parameter

<i>graph1</i>	first graph
<i>graph2</i>	second graph

**Rückgabe**

the new graph, which contains the given two

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



**6.3.3.9** `static IGraph GraphFramework.Graph.MergeGraphs ( IGraph graph1, IGraph graph2, INode graphOneConnectionNode, INode graphTwoConnectionNode, params IAttribute[] attributes ) [static]`

merges the two graphs to one new and connects the two given nodes

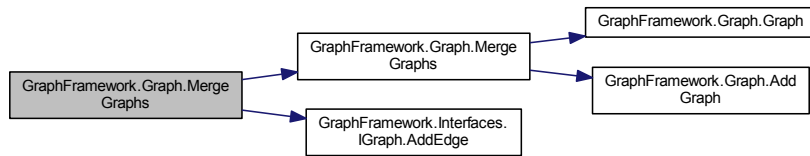
**Parameter**

<i>graph1</i>	first graph
<i>graph2</i>	second graph
<i>graphOne</i> ↔ <i>ConnectionNode</i>	node in graph one to connect with node in graph two
<i>graphTwo</i> ↔ <i>ConnectionNode</i>	node in graph two to connect with node in graph one
<i>attributes</i>	attributes to add to the edge between <i>graphOneConnectionNode</i> and <i>graphTwoConnectionNode</i> ↔ Node

**Rückgabe**

the new graph, which contains the given two and a edge between the given nodes

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



### 6.3.3.10 void GraphFramework.Graph.RemoveEdge ( params IEdge[] edgesToRemove )

removes the edge identified by its reference from the graph

**Parameter**

<i>edgesToRemove</i>	reference of the edges to remove
----------------------	----------------------------------

Implementiert [GraphFramework.Interfaces.IGraph](#).

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



### 6.3.3.11 void GraphFramework.Graph.RemoveNode ( params INode[] nodesToRemove )

removes the nodes identified by its reference from the graph

**Parameter**

<i>nodesToRemove</i>	reference of the nodes to remove
----------------------	----------------------------------

Implementiert [GraphFramework.Interfaces.IGraph](#).

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



### 6.3.4 Dokumentation der Propertys

#### 6.3.4.1 IEnumerable<IEdge> GraphFramework.Graph.Edges [get]

contains all edges

#### 6.3.4.2 IEnumerable<INode> GraphFramework.Graph.Nodes [get]

contains all nodes

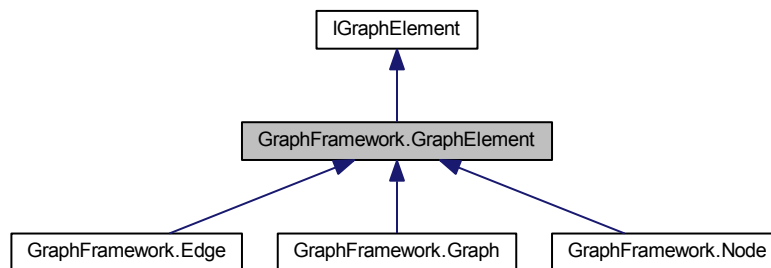
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Graph.cs](#)

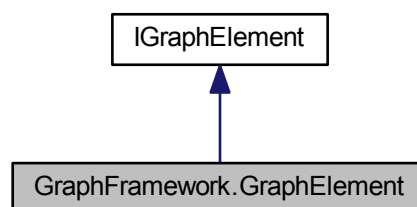
## 6.4 GraphFramework.GraphElement Klassenreferenz

the abstract graphelement class nearly every framework class inherits from. The class generalize the relation between classes that can have attributes and the attribute class

Klassendiagramm für GraphFramework.GraphElement:



Zusammengehörigkeiten von GraphFramework.GraphElement:



### Öffentliche Methoden

- void [AddAttribute](#) (IAttribute attribute)

- Adds an attribute to the node. If the element already has an attribute with this name, it is not added*
  - void [RemoveAttribute](#) (string nameOfAttributeToRemove)
  - Removes an attribute with the defined name from the element. (if the element has an attribute with this name)*
  - void [RemoveAttribute](#) ([IAttribute](#) attributeToRemove)
  - Removes the defined attribute if the element holds it*
  - [IAttribute](#) [GetAttributeByName](#) (string name)
  - Returns the attribute with the given name. Warning: can be also null*

## Geschützte Methoden

- [GraphElement](#) (params [IAttribute](#)[] attributes)
  - Special constructor which adds multiple attributes to the element*

## Geschützte Attribute

- List< [IAttribute](#) > [mAttributes](#)
  - The attributes of the element, sorted by their names*

## Propertys

- IEnumerable< [IAttribute](#) > [Attributes](#) [get]
  - Gets the element's attributes as a collection*

### 6.4.1 Ausführliche Beschreibung

the abstract graphelement class nearly every framework class inherits from. The class generalize the relation between classes that can have attributes and the attribute class

### 6.4.2 Beschreibung der Konstruktoren und Destruktoren

#### 6.4.2.1 [GraphFramework.GraphElement.GraphElement](#) ( params [IAttribute](#)[] *attributes* ) [protected]

Special constructor which adds multiple attributes to the element

Parameter

<i>attributes</i>	the attributes you want to add
-------------------	--------------------------------

### 6.4.3 Dokumentation der Elementfunktionen

#### 6.4.3.1 void [GraphFramework.GraphElement.AddAttribute](#) ( [IAttribute](#) *attribute* )

Adds an attribute to the node. If the element already has an attribute with this name, it is not added

Parameter

<i>attribute</i>	the attribute you want to add
------------------	-------------------------------

Implementiert [GraphFramework.Interfaces.IGraphElement](#).

#### 6.4.3.2 [IAttribute](#) [GraphFramework.GraphElement.GetAttributeByName](#) ( string *name* )

Returns the attribute with the given name. Warning: can be also null

## Parameter

<i>name</i>	
-------------	--

## Rückgabe

null or the attribute with the given name

Implementiert [GraphFramework.Interfaces.IGraphElement](#).

#### 6.4.3.3 void GraphFramework.GraphElement.RemoveAttribute ( string *nameOfAttributeToRemove* )

Removes an attribute with the defined name from the element. (if the element has an attribute with this name)

## Parameter

<i>nameOfAttributeToRemove</i>	the name of the attribute you want to remove
--------------------------------	--

Implementiert [GraphFramework.Interfaces.IGraphElement](#).

#### 6.4.3.4 void GraphFramework.GraphElement.RemoveAttribute ( IAttribute *attributeToRemove* )

Removes the defined attribute if the element holds it

## Parameter

<i>attributeToRemove</i>	
--------------------------	--

Implementiert [GraphFramework.Interfaces.IGraphElement](#).

### 6.4.4 Dokumentation der Datenelemente

#### 6.4.4.1 List<IAttribute> GraphFramework.GraphElement.mAttributes [protected]

The attributes of the element, sorted by their names

### 6.4.5 Dokumentation der Propertys

#### 6.4.5.1 IEnumerable<IAttribute> GraphFramework.GraphElement.Attributes [get]

Gets the element's attributes as a collection

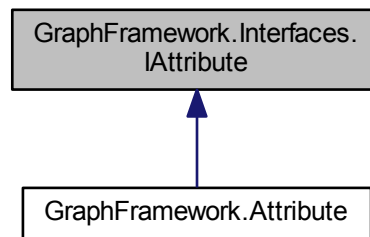
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [GraphElement.cs](#)

## 6.5 GraphFramework.Interfaces.IAttribute Schnittstellenreferenz

An interface for classes that contain a value that has a name. Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraphElement](#).

Klassendiagramm für GraphFramework.Interfaces.IAttribute:



## Propertys

- object `Value` [get, set]  
*Value of the [IAttribute](#).*
- Type `Type` [get]  
*Type of the property Value of the [IAttribute](#).*
- string `Name` [get]  
*Name to identify the [IAttribute](#).*

### 6.5.1 Ausführliche Beschreibung

An interface for classes that contain a value that has a name. Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraphElement](#).

### 6.5.2 Dokumentation der Propertys

#### 6.5.2.1 string GraphFramework.Interfaces.IAttribute.Name [get]

Name to identify the [IAttribute](#).

#### 6.5.2.2 Type GraphFramework.Interfaces.IAttribute.Type [get]

Type of the property Value of the [IAttribute](#).

#### 6.5.2.3 object GraphFramework.Interfaces.IAttribute.Value [get], [set]

Value of the [IAttribute](#).

Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

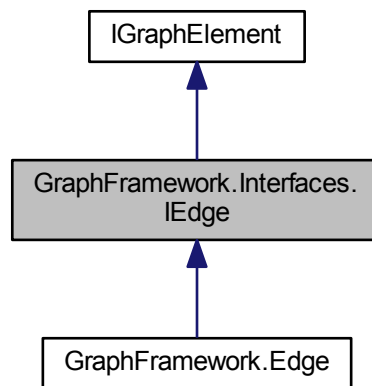
- [Interfaces/IAttribute.cs](#)



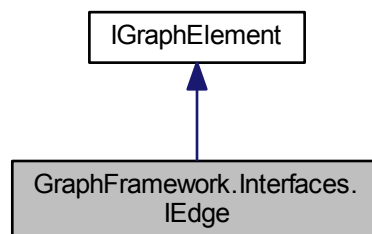
## 6.6 GraphFramework.Interfaces.IEdge Schnittstellenreferenz

An interface for classes that are able to connect two objects of a class that implements the interface [INode](#). Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraph](#).

Klassendiagramm für GraphFramework.Interfaces.IEdge:



Zusammengehörigkeiten von GraphFramework.Interfaces.IEdge:



### Propertys

- [INode Node1](#) [get]  
*The first [INode](#) to which the [IEdge](#) is connected.*
- [INode Node2](#) [get]  
*The second [INode](#) to which the [IEdge](#) is connected.*

### Weitere Geerbte Elemente

### 6.6.1 Ausführliche Beschreibung

An interface for classes that are able to connect two objects of a class that implements the interface [INode](#). Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraph](#).

### 6.6.2 Dokumentation der Propertys

#### 6.6.2.1 [INode](#) `GraphFramework.Interfaces.Edge.Node1` [get]

The first [INode](#) to which the [IEdge](#) is connected.

#### 6.6.2.2 [INode](#) `GraphFramework.Interfaces.Edge.Node2` [get]

The second [INode](#) to which the [IEdge](#) is connected.

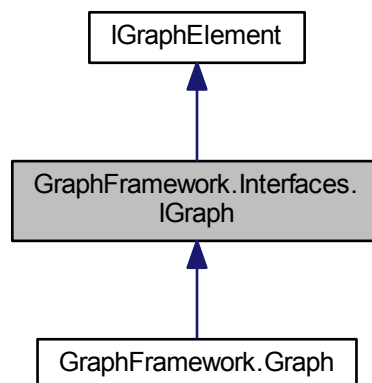
Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

- [Interfaces/IEdge.cs](#)

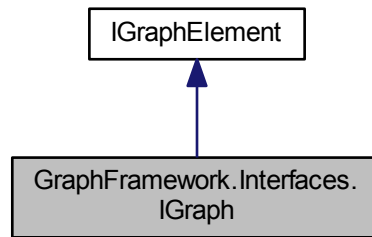
## 6.7 `GraphFramework.Interfaces.IGraph` Schnittstellenreferenz

Classes that implement this interface are a representation of a set of objects that implement the interfaces [INode](#) and [IEdge](#) and the relations between them. It is able to perform actions on this objects.

Klassendiagramm für `GraphFramework.Interfaces.IGraph`:



Zusammengehörigkeiten von GraphFramework.Interfaces.IGraph:



## Öffentliche Methoden

- void **AddNode** (INode node)  
*Adds the INode to property Nodes of the IGraph.*
- void **RemoveNode** (params INode[] nodesToRemove)  
*Removes the INode from the property Nodes of the IGraph.*
- void **AddEdge** (INode node1, INode node2, params IAttribute[] attributes)  
*Adds a new IEdge to the property Edges of the IGraph.*
- void **AddEdge** (IEdge edge)  
*Adds the IEdge edge to the property Edges of the IGraph.*
- void **RemoveEdge** (params IEdge[] edgesToRemove)  
*Removes the IEdge from the property Edges of the IGraph.*
- IEnumerable< INode > **GetSingleNodes** ()  
*Returns all INode objects of this IGraph that are not connected by any IEdges.*
- void **AddGraph** (IGraph graphToAdd)  
*Adds one IGraph object to this IGraph.*
- void **AddGraph** (IGraph graphToAdd, INode thisGraphConnectionNode, INode graphToAddConnectionNode, params IAttribute[] attributes)  
*Adds one IGraph object to this IGraph and connects it with an IEdge.*
- IEnumerable< IEdge > **GetEdges** (INode node1, INode node2)  
*Gets the IEdge objects which are connected to the INode objects node1 and node2.*

## Propertys

- IEnumerable< INode > **Nodes** [get]  
*Contains all INode objects that are part of the IGraph.*
- IEnumerable< IEdge > **Edges** [get]  
*Contains all IEdge objects that are part of the IGraph.*

### 6.7.1 Ausführliche Beschreibung

Classes that implement this interface are a representation of a set of objects that implement the interfaces **INode** and **IEdge** and the relations between them. It is able to perform actions on this objects.

## 6.7.2 Dokumentation der Elementfunktionen

6.7.2.1 void GraphFramework.Interfaces.IGraph.AddEdge ( INode *node1*, INode *node2*, params IAttribute[] *attributes* )

Adds a new [IEdge](#) to the property Edges of the [IGraph](#).

## Parameter

<i>node1</i>	first <a href="#">INode</a> to which the <a href="#">IEdge</a> is connected
<i>node2</i>	second <a href="#">INode</a> to which the <a href="#">IEdge</a> is connected
<i>attributes</i>	<a href="#">IAttribute</a> objects of the <a href="#">IEdge</a>

Implementiert in [GraphFramework.Graph](#).

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



### 6.7.2.2 void GraphFramework.Interfaces.IGraph.AddEdge ( [IEdge](#) *edge* )

Adds the [IEdge](#) edge to the property Edges of the [IGraph](#).

## Parameter

<i>edge</i>	<a href="#">IEdge</a> to add to <a href="#">IGraph</a>
-------------	--

Implementiert in [GraphFramework.Graph](#).

### 6.7.2.3 void GraphFramework.Interfaces.IGraph.AddGraph ( [IGraph](#) *graphToAdd* )

Adds one [IGraph](#) object to this [IGraph](#).

## Parameter

<i>graphToAdd</i>	<a href="#">IGraph</a> object that will be added to this
-------------------	--

Implementiert in [GraphFramework.Graph](#).

### 6.7.2.4 void GraphFramework.Interfaces.IGraph.AddGraph ( [IGraph](#) *graphToAdd*, [INode](#) *thisGraphConnectionNode*, [INode](#) *graphToAddConnectionNode*, params [IAttribute](#)[] *attributes* )

Adds one [IGraph](#) object to this [IGraph](#) and connects it with an [IEdge](#).

## Parameter

<i>graphToAdd</i>	<a href="#">IGraph</a> object that will be added to this
<i>thisGraph↔ ConnectionNode</i>	first <a href="#">INode</a> for new <a href="#">IEdge</a>
<i>graphToAdd↔ ConnectionNode</i>	second <a href="#">INode</a> for new <a href="#">IEdge</a>
<i>attributes</i>	<a href="#">IAttribute</a> objects of new <a href="#">IEdge</a>

Implementiert in [GraphFramework.Graph](#).

### 6.7.2.5 void GraphFramework.Interfaces.IGraph.AddNode ( [INode](#) *node* )

Adds the [INode](#) to property Nodes of the [IGraph](#).

## Parameter

<i>node</i>	reference of the <a href="#">INode</a> to add
-------------	---

Implementiert in [GraphFramework.Graph](#).

#### 6.7.2.6 IEnumerable<IEdge> GraphFramework.Interfaces.IGraph.GetEdges ( [INode](#) *node1*, [INode](#) *node2* )

Gets the [IEdge](#) objects which are connected to the [INode](#) objects *node1* and *node2*.

## Parameter

<i>node1</i>	first <a href="#">INode</a> to which the <a href="#">IEdge</a> has to be connected
<i>node2</i>	second <a href="#">INode</a> to which the <a href="#">IEdge</a> has to be connected

## Rückgabe

all the [IEdge](#) objects which connect the [INode](#) objects *node1* and *node2*

Implementiert in [GraphFramework.Graph](#).

#### 6.7.2.7 IEnumerable<INode> GraphFramework.Interfaces.IGraph.GetSingleNodes ( )

Returns all [INode](#) objects of this [IGraph](#) that are not connected by any [IEdges](#).

## Rückgabe

returns all single [INode](#) objects

Implementiert in [GraphFramework.Graph](#).

#### 6.7.2.8 void GraphFramework.Interfaces.IGraph.RemoveEdge ( params [IEdge](#)[] *edgesToRemove* )

Removes the [IEdge](#) from the property *Edges* of the [IGraph](#).

## Parameter

<i>edgesToRemove</i>	reference of the <a href="#">IEdge</a> objects to remove
----------------------	--

Implementiert in [GraphFramework.Graph](#).

#### 6.7.2.9 void GraphFramework.Interfaces.IGraph.RemoveNode ( params [INode](#)[] *nodesToRemove* )

Removes the [INode](#) from the property *Nodes* of the [IGraph](#).

## Parameter

<i>nodesToRemove</i>	reference of the <a href="#">INode</a> objects to remove
----------------------	--

Implementiert in [GraphFramework.Graph](#).

### 6.7.3 Dokumentation der Propertys

#### 6.7.3.1 IEnumerable<IEdge> GraphFramework.Interfaces.IGraph.Edges [get]

Contains all [IEdge](#) objects that are part of the [IGraph](#).

## 6.7.3.2 IEnumerable&lt;INode&gt; GraphFramework.Interfaces.IGraph.Nodes [get]

Contains all [INode](#) objects that are part of the [IGraph](#).

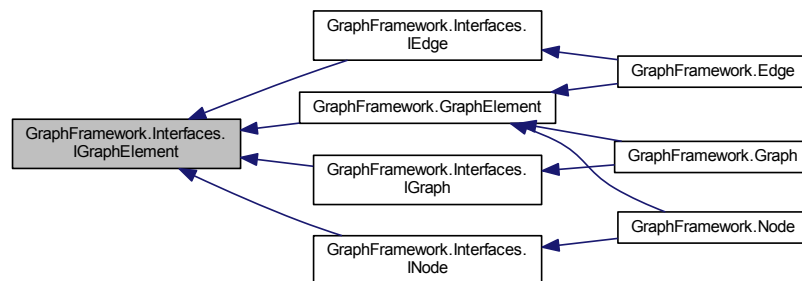
Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

- [Interfaces/IGraph.cs](#)

## 6.8 GraphFramework.Interfaces.IGraphElement Schnittstellenreferenz

An interface for classes that are able to hold objects of classes that implement the interface [IAttribute](#). The interfaces [INode](#), [IEdge](#) and [IGraph](#) inherit from this interface.

Klassendiagramm für GraphFramework.Interfaces.IGraphElement:



### Öffentliche Methoden

- void [AddAttribute](#) ([IAttribute](#) attribute)  
Adds the [IAttribute](#) to the property `Attributes` of the [IGraphElement](#).
- void [RemoveAttribute](#) (string nameOfAttributeToRemove)  
Removes the [IAttribute](#) identified by its name from the property `Attributes`.
- void [RemoveAttribute](#) ([IAttribute](#) attributeToRemove)  
Removes the [IAttribute](#) from the property `Attributes`.
- [IAttribute](#) [GetAttributeByName](#) (string name)  
Returns the attribute with the given name. Warning: can be also null

### Propertys

- IEnumerable< [IAttribute](#) > `Attributes` [get]  
Contains all [IAttribute](#) objects of the [IGraphElement](#).

### 6.8.1 Ausführliche Beschreibung

An interface for classes that are able to hold objects of classes that implement the interface [IAttribute](#). The interfaces [INode](#), [IEdge](#) and [IGraph](#) inherit from this interface.

## 6.8.2 Dokumentation der Elementfunktionen

### 6.8.2.1 void GraphFramework.Interfaces.IGraphElement.AddAttribute ( IAttribute *attribute* )

Adds the [IAttribute](#) to the property Attributes of the [IGraphElement](#).



## Parameter

<i>attribute</i>	reference of the <a href="#">IAttribute</a> to add to the property Attributes
------------------	---

Implementiert in [GraphFramework.GraphElement](#).

### 6.8.2.2 IAttribute GraphFramework.Interfaces.IGraphElement.GetAttributeByName ( string name )

Returns the attribute with the given name. Warning: can be also null

## Parameter

<i>name</i>	
-------------	--

## Rückgabe

null or the attribute with the given name

Implementiert in [GraphFramework.GraphElement](#).

### 6.8.2.3 void GraphFramework.Interfaces.IGraphElement.RemoveAttribute ( string nameOfAttributeToRemove )

Removes the [IAttribute](#) identified by its name from the property Attributes.

## Parameter

<i>nameOfAttributeToRemove</i>	property Name of the <a href="#">IAttribute</a> to remove
--------------------------------	---

Implementiert in [GraphFramework.GraphElement](#).

### 6.8.2.4 void GraphFramework.Interfaces.IGraphElement.RemoveAttribute ( IAttribute attributeToRemove )

Removes the [IAttribute](#) from the property Attributes.

## Parameter

<i>attributeToRemove</i>	reference of the <a href="#">IAttribute</a> to remove
--------------------------	---

Implementiert in [GraphFramework.GraphElement](#).

## 6.8.3 Dokumentation der Propertys

### 6.8.3.1 IEnumerable<IAttribute> GraphFramework.Interfaces.IGraphElement.Attributes [get]

Contains all [IAttribute](#) objects of the [IGraphElement](#).

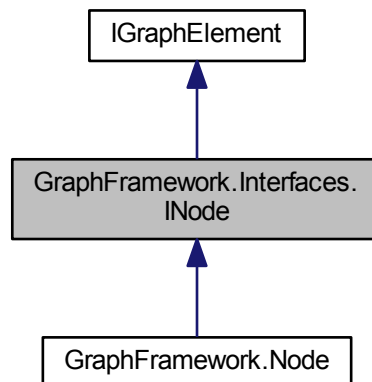
Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

- [Interfaces/IGraphElement.cs](#)

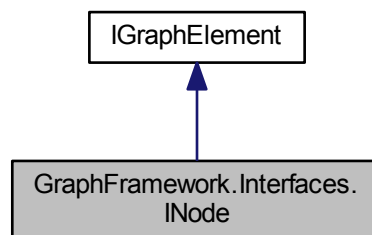
## 6.9 GraphFramework.Interfaces.INode Schnittstellenreferenz

An interface for a class that is able to hold objects of a class that implements the interface [IAttribute](#). Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraph](#).

Klassendiagramm für GraphFramework.Interfaces.INode:



Zusammengehörigkeiten von GraphFramework.Interfaces.INode:



## Weitere Geerbte Elemente

### 6.9.1 Ausführliche Beschreibung

An interface for a class that is able to hold objects of a class that implements the interface [IAttribute](#). Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraph](#).

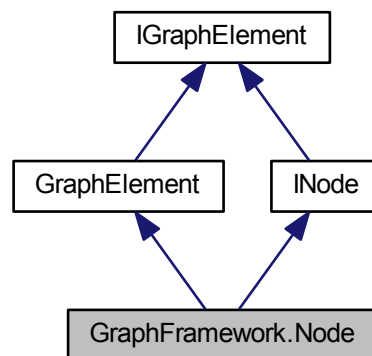
Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

- [Interfaces/INode.cs](#)

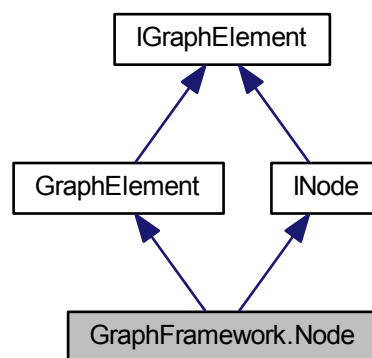
## 6.10 GraphFramework.Node Klassenreferenz

[Node](#) class

Klassendiagramm für GraphFramework.Node:



Zusammengehörigkeiten von GraphFramework.Node:



## Öffentliche Methoden

- `Node` (params `IAttribute[]` attributes)  
*Adds also attributes to the node*

## Weitere Geerbte Elemente

### 6.10.1 Ausführliche Beschreibung

`Node` class

## 6.10.2 Beschreibung der Konstruktoren und Destruktoren

### 6.10.2.1 GraphFramework.Node.Node ( params IAttribute[] *attributes* )

Adds also attributes to the node

Parameter

<i>attributes</i>	the attributes you want to add
-------------------	--------------------------------

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Node.cs](#)

# Kapitel 7

## Datei-Dokumentation

### 7.1 Attribute.cs-Dateireferenz

#### Klassen

- class [GraphFramework.Attribute](#)

*The attribute class. A attribute is a holder of a generic value which has a name*

#### Namensbereiche

- package [GraphFramework](#)

### 7.2 Edge.cs-Dateireferenz

#### Klassen

- class [GraphFramework.Edge](#)

*The edge class*

#### Namensbereiche

- package [GraphFramework](#)

### 7.3 Graph.cs-Dateireferenz

#### Klassen

- class [GraphFramework.Graph](#)

*the graph class: holder of edges and nodes + methods to modify the graph*

#### Namensbereiche

- package [GraphFramework](#)

## 7.4 GraphElement.cs-Dateireferenz

### Klassen

- class [GraphFramework.GraphElement](#)  
*the abstract graphelement class nearly every framework class inherits from. The class generalize the relation between classes that can have attributes and the attribute class*

### Namensbereiche

- package [GraphFramework](#)

## 7.5 Interfaces/IAttribute.cs-Dateireferenz

### Klassen

- interface [GraphFramework.Interfaces.IAttribute](#)  
*An interface for classes that contain a value that has a name. Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraphElement](#).*

### Namensbereiche

- package [GraphFramework.Interfaces](#)

## 7.6 Interfaces/IEdge.cs-Dateireferenz

### Klassen

- interface [GraphFramework.Interfaces.IEdge](#)  
*An interface for classes that are able to connect two objects of a class that implements the interface [INode](#). Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraph](#).*

### Namensbereiche

- package [GraphFramework.Interfaces](#)

## 7.7 Interfaces/IGraph.cs-Dateireferenz

### Klassen

- interface [GraphFramework.Interfaces.IGraph](#)  
*Classes that implement this interface are a representation of a set of objects that implement the interfaces [INode](#) and [IEdge](#) and the relations between them. It is able to perform actions on this objects.*

### Namensbereiche

- package [GraphFramework.Interfaces](#)

## 7.8 Interfaces/IGraphElement.cs-Dateireferenz

### Klassen

- interface [GraphFramework.Interfaces.IGraphElement](#)  
*An interface for classes that are able to hold objects of classes that implement the interface [IAttribute](#). The interfaces [INode](#), [IEdge](#) and [IGraph](#) inherit from this interface.*

### Namensbereiche

- package [GraphFramework.Interfaces](#)

## 7.9 Interfaces/INode.cs-Dateireferenz

### Klassen

- interface [GraphFramework.Interfaces.INode](#)  
*An interface for a class that is able to hold objects of a class that implements the interface [IAttribute](#). Objects of classes that implement this interface can be added to objects of classes that implement the interface [IGraph](#).*

### Namensbereiche

- package [GraphFramework.Interfaces](#)

## 7.10 Node.cs-Dateireferenz

### Klassen

- class [GraphFramework.Node](#)  
*[Node](#) class*

### Namensbereiche

- package [GraphFramework](#)

## 7.11 Properties/AssemblyInfo.cs-Dateireferenz





# Index

- AddAttribute
  - GraphFramework::GraphElement, [24](#)
  - GraphFramework::Interfaces::IGraphElement, [34](#)
- AddEdge
  - GraphFramework::Graph, [17](#), [18](#)
  - GraphFramework::Interfaces::IGraph, [30](#), [31](#)
- AddGraph
  - GraphFramework::Graph, [18](#), [19](#)
  - GraphFramework::Interfaces::IGraph, [31](#)
- AddNode
  - GraphFramework::Graph, [19](#)
  - GraphFramework::Interfaces::IGraph, [31](#)
- Attribute
  - GraphFramework::Attribute, [12](#)
- Attribute.cs, [39](#)
- Attributes
  - GraphFramework::GraphElement, [25](#)
  - GraphFramework::Interfaces::IGraphElement, [35](#)
- Edge
  - GraphFramework::Edge, [14](#)
- Edge.cs, [39](#)
- Edges
  - GraphFramework::Graph, [23](#)
  - GraphFramework::Interfaces::IGraph, [32](#)
- GetAttributeByName
  - GraphFramework::GraphElement, [24](#)
  - GraphFramework::Interfaces::IGraphElement, [35](#)
- GetEdges
  - GraphFramework::Graph, [20](#)
  - GraphFramework::Interfaces::IGraph, [32](#)
- GetSingleNodes
  - GraphFramework::Graph, [20](#)
  - GraphFramework::Interfaces::IGraph, [32](#)
- Graph
  - GraphFramework::Graph, [16](#)
- Graph.cs, [39](#)
- GraphElement
  - GraphFramework::GraphElement, [24](#)
- GraphElement.cs, [40](#)
- GraphFramework, [9](#)
- GraphFramework.Attribute, [11](#)
- GraphFramework.Edge, [12](#)
- GraphFramework.Graph, [14](#)
- GraphFramework.GraphElement, [23](#)
- GraphFramework.Interfaces, [9](#)
- GraphFramework.Interfaces.IAttribute, [25](#)
- GraphFramework.Interfaces.Edge, [27](#)
- GraphFramework.Interfaces.IGraph, [28](#)
- GraphFramework.Interfaces.IGraphElement, [33](#)
- GraphFramework.Interfaces.INode, [35](#)
- GraphFramework.Node, [36](#)
- GraphFramework::Attribute
  - Attribute, [12](#)
  - Name, [12](#)
  - Type, [12](#)
  - Value, [12](#)
- GraphFramework::Edge
  - Edge, [14](#)
  - Node1, [14](#)
  - Node2, [14](#)
- GraphFramework::Graph
  - AddEdge, [17](#), [18](#)
  - AddGraph, [18](#), [19](#)
  - AddNode, [19](#)
  - Edges, [23](#)
  - GetEdges, [20](#)
  - GetSingleNodes, [20](#)
  - Graph, [16](#)
  - MergeGraphs, [20](#), [21](#)
  - Nodes, [23](#)
  - RemoveEdge, [22](#)
  - RemoveNode, [22](#)
- GraphFramework::GraphElement
  - AddAttribute, [24](#)
  - Attributes, [25](#)
  - GetAttributeByName, [24](#)
  - GraphElement, [24](#)
  - mAttributes, [25](#)
  - RemoveAttribute, [25](#)
- GraphFramework::Interfaces::IAttribute
  - Name, [26](#)
  - Type, [26](#)
  - Value, [26](#)
- GraphFramework::Interfaces::IEdge
  - Node1, [28](#)
  - Node2, [28](#)
- GraphFramework::Interfaces::IGraph
  - AddEdge, [30](#), [31](#)
  - AddGraph, [31](#)
  - AddNode, [31](#)
  - Edges, [32](#)
  - GetEdges, [32](#)
  - GetSingleNodes, [32](#)
  - Nodes, [32](#)
  - RemoveEdge, [32](#)
  - RemoveNode, [32](#)
- GraphFramework::Interfaces::IGraphElement

- AddAttribute, [34](#)
- Attributes, [35](#)
- GetAttributeByName, [35](#)
- RemoveAttribute, [35](#)
- GraphFramework::Node
  - Node, [38](#)
- Interfaces/IAttribute.cs, [40](#)
- Interfaces/IEdge.cs, [40](#)
- Interfaces/IGraph.cs, [40](#)
- Interfaces/IGraphElement.cs, [41](#)
- Interfaces/INode.cs, [41](#)
- mAttributes
  - GraphFramework::GraphElement, [25](#)
- MergeGraphs
  - GraphFramework::Graph, [20](#), [21](#)
- Name
  - GraphFramework::Attribute, [12](#)
  - GraphFramework::Interfaces::IAttribute, [26](#)
- Node
  - GraphFramework::Node, [38](#)
- Node.cs, [41](#)
- Node1
  - GraphFramework::Edge, [14](#)
  - GraphFramework::Interfaces::IEdge, [28](#)
- Node2
  - GraphFramework::Edge, [14](#)
  - GraphFramework::Interfaces::IEdge, [28](#)
- Nodes
  - GraphFramework::Graph, [23](#)
  - GraphFramework::Interfaces::IGraph, [32](#)
- Properties/AssemblyInfo.cs, [41](#)
- RemoveAttribute
  - GraphFramework::GraphElement, [25](#)
  - GraphFramework::Interfaces::IGraphElement, [35](#)
- RemoveEdge
  - GraphFramework::Graph, [22](#)
  - GraphFramework::Interfaces::IGraph, [32](#)
- RemoveNode
  - GraphFramework::Graph, [22](#)
  - GraphFramework::Interfaces::IGraph, [32](#)
- Type
  - GraphFramework::Attribute, [12](#)
  - GraphFramework::Interfaces::IAttribute, [26](#)
- Value
  - GraphFramework::Attribute, [12](#)
  - GraphFramework::Interfaces::IAttribute, [26](#)