Ingegneria del Software T

XML *Programming*

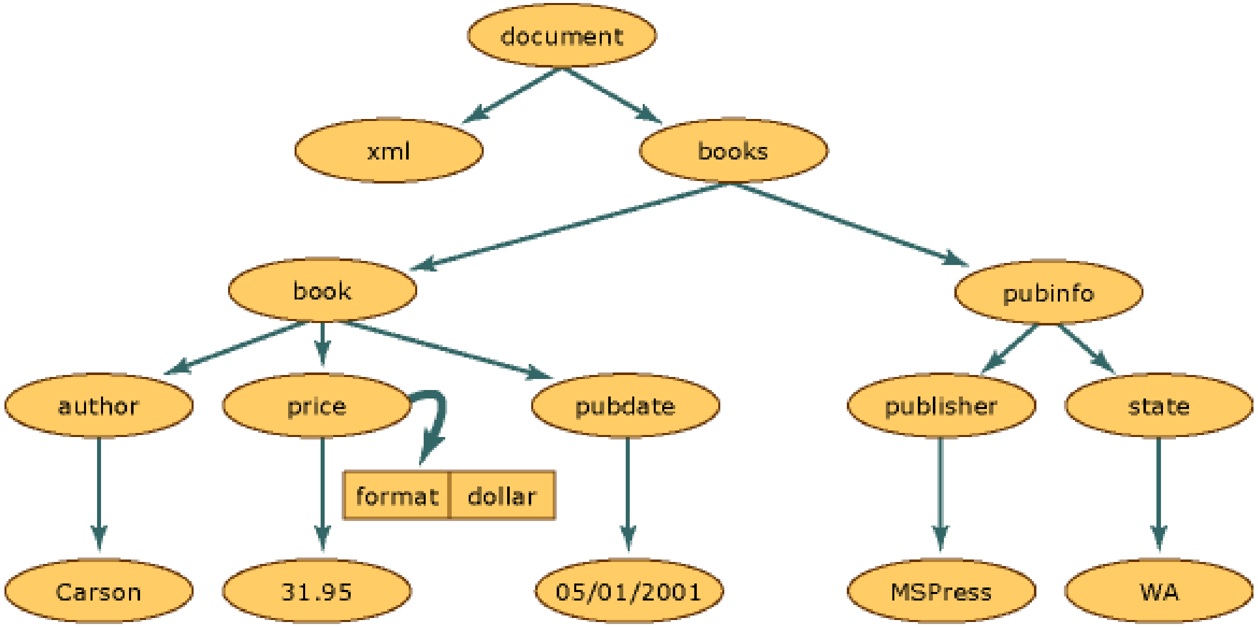
XML Document Object Model

- An in-memory representation of an XML document
- The DOM allows you to programmatically
  - Load
  - Modify
  - Save

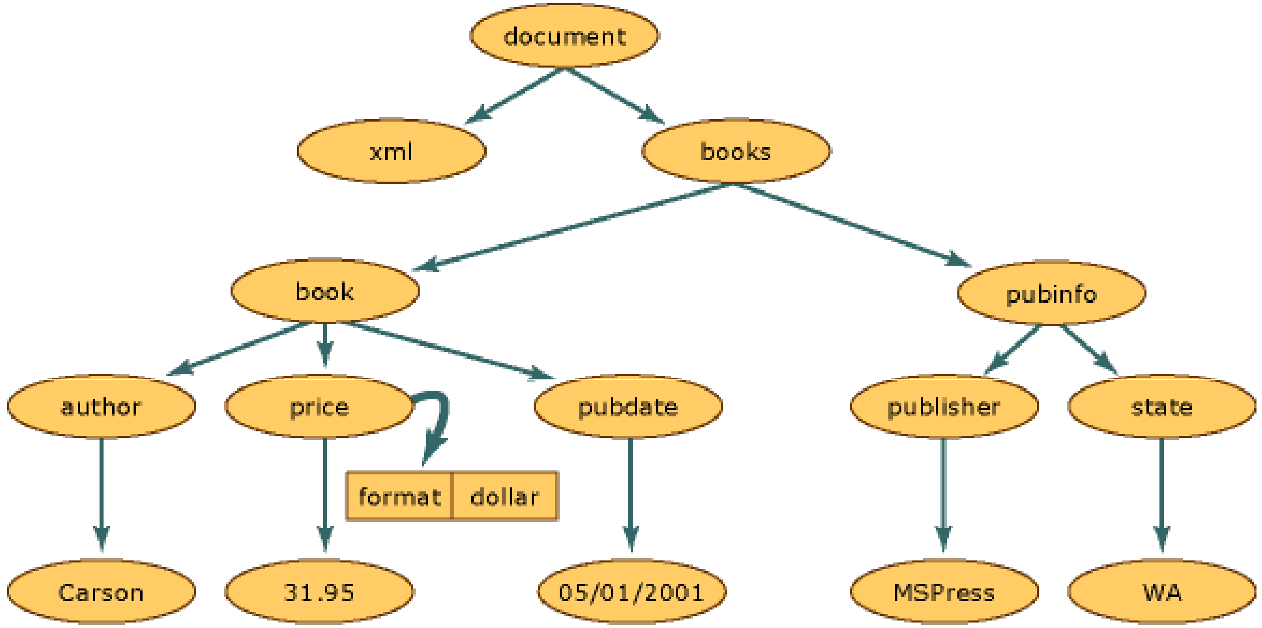
  an XML document

XML Document Object Model

```
<?xml version="1.0"?>
<books>
  <book>
    <author>Carson</author>
    <price format="dollar">31.95</price>
    <pubdate>05/01/2001</pubdate>
  </book>
  <pubinfo>
    <publisher>MSPress</publisher>
    <state>WA</state>
  </pubinfo>
</books>
```

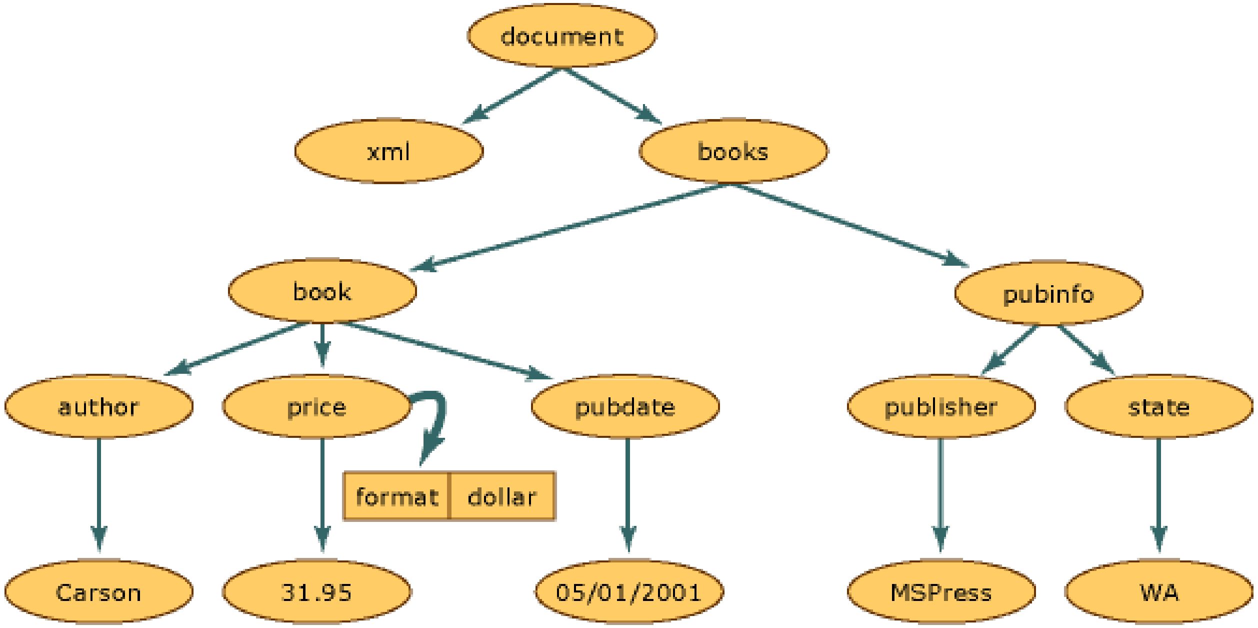


XML Document Object Model



- Nodes have a single **parent node**, a parent node being a node directly above it (the only node that do not have a parent is the “document” node)
- Most nodes can have multiple **child nodes**, which are nodes directly below it
- Nodes that are at the same level are **siblings** (the “book” and “pubinfo” nodes, …)

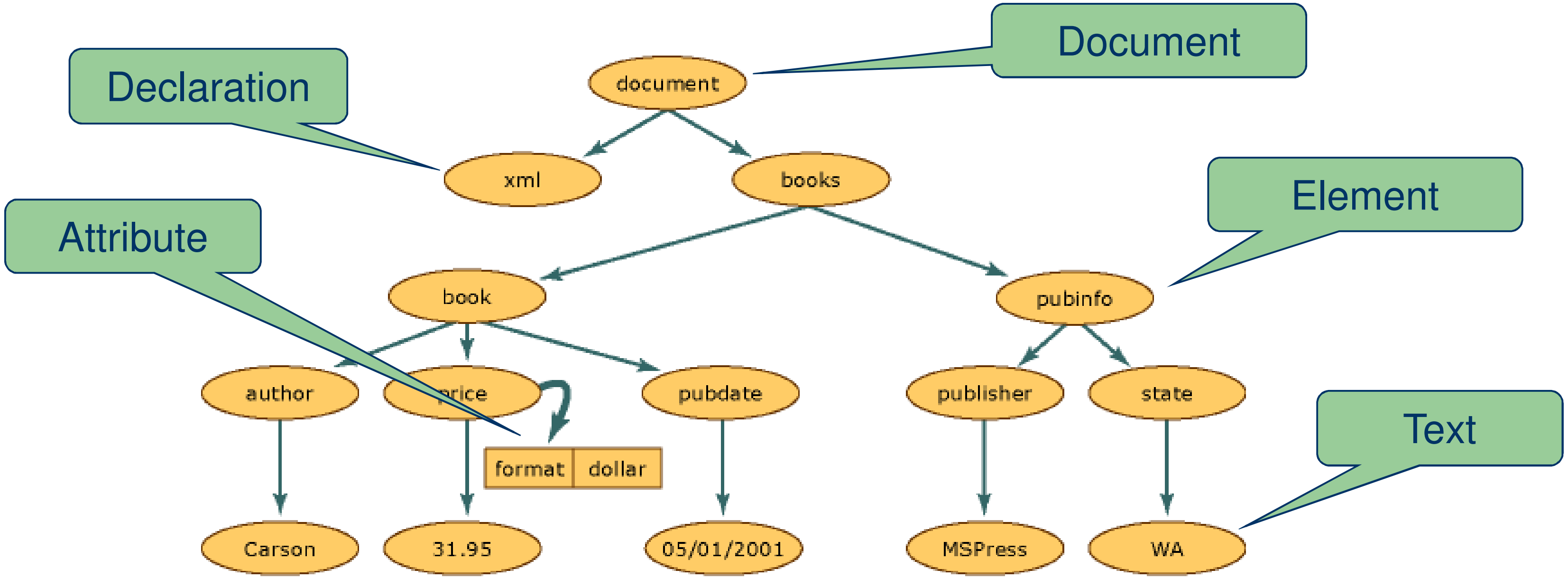
XML Document Object Model



- Gli attributi non fanno parte delle relazioni *parent*, *child* e *sibling*
- Gli attributi vengono considerati **proprietà dei nodi di tipo element**, e sono costituiti da una **coppia nome-valore**
- Nell’esempio:
  - la parola “**`format`**” è il nome dell’attributo
  - la stringa “**`dollar`**” è il valore dell’attributo **`format`**

XML Document Object Model

- As XML is read into memory, nodes are created
- However, not all nodes are the same type
- An element, in XML, has different rules and syntax than a processing instruction
- So as various data is read, a **node type** is assigned to each node
- This node type determines the characteristics and functionality of the node



XML Document Object Model

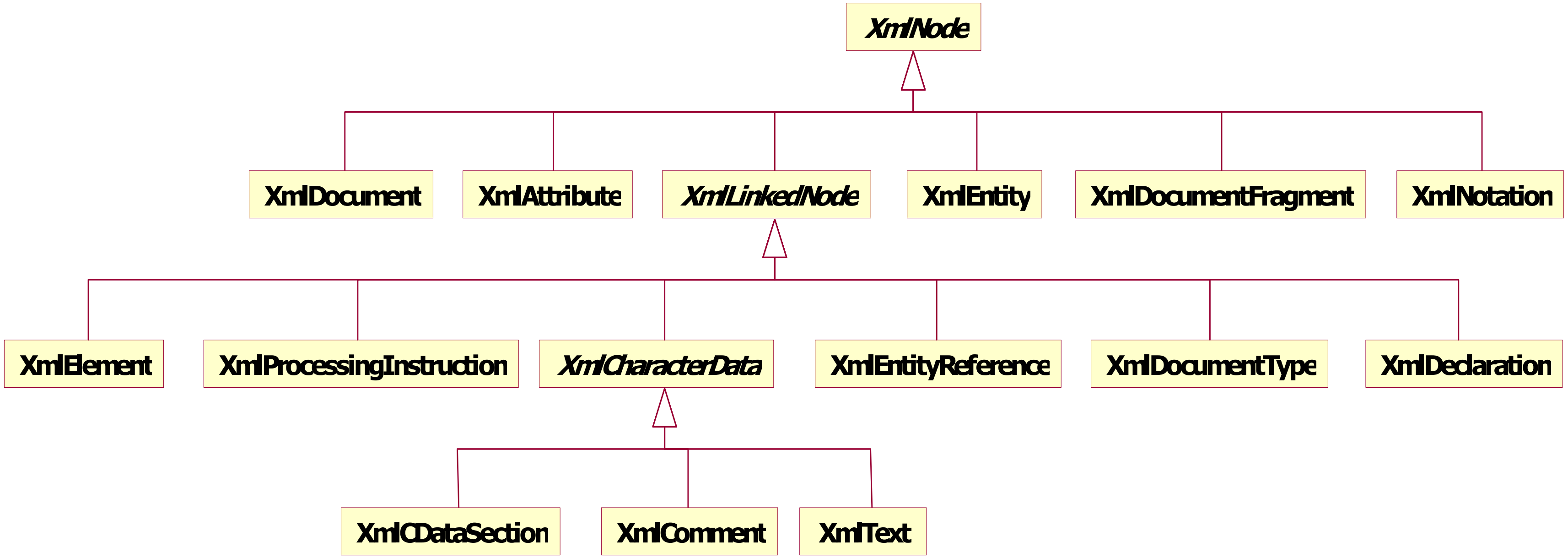
DOM Node Type	Classe	Descrizione
Document	`XmlDocument`	The container of all the nodes in the tree
Element	`XmlElement`	Represents an element node: <item>…</item>
Attribute	`XmlAttribute`	Represents an attribute of an element: <… id="123“>
Comment	`XmlComment`	Represents a comment node: <!-- my comment -->
Text	`XmlText`	Text belonging to an element or attribute
CDATA	`XmlCDataSection`	Represents CDATA: <![CDATA[…escaped text…]]>
Declaration	`XmlDeclaration`	Represents the declaration node: <?xml version="1.0"...>

XML Document Object Model

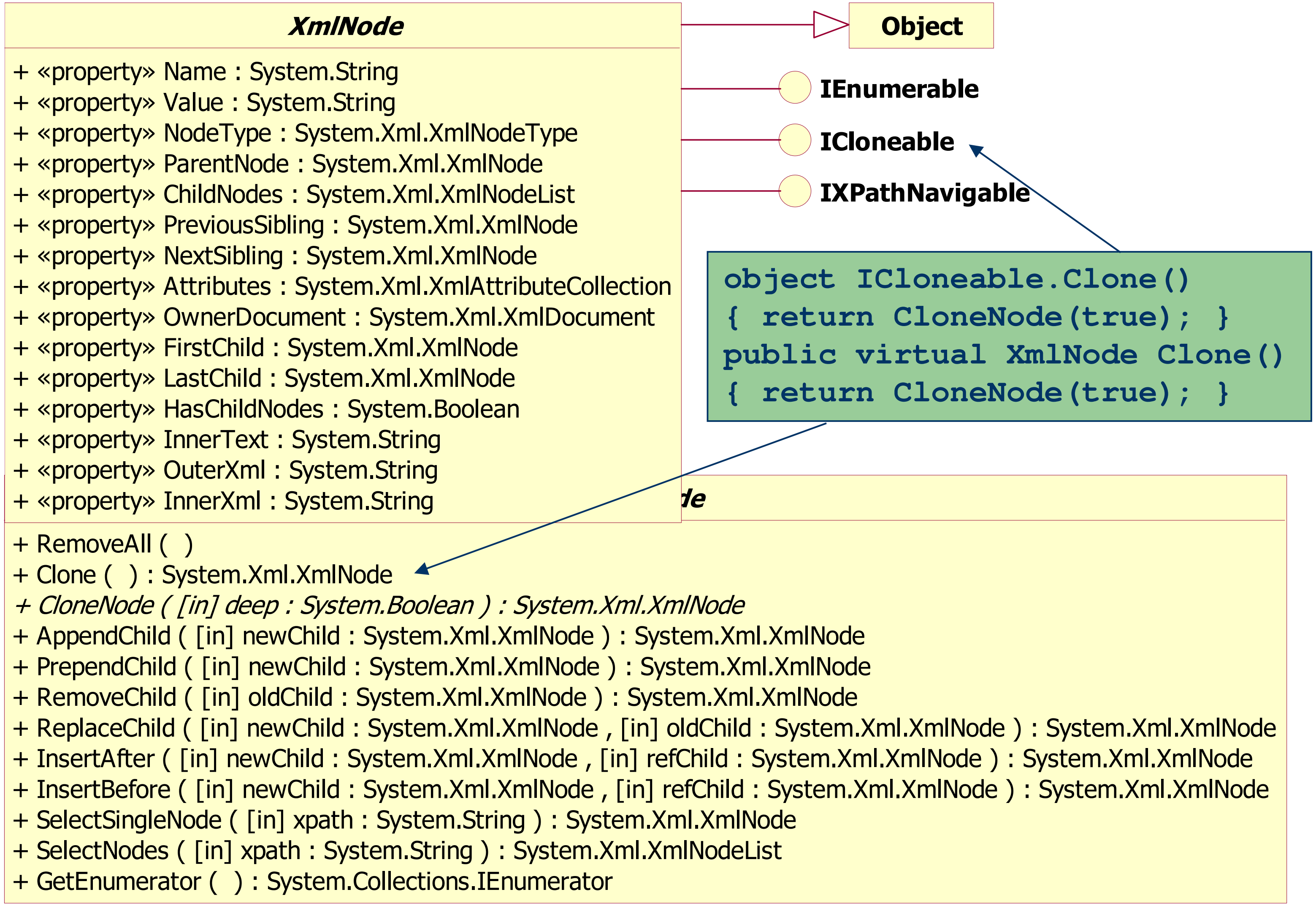
DOM Node Type	Classe	Descrizione
DocumentFragment	`XmlDocumentFragment`	A temporary bag containing one or more nodes without any tree structure
DocumentType	`XmlDocumentType`	Represents the <!DOCTYPE...> node
EntityReference	`XmlEntityReference`	Represents the non-expanded entity reference text
ProcessingInstruction	`XmlProcessingInstruction`	Is a processing instruction node
Entity	`XmlEntity`	Represents the <!ENTITY...> declarations in an XML document, either from an internal document type definition (DTD) subset or from external DTDs and parameter entities
Notation	`XmlNotation`	Represents a notation declared in the DTD

XML Document Object Model

- The **`XmlNode`** class is the basic class in the DOM tree



XML Document Object Model



XmlNode

+ «property» Name : System.String
+ «property» Value : System.String
+ «property» NodeType : System.Xml.XmlNodeType
+ «property» ParentNode : System.Xml.XmlNode
+ «property» ChildNodes : System.Xml.XmlNodeList
+ «property» PreviousSibling : System.Xml.XmlNode
+ «property» NextSibling : System.Xml.XmlNode
+ «property» Attributes : System.Xml.XmlAttributeCollection
+ «property» OwnerDocument : System.Xml.XmlDocument
+ «property» FirstChild : System.Xml.XmlNode
+ «property» LastChild : System.Xml.XmlNode
+ «property» HasChildNodes : System.Boolean
+ «property» InnerText : System.String
+ «property» OuterXml : System.String
+ «property» InnerXml : System.String

Object

IEnumerable

ICloneable

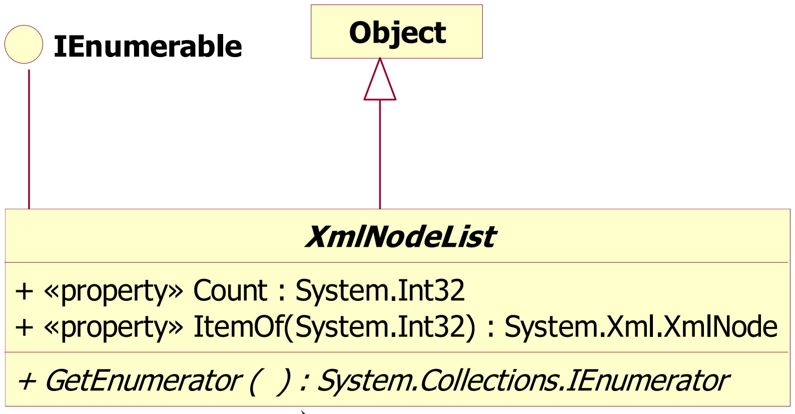
IXPathNavigable

```
object ICloneable.Clone()
{ return CloneNode(true); }
public virtual XmlNode Clone()
{ return CloneNode(true); }
```

de

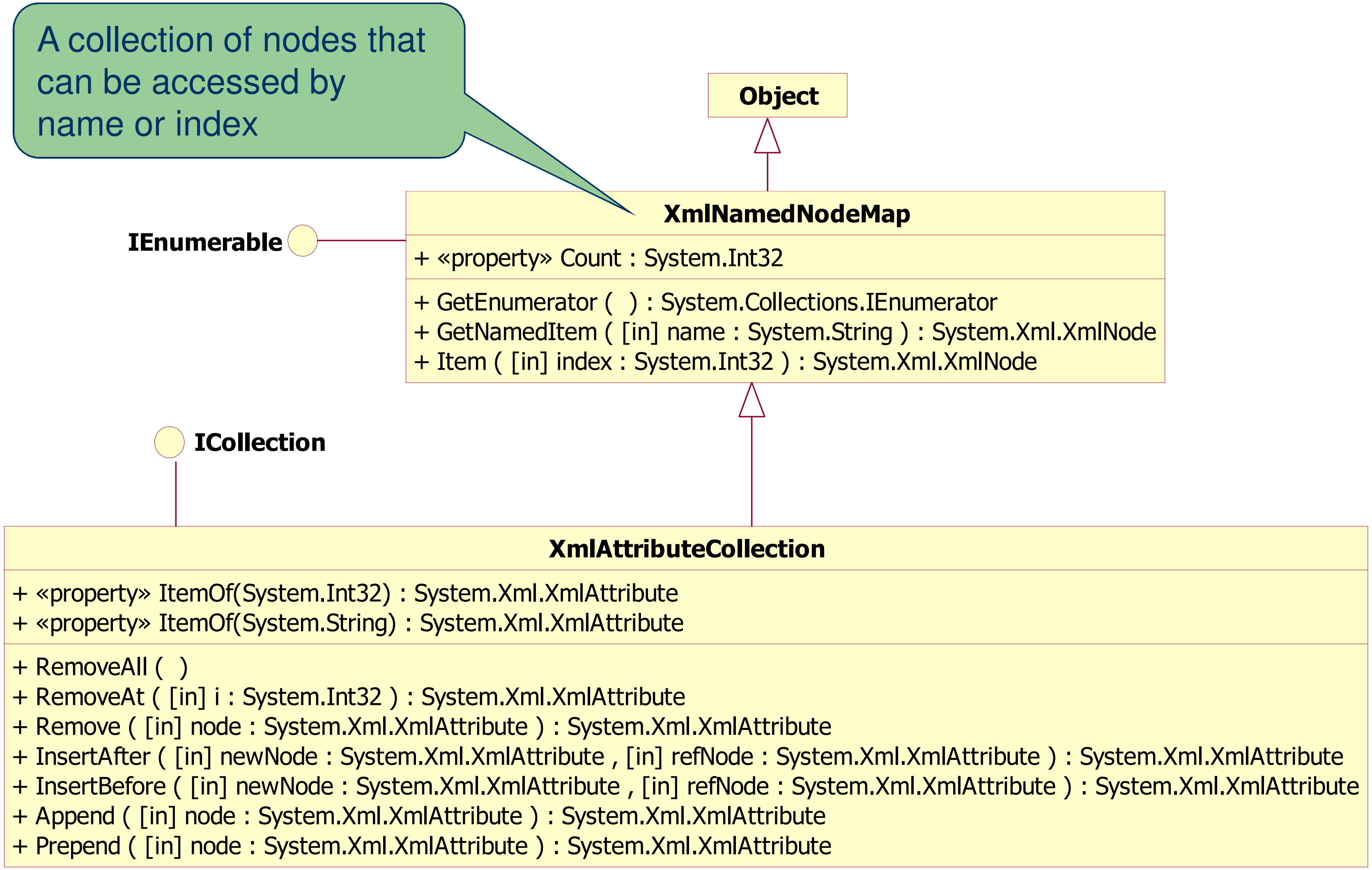
+ RemoveAll ( )
+ Clone ( ) : System.Xml.XmlNode
+ *CloneNode ([in] deep : System.Boolean) : System.Xml.XmlNode*
+ AppendChild ([in] newChild : System.Xml.XmlNode) : System.Xml.XmlNode
+ PrependChild ([in] newChild : System.Xml.XmlNode) : System.Xml.XmlNode
+ RemoveChild ([in] oldChild : System.Xml.XmlNode) : System.Xml.XmlNode
+ ReplaceChild ([in] newChild : System.Xml.XmlNode , [in] oldChild : System.Xml.XmlNode) : System.Xml.XmlNode
+ InsertAfter ([in] newChild : System.Xml.XmlNode , [in] refChild : System.Xml.XmlNode) : System.Xml.XmlNode
+ InsertBefore ([in] newChild : System.Xml.XmlNode , [in] refChild : System.Xml.XmlNode) : System.Xml.XmlNode
+ SelectSingleNode ([in] xpath : System.String) : System.Xml.XmlNode
+ SelectNodes ([in] xpath : System.String) : System.Xml.XmlNodeList
+ GetEnumerator ( ) : System.Collections.IEnumerator

XML Document Object Model


IEnumerable
Object
XmlNodeList
+ «property» Count : System.Int32
+ «property» ItemOf(System.Int32) : System.Xml.XmlNode
+ GetEnumerator ( ) : System.Collections.IEnumerator

An ordered collection of nodes

XML Document Object Model



A collection of nodes that can be accessed by name or index

Object

IEnumerable

ICollection

XmlNamedNodeMap

+ «property» Count : System.Int32

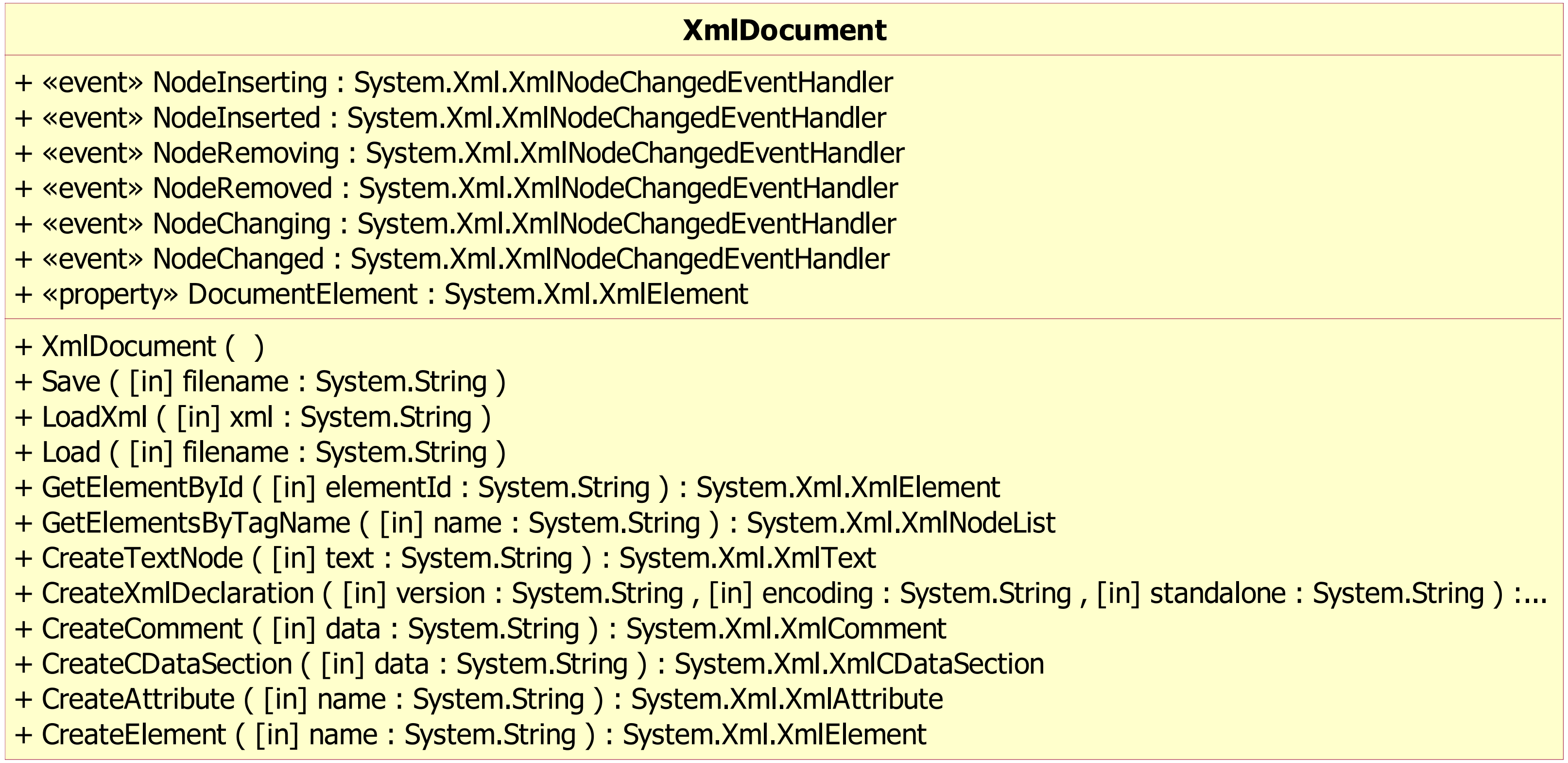
+ GetEnumerator ( ) : System.Collections.IEnumerator
+ GetNamedItem ([in] name : System.String) : System.Xml.XmlNode
+ Item ([in] index : System.Int32) : System.Xml.XmlNode

XmlAttributeCollection

+ «property» ItemOf(System.Int32) : System.Xml.XmlAttribute
+ «property» ItemOf(System.String) : System.Xml.XmlAttribute

+ RemoveAll ( )
+ RemoveAt ([in] i : System.Int32) : System.Xml.XmlAttribute
+ Remove ([in] node : System.Xml.XmlAttribute) : System.Xml.XmlAttribute
+ InsertAfter ([in] newNode : System.Xml.XmlAttribute , [in] refNode : System.Xml.XmlAttribute) : System.Xml.XmlAttribute
+ InsertBefore ([in] newNode : System.Xml.XmlAttribute , [in] refNode : System.Xml.XmlAttribute) : System.Xml.XmlAttribute
+ Append ([in] node : System.Xml.XmlAttribute) : System.Xml.XmlAttribute
+ Prepend ([in] node : System.Xml.XmlAttribute) : System.Xml.XmlAttribute

XML Document Object Model



XmlDocument

+ «event» NodeInserting : System.Xml.XmlNodeChangedEventHandler
+ «event» NodeInserted : System.Xml.XmlNodeChangedEventHandler
+ «event» NodeRemoving : System.Xml.XmlNodeChangedEventHandler
+ «event» NodeRemoved : System.Xml.XmlNodeChangedEventHandler
+ «event» NodeChanging : System.Xml.XmlNodeChangedEventHandler
+ «event» NodeChanged : System.Xml.XmlNodeChangedEventHandler
+ «property» DocumentElement : System.Xml.XmlElement

+ XmlDocument ( )
+ Save ([in] filename : System.String)
+ LoadXml ([in] xml : System.String)
+ Load ([in] filename : System.String)
+ GetElementById ([in] elementId : System.String) : System.Xml.XmlElement
+ GetElementsByTagName ([in] name : System.String) : System.Xml.XmlNodeList
+ CreateTextNode ([in] text : System.String) : System.Xml.XmlText
+ CreateXmlDeclaration ([in] version : System.String , [in] encoding : System.String , [in] standalone : System.String) :...
+ CreateComment ([in] data : System.String) : System.Xml.XmlComment
+ CreateCDataSection ([in] data : System.String) : System.Xml.XmlCDataSection
+ CreateAttribute ([in] name : System.String) : System.Xml.XmlAttribute
+ CreateElement ([in] name : System.String) : System.Xml.XmlElement

XML Document Object Model

- **Lettura** (sincrona) di un documento XML da file (in caso di errore: `XmlException`)

```
XmlDocument document = new XmlDocument();
document.Load(fileName);
```

- **Reperimento** elemento radice di un documento XML

```
XmlElement root = document.DocumentElement;
```

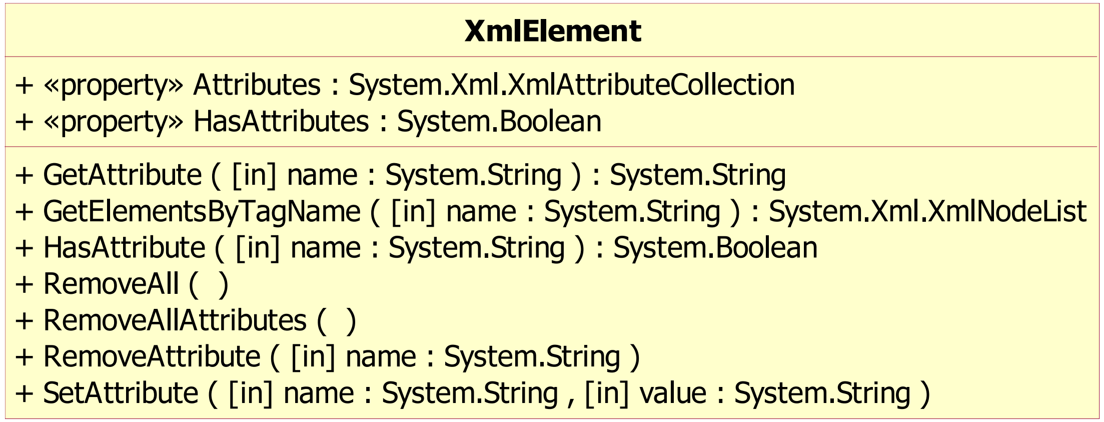
- **Creazione di un nuovo documento** XML

```
XmlDocument doc = new XmlDocument();
XmlNode node = doc.CreateXmlDeclaration("1.0","","");
doc.AppendChild(node);
XmlNode root = doc.CreateElement("XmlNodeType");
doc.AppendChild(root);
//  Inserimento di tutti gli altri nodi in root
```

- **Salvataggio** di un documento XML su file

```
doc.Save(fileName);
```

XML Document Object Model



XmlElement

+ «property» Attributes : System.Xml.XmlAttributeCollection
+ «property» HasAttributes : System.Boolean

+ GetAttribute ([in] name : System.String) : System.String
+ GetElementsByTagName ([in] name : System.String) : System.Xml.XmlNodeList
+ HasAttribute ([in] name : System.String) : System.Boolean
+ RemoveAll ( )
+ RemoveAllAttributes ( )
+ RemoveAttribute ([in] name : System.String)
+ SetAttribute ([in] name : System.String , [in] value : System.String)

- **`GetAttribute(nomeAttributo)`**
  - Se l’attributo esiste, restituisce il valore dell’attributo
  - In caso contrario, restituisce una stringa vuota
- **`SetAttribute(nomeAttributo,valoreAttributo)`**
  - Se l’attributo esiste, ne cambia il valore
  - In caso contrario, crea un nuovo attributo con il valore specificato
- **`RemoveAttribute(nomeAttributo)`**
  - Se l’attributo esiste, lo elimina
  - In caso contrario, nop

XML Document Object Model

- **`XmlNodeList SelectNodes(string xpath);`**
  Selects a list of nodes matching the XPath expression
- **`XmlNode SelectSingleNode(string xpath);`**
  Selects the first **`XmlNode`** that matches the XPath expression

```
<?xml version="1.0" encoding="utf-8" ?>
<Gruppo>
  <Item id="1">Pippo</Item>
  <Item id="2">Topolino</Item>
  <Item id="5">Paperino</Item>
  <Item id="7">Gastone</Item>
</Gruppo>
```

- Semplici espressioni XPath:
  - **`/Gruppo/Item`** → restituisce tutti gli **`Item`**
  - **`/Gruppo/Item[@id >= 5]`** → restituisce 2 **`Item`**
  - **`/Gruppo/Item[text() = 'Topolino']`** → restituisce 1 **`Item`**