Programmazione Orientata agli Oggetti in Linguaggio Java

Programmazione su XML: C#

versione 2.0

G. Mecca – Università della Basilicata – mecca@unibas.it

Sommario

- Introduzione
- System.Xml
- Utilizzo dei File

Introduzione

- XML
  - una tecnologia fondamentale della piattaforma .NET
  - utilizzata per vari scopi (es: configurazione delle applicazioni, persistenza, servizi Web...)
  - il supporto è decisamente completo
- La libreria di sistema
  - System.Xml

Introduzione

- In particolare
  - System.Xml contiene le classi che implementano DOM
  - System.Xml.Schema: supporto per XMLSchema
  - System.Xml.Xpath: supporto a XPath
  - System.Xml.Xsl: supporto a XSL
  - System.Xml.Serialization: supporta la trasmissione di XML in rete

Introduzione

>> questionari

- Librerie alternative
  - dom4net – http://www.dom4net.org
  - non esiste una implementazione per .NET equivalente a JDOM
- In questa lezione
  - il package standard System.Xml
  - implementazione di DOM
  - codice molto simile alla versione JAXP dell'applicazione

System.Xml

- Le classi principali
  - System.Xml.XmlDocument
  - System.Xml.XmlElement
  - System.Xml.XmlAttribute
  - System.Xml.XmlText
  - System.Xml.XmlNodeList

System.Xml

- Inoltre
  - ⇨System.Xml.TextReader
  - ⇨System.Xml.ValidatingReader
  - ⇨System.Xml.XmlException
  - ⇨System.Xml.TextWriter
  - ⇨System.Xml.Formatting
- Esempi
  - ⇨DAOQuestionario, DAORisposte

```
namespace Unibas.Questionari.Persistenza {

using System.Xml;
using Unibas.Questionari.Modello;

public class DAOQuestionario {

   public static Questionario CaricaQuestionario(string nomeFile) {
      System.Xml.XmlDocument document =
                                          CostruisciDOM(nomeFile);
      Questionario questionario = new Questionario();
      AnalizzaRadice(document, questionario);
      RiempiArgomenti(document, questionario);
      RiempiDifficolta(document, questionario);
      RiempiQuesiti(document, questionario);
      return questionario;
   }
```

```
private static XmlDocument CostruisciDOM(string nomeFile) {
    XmlDocument document = new XmlDocument();
    XmlTextReader textReader = null;
    XmlValidatingReader reader = null;
    try {
        textReader = new XmlTextReader(nomeFile);
        reader = new XmlValidatingReader(textReader);
        document.Load(reader);
    } catch (XmlException xe) {
        throw new DAOException(xe.ToString());
    } catch (System.Xml.Schema.XmlSchemaException se) {
        throw new DAOException(se.ToString());
    } catch (System.IO.IOException ioe) {
        throw new DAOException(ioe.ToString());
    } finally {
        if (reader != null) {
            reader.Close();
        }
    }
    return document;
}
```


```
private static void AnalizzaRadice(XmlDocument document,
                                   Questionario questionario) {
    XmlElement radice = document.DocumentElement;
    string id = radice.GetAttribute("id");
    string nomeDisciplina = radice.GetAttribute("disciplina");
    questionario.Id = id;
    questionario.Disciplina = nomeDisciplina;
}

private static void RiempiArgomenti(XmlDocument document,
                                    Questionario questionario) {
    XmlElement radice = document.DocumentElement;
    XmlElement listaArgomenti = GetFirstChildByName(radice, "listaArgomenti");
    XmlNodeList argomenti = listaArgomenti.GetElementsByTagName("argomento");
    for(int i = 0; i < argomenti.Count; i++){
        XmlElement elementoArgomento = (XmlElement)argomenti[i];
        Argomento argomento = RiempiArgomento(elementoArgomento);
        questionario.AddArgomento(argomento);
    }
}
```

```
private static XmlElement GetFirstChildByName(XmlElement elem,
                                              string nome) {
    XmlNodeList figli = elem.ChildNodes;
    for (int i = 0; i < figli.Count; i++) {
        XmlNode figlio = figli[i];
        if (figlio.Name.Equals(nome)) {
            return (XmlElement)figlio;
        }
      }
      return null;
}
```



```
namespace Unibas.Questionari.Persistenza {
using System.Xml;
using Unibas.Questionari.Modello;

public class DAORisposte {

   public static void SalvaRisposte(Risposte risposte, string nomeFile)  {
      if (risposte == null) { throw new DAOException("Risposte nulle");}
      XmlDocument document = CreaDOM();
      CompletaRadice(risposte, document);
      AggiungiListaRisposte(risposte, document);
      SalvaDocumento(document, nomeFile);
      string percorso = DAOUtilita.EstraiPercorso(nomeFile);
      DAOUtilita.CopiaDTD(percorso);
   }

   private static XmlDocument CreaDOM() {
      XmlDocument document = new XmlDocument();
      XmlElement radice = (XmlElement)document.CreateElement("risposte");
      document.AppendChild(radice);
      return document;
   }
```

```
private static void SalvaDocumento(XmlDocument document,
                                   string nomeFile) {
    XmlTextWriter writer = new XmlTextWriter(nomeFile, null);
    writer.Formatting = Formatting.Indented;
    try {
        document.Save(writer);
    } catch (XmlException e) {
        throw new DAOException(e.ToString());
    } finally {
        if (writer != null) {
            writer.Close();
        }
    }
}
```


Riferimenti ai File

- Un deciso miglioramento rispetto a Java
  - ⇨in .NET è possibile utilizzare riferimenti relativi ai file
  - ⇨i riferimenti vengono espansi relativamente alla cartella in cui è contenuto l'assembly in cui viene usato il riferimento
  - ⇨questo semplifica decisamente le operazioni sui file

Riferimenti ai File

- Inoltre, nella classe System.IO.File
  - ⇨un metodo statico Copy() per la copia dei file
  - ⇨due versioni
  - ⇨static void Copy(string file1, string file2)
  - ⇨static void Copy(string file1, string file2, bool overwr)
  - ⇨il terzo parametro consente di specificare se sovrascrivere il file di destinazione (true) o meno
  - ⇨nella prima versione il parametro è false

```
namespace Unibas.Questionari.Persistenza {
using Unibas.Questionari.Modello;

public class DAOUtilita {

   private const string PERCORSODTD = "..\..\varie\";
   private const string NOMEDTD = "risposte.dtd";

   public static void CopiaDTD(string percorso) {
      try {
         string fileDTDOriginale = PERCORSODTD + NOMEDTD;
         string nuovoFileDTD = percorso + NOMEDTD;
         System.IO.File.Copy(fileDTDOriginale, nuovoFileDTD, true);
      } catch (System.IO.IOException ioe) {
         throw new DAOException(ioe);
      }
   }

   public static string EstraiPercorso(string nomeFile) {
      string percorsoCompleto = System.IO.Path.GetFullPath(nomeFile);
      int ultimaPosizioneSeparatore = percorsoCompleto.LastIndexOf("\");
      string percorso = percorsoCompleto.Substring(0, ultimaPosizioneSeparatore + 1);
      return percorso;
   }

}
}
```

Riferimenti ai File

>> questionari.zip

- Distribuzione di applicazioni .NET
  - ⇨decisamente semplificata
  - ⇨gli assembly sono immediatamente eseguibili da Windows (non su altre piattaforme)
  - ⇨i riferimenti ai file sono facilmente gestibili in modo relativo
  - ⇨la struttura delle cartelle è approssimativamente corrispondente a quella utilizzata per Java

Riassumendo

- Introduzione
- System.Xml
- Utilizzo dei File

Termini della Licenza