

Corso sul linguaggio Java

Modulo JAVA9

A1 – Classe File

M. Malatesta A1-Classe File-06

1
12/04/2013

Prerequisiti

- Programmazione base in Java
- Definizione e funzioni del file system
- Programmare con gli *stream*

M. Malatesta A1-Classe File-06

2
12/04/2013

Introduzione

Il package **java.io** include classi, interfacce ed eccezioni per la gestione dell'I/O.

Distinguiamo due classi di funzionalità:

- funzionalità per leggere e scrivere dati su memoria di massa (argomento trattato in precedenza parlando di *stream* **Reader** e **Writer**);
- funzionalità per accedere ad informazioni relative ai file ed alle directory e che sono tipiche della classe **File** presentata in questa Unità.

M. Malatesta A1-Classe File-06

3
12/04/2013

La classe **File**

La classe **File** che presentiamo, consente di scrivere applicazioni che operano sul *file system*, per accedere ad informazioni su file e directory come, ad esempio, calcolare la dimensione di un file, elencare i file contenuti in una directory, e così via.

Di questa classe, presentiamo i costruttori ed i metodi più comuni per operare con i suoi oggetti.

M. Malatesta A1-Classe File-06

4
12/04/2013

La classe File

- I costruttori

La classe **File** definisce oggetti file che vengono associati ai file fisici del *file system* e un insieme di metodi per creare, cancellare e gestire file. I costruttori presentano tre overload:

```
File f = new File (percorso);           // Crea file in percorso
File f = new File (percorso, nomefile);  // Crea nomefile in percorso
File f = new File (directory, nomefile); // Crea nomefile in directory
```

Il *percorso* utilizza il carattere */*. Ad esempio per creare o aprire un file *lettera* nella directory *C:\doc* si può scrivere:

```
File f = new File ("C:/doc", "lettera")
```

Nella classe **File**, per quanto detto in precedenza, non sono definiti metodi per la lettura e scrittura su file.

M. Malatesta A1-Classe File-06

5
12/04/2013

La classe File

- I metodi più importanti di File

I più comuni metodi di **File** sono riportati nella seguente tabella:

Funzioni classe File	
METODO	EFFETTO
File f = new File (percorso);	Crea un file <i>f</i> in <i>percorso</i>
File f = new File (percorso, nomefile);	Crea in <i>percorso</i> , un file <i>f</i> avente nome <i>nomefile</i>
File f = new File (directory, nomefile);	Crea nella cartella <i>directory</i> un file <i>f</i> di nome <i>nomefile</i>
Boolean createNewFile();	Crea un oggetto file nella cartella
public boolean exists();	Verifica l'esistenza di un file
public boolean isFile();	Verifica se l'oggetto è un file
public boolean isDirectory();	Verifica se l'oggetto è una directory
public long length();	Restituisce la lunghezza del file
public boolean renameTo(dest)	Cambia il nome del file.
public boolean delete();	Cancella il file dal disco
public String getName();	Restituisce il nome del file
public String getPath();	Restituisce il <i>pathname</i> completo del file
String[] list();	Dà un array di stringhe con i nomi dei file in directory
boolean mkdir();	Crea una directory e restituisce l'esito

M. Malatesta A1-Classe File-06

6
12/04/2013

La classe **File**

- I metodi più importanti di **File**

Per le directory sono usati spesso i seguenti metodi:

- **String[] list()** // restituisce un array di stringhe che corrispondono
// ai nomi dei file contenuti nella directory.
// Esempio: **String []** Listafile = **f.list()**;
- **boolean mkdir()** // crea una directory in base alle informazioni
// dell'oggetto su cui è invocata e ritorna un valore
// booleano che sarà **true** se l'operazione è stata
// effettuata.

M. Malatesta A1-Classe File-06

7
12/04/2013

La classe **File**

- Schemi di utilizzo

```
try { File file = new File (percorso);  
    boolean success = file.createNewFile();  
    }  
catch (IOException e) { }
```

Codice di esempio per la
creazione di un file.

```
try { File file = new File (percorso);  
    boolean success = file.delete();  
    }  
catch (IOException e) { }
```

Codice di esempio per la
cancellazione di un file.

M. Malatesta A1-Classe File-06

8
12/04/2013

La classe File

- Schemi di utilizzo

Sebbene non prevedano l'uso della classe **File** li riportiamo per uniformità

```
try { BufferedReader infile =  
    new BufferedReader (new FileReader(percorso));  
    String str;  
    while ((str = infile.readLine()) != null) { elabora str; }  
    infile.close();  
}  
catch (IOException e) { }
```

Codice di esempio per la lettura da file.

```
try { BufferedWriter outfile =  
    new BufferedWriter (new FileWriter (percorso));  
    outfile.write(str);  
    outfile.close();  
}  
catch (IOException e) { }
```

Codice di esempio per la scrittura su file.

M. Malatesta A1-Classe File-06

9
12/04/2013

La classe File

- Esempio di uso di File

```
import java.io.*;  
public class Dir  
{ public static void main (String args[])  
    { File directory = new File("C:/Documents and Settings/Mauro/Desktop");  
      String[] lista = directory.list();  
      if (lista == null)  
          { System.out.println("La directory non esiste"); }  
      else  
          { for (int i=0; i<lista.length; i++)  
              System.out.println (lista[i]);  
          }  
      }  
}
```

Questa applicazione elenca i file presenti in una directory assegnata.

M. Malatesta A1-Classe File-06

10
12/04/2013

Argomenti

- La classe **File**
 - I costruttori
 - I metodi più importanti di **File**
 - Schemi di utilizzo
 - Esempio di uso di **File**

M. Malatesta A1-Classe File-06

11
12/04/2013

Altre fonti di informazione

- P.Gallo, F.Salerno – Informatica Generale 2, ed. Minerva Italica
- Camagni – Nicolassy – Interfacce grafiche e programmazione concorrente, ed. Hoepli

M. Malatesta A1-Classe File-06

12
12/04/2013