

## (A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti:

- Classe **File**
- **public boolean exists()**
- **public boolean isFile()**
- **public boolean isDirectory()**
- **public long length()**
- **public boolean renameTo(dest)**
- **public boolean delete()**
- **public String getName()**
- **public String getPath()**
- **String[ ] list ()**
- **boolean mkdir()**
- **File f = new File (percorso);**
- **File f = new File (percorso, nomefile);**
- **File f = new File (directory, nomefile);**

## (B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

1) *Conoscenza*

1. Che *relazione esiste* tra la classe **File** e il sistema operativo?
2. Sono possibili operazioni di input e/o output sulla classe **File**? Perché?

2) *Competenza*

1. Qual è la *sintassi dei costruttori* della classe **File**?
2. Qual è il compito dei principali metodi della classe **File**?

## (C) ESERCIZI DI COMPrensione

1. La classe **File**, consente di scrivere applicazioni che operano sul ..... e che consentono , per accedere ad informazioni su ..... e su .....
2. Nella classe **File**, , non sono definiti metodi per la ..... e ..... che sono invece trattati dalle classi ..... ed .....
3. Scrivere il codice di esempio per la creazione di un file:

4. Scrivere il codice di esempio per la cancellazione di un file:

5. Scrivere il codice di esempio per la lettura da file:

6. Scrivere il codice di esempio per la scrittura su file:

7. Scrivere la sintassi e l'effetto dei costruttori della classe **File**:

Sintassi	Effetto

8. Associare, per ciascuno dei metodi indicati a destra, la corrispondente azione indicata a sinistra, scrivendo la lettera corrispondente nella prima colonna.

1		verifica l'esistenza di un file	A	<b>String[] list()</b>
2		cancella il file dal disco	B	<b>public boolean isFile()</b>
3		verifica se è un file normale	C	<b>public boolean delete()</b>
4		Crea una directory	D	<b>public long length()</b>
5		verifica se è una directory	E	<b>public String getPath()</b>
6		elenco dei file presenti	F	<b>public boolean exists()</b>
7		cambia il nome del file	G	<b>public boolean isDirectory()</b>
8		restituisce il <i>pathname</i>	H	<b>boolean mkdir()</b>
9		restituisce il nome del file	I	<b>public boolean renameTo(<i>dest</i>)</b>
10		restituisce la lunghezza del file	L	<b>public String getName()</b>

9. Scrivere l'effetto di ciascuno dei costruttori insequenti:

Costruttore	Effetto
<b>File f = new File("C:\prova.txt");</b>	
<b>File cartella = new File("C:\cartella");</b>	
<b>File f = new File("C:\", "prova.txt");</b>	
<b>File f = new File(cartella, "prova.txt");</b>	

#### (D) ESERCIZI DI APPLICAZIONE

1. (**Esercizio risolto**). Scrivere un'applicazione *FileSize* che conti il numero di caratteri presenti in un file. Se il file è una directory, il programma deve elencare i nomi dei file che sono contenuti in essa.  
**Soluzione.** Supponiamo che il programma debba essere lanciato da linea di comando, per cui il nome del file viene immesso in `args[0]`. Dopo aver controllato che questo parametro sia presente, si può istanziare una funzione *leggi(filename)* che crea un oggetto **File** con nome *filename*, ne controlla l'esistenza e se è una directory, elenca i file in essa presenti con il metodo **list()**, se è un file ne stampa la lunghezza tramite il metodo **length()**;

```
import java.io.*;
public class FileSize
{
    static public void main (String argv[])
    {
        if(argv.length != 1)
            System.err.println("Uso: java FileSize nomefile");
        else
            leggi(argv[0]);
    }
    static private void leggi(String filename)
    {
        long dim = 0;
        File f = new File(filename);
        if (f.exists())
        {
            if (f.isDirectory())
            {
                String lista[] = f.list();
                for (int i = 0; i < lista.length; i++)
                    System.out.println("nome file = " + filename + "/" + lista[i]);
            }
        }
    }
}
```

```
        else
        {   dim = f.length();
            System.out.println("Dimensione " + filename + " = " + dim);
        }
    }
} // fine metodo leggi
}
```

2. Scrivere un'applicazione che crei una directory con nome e percorso dati, dopo aver controllato che non esista. Se esiste, viene creato un messaggio a video.
3. Scrivere un'applicazione che cancelli un file con nome e percorso dati, dopo aver controllato che esista. Se non esiste, viene creato un messaggio a video.
4. Scrivere un'applicazione che dato il nome e il percorso di un file, consenta di rinominarlo, dopo aver controllato che esista. Se non esiste, viene creato un messaggio a video.