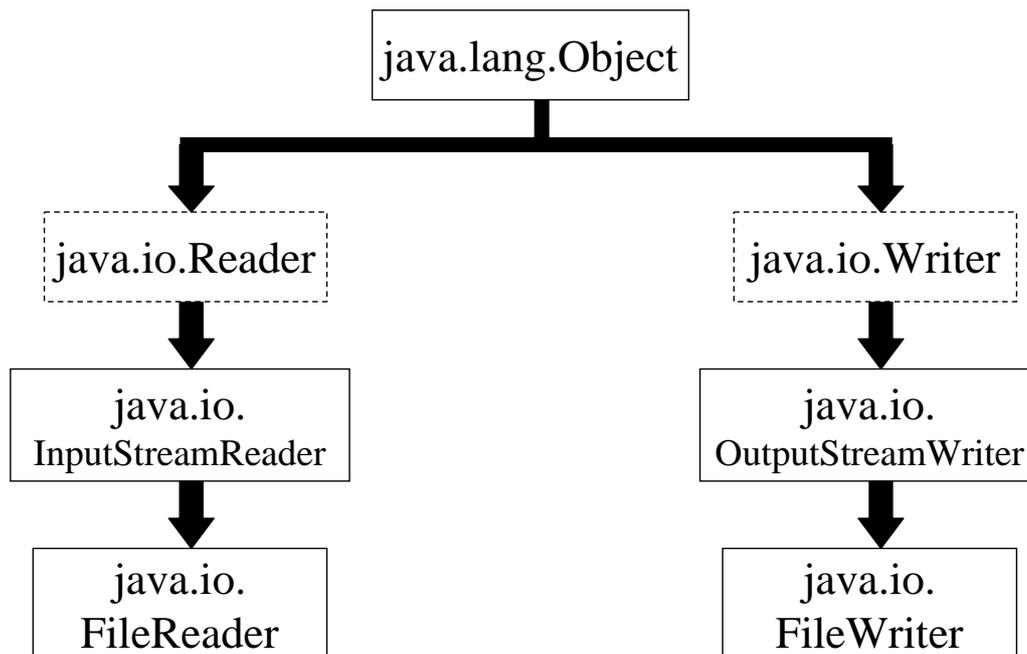


Esercitazione n° 4

Obiettivi:

- Capacità di *analisi* e di *estensione* di progetti esistenti
- Il concetto di *filtro*
- Linguaggio Java:
 - Gestione di *input/output* da console e da file (*package java.io*)
 - Utilizzo di classi per I/O *a carattere*:
 - *Reader/Writer* e loro specializzazioni (*InputStreamReader, FileReader; OutputStreamWriter, FileWriter*)

Il package `java.io`: classi per I/O a carattere



Esercizio (1)

- Utilizzo delle classi di `java.io` per *I/O a carattere*
- *Conteggio* del numero di linee contenute in file e realizzazione di un *filtro*

Numero qualsiasi di parametri di invocazione **N**:

- N >= 2** i primi (N-1) sono nomi di file di ingresso
ultimo è nome del file di uscita
- N = 1** il parametro è il file di uscita
ingresso da console (**System.in**)
- N = 0** ingresso da **System.in** e uscita su **System.out**

Esercizio (2)

- *Conteggio* del numero totale di linee presenti nell'insieme dei file di ingresso
 - utilizzo, per ciascuno dei file, del metodo statico:
`public static int quanteRighe(String fileName)`
- *Filtro*: scrive sul file di uscita il risultato di una *trasformazione* sui caratteri del file di ingresso (compreso il caso di ingresso da **System.in**)
 - utilizzo del metodo statico:
`public static void filtra(String in, String out)`
`public static void filtra(String fileOut)`
`public static void filtra()`

La classe **Trasformazione**

- La *trasformazione* da operare sui singoli caratteri dei file di ingresso è contenuta nella classe di utilità **Esercitazione4.Trasformazione**

```
public class Trasformazione {  
  
    public static char trasforma(char ch) {  
  
        if (ch == 'p') return '?';  
        else return ch;  
    }  
}
```

La classe **MainClass**: il metodo **main()**

```
public class MainClass {  
  
    public static void main(String[] args) {  
  
        int temp=0, totale=0;  
  
        if (args.length >= 2)  
            for (int i=0; i < args.length-1; i++) {  
                temp = quanteRighe(args[i]);  
                System.out.println("Il file "+args[i]+"  
                    contiene: "+temp+" linee.");  
                totale = totale + temp;  
                filtra(args[i], args[args.length-1]);  
            }  
    }  
}
```

La classe `MainClass`: il metodo `main()`

```
public class MainClass {  
  
    ...  
  
    else if (args.length == 1)  
        filtra(args[0]);  
    else filtra();  
}
```

La classe `MainClass`: il metodo `quanteRighe()`

```
public static int quanteRighe(String fileName) {  
  
    FileReader in = null;  
    int lineCount; int x;  
  
    try { in = new FileReader(fileName);  
    }  
    catch (FileNotFoundException e) {  
        System.out.println("Errore. Non riesco ad  
            aprire il file "+fileName);  
        System.exit(1);  
    }  
    lineCount = 1;
```

La classe `MainClass`: il metodo `quanteRighe()`

```
try { while ((x = in.read()) >= 0) {
    char ch = (char) x;
    if (ch=='\n') lineCount++; } }
catch (IOException e) {
    System.out.println("Errore. Problema nella
        lettura dal file "+fileName);
    System.exit(2); }
return lineCount;
try { in.close(); }
catch (IOException e) {
    System.out.println("Errore. Problema nella
        chiusura del file "+fileName);
    System.exit(3); }
```

Sintesi di codice: 3 versioni del metodo `filtra()`

- 1) `public static void filtra(String in,
String out)`
- 2) `public static void filtra(String
fileOut)`
- 3) `public static void filtra()`

Parte II: e se avessi scritto `MainClass.main()`...

```
public static void main(String[] args) {  
  
    int temp=0, totale=0;  
    FileWriter out = null;  
  
    if (args.length >= 2) {  
        try {  
            out = new FileWriter(args[args.length-1],  
                true); }  
        catch (IOException e) {  
            System.out.println("Errore. Non riesco ad  
                aprire il file in scrittura.");  
            System.exit(1); }  
    }  
}
```

Parte II: e se avessi scritto `MainClass.main()`...

```
for (int i=0; i < args.length-1; i++) {  
    FileReader in = null;  
    try {  
        in = new FileReader(args[i]); }  
    catch (FileNotFoundException e) {  
        System.out.println("Errore. Non riesco  
            ad aprire il file " + args[i]);  
        System.exit(2); }  
    temp = quanteRighe(in);  
    System.out.println("Il file "+args[i]+  
        "contiene: "+temp+" linee.");  
    ...  
}
```

Parte II: e se avessi scritto `MainClass.main()`...

```
try { in.close();
    in = new FileReader(args[i]); }
// semplicemente riporta il file all'inizio
catch (IOException e) {
    System.out.println("Errore. Non riesco
        a riaprire il file in lettura.");
    System.exit(3);
}
filtra(in, out); }
System.out.println("Totale linee: " + totale);
try { out.close(); }
catch (IOException e) {
    System.out.println("Errore."); ... }
```

Parte II: e se avessi scritto `MainClass.main()`...

```
else if (args.length == 1) {
    try {out = new FileWriter(args[0],true); }
    catch (IOException e) {
        System.out.println("Errore. Non riesco
            ad aprire il file in scrittura.");
        System.exit(3); }
    filtra(System.in, out);
    }
    try { out.close(); }
    catch (IOException e) {
        System.out.println("Errore."); ... }
else filtra(System.in, System.out);
}
```

Parte II: scrivere 3 versioni di `filtra()`...

- 1) `public static void filtra`
`(InputStreamReader in, OutputStreamWriter out)`

- 2) `public static void filtra`
`(InputStream in, OutputStreamWriter out)`

- 3) `public static void filtra`
`(InputStream in, OutputStream out)`