



Agent and Object Technology Lab  
Dipartimento di Ingegneria dell'Informazione  
Università degli Studi di Parma



Ingegneria del software A

Input e Output

**Prof. Agostino Poggi**

- ◆ Java fornisce una libreria, corrispondente al package `java.io`, contenente un insieme di classi per gestire le operazioni di input/output
- ◆ Questa libreria ha due scopi:
  - Fornire gli strumenti per accedere al file system
  - Fornire gli strumenti per leggere e scrivere dati

- ◆ Un'istanza della classe `File` rappresenta un descrittore di un file usato all'interno di un programma
- ◆ Un descrittore di file viene creato in base al **pathname** del file:

```
File f = new File(pathname);
```

- ◆ Se questo file non esiste nel file system, allora può essere creato:

```
F.createNewFile();
```

# Esempio: stampa informazioni su un file

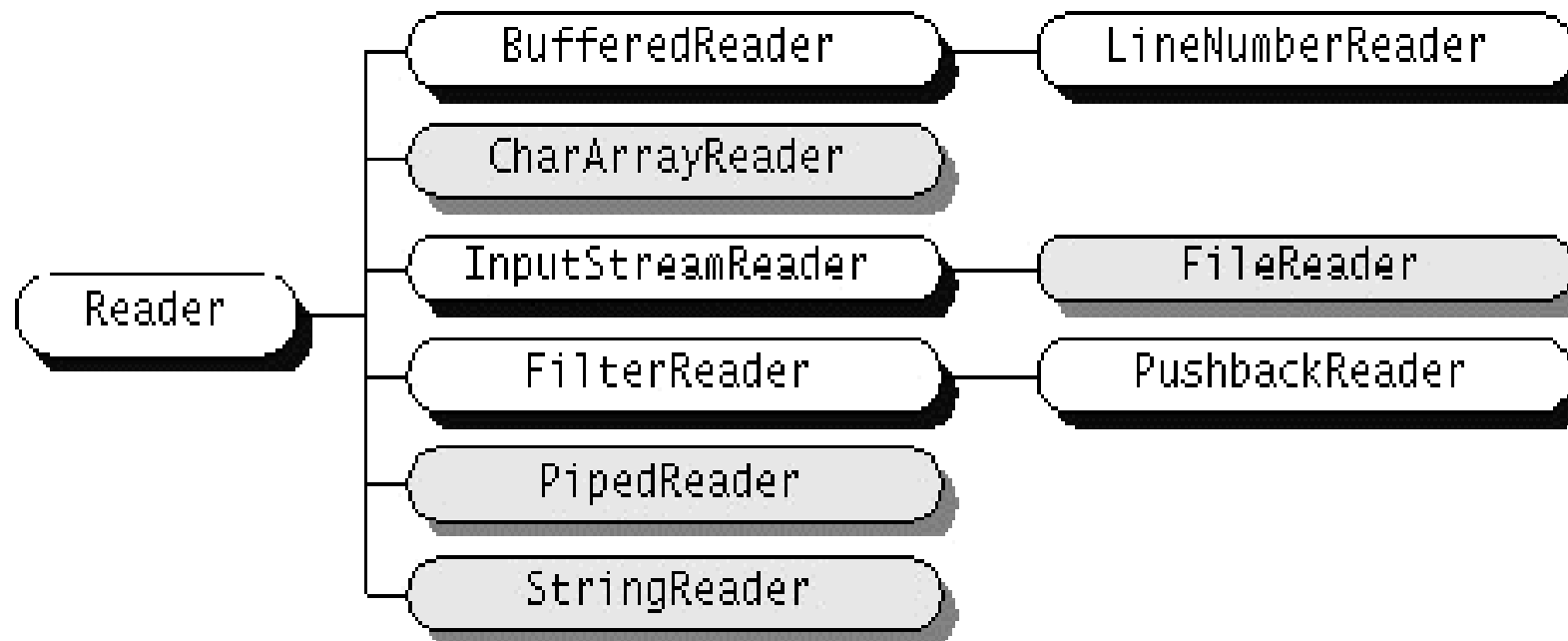
---

```
import java.io.*;

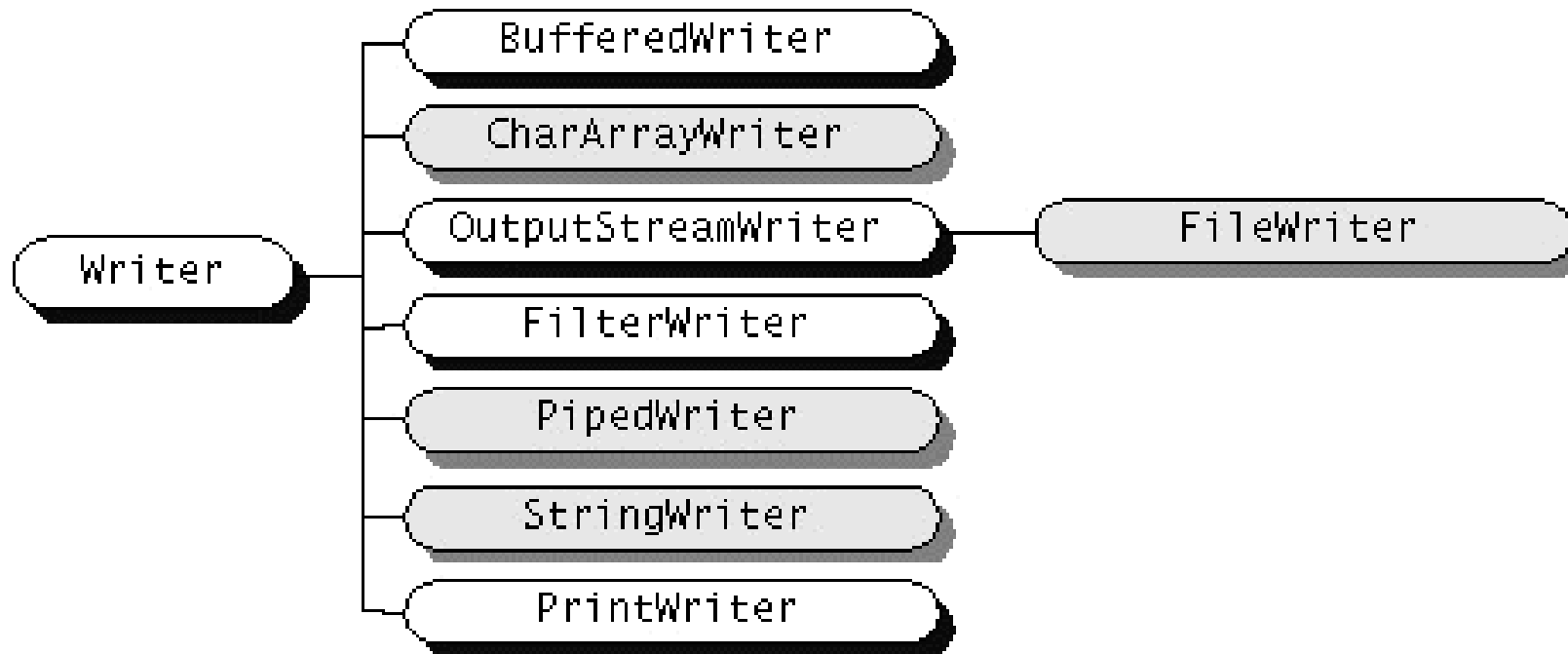
class FileInfo {
    public static void main (String[] args) throws IOException {
        File f = new File("pippo.txt");
        System.out.println(f.exists());
        if (!f.exists())
            f.createNewFile();
        System.out.println(f.exists());
        System.out.println(f.isFile());
        System.out.println(f.isDirectory());
    }
}
```

23/FileInfo.java

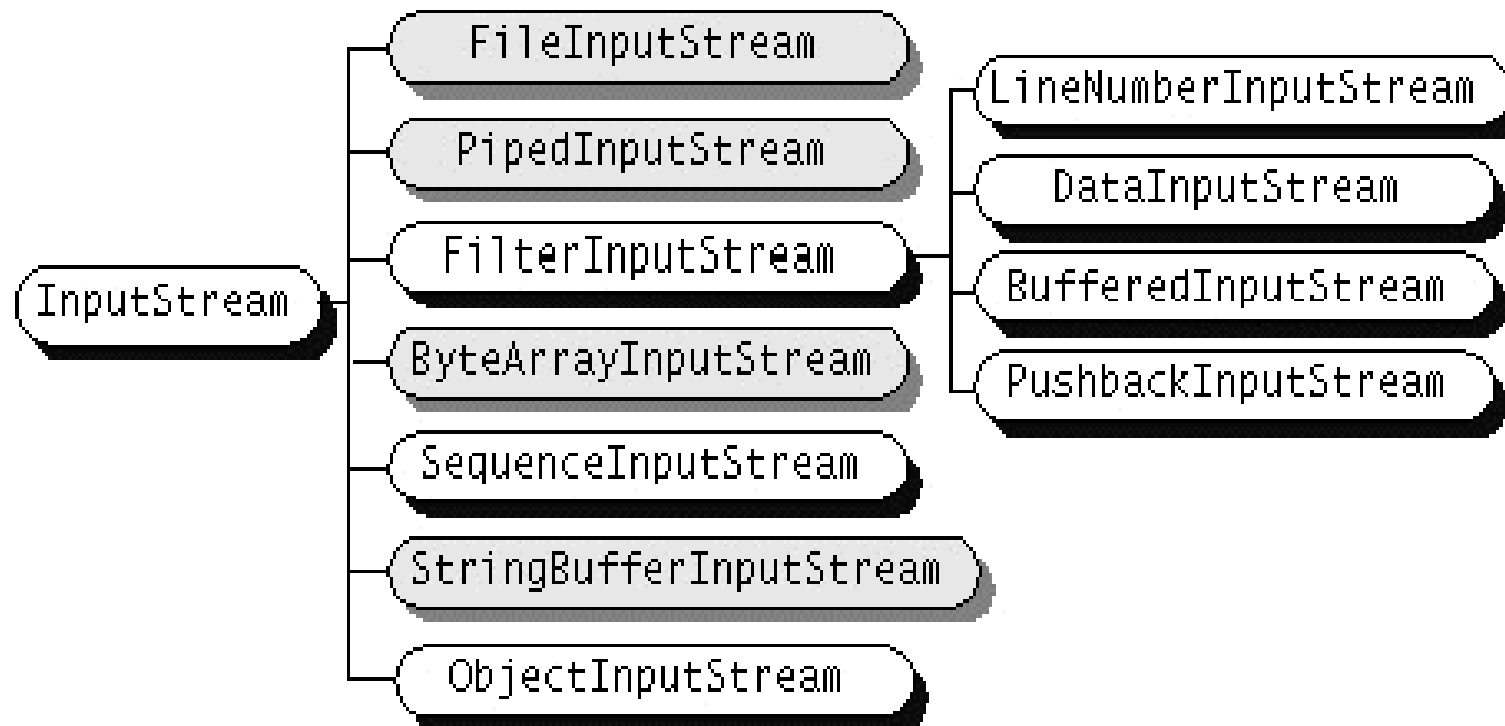
- ◆ La lettura e la scrittura dei dati si basa sul concetto di **stream**
- ◆ Gli stream si dividono in input e output stream e individuano i due lati estremi di un canale di comunicazione mono-direzionale
- ◆ Questi canali non rappresentano solo i file, ma anche: i collegamenti via rete, i buffer in memoria, le tastiera e i video di un computer, ...
- ◆ Gli stream offrono un'interfaccia uniforme per le operazioni di input/output



```
int read()  
int read(char cbuf[])  
int read(char cbuf[], int offset, int length)
```

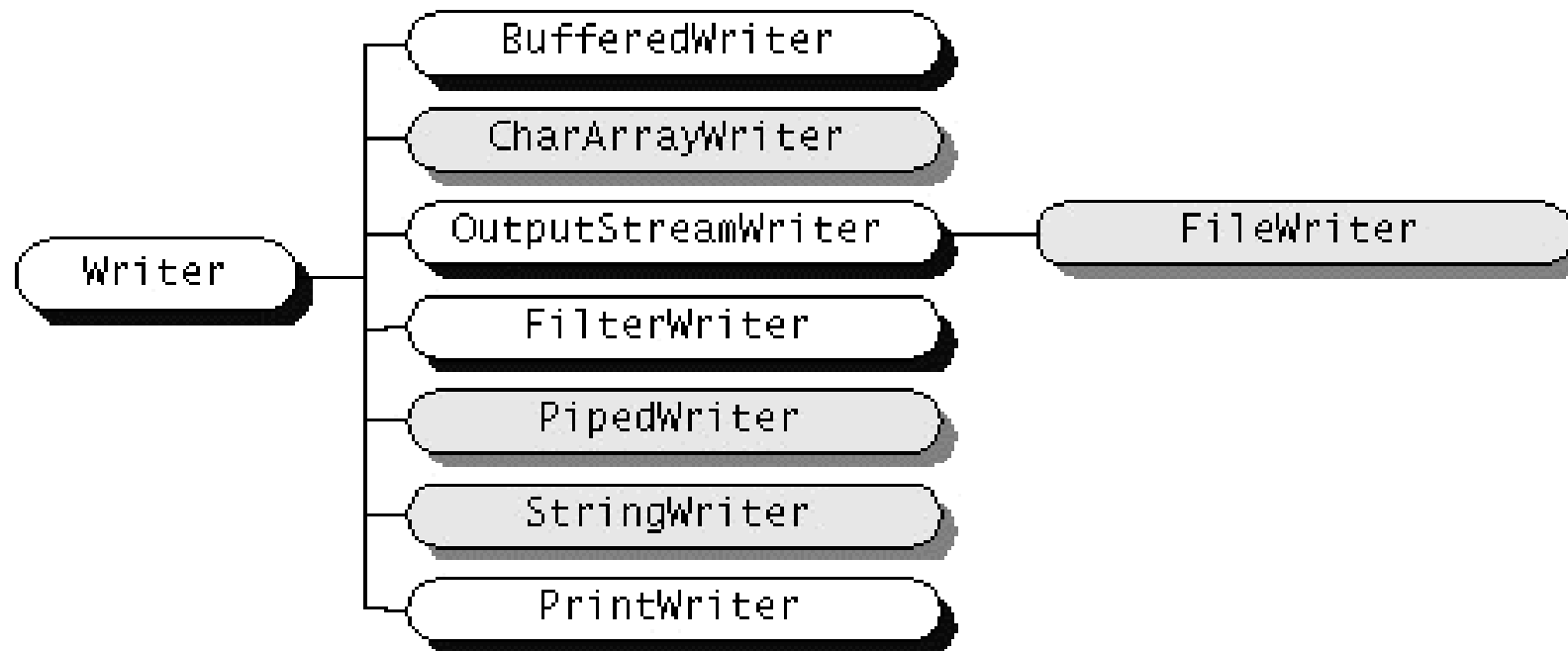


```
void write(char c)
void write(char cbuf[])
void write(char cbuf[], int offset, int length)
```



```
int read()
int read(byte cbuf[])
int read(byte cbuf[], int offset, int length)
```





```
void write(int b)
void write(byte cbuf[])
void write(byte cbuf[], int offset, int length)
```

```
Import java.io.*;

public class Main {
    public static void main(String[] args) {
        if (args.length != 2) {
            System.err.println("Usage: java Main <from> <to>");
            System.exit(-1); }

        File inputFile = new File(args[0]);
        File outputFile = new File(args[1]);

        try {
            FileReader in = new FileReader(inputFile);
            FileWriter out = new FileWriter(outputFile);
            int c;
            while ((c = in.read()) != -1) out.write(c);
            in.close(); out.close();
        } catch (FileNotFoundException fnfe) {
            System.err.println("File not found: " + args[0]);
        } catch (IOException e) {
            System.err.println("Unable to write file: " + args[1]); }
    }
}
```

24/Main.java

```
import java.io.*;
import java.util.Random;

public class Main {
    public static void main(String argv[]) throws IOException {
        BufferedReader console =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Quanti interi vuoi?");
        String input = console.readLine();
        int n = Integer.parseInt(input);
        System.out.println("Hai scritto " + n);

        Random randomGenerator = new Random();
        for (int i = 0; i < n; i++) {
            int randomInt = randomGenerator.nextInt(100);
            System.out.println("Generato : " + randomInt);
        }
    }
}
```

25/Main.java