

# 3.2 Escritura de Archivos en Java

Programación Avanzada

Junio 2017

Tecnologías Computacionales

Facultad de Estadística e Informática UV

# Objetivo

- ▶ Comprender los flujos de entrada a través de escritura de archivos

# Recordatorio con el Uso de los Flujos

## ▶ Lectura

- ▶ 1. Abrir flujo de datos (creación del objeto stream)
  - ▶ Teclado, Archivos, socket
- ▶ 2. Mientras existan datos disponibles
  - ▶ Leer datos
- ▶ 3. Cerrar el flujo

## ▶ Salida

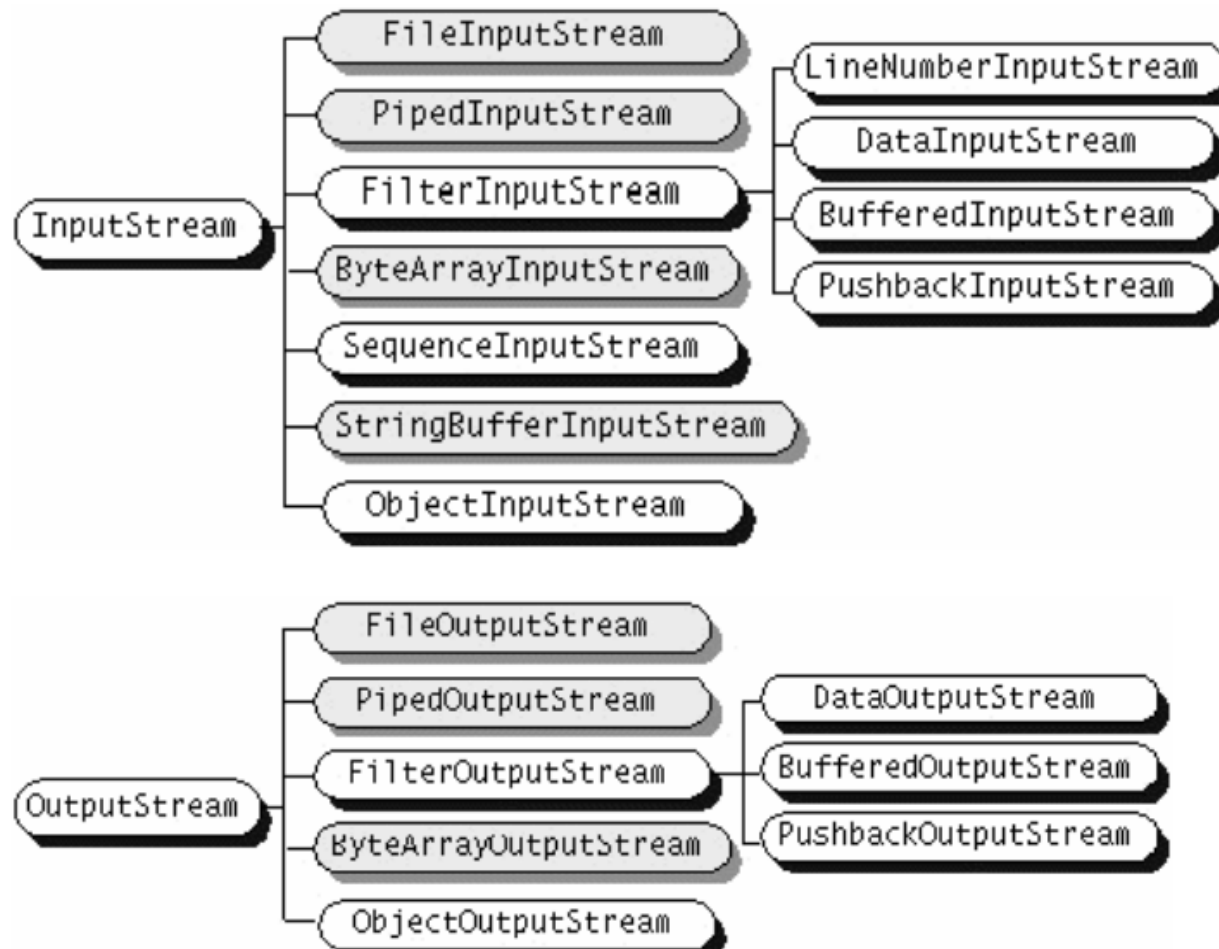
- ▶ 1. Abrir flujo de datos (creación del objeto stream)
  - ▶ Pantalla, archivo, socket
- ▶ 2. Mientras existan datos disponibles
  - ▶ Escribir datos
- ▶ 3. Cerrar el flujo
- ▶ Si ocurre un error se genera una excepción IOException

- ▶ \*En Java los flujos se implementan en el paquete java.io
- ▶ Fuentes de datos java.io.Reader y java.io.Writer

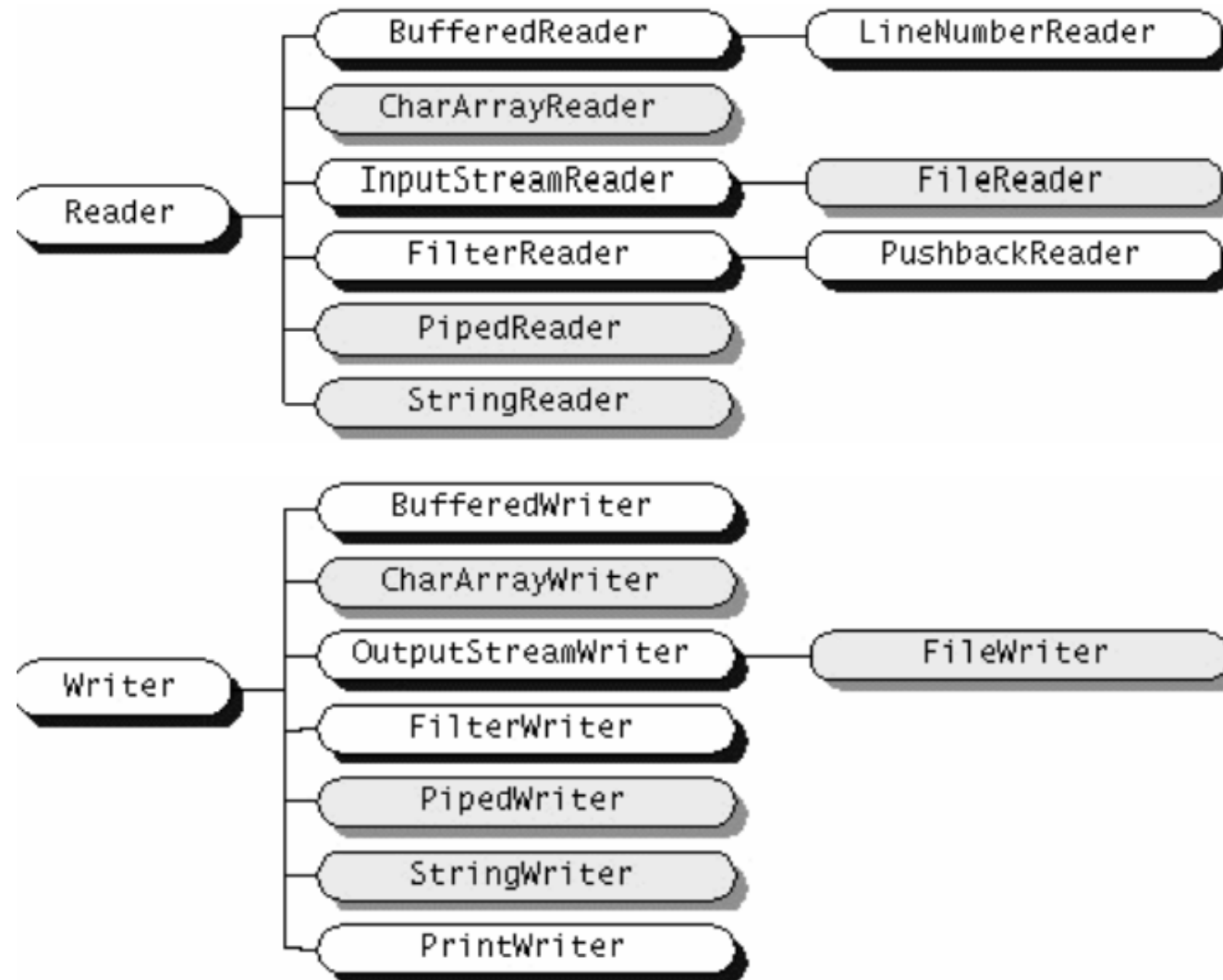
# Clasificación de los flujos

- ▶ Representación de la información
  - ▶ Flujos de bytes: clases InputStream y OutputStream
  - ▶ Flujos de caracteres: clases Reader y Writer
    - ▶ Es posible pasar de un flujo de bytes a uno de caracteres a través de InputStreamReader y OutputStreamWriter
- ▶ Propósito
  - ▶ Entrada: InputStream, Reader
  - ▶ Salida: OutputStream, Writer
  - ▶ Lectura/Entrada: RandomAccessFile
  - ▶ Transformación de los datos: Realizar algún tipo de procesamiento sobre los datos (BufferedReader y BufferedWriter)
- ▶ Acceso
  - ▶ Secuencial
  - ▶ Aleatorio (RandomAccessFile)

# Jerarquía de Clases para el Flujo de Bytes



# Jerarquía de Clases para el Flujo de Caracteres



# Ejemplo de Escritura (1/2)

## ► Ejemplo en Java:

```
import java.io.*;
public class Escritura{
    public static void main(String args[]){
        File f = new File("ejemplo.txt");
        //Escritura
        try{
            FileWriter w = new FileWriter(f);
            BufferedWriter bw = new BufferedWriter(w);
            PrintWriter pw = new PrintWriter(bw);
            pw.write(args[0]);
            pw.write(": continuando...");
            pw.close();
            bw.close();

        }catch(IOException e){
            System.out.println("Error en la escritura de archivo: " + e);
        }
    }
}
```

C:\WINDOWS\system32\cmd.exe

```
C:\Programacion\práctica 9 - Escritura de Archivos>java Escritura hola
C:\Programacion\práctica 9 - Escritura de Archivos>type ejemplo.txt
hola: continuando...
C:\Programacion\práctica 9 - Escritura de Archivos>
```

# Ejemplo de Escritura (2/2)

- ▶ En el ejemplo anterior es necesario incluir el código dentro de un **try** y un **catch** para el manejo de errores
- ▶ Será necesario el uso de 4 clases especiales para poder escribir, la clase **File**, **FileWriter**, **BufferedWriter** y **PrintWriter**, cada una hace lo siguiente:
  - ▶ **File**: esta clase es la esencial para crear/abrir un nuevo archivo, si un archivo con el mismo nombre se podría escribir información sobre el mismo
  - ▶ **FileWriter**: es un objeto que tiene como función escribir datos en un archivo
  - ▶ **BufferedWriter**: objeto que reserva un espacio en memoria donde se guarda la información antes de ser escrita en un archivo
  - ▶ **PrintWriter**: Es el objeto utilizado para escribir directamente sobre el archivo de texto



# Ejemplo de Escritura Agregando a un Archivo Existente

## ► Ejemplo en Java:

```
import java.io.*;

public class Escritura{
    public static void main(String args[]){
        File f = new File("ejemplo.txt");
        //Escritura
        try{
            FileWriter w = new FileWriter(f, true);
            BufferedWriter bw = new BufferedWriter(w);
            PrintWriter pw = new PrintWriter(bw);
            pw.write(args[0]);
            pw.write(": continuando...\n");
            pw.close();
            bw.close();

        }catch(IOException e){
            System.out.println("Error en la escritura de archivo: " + e);
        }
    }
}
```

C:\WINDOWS\system32\cmd.exe

```
C:\Programacion\práctica 9 - Escritura de Archivos>java EscrituraAgregar Linea1
C:\Programacion\práctica 9 - Escritura de Archivos>java EscrituraAgregar Linea2
C:\Programacion\práctica 9 - Escritura de Archivos>java EscrituraAgregar Linea3
C:\Programacion\práctica 9 - Escritura de Archivos>type ejemplo_agregar.txt
Linea1: continuando...
Linea2: continuando...
Linea3: continuando...
C:\Programacion\práctica 9 - Escritura de Archivos>_
```