



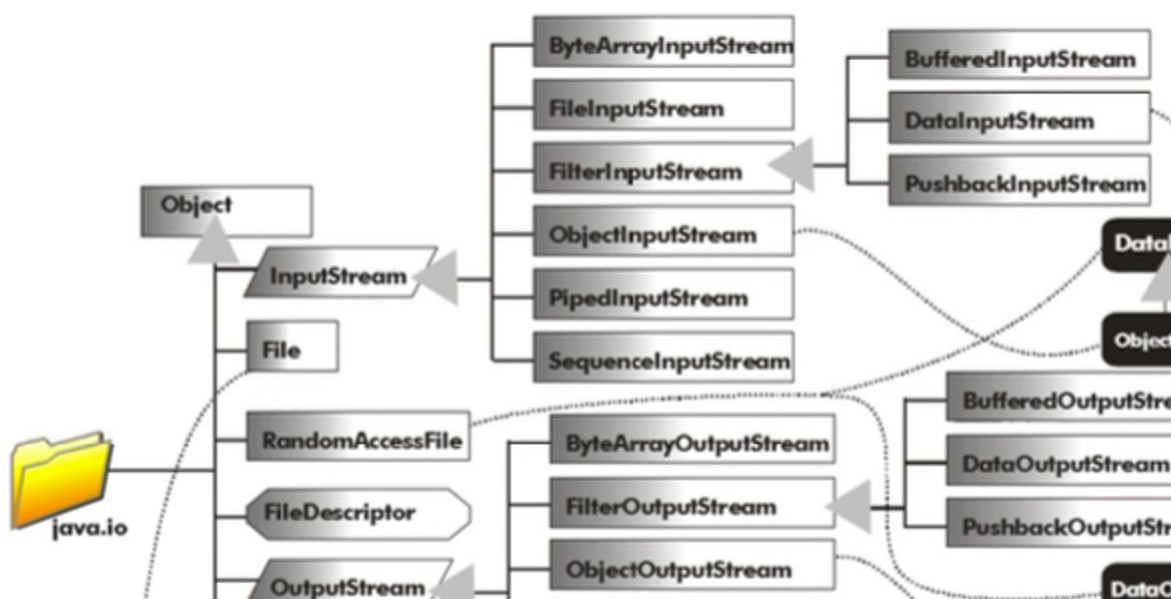


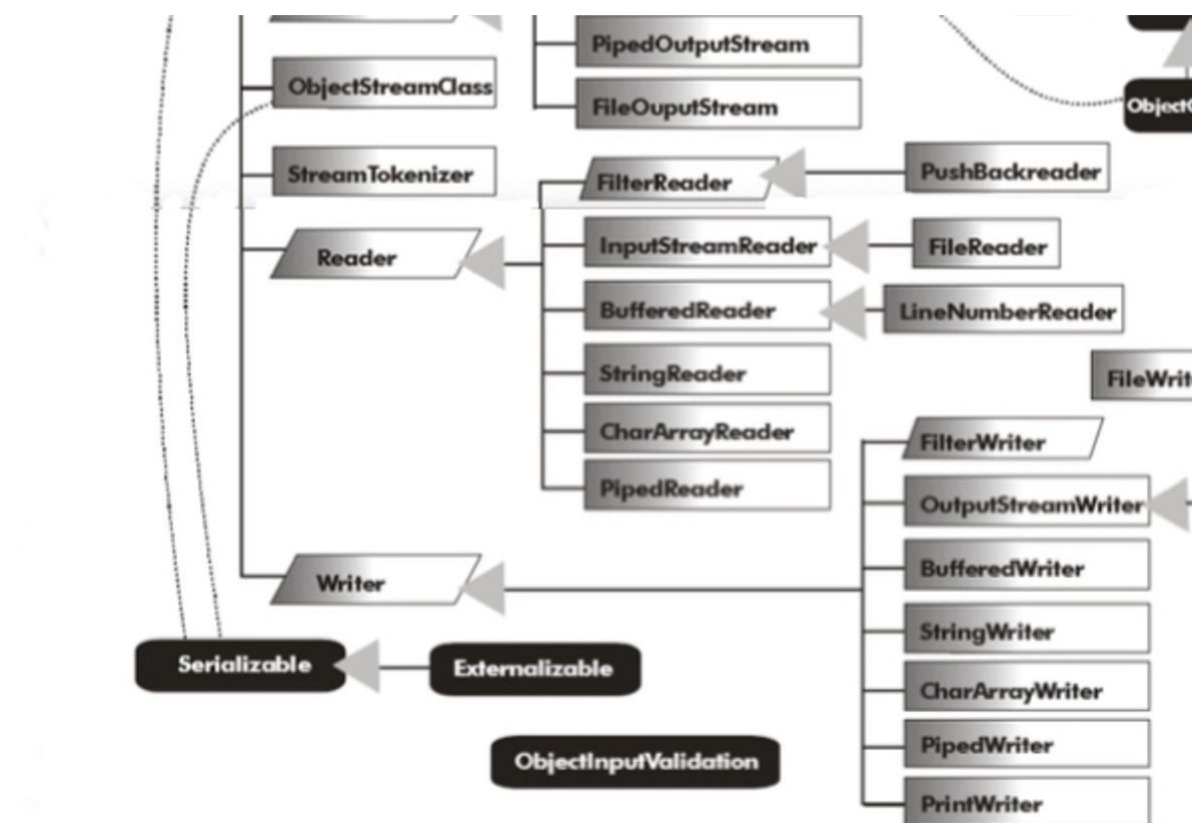


2. Entradas/Salida por teclado

 Date	Empty
 Status	Not started
 Type	Empty
Units	 8. Flujos de datos

Jerarquía de clases





Entrada y salida por teclado

Para la entrada de datos se utilizan las clases de Scanner y BufferedReader

Scanner

El uso de la clase Scanner es una de las mejores maneras de ingresar datos por teclado en Java.

Scanner es una clase en el paquete java.util utilizada para obtener la entrada de los tipos primitivos como int, double etc. y también String. Es la forma más fácil de leer datos en un programa Java

Para poder definir una variable de tipo Scanner en una aplicación se tendrá que importar dicha clase en la aplicación de la siguiente manera:

```
Scanner in = new Scanner(System.in);
```

Existen métodos específicos para cada uno de los datos los cuales son los siguientes:

```
a = in.next(); //a es de tipo String
a = in.nextByte(); //a es de tipo byte
a = in.nextShort(); //a es de tipo short
a = in.nextInt(); //a es de tipo int
a = in.nextLong(); //a es de tipo long
a = in.nextFloat(); //a es de tipo float
introducido con , a = in.nextDouble(); //a es de tipo double
introducido con , a = in.nextBoolean(); //a es de tipo boolean
a = in.next().charAt(0); //a es de tipo char
a = in.nextLine(); //a es de tipo String
// la ejecución de in.nextLine() vacía el buffer de lectura recomendado tras la última lectura
```

BufferedReader

Clase similar a la anterior con la diferencia que ofrece menos métodos con los que poder trabajar, ya que todas las lecturas que realiza las convierte a String

```
BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
try { String nombre = bufferedReader.readLine(); } catch (IOException e) { e.printStackTrace(); }
}
```

Para la salida de datos se utiliza la clase `System.out.print`