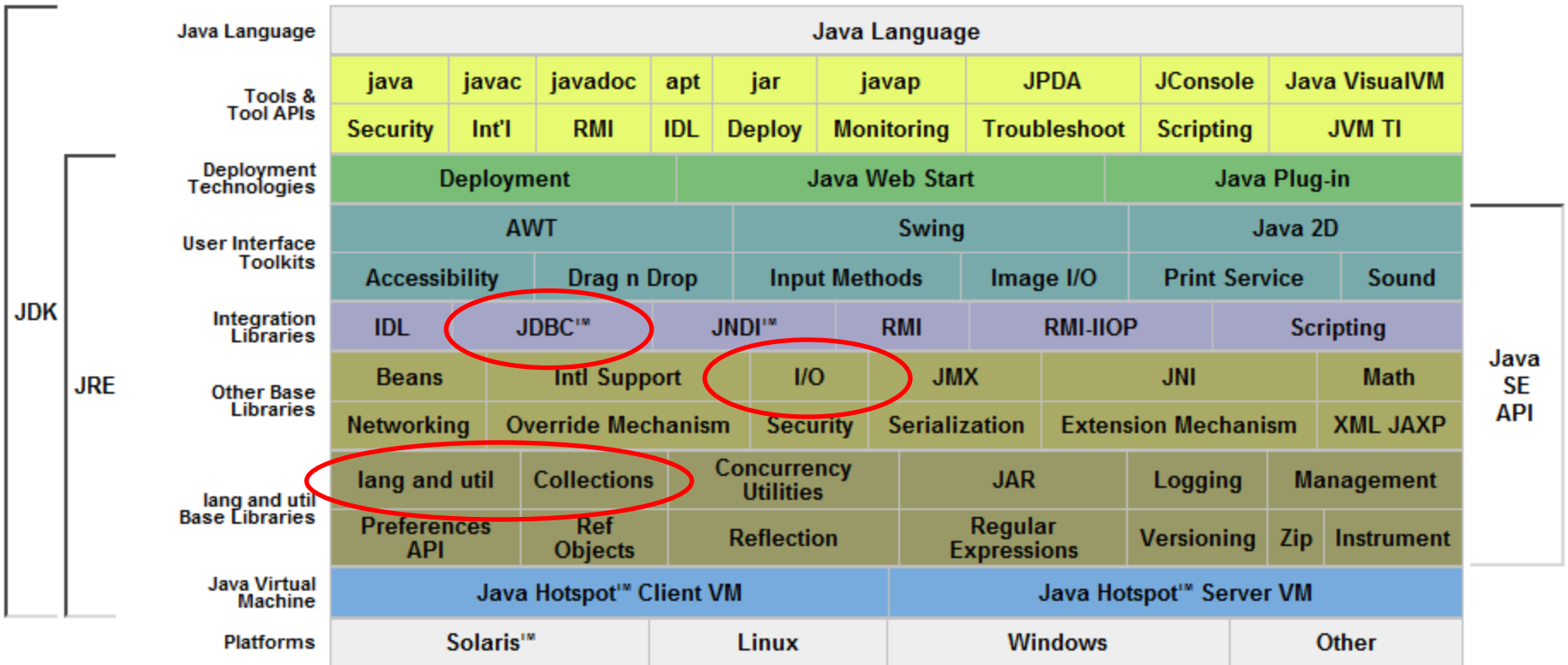


**MASTER EN MODELIZACIÓN
MATEMÁTICA, ESTADÍSTICA Y
COMPUTACIÓN
2016-2017**

Curso: Bases de datos y programación
orientada a objetos
Parte POO

4 - Biblioteca de clases (clases de interés)

Entorno de desarrollo de Java



The screenshot displays the Java Platform Standard Ed. 7 documentation for the `java.lang.Object` class. The interface includes a navigation menu at the top with options like 'Overview', 'Package', 'Class', 'Use', 'Tree', 'Deprecated', 'Index', and 'Help'. The 'Class' tab is active, showing the class name and a brief description: 'Class Object is the root of the class hierarchy. Every class has Object as a superclass. All objects, including arrays, implement the methods of this class.' Below this, there are sections for 'Constructor Summary' and 'Method Summary'. The 'Constructor Summary' section shows a table with one entry: 'Object ()'. The 'Method Summary' section shows a table with three entries: 'clone ()', 'equals (Object obj)', and 'finalize ()'. The left sidebar contains a list of packages and classes, with 'Object' highlighted. Two red arrows point from the sidebar to the 'Object' class name in the main content area.

Clases básicas

java.lang.Object

- ✓•Clone
- ✓•Equals
- ✓•toString
- ✓•hashCode
- Finalize
- getClass

java.lang.System

- ✓•getenv
- ✓•getProperty, getProperties, setProperty, setProperties y clearProperty
- ✓•Exit
- ✓•Arraycopy
 - setIn, setOut y setErr

java.lang.<objetos asociados a tipos>

- ✓java.lang.String (operador +, main); java.lang.StringBuffer✓
- java.lang.Math; java.lang.StrictMath;
- java.lang.Throwable
- (otras)✓Thread, Process, SecurityManager, ClassLoader, Compiler, Runtime, etc

java.lang

- ✓•Java.util.Date; java.util.Calendar;✓
- Java.util.BitSet
- Java.util.Random
- Java.util.Timer; java.util.TimerTask

✓•Java.util.Properties

- Java.util.ResourceBundle

Java.util.Scanner; java.util.Formatter✓

FRAMEWORK COLECCIONES

- Interfaz List y clases Vector, Stack, ArrayList y LinkedList
- Interfaces Map y SortedMap, y clases Hashtable, HashMap, LinkedHashMap y TreeMap
- Interfaces Set y SortedSet, y clases HashSet, LinkedHashSet y TreeSet
- El interfaz Queue y la clase PriorityQueue.

- ✓•El interfaz Comparator.

- ✓•Arrays (search sort)

EL SUBPAQUETE java.util.zip✓

java.util

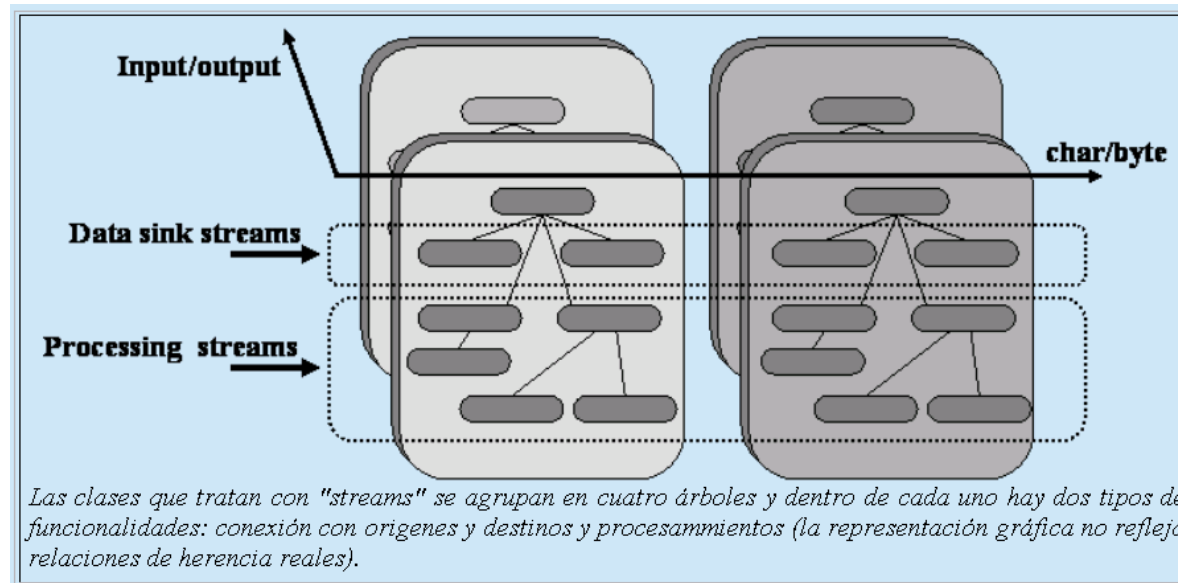
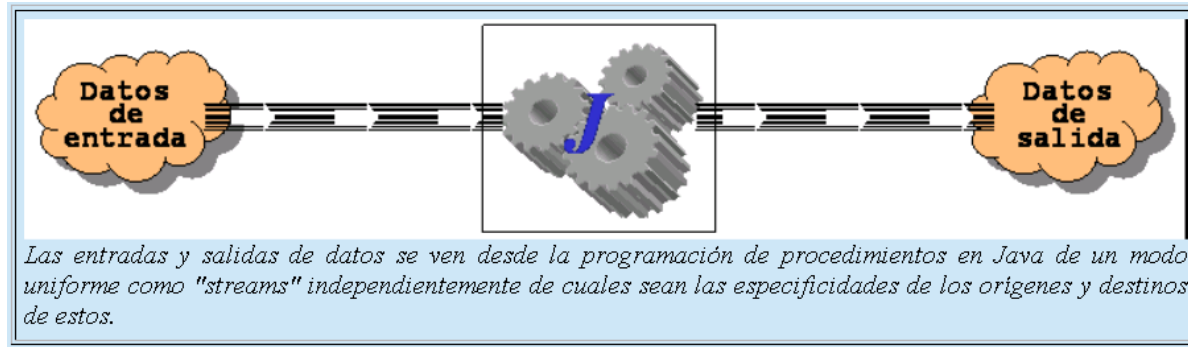
El "for" para colecciones

```
for (E e : Colección<E>) {}
```

p.ej.

```
List<String> sl=new ArrayList();  
//introducir elementos en sl  
for (String s:sl){  
    System.out.println(s);  
}
```

Clases básicas: entrada/salida



**Reader
Writer**

**InputStream
OutputStream**

| Origen/Destino | Streams de caracteres | Streams de Bytes |
|----------------|------------------------------------|---|
| Memoria | CharArrayReader CharArrayWriter | ByteArrayInputStream ByteArrayOutputStream |
| | StringReader ✓ StringWriter | StringBufferInputStream |
| "Pipes" | PipedReader PipedWriter | PipedInputStream PipedOutputStream |
| Ficheros ✓ | FileReader FileWriter | FileInputStream FileOutputStream |

Clases de entrada y salida de los orígenes y destinos básicos.

| Procesamientos | Streams de caracteres | Streams de Bytes |
|------------------------------------|---|---|
| Conversión de bytes a caracteres ✓ | InputStreamReader OutputStreamWriter | |
| Buffering ✓ | BufferedReader BufferedWriter | BufferedInputStream BufferedOutputStream |
| Filtrado ✓ | FilterReader FilterWriter | FilterInputStream ✓ FilterOutputStream |
| Concatenación ✓ | | SequenceInputStream |
| Conversión de datos | | DataInputStream DataOutputStream |
| Conteo | LineNumberReader | LineNumberInputStream |
| Peeking Ahead | PushbackReader | PushbackInputStream |
| Impresión ✓ | PrintWriter | PrintStream |
| Serialización de objetos ✓ | | ObjectInputStream ObjectOutputStream |

Clases para entradas y salidas con procesamiento de datos.

Un comentario sobre

- java.io.File
- java.io.RandomAccessFile



```
1- InputStreamReader entrada=new InputStreamReader(System.in);
2- BufferedReader entradaB=new BufferedReader(entrada);
3- //...
4- String s=entradaB.readLine();
```

```
1- BufferedReader entradaB=new BufferedReader(new InputStreamReader(System.in));
2- //...
3- String s=entradaB.readLine();
```

```
BufferedReader br=null;
try {
    br = new BufferedReader(new FileReader("MiFichero.txt"));
    String linea;
    while ( (linea=br.readLine()) != null){
        if (linea.trim().length()==0) continue;
        //TODO aquí lo que se quiera hacer con cada línea
    }
} catch (IOException ex) {
    ex.printStackTrace();
} finally {
    try {
        br.close();
    } catch (IOException ignore) {
        //se da tras FileNotFoundException
    }
}
```



```
try ( BufferedReader br = new BufferedReader(new FileReader("MiFichero.txt")) ){
    String linea;
    while ( (linea=br.readLine()) != null){
        if (linea.trim().length()==0) continue;
        //TODO aquí lo que se quiera hacer con cada línea
    }
} catch (IOException ex) {
    ex.printStackTrace();
}
```