MASTER EN MODELIZACIÓN MATEMÁTICA, ESTADÍSTICA Y COMPUTACIÓN
2011-2012

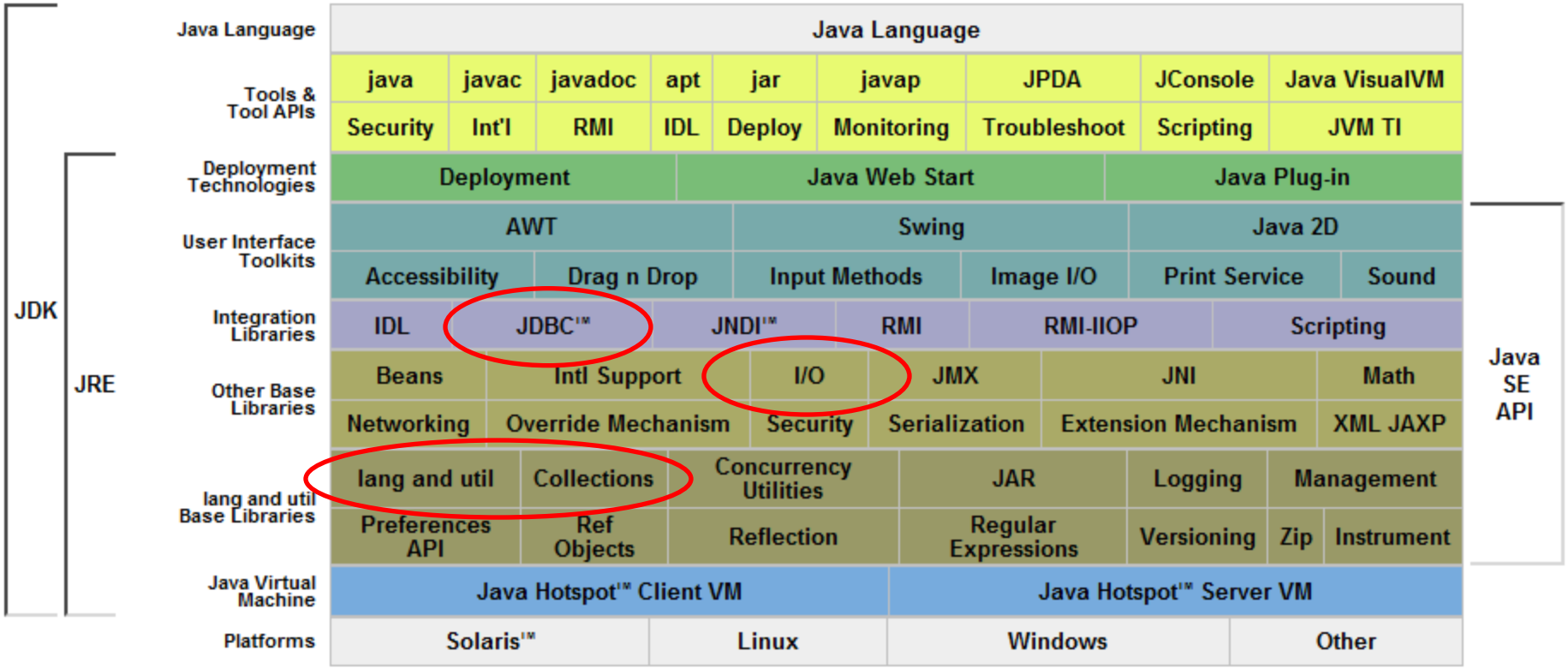
Curso: Bases de datos y programación orientada a objetos
Parte POO

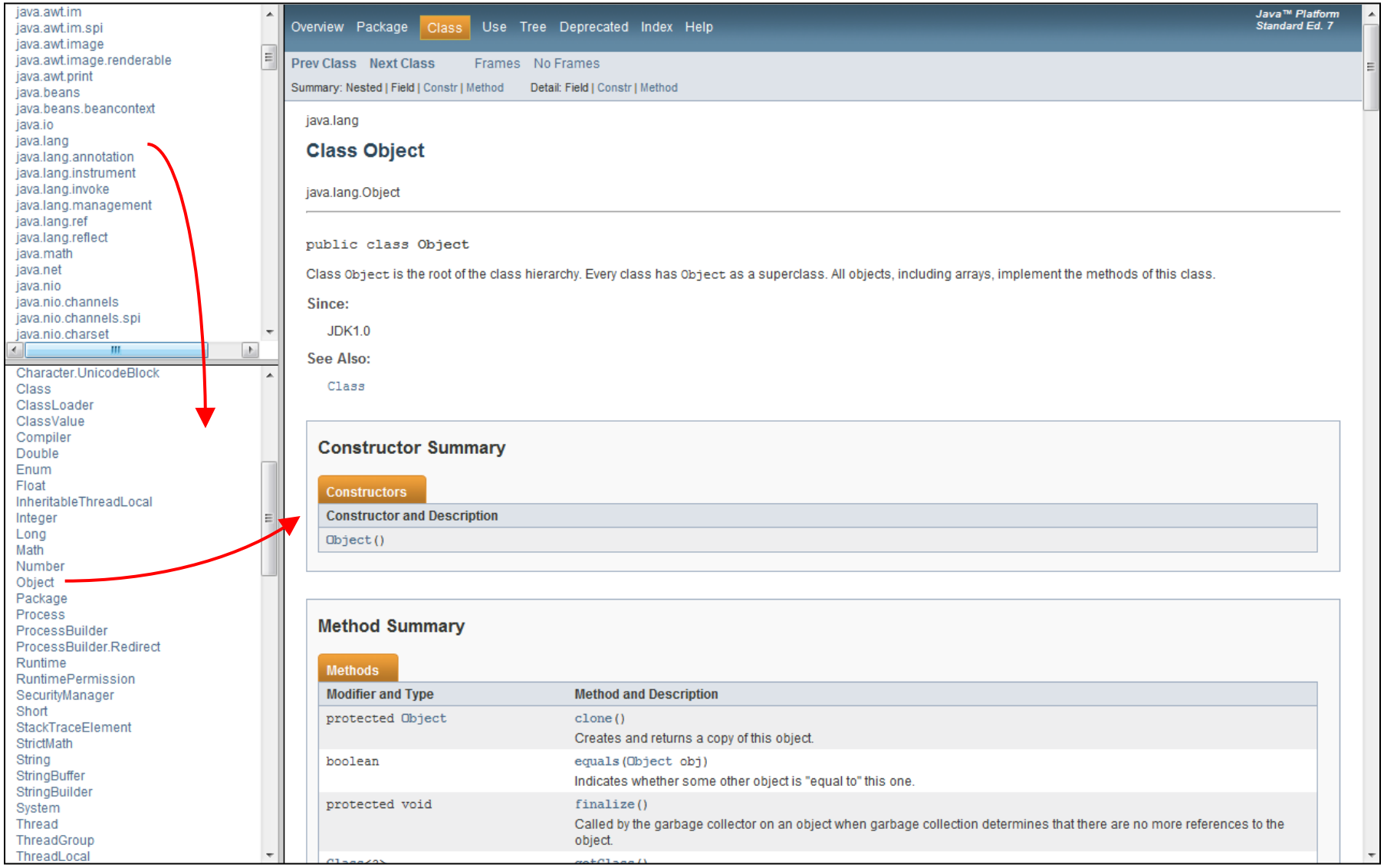
Librerías, estructura disponibilidad y uso.
Entrada / salida (I/O)

german.bordel@ehu.es
http://gtts.ehu.es/German

Entorno de desarrollo de Java

Imagen global del marco de trabajo Java




java.awt.im
java.awt.im.spi
java.awt.image
java.awt.image.renderable
java.awt.print
java.beans
java.beans.beancontext
java.io
java.lang
java.lang.annotation
java.lang.instrument
java.lang.invoke
java.lang.management
java.lang.ref
java.lang.reflect
java.math
java.net
java.nio
java.nio.channels
java.nio.channels.spi
java.nio.charset
Character.UnicodeBlock
Class
ClassLoader
ClassValue
Compiler
Double
Enum
Float
InheritableThreadLocal
Integer
Long
Math
Number
Object
Package
Process
ProcessBuilder
ProcessBuilder.Redirect
Runtime
RuntimePermission
SecurityManager
Short
StackTraceElement
StrictMath
String
StringBuffer
StringBuilder
System
Thread
ThreadGroup
ThreadLocal
Overview Package Class Use Tree Deprecated Index Help
Java™ Platform Standard Ed. 7
Prev Class Next Class Frames No Frames
Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method
java.lang
Class Object
java.lang.Object
public class Object
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.
Since:
JDK1.0
See Also:
Class
Constructor Summary
Constructors
Constructor and Description
Object()
Method Summary
Methods
Modifier and Type
Method and Description
protected Object
clone()
Creates and returns a copy of this object.
boolean
equals(Object obj)
Indicates whether some other object is "equal to" this one.
protected void
finalize()
Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.

Clases básicas

java.lang

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,ClassLoade,Compiler, Runtime,etc

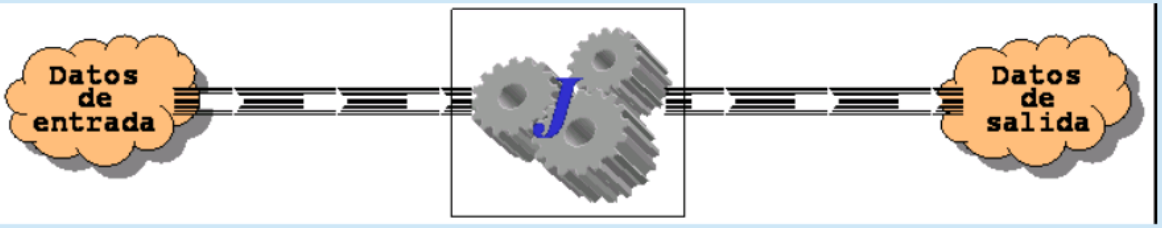
java.util

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

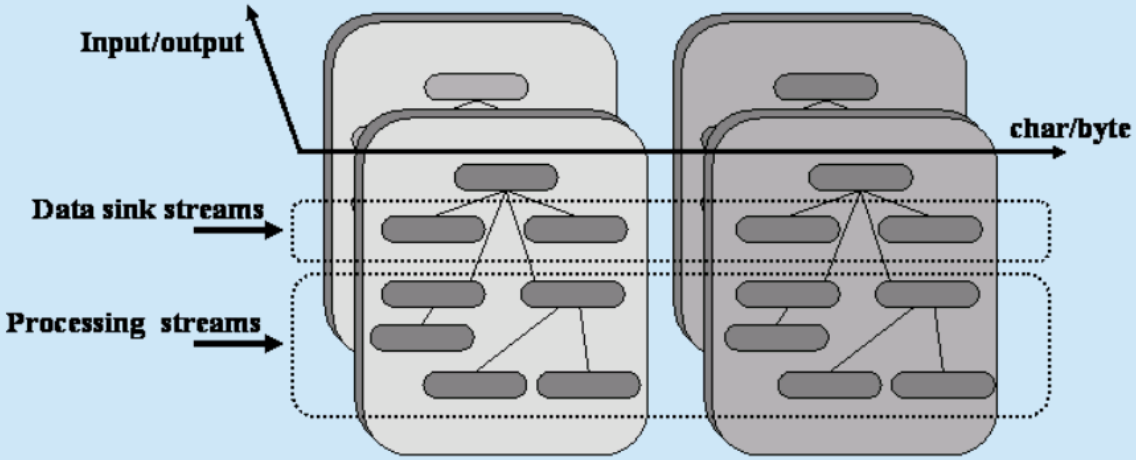
Java.util.Scanner; java.util.Formatter
FRAMEWORK COLECCIONES
- •Interfaz List y clases Vector, Stack, ArrayList y LinkedList
- •Interfaces Map y SortedMap, y clases Hashtable, HashMap, linkedHashMap y TreeMap
- •Interfaces Set y SortedSet, y clases HashSet, LinkedHashSet y Treeset
- •El interfaz Queue y la clase PriorityQueue.
- •El interfaz Comparator.
- •Arrays (search sort)

EL SUBPAQUETE java.util.zip

Clases básicas: entrada/salida



Las entradas y salidas de datos se ven desde la programación de procedimientos en Java de un modo uniforme como "streams" independientemente de cuales sean las especificidades de los orígenes y destinos de estos.



Las clases que tratan con "streams" se agrupan en cuatro árboles y dentro de cada uno hay dos tipos de funcionalidades: conexión con orígenes y destinos y procesammientos (la representación gráfica no refleja relaciones de herencia reales).

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.

```
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();
```

Comentarios sobre

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