

Prácticas POO

Curso 10/11

Alejandro Bellogín

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Marzo 2011

<http://www.eps.uam.es/~abellogin>

Esquema

- IDE
- Práctica 2
 - Java
 - Enunciado
 - Tareas
 - Librerías externas
- Más cosas de Java

Esquema

- **IDE**
- Práctica 2
 - Java
 - Enunciado
 - Tareas
 - Librerías externas
- Más cosas de Java

IDE

- Deberíais saber
 - Crear proyectos (desde cero y con código)
 - Ejecutar
 - Depurar (breakpoint, watch, variables locales)
 - Generar Javadoc

Esquema

- IDE
- **Práctica 2**
 - Java
 - Enunciado
 - Tareas
 - Librerías externas
- Más cosas de Java

Práctica 2: Java

- Útil para esta práctica:
 - Constructores
 - Interfaces / Herencia
 - Control de acceso
 - ArrayList
 - HashMap
 - Enhanced for
 - Hilos
 - Excepciones
 - Entrada / salida

Constructores

- Métodos especiales que crean un objeto
 - Se invoca de manera automática
 - La JVM reserva memoria para ese objeto
 - Se devuelve una referencia a dicho objeto
 - Palabra reservada: *new*
- Inicializan las variables del objeto
- Puede haber más de un constructor
- Sin argumentos: constructor por defecto
 - Definido por defecto (salvo que se definan otros)
- Siempre hay que llamar a algún constructor

Interfaces y Herencia

- Interfaces

```
public interface Dibujo {  
  
    public void resize();  
  
}
```

- Imponen un protocolo de métodos a implementar

- Herencia

- Clases abstractas

- Tipo especial de herencia, donde se definen métodos (que pueden ser llamados) pero no se implementan

```
public abstract class Figura {  
  
    public abstract double calculaArea();  
  
    @Override  
    public String toString() {  
        return "Figura con área " + calculaArea();  
    }  
  
}
```

```
public class Circulo extends Figura {  
  
    private double radio;  
  
    public double calculaArea() {  
        return Math.PI * radio * radio;  
    }  
  
}
```

Interfaces vs Herencia

- No siempre es fácil elegir
- En la práctica:
 - Sólo se puede heredar de una clase
 - Se pueden implementar muchas interfaces

Control de acceso

- Ocultación de
 - Variables
 - Métodos
 - Constructores
- Todas variables public ==> mal implementado
(normalmente)

ArrayList

- Conjunto variable de cualquier tipo de objetos
- Similar a array, pero su capacidad aumenta o disminuye **dinámicamente**
- Desde 1.5: arrays tipados

```
protected ArrayList<Evento> eventos;
```

(en tiempo de compilación nos aseguramos el tipo del contenido)

HashMap

- Manera sencilla de tener una tabla (hash)
 - Función hash: hashCode
- Desde 1.5: tablas tipadas

```
private HashMap<String, Cliente> clientes;  
// ...  
clientes.put(cliente.getLogin(), cliente);
```

clientes ==

Login (String)	Cliente (Cliente)
"a"	Alguien{ Login="a"; nombre=Yo}
...	...

Enhanced for

- Tipos de iteración:
 - Con iterador (clase Iterator)
 - Sin iterador (usando una variable como índice)
 - Enhanced for

```
for (Articulo a : cliente.getHistorialFechas(desde, hasta)){  
    System.out.println(a.getNombre());  
}
```

Hilos

- El intérprete de Java hace un uso intensivo de hilos.
- Esto provoca situaciones raras:

```
Escribe su dirección:  
Escribe su teléfono:  
aaa  
aaaa
```

- Veremos más cosas en la P3 (GUI)

Excepciones

- Cómo (y cuándo) lanzarlas

```
public void cerrarExpediente() throws IllegalArgumentException {  
    if (!esPosibleCerrarExpediente()) {  
        throw new IllegalArgumentException("No es posible cerrar el expediente");  
    }  
}
```

- Throws: en la definición del método
- Throw: dentro del método

- Cómo (y cuándo) tratarlas

```
try {  
    cerrarExpediente();  
} catch (IllegalArgumentException e) {  
    // nada  
}
```

- Try-catch: se capturan las que se quieren, las demás se lanzan

Entrada / salida

- En esta práctica:

```
public static String leeLineaDeTeclado() {  
    BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));  
    String s = null;  
    try {  
        s = entrada.readLine();  
    } catch (Exception e) {  
    }  
    return s;  
}
```

Esquema

- IDE
- Práctica 2
 - Java
 - **Enunciado**
 - **Tareas**
 - Librerías externas
- Más cosas de Java

Práctica 2

- Implementar un *sistema de gestión de una tienda virtual*
 - Partir de la P1
 - Y de las interfaces/clases abstractas entregadas
- Comprobar mínima funcionalidad con mains de prueba
 - Aconsejable hacer (y entregar) pruebas extra

Práctica 2 - Artículos

- Libros
- Discos
- Películas
- ...

Práctica 2 - Cliente

- Registrarse
- Aumentar saldo
- Buscar artículos
- Gestionar carrito
- Comprar carrito
- Visualizar histórico
- ...

Práctica 2 - Administrador

- Mostrar inventario ordenado
- Buscar artículos
- Aumentar stock
- Mostrar clientes

Esquema

- IDE
- Práctica 2
 - Java
 - Enunciado
 - Tareas
 - **Librerías externas**
- Más cosas de Java

Librerías externas

- Conjunto de clases reutilizables (*jars*)
- En esta práctica: JDOM
 - Documentación y ejemplos en la web
- NetBeans:
 - propiedades del proyecto -> librerías -> añadir JAR

Librerías: pasos *generales* a seguir

1. Elegir librería a usar
2. Conseguir *jars*
 - Tener documentación a mano
3. Incluir en el proyecto
4. Identificar clases útiles de la librería
5. Probar ejemplos simples de uso
6. Integrar en la aplicación

Librería JDOM

- Manipular ficheros en formato XML

- <tienda nombre="Tienda virtual de prueba">

- <usuarios>

<usuario dni="0000001" nombre="Jose" apellidos="Lopez" saldo="300"/>

<usuario dni="0001002" nombre="Pedro" apellidos="Martinez" saldo="20.4"/>

</usuarios>

- <articulos>

<articulo id="02" tipo="Libro" titulo="El hobbit" autor="Tolkien, J.R.R" editorial="Minotauro" stock="6" coste="20.1"/>

<articulo id="01" tipo="Libro" titulo="La historia interminable" autor="Ende, Michael" editorial="Alfaguara" stock="1" coste="26.5"/>

<articulo id="06" tipo="Libro" titulo="El hereje" autor="Delibes, Miguel" editorial="Planeta DeAgostini" stock="4" coste="30"/>

<articulo id="03" tipo="Disco" titulo="Crush" interprete="Bon Jovi" agno="2000" stock="17" coste="10.25"/>

<articulo id="05" tipo="Disco" titulo="Fallen" interprete="Evanescence" agno="2003" stock="0" coste="9"/>

<articulo id="05" tipo="Disco" titulo="Sin noticias de Holanda" interprete="Melendi" agno="2003" stock="10" coste="6"/>

<articulo id="25" tipo="Disco" titulo="Que el cielo espere sentao" interprete="Melendi" agno="2005" stock="3" coste="6"/>

<articulo id="26" tipo="Libro" titulo="El gen egoista" autor

- <articulo id="10" tipo="Pelicula" titulo="Identidad" genero=

<actor>John Cusack</actor>

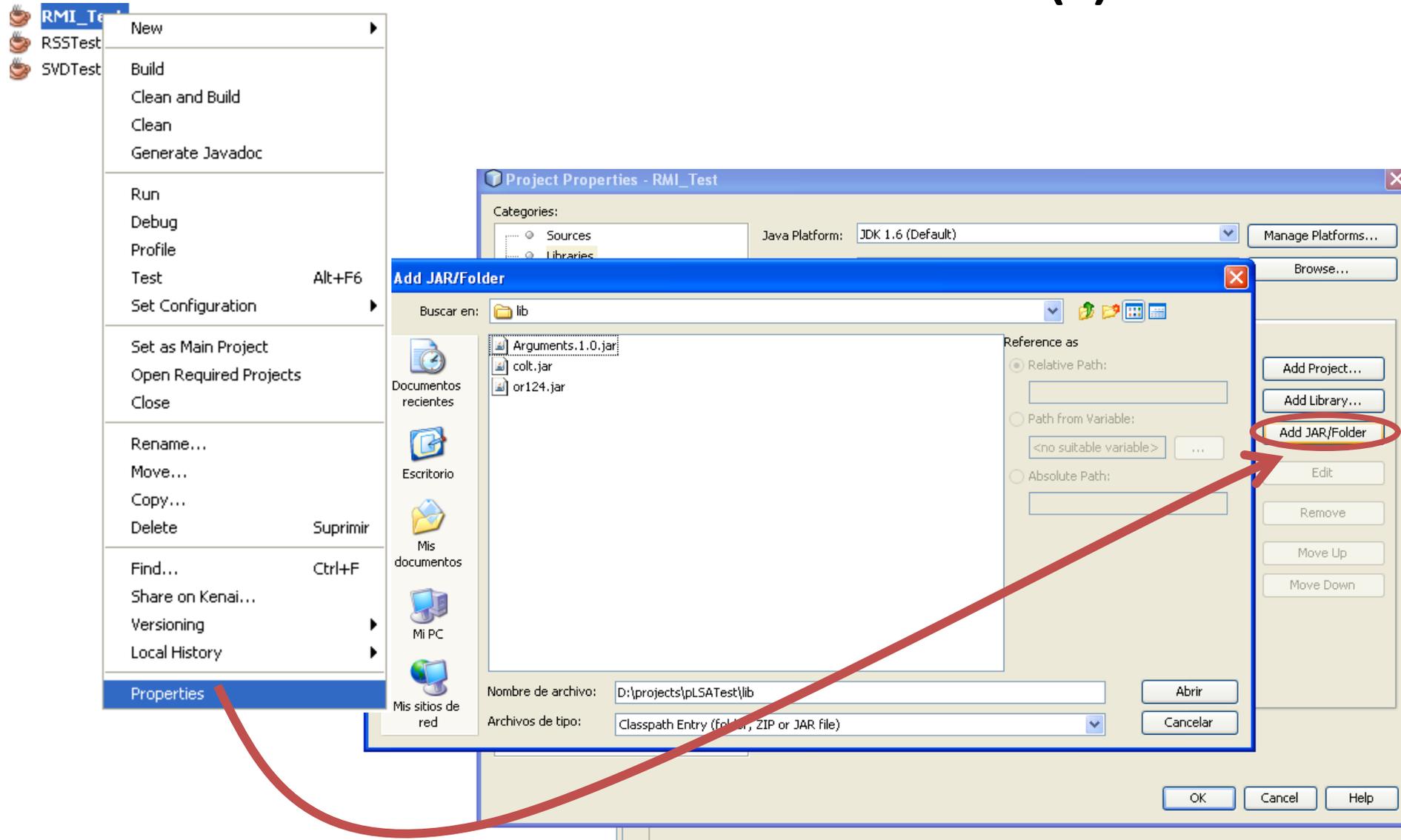
<actor>Ray Liotta</actor>

```
SAXBuilder parser = new SAXBuilder();
Document doc = parser.build("datos/tienda.xml");
Element datos;

System.out.println("Listado de usuarios:");
Iterator<Element> usuarios = doc.getDescendants(new ElementFilter("usuario"));
while (usuarios.hasNext()) {
    datos = usuarios.next();
    System.out.println("\t" + datos.getAttribute("dni").getValue() + ": " +
        datos.getAttribute("nombre").getValue() + " " +
        datos.getAttribute("apellidos").getValue());
}

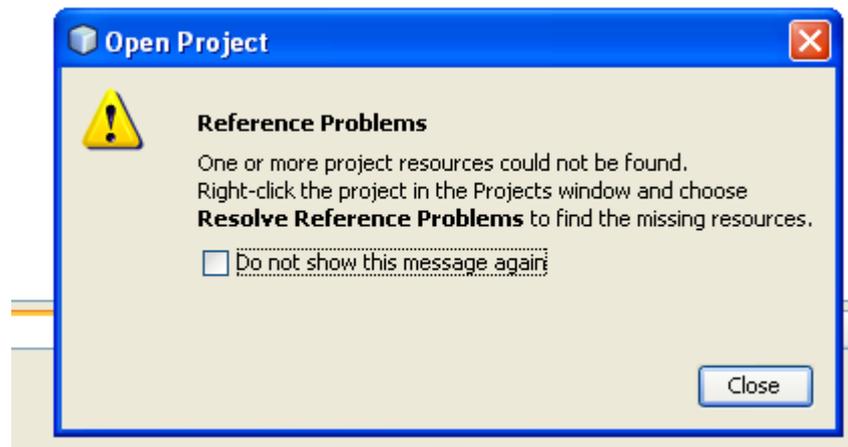
System.out.println("Listado de articulos:");
Iterator<Element> articulos = doc.getDescendants(new ElementFilter("articulo"));
while (articulos.hasNext()) {
    datos = articulos.next();
    System.out.println("\t" + datos.getAttribute("nombre").getValue() + " (" +
        datos.getAttribute("tipo").getValue() + ")");
}
```

Librerías + NetBeans (I)



Librerías + NetBeans (II)

- Si NetBeans no encuentra las librerías:



- Solución:
 - ¿están incluidas las librerías?
 - ¿la ruta es la correcta?

Esquema

- IDE
- Práctica 2
 - Java
 - Enunciado
 - Tareas
 - Librerías externas
- **Más cosas de Java**

Más cosas de Java

- Interfaces: Cloneable, Iterable, Comparable
- Ordenación, búsqueda, etc.: `java.util.Collections`
- Funciones matemáticas: `java.lang.Math`
- Entrada / Salida alternativa: `java.io.Console`
- Patrones de diseño:
 - Singleton
 - Factory
 - ...