

Índice

1. Introducción	1
1.1 ¿Qué vamos a hacer durante 5 prácticas?	1
1.2 ¿Qué vamos a utilizar?	1
1.3 ¿Con qué información contaremos para la realización de las prácticas?	. 1
2. ¿Qué vamos a ver en esta práctica 0?	2
3. Descarga de las herramientas	2
3.1 La plataforma J2SE y su documentación	2
3.2 Eclipse	2
4. Las aplicaciones básicas del J2SE SDK 1.4.2 de Sun	2
4.1 javac	3
4.2 java	3
5. El entorno de desarrollo Eclipse	4
5.1 ¿Qué es Eclipse?	4
5.2 Trabajando con Eclipse	4
5.2.1 Creación de un proyecto	5
5.2.2 Creando clases	5
5.2.3 Ejecutando el programa	6
5.2.4 Depuración de programas	7
5.2.5 Otras herramientas interesantes	8

1. Introducción

1.1 ¿Qué vamos a hacer durante 5 prácticas?

Las prácticas se realizarán utilizando el lenguaje Java.

Se van a plantear prácticas para trabajar los aspectos generales propios de un lenguaje orientado a objetos (clases, objetos, herencia, polimorfismo,...) y se revisarán los aspectos específicos que tiene este lenguaje (herramientas necesarias para compilar y ejecutar las aplicaciones, ejecución en una máquina virtual, interfaces, paquetes, soporte para la programación concurrente).

1.2 ¿ Qué vamos a utilizar?

Para la realización de las prácticas vamos a utilizar las siguientes herramientas:

- La plataforma Java 2 Standard Edition (J2SE) v.1.4.2 (J2SE 1.4.2)
- La documentación que acompaña a esta plataforma
- Un entorno de desarrollo: Eclipse 3.0

1.3 ¿Con qué información contaremos para la realización de las prácticas?

Para cada práctica se proporcionarán enunciados (similares a este) que serán colocados en la página web de la asignatura: http://informatica.uv.es/iiguia/LP/laboratorio.html.



Cuando sea necesario, además del enunciado, se colocarán programas de ejemplo que servirán como guía o esqueleto para realizar la práctica. Es muy conveniente leer el enunciado y echar un vistazo a los programas de ejemplo antes de la sesión. Los enunciados contendrán información adicional a la proporcionada en las clases de teoría por lo que es importante procesar la información que contienen y llevarlos a las sesiones de laboratorio.

2. ¿Qué vamos a ver en esta práctica 0?

Se pretende:

- Mostrar de dónde se pueden descargar las herramientas necesarias para las prácticas, herramientas enumeradas en el epígrafe 1.2
- Presentar las aplicaciones básicas de la plataforma J2SE: javac (compilación) y java (ejecución)
- Presentar el entorno de desarrollo Eclipse 3.0

3. Descarga de las herramientas

3.1 La plataforma J2SE y su documentación

Este software y su documentación se publican en el sitio java.sun. Concretamente, la versión 1.4.2 que vamos a utilizar se puede encontrar encontrarla en: http://java.sun.com/j2se/1.4.2/download.html

3.2 Eclipse

Podemos encontrar Eclipse en la siguiente dirección: http://www.eclipse.org/downloads/index.php

4. Las aplicaciones básicas del J2SE SDK 1.4.2 de Sun

La explicación en este apartado asume que el código fuente está en un fichero llamado Clase.java y que contiene diversas clases ClaseAux1, ClaseAux2 y Clase, siendo esta última la que contiene el método main (=punto de entrada de la ejecución de una aplicación Java en consola).

Las aplicaciones básicas que se explican a continuación se encuentran en el directorio DIR_JDK\bin, donde DIR_JDK representa el directorio donde está instalado el JDK 1.4.2.

Lenguajes de Programación



4.1 javac

n.

Programa que transforma el código fuente en bytecode.

El bytecode es el lenguaje que ejecuta la Máquina Virtual de Java (MVJ). El compilador de java lee un fichero fuente Java (fichero.java) y traduce el este código a bytecode escribiéndolo en un fichero (o ficheros) .class. Se generan tantos ficheros .class como clases hay en el fichero.

Como ejemplo de compilación:

javac Clase.java

esto genera (en nuestro ejemplo) tres ficheros:

Clase.class ClaseAux1.class ClaseAux2.class

4.2 java

Inicia una Máquina Virtual de Java.

La MVJ es un computador abstracto que ejecuta carga y ejecuta programas Java compilados. La MVJ es virtual ya que está implementada en software por encima de una plataforma hardware real y un sistema operativo. Por tanto, la MVJ debe estar implementada en una plataforma particular antes de que los programas se puedan ejecutar sobre esa plataforma.

La MVJ proporciona una capa de abstracción entre los programas Java compilados (el bytecode) y la plataforma sobre la que se ejecuta. Esto es importante para asegurar la portabilidad ya que los programas se ejecutan sobre la MVJ, independientemente del hardware sobre el que se está instalada la MVJ.

Para el ejemplo que nos ocupa, llamamos a la MVJ pasándole como argumento el nombre de la clase:

java Clase

Veamos qué secuencia de acciones desencadena la orden anterior:

1. Se inicia una MVJ

2. Se carga la representación binaria de la clase (el fichero .class con el bytecode). De este proceso se encarga la clase ClassLoader.

3. Enlace (*linking*), este paso se subdivide en 3:

- La <u>verificación</u> asegura que la representación binaria es correcta estructuralmente
- La <u>preparación</u> supone la creación de miembros estáticos y otras optimizaciones
- La <u>resolución</u> : La representación binaria contiene referencias a otras clases o interfaces, a sus miembros y métodos, y a los constructores. Antes de que una referencia simbólica pueda usarse debe someterse a resolución, donde se comprueba que la referencia es correcta y se reemplaza por una referencia directa.
- 4. Iniciación

5. Llamada al método Clase.main. El método main debe ser declarado como público y estático, debe devolver void y acepta como argumento un vector de cadenas (de *strings*). Todos los ficheros class a los que se hace referencia en nuestra clase deben ser accesibles por la MVJ (para ser cargados en memoria) cuando se ejecuta la clase. Si se



produce algún error durante alguno de esos pasos se lanza una excepción y el proceso se detiene.

Se verán más programas a lo largo de curso y se irán introduciendo según se necesiten.

5. El entorno de desarrollo Eclipse

5.1 ¿Qué es Eclipse?

Eclipse es una plataforma de desarrollo *open source* basada en Java. Es un desarrollo de IBM cuyo código fuente fue puesto a disposición de los usuarios. En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plug-in).

Hay plug-ins para el desarrollo de Java (JDT: Java Development Tools) así como para el desarrollo en C/C++, COBOL, etc. La versión instalada en el laboratorio incluye el plug-in JDT.

5.2 Trabajando con Eclipse

Al ejecutar Eclipse aparece una ventana como la mostrada en la figura_1

🚰 Resource - Eclipse Platform					
Eile Edit Navigate Search Project Run Windo	w Hel)			
] 🗈 • 🗟 🗎 💁 - 🖓 🛛 🍫 🗇 - 🗢					Resource
Navigator X					
· 후 후 🗉 🕼 수 수					
E Outline 🛛 🗖 🗖					
An outline is not available.					
	Tas	xe 52			
	0 Rems				
	•	Description	Resource	In Folder	Location
	6. 41	4			

figura 1 : el entorno eclipse

El desarrollo con Eclipse está organizado en *perspectivas* cada una de las cuales proporciona funcionalidades específicas para un tipo de tarea.

Lenguajes de Programación



Por ejemplo, la perspectiva *Java* combina un conjunto de vistas que permiten ver información útil cuando se está escribiendo código fuente, mientras que la perspectiva de *Depuración* contiene vistas que muestran información útil para la depuración de los programas Java.

La barra de herramientas *de perspectivas* muestra las perspectivas abiertas y permite, pulsando sobre ellas, cambiar de una a otra.

5.2.1 Creación de un proyecto

Eclipse permite organizar los ficheros en forma de proyecto. Para crear un proyecto Java se procede del siguiente modo:

- O bien seleccionar en el menú File >> New >> Project; o bien, en la perspectiva Resource, pulsar con el botón derecho del ratón sobre la vista Navigator y seleccionar New >> Project
- 2. Aparece una ventana en la que se puede seleccionar el tipo de proyecto. En este caso seleccionar Java Project; a continuación pulsar sobre el botón Next
- 3. A continuación se piden los datos sobre el proyecto (nombre y ruta donde se almacenarán los ficheros asociados al proyecto). Una vez introducidos, pulsar sobre el botón Finish http://java.sun.com/j2se/1.4.2/download.html
- 4. Eclipse abre automáticamente (o pide autorización para abrir) la perspectiva *Java* cuando se crea un proyecto Java. También, crea un directorio del proyecto en la ruta especificada en el punto anterior y en él crea dos ficheros .project y .classpath que contienen información sobre el proyecto.

La vista *Navigator* de la perspectiva *Resource* y la vista *Package Explorer* de la perspectiva *Java* contienen un directorio con el nombre del proyecto.

Como se comentó anteriormente se puede pasar de una perspectiva a otra pulsando sobre los botones de la barra horizontal superior.

- 5. Asociadas con Java, hay dos perspectivas más que serán de nuestro interés; se pueden abrir o bien con la opción de menú: Window >> Open perspective; o bien con el icono *Open a perspective* de la barra de herramientas *de perspectivas*
 - Perspectiva *Java Browsing*: en esta perspectiva aparecen vistas correspondientes al proyecto, a paquetes, a clases e interfaces y a los miembros de estas
 - Perspectiva *Debug*: como hemos comentado antes en esta guía, contiene vistas que muestran información útil para la depuración de los programas Java (puntos de ruptura, valor de las variables en cada punto de la ejecución, etc.).

5.2.2 Creando clases

Las clases se pueden crear de dos formas diferentes:

 Utilizando un asistente para crear el esqueleto de la clase Para ello, en la vista Package Explorer de la perspectiva Java, pulsar con el botón derecho sobre: <nombre del proyecto >> New >> Class Aparece el asistente. Al rellenar los campos y pulsar sobre el botón Finish se crea un fichero <nombre_de_la_clase>.java con el código que constituye el esqueleto de la clase.

El asistente simplemente ha creado un esqueleto; a continuación se debe teclear el resto del código

• Tecleando todo el código de la clase

Para ello, en la vista *Package Explorer* de la perspectiva *Java*, pulsar con el botón derecho sobre: <nombre_del_proyecto >> New >> File

El programa pide el nombre del fichero, al que llamaremos <nombre de la clase>.java para que lo reconozca como una clase. Al pulsar sobre el botón Finish se crea un fichero vacío <nombre de la clase>.java

Si se explora el contenido del directorio del proyecto se puede ver que, además del fichero <nombre_de_la_clase>.java hay otro fichero <nombre_de_la_clase>.class, este fichero es el que contiene el código compilado a partir de <nombre_de_la_clase>.java (Eclipse compila el proyecto cada vez que se guarda)

Ambas maneras de proceder son equivalentes e igualmente válidas para crear clases aunque, como recomendación, al principio es conveniente escribir todo el código de la clase para ser más consciente de lo que se está haciendo; además jen el examen no habrá ningún asistente!

5.2.3 Ejecutando el programa

Los programas se pueden ejecutar dentro de Eclipse: con la perspectiva *Java* abierta, seleccionar en el menú: Run >> Run...

En la ventana que se abre, pulsar dos veces sobre *Java Application*. En el panel Main rellenar la información tal y como aparece en la figura_2.

Prácticas de Lenguajes de Programación (Java) Ingeniería Informática Curso 2004-2005 . Práctica 0 Universitat de València

C Run		
Create, manage, and run con Create a configuration that will	nfigurations Iaunch a Java virtual machine.	
Configurations:	Main Main Arguments IRE Classpath Source Image: Elements Project: prueba Image: Elements	nviroi
Ne <u>w</u> Dele <u>t</u> e	Apply	Revert
	Run	Close

figura 2 : ejecución de una aplicación Java de consola

Puesto que en el ejemplo propuesto se pasa un argumento a la aplicación, pulsar sobre el panel *Arguments* y escribir el argumento que vamos a pasar al programa: en este caso el literal "Hello world !".

Una vez proporcionada la información necesaria (hay más paneles pero por ahora no es necesario conocerlos), pulsar sobre el botón Run.

Esta aplicación simplemente muestra un mensaje por consola. La consola está integrada dentro de Eclipse (es una vista más).

5.2.4 Depuración de programas

16

Dentro del entorno de desarrollo de Eclipse se puede depurar programas desarrollados en Java: Window >> Open perspective >> Debug.

Aparecen una serie de vistas similares a las de la figura_3.





figura 3 : perspectiva de depuración

Desde la perspectiva *Debug* podemos pedir parar la ejecución del programa en una determinada línea (poniendo un punto de ruptura) e inspeccionar las variables locales. Para poner un punto de ruptura, en la vista donde se encuentra el código fuente, marcar la línea donde queremos que se detenga la ejecución y seleccionar en el menú Run >> Add/remove Breakpoint. El punto azul en la parte izquierda de la línea indica la existencia de un punto de ruptura.

A continuación, lanzar el depurador: seleccionar en el menú Run >> Debug . La ejecución del programa se detiene en el primer punto de ruptura. Una vez el programa está detenido se puede ver el valor de las variables y los puntos de ruptura definidos.

Una vez inspeccionado el código donde está el problema se puede optar por ejecutar el programa hasta que termine (Run >> Resume) o terminar el programa inmediatamente (Run >> Terminate).

5.2.5 Otras herramientas interesantes

El editor de Java ofrece correcciones a problemas encontrados mientras se escribe el código y tras compilar. El editor muestra que existen propuestas para la corrección de un problema o aviso mediante una bombilla visible en la parte izquierda del editor. Si se pulsa con el botón izquierdo sobre esta bombilla (o también mediante Edit >> Quick Fix) se muestran las propuestas para el problema en la posición del cursor.



Si se desea importar recursos a un proyecto se puede realizar del siguiente modo: en la vista *Navigator* pulsar sobre el botón derecho y aparece un menú, seleccionar Import y aparece una ventana desde la que se puede seleccionar el directorio donde están los recursos y cuales se desea incorporar al proyecto.

Desde un proyecto se pueden exportar todos o algunos de los ficheros que lo conforman. Para ello, en la vista *Navigator*, pulsar sobre el botón derecho; aparece un menú. Seleccionar Export; aparece una ventana en la que se puede indicar cómo se va a exportar (un fichero zip, tal cual aparecen en el proyecto, etc.). La siguiente ventana sirve para seleccionar los ficheros que se desea exportar y dónde.

Si se coloca el ratón sobre un método (sin pulsar) se muestra la declaración del método (qué devuelve y qué parámetros acepta). Si se coloca el ratón (sin pulsar) sobre una variable aparece información sobre el tipo de la variable. Al escribir código podemos pulsar Ctrl + espacio y nos aparece un menú con posibles formas de finalizar la sentencia que se está escribiendo.