



PRACTICA 12: Entorno de Desarrollo: Eclipse.

1. OBJETIVO

El objetivo de esta práctica es mostrar un entorno de desarrollo para la programación de aplicaciones en el lenguaje de programación Java. Para ello, se presenta el entorno de desarrollo Eclipse.

2. PLATAFORMA DE DESARROLLO ECLIPSE

Eclipse es una comunidad de código abierto orientada a construir una plataforma de desarrollo abierta, compuesta de frameworks extensibles, herramientas, etc., para construir, desarrollar y gestionar software a lo largo de todo su ciclo de vida. Eclipse puede significar cosas distintas para usuarios distintos.

Eclipse es abierto porque su diseño permite una sencilla extensión de sus funcionalidades por terceros. Es un IDE (Entorno de Desarrollo Integrado o *Integrated Development Enviroment*) porque proporciona herramientas para manejar los espacios de trabajo, construir, lanzar y depurar aplicaciones. También permite una fácil integración para desarrollo en equipos con control de versiones, etc. Se dice también que es una **plataforma** porque está preparada para ser extendida con nuevas herramientas.

2.1 EJECUCIÓN DE ECLIPSE

Eclipse tiene varias versiones, cada una con un nombre en clave: Europa (versión 3.3), Ganymede (v3.4), Galileo (v3.5), Helios (v3.6), etc. Un requisito indispensable para la ejecución de Eclipse es tener instalada la máquina virtual de Java. Además, para la versión que utilizaremos de Eclipse es necesario disponer, al menos de la versión 6 de la JDK de Java. Para comprobarlo, podemos abrir un terminal y ejecutar lo siguiente:

```
java -version
```

La salida del comando anterior será algo similar a:

```
java version "1.6.0_26"  
Java(TM) SE Runtime Environment (build 1.6.0_26-b03)  
Java HotSpot(TM) Client VM (build 20.1-b02, mixed mode, sharing)
```

La instalación de Eclipse consiste solamente en la descompresión del paquete en una carpeta determinada, y para ser ejecutado sólo deberá invocar el ejecutable de nombre eclipse:

```
eclipse &
```

Aparecerá en la pantalla la ventana de inicio de Eclipse, mostrando la versión concreta de la plataforma.

La versión clásica de Eclipse incluye los plugins para desarrollo Java JDT (Java Developmet Tool) y PDE (Plug-in Development Environment). A partir de estos podrá instalar todos aquellos que vaya necesitando mediante la opción de menú "**Help > Install New Software**". La primera pantalla que aparece tras la ejecución de Eclipse es la correspondiente a la selección del espacio de trabajo (workspace).

Para el uso del espacio de trabajo, realice lo siguiente:

1º) Cree el directorio ***./eclipse/P01***

2º) Pulse en "**Browse...**" para configurar dicha carpeta como su espacio de trabajo durante esta práctica.

El workspace es la localización física donde se trabaja. Se puede elegir el workspace durante el inicio de Eclipse o a través del menú **File > Switch Workspace > Others**. Todos los proyectos, ficheros fuentes, imágenes y otros artefactos serán almacenados en el workspace. Se puede predefinir el workspace a través del parámetro ***-data path_to_workspace***, por ejemplo:

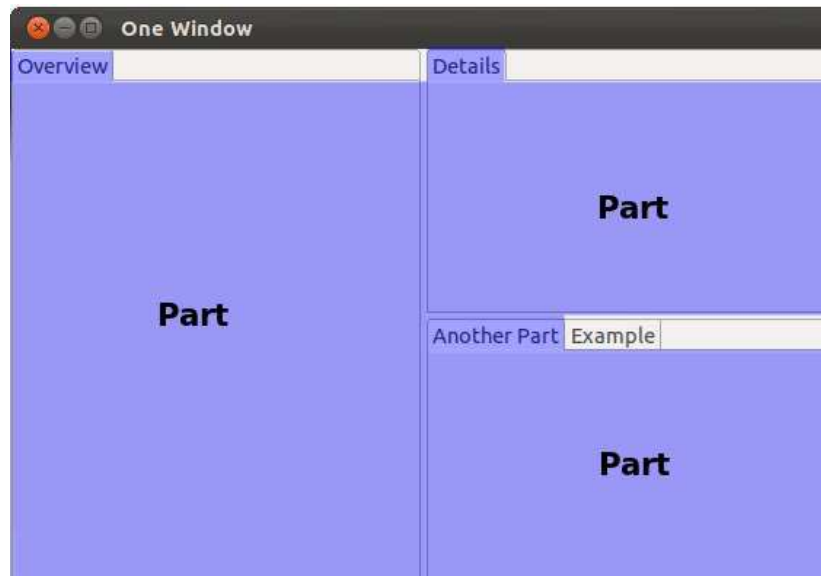
```
:~$ eclipse -data "./workspace/P01" &
```

2.2 WORKBENCH, PERSPECTIVAS Y VISTAS

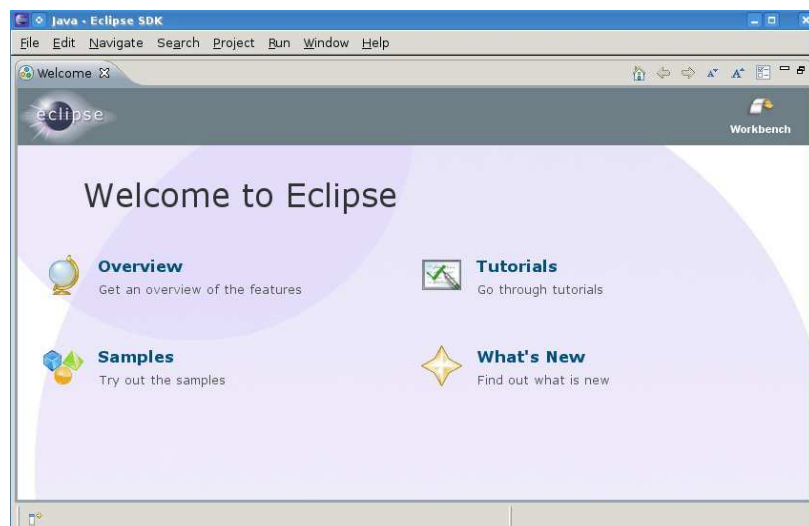
El término **Workbench** se refiere al entorno de desarrollo. Cada workbench contiene una o más perspectivas. Cada perspectiva contiene vistas, editores y controles que aparecen en determinados menús y barras de herramientas. Estas perspectivas proporcionan las herramientas de desarrollo habituales para desarrollar una cierta actividad, aunque se puede modificar su apariencia, etc.

Cuando se inicia el workbench, lo primero que se observa es un diálogo que permite seleccionar el directorio donde se almacenará nuestro trabajo, el workspace o zona de trabajo. Esto es lo que hemos hecho en el punto anterior.

Una perspectiva está dividida en partes: editores y vistas (como el "Project Explorer" o explorador de proyectos).



Inicialmente suele mostrarse la perspectiva Java, y en la primera ejecución sólo está visible la ventana de Bienvenida:



Haga clic sobre el icono etiquetado como **WorkBench** para poder acceder a la perspectiva Java y al resto de vistas. Podemos regresar a la vista de Bienvenida seleccionando en el menú principal **Help > Welcome**.

En la esquina superior derecha aparece una barra de atajos para las perspectivas, que permite abrir nuevas o conmutar entre las ya abiertas. La perspectiva activa se muestra en el título de la ventana. Si por ejemplo cambiamos a la perspectiva Resource (**Window > Open Perspective > Other... > Resource**) se nos mostrará el explorador de proyectos y las vistas **Outline** y **Tasks**. Dentro de una ventana, cada perspectiva puede tener un conjunto de vistas diferentes, pero todas las vistas comparten el mismo conjunto de editores.

Durante el uso de Eclipse, abrirá, moverá, cerrará o cambiará de tamaño vistas. Si desea restaurar una perspectiva a su estado original puede seleccionar el menú **Window > Reset Perspective**.

Pruebe a cambiar el tamaño de alguna vista, y después devuelva la perspectiva al estado original. Otro de los componentes visuales de nuestro entorno son los editores, usados para editar o mostrar un recurso. La presentación del recurso puede ser en modo texto o diagrama. Los editores se pueden abrir haciendo clic sobre un recurso en alguna vista. ***Las modificaciones realizadas en los recursos utilizando los editores necesitan ser guardadas explícitamente.*** Dependiendo del tipo de fichero, se mostrará el editor adecuado. Si, por ejemplo, estamos editando un fichero de texto, se abrirá un editor de texto en el área de edición, en la zona central. El nombre del fichero aparece en la pestaña del editor, y un asterisco en la pestaña indica que los cambios no se han guardado en fichero. Por supuesto, si se intenta cerrar el editor en esa situación, se mostrará un mensaje indicando si desea guardar los cambios no reflejados.

Cuando un editor está activo, la barra de menú y la barra de herramientas reflejan las operaciones posibles con ese editor. Cuando otra vista está activa, esas operaciones se deshabilitan. Las operaciones comunes a un editor y a una determinada vista permanecen habilitadas.

Los editores pueden ser apilados en el área de edición, y cada uno de ellos se puede activar haciendo clic sobre el editor o la pestaña con su nombre. Los editores se pueden organizar para poder ver el contenido de dos ficheros en columnas, o uno debajo de otro, etc.

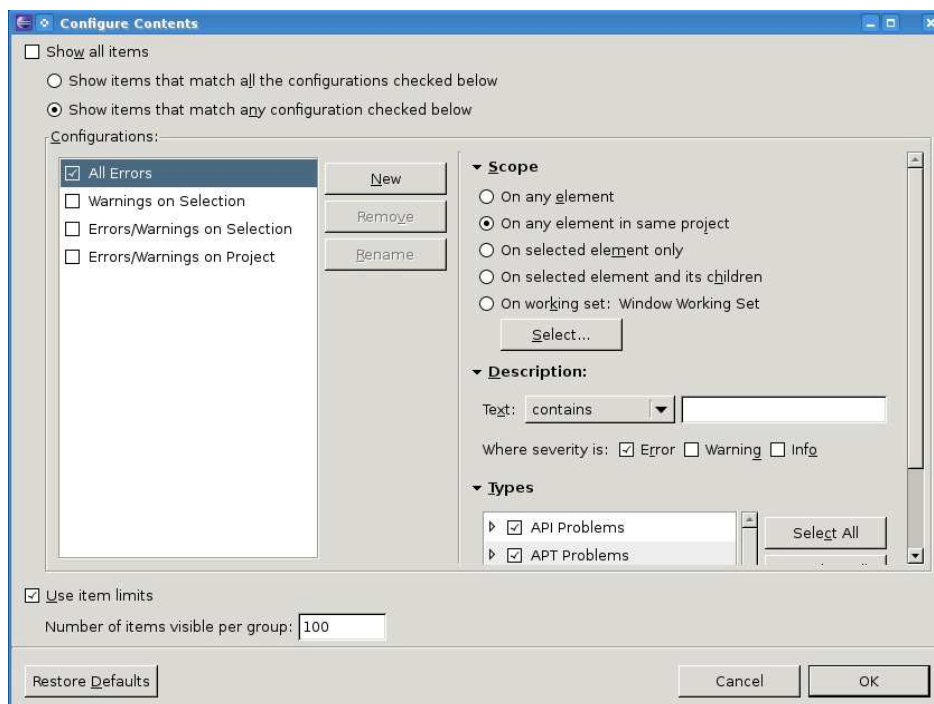
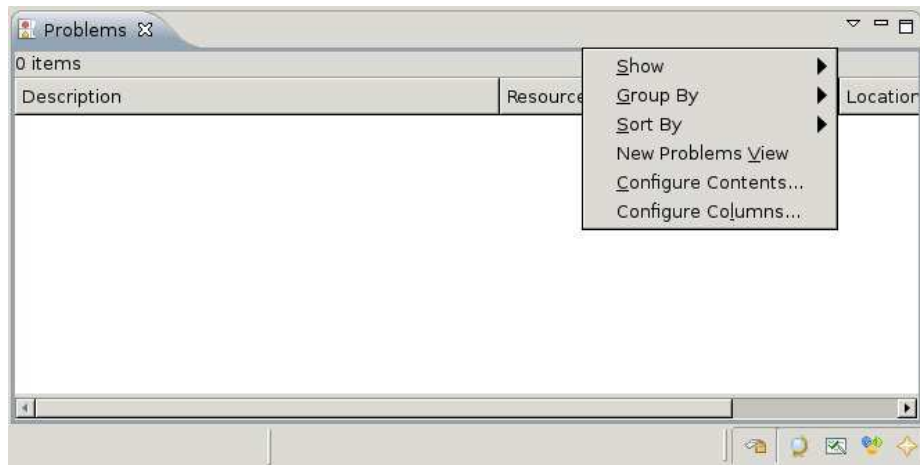
Resumiendo, mientras que los editores normalmente permiten realizar una tarea completa, las vistas proporcionan funciones de apoyo. Una perspectiva de Eclipse es una agrupación de vistas y editores de manera que den apoyo a una actividad completa del proceso de desarrollo software. Sin embargo, es posible crear perspectivas propias añadiendo nuevas vistas y cambiando su distribución en la pantalla. Las perspectivas pueden seleccionarse haciendo clic en los iconos de perspectiva del lateral izquierdo o eligiendo "**Window > Open Perspective**" del menú.

Por ejemplo, la perspectiva **Resource** está estrechamente relacionada con el sistema de archivos, y representa la localización física de los recursos almacenados dentro de los proyectos. La perspectiva Java se centra en tareas de programación, mostrando paquetes, clases, métodos y atributos en sus vistas asociadas. De las vistas destacaremos la vista del explorador que muestra la estructura lógica de paquetes y clases Java almacenados en los distintos proyectos. Las carpetas fuente (que deben almacenar los archivos fuente ".java") se muestran aquí decoradas con el icono de un paquete contenido. Los archivos Java también pueden ser expandidos de modo que muestren sus métodos y atributos internos al pulsar el botón "<". La vista del navegador de recursos ("**Navigator**") permite echar un vistazo a la estructura de archivos de los proyectos definidos. Nótese que esta vista es la única que muestra la carpeta de salida ("bin") así como los archivos Java compilados (".class").

La vista "**Outline View**" permite de forma visual ver qué métodos y atributos se encuentran definidos dentro de una clase de Java. Los iconos asociados proporcionan información adicional de acuerdo con la visibilidad del atributo o método en cuestión. Y sólo con hacer clic en cualquiera de estos iconos conducirá a la línea de código exacta en que dicho atributo o método está definido. La vista de resumen es una herramienta esencial para entender y navegar archivos voluminosos. También puede reflejar la estructura de un fichero html o xml.

Antes o después, nos encontraremos con problemas en la creación del código o proyecto. Para ver los problemas en el proyecto, se puede utilizar la vista “**Problems**”, que es parte de la perspectiva Java estándar.

Se puede configurar el contenido de la vista “Problems”. Se puede seleccionar “**Configure Contents**” y ajustar el Scope a “*On any element in the same project*”, para ver los problemas en el proyecto seleccionado actualmente.



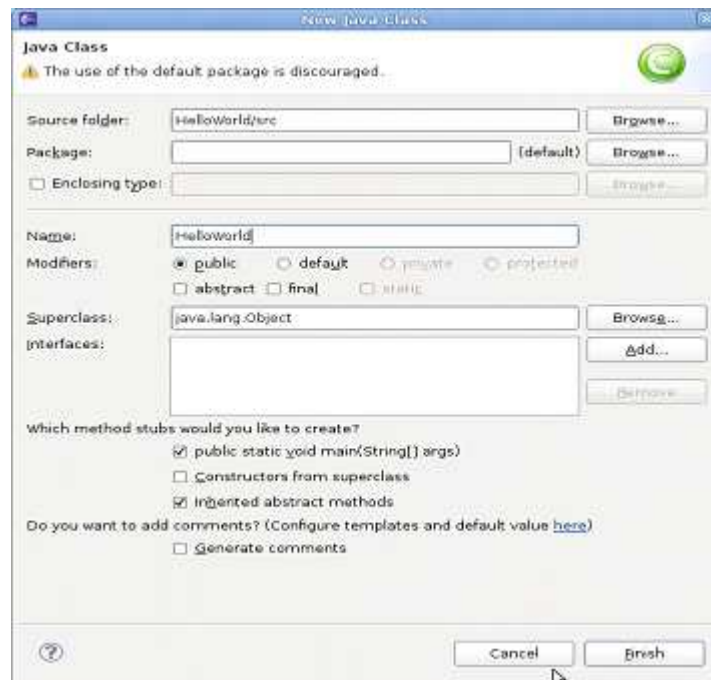
2.3 USO BÁSICO DE ECLIPSE

A continuación, describiremos cómo crear un programa Java mínimo utilizando Eclipse. Como ya es tradición en el mundo de la programación, crearemos un pequeño programa que escriba “*Hello word!*” en la consola. Este ejemplo servirá para familiarizarnos con la forma de trabajar con Eclipse.

Una vez en el Workbench, lo primero que haremos, si no está seleccionada ya (icono en la parte superior derecha en la pantalla de Bienvenida), es utilizar la perspectiva Java. Para ello seleccionamos en el menú principal **Window > Open Perspective > Java**. Antes de crear una clase Java necesitaremos un proyecto donde incluirla. En la barra de herramientas principal, haga click sobre el botón *New Java Project*, proporcione el nombre *HelloWorld* al proyecto y pulse Finish. Puede acceder también a través del menú principal, **File > New > Java Project**.

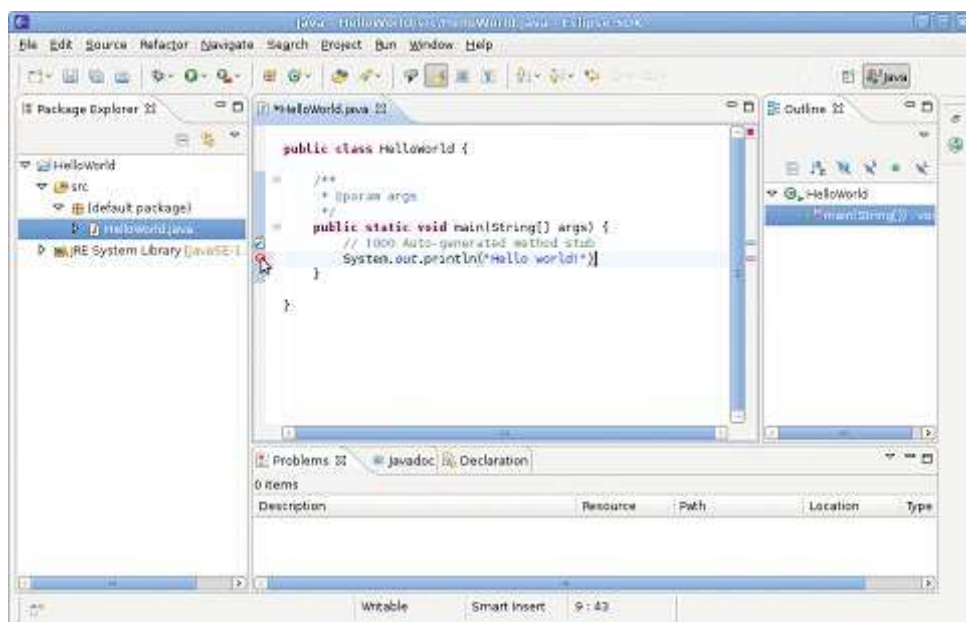


Para crear una clase, pulse el botón *New Java Class* de la barra de herramientas principal (o acceda mediante el menú principal **File > New > Class**). Seleccione *HelloWorld/src* como directorio fuente. Introduzca *HelloWorld* como nombre de la clase, y no olvide seleccionar la opción para crear el método **main**. Finalmente, pulse *Finish*.



El editor Java se abrirá de forma automática y nos mostrará la clase creada. Añada en el método main la siguiente sentencia:

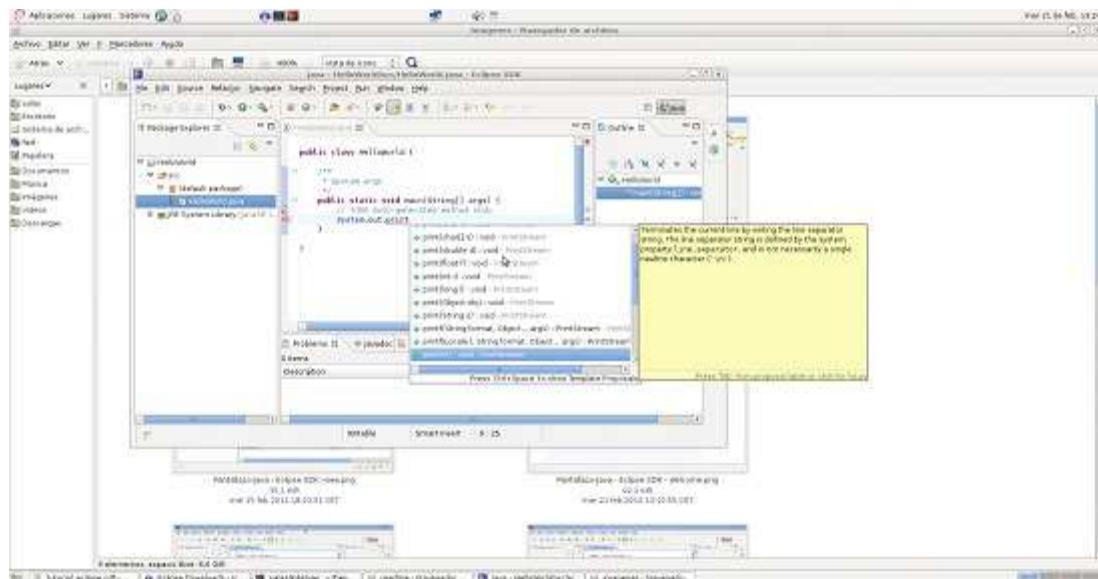
`System.out.println("Hello world!");`



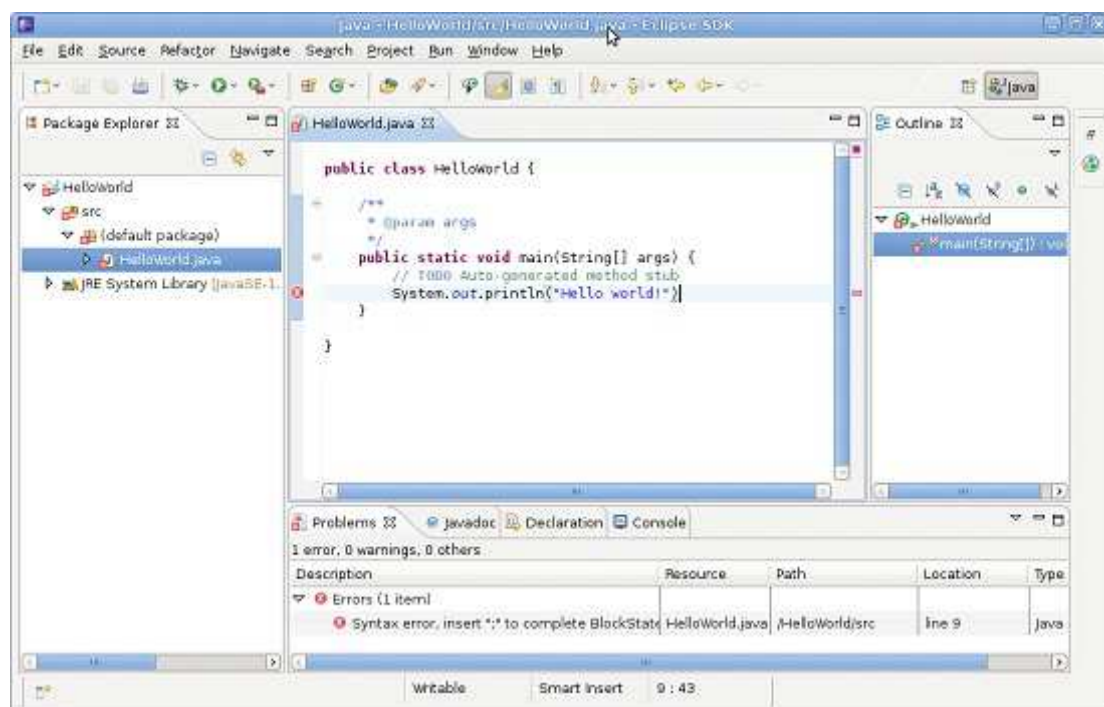
Puede recibir ayuda de autocompletado. Observe por ejemplo que si escribe **Sys** y pulsa la combinación **<Ctrl> + espacio** el editor le sugiere nombres de clases (en verde) o de interfaces (en morado). Haciendo clic se completa, y se añaden las sentencias de importación si son necesarias. Algo similar ocurre con variables locales, atributos, métodos, etc.

Incluso podrá autocompletar estructuras del lenguaje. Por ejemplo, si escribe **do** o **for** e intenta autocompletar, además de otras opciones le propone rellenar una estructura **do while** o **for** genéricas.

A la hora de ayudarlo con los posibles errores, por ejemplo, si olvida el punto y coma final aparece una advertencia en esa línea, y posicionado el ratón sobre la misma nos indica el posible error.



Observe que si elimina el punto y coma final de la sentencia, y guarda el fichero, en la parte inferior aparecen los errores encontrados: *al guardar los cambios la clase se compila de forma automática.*



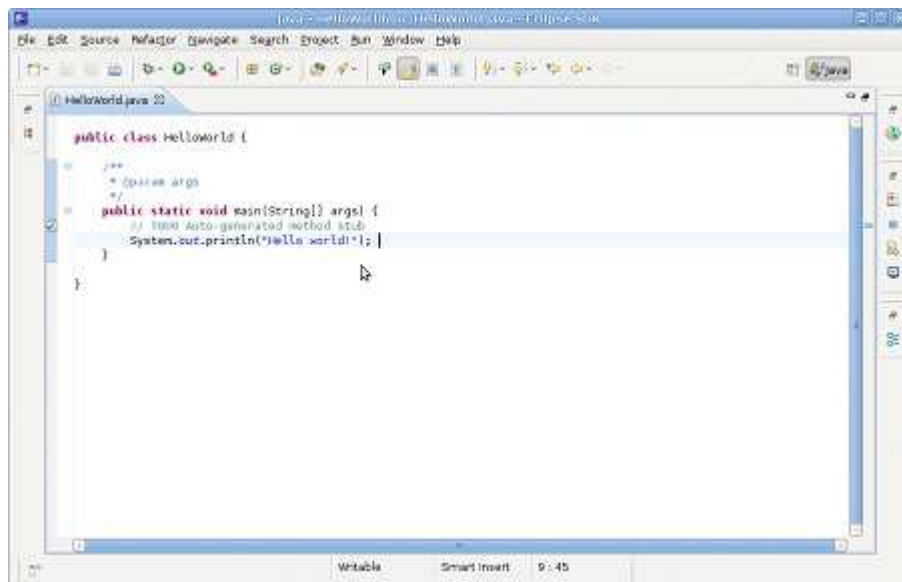
Es decir, que los errores de compilación se muestran en tiempo real subrayando el fragmento de código adecuado con una línea. También los errores y advertencias presentes en archivos ya guardados se muestran en la vista **Problems**, justo debajo de la ventana de edición. Haciendo clic en cualquiera de los dos tipos de marcadores de error le llevará automáticamente hasta la línea en la que el error está presente. Las advertencias (*warnings*) se muestran de la misma manera, pero con marcas amarillas.

Incluso es posible que nos intente ayudar en el posible error posicionándonos sobre la bombilla que aparece junto al mensaje de error.

Intente observar esta ayuda, por ejemplo, olvidando poner la letra p en la línea:

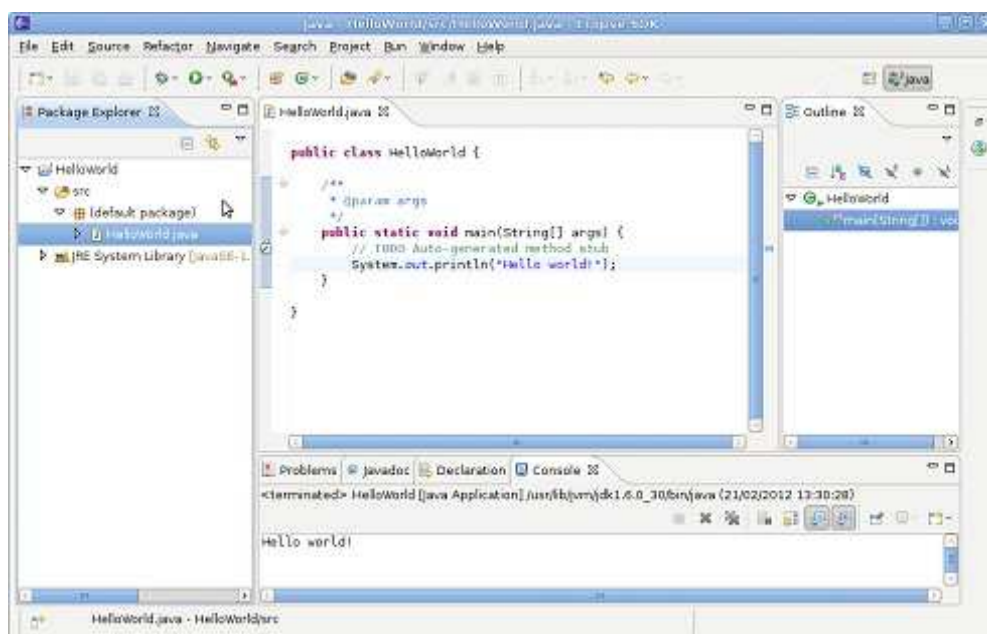
```
System.out.rintln("Hello World");
```

Si queremos maximizar el editor (o cualquier otra ventana) basta hacer doble clic sobre la pestaña del título de esa ventana.

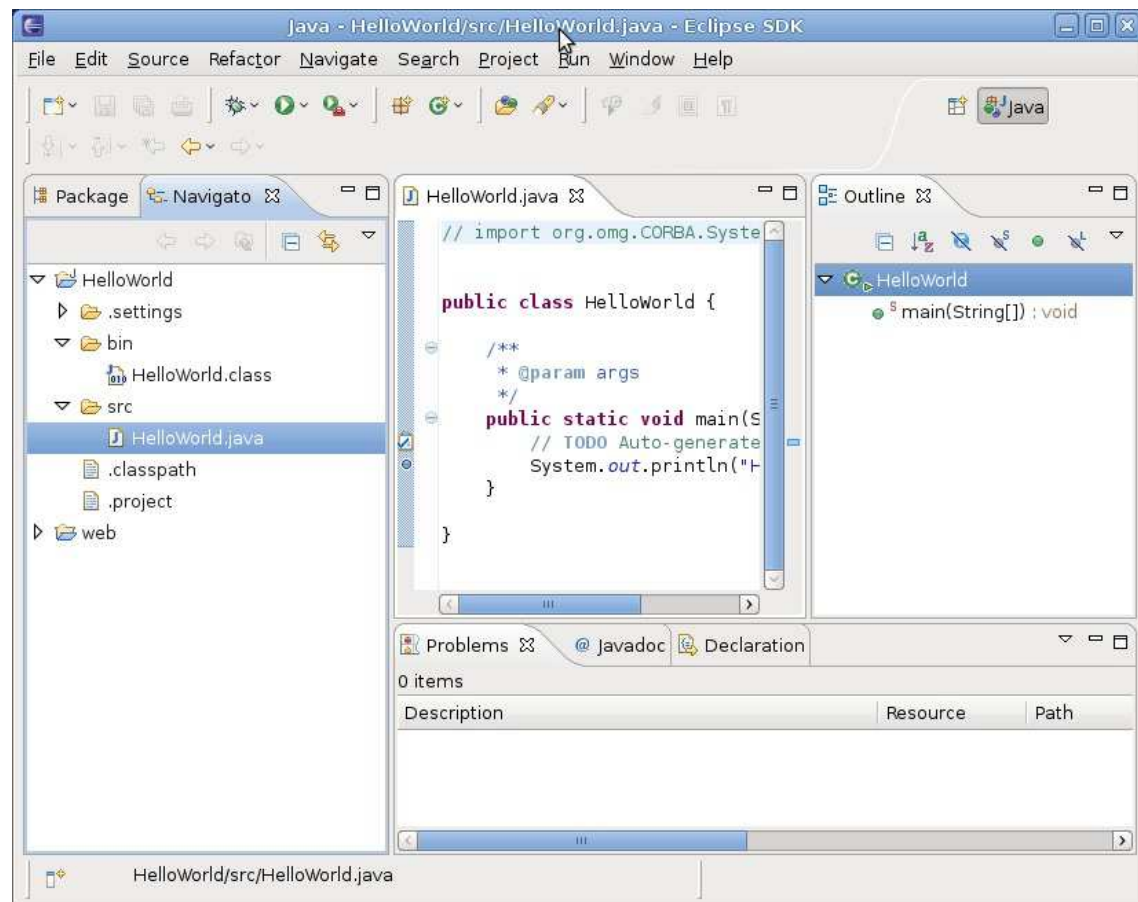


Para restaurar la vista previa basta hacer doble clic sobre el título de la ventana. Puede usar la combinación **<Ctrl>+M**.

Para ejecutar nuestra aplicación, haga clic con el botón derecho sobre el nombre de la clase en el package Explorer, y seleccione **Run As > Java Application**. La vista de consola en la parte inferior mostrará el mensaje de salida.

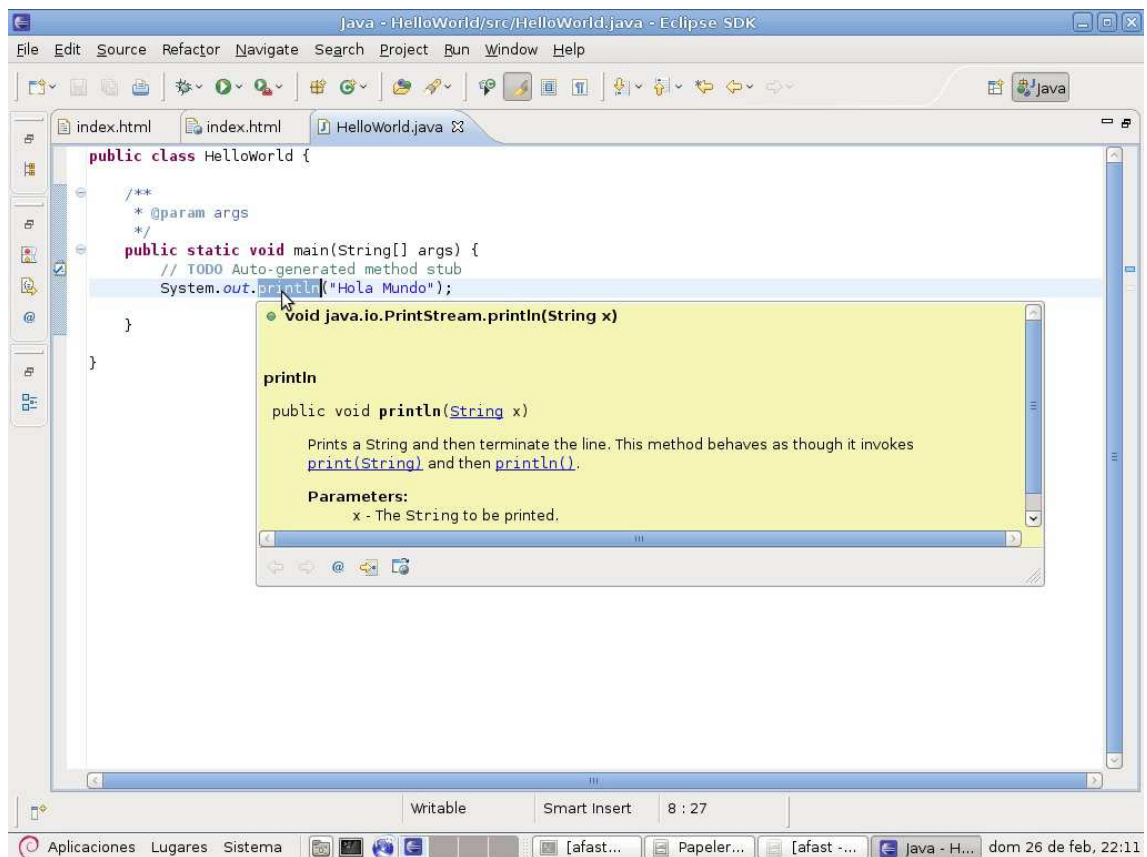


Observe que en la vista de proyecto no puede ver el fichero *HelloWorld.class* generado. Para ello, deberá abrir la vista “Navigator”, accediendo al menú **Windows > Show View > Navigator**. Esta vista se apilará junto a la de proyectos, y en ella podrá observar la estructura de ficheros del proyecto.



Observe a la derecha la vista resumen o “**Outline view**”, en la que podemos ver los métodos o atributos definidos dentro de una clase Java. Los iconos asociados proporcionan información adicional de acuerdo con la visibilidad del atributo o método en cuestión. Y sólo con hacer clic en cualquiera de estos iconos conducirá a la línea de código exacta en que dicho atributo o método está definido. La vista de resumen es una herramienta esencial para entender y navegar en archivos Java voluminosos.

Por último, la documentación Javadoc del código que se esté actualmente programando puede ser consultada en tiempo real simplemente colocando el cursor o el puntero del ratón sobre el elemento elegido. Para expandir la ventana con esta documentación basta con pulsar la tecla de función F2. La documentación Javadoc externa como, por ejemplo, la documentación oficial de las clases de Java, puede consultarse dejando el cursor sobre el nombre de la clase y pulsando “Mayúsculas + F2”: se abrirá la documentación por el punto adecuado.



2.4 TAREA A REALIZAR

Ejercicio 1. Cree una clase genérica denominada “**ExpendedoraGenerica**” tal y como lo ha desarrollado en el Ejercicio 1 de la práctica 9, en el entorno de desarrollo de Eclipse.