Eclipse

Guillermo Román

F. Informática UPM

9 de septiembre de 2009

Índice

Índice

Requisitos e Instalación

Requisitos

- SunJDK 6.0 (`http://java.sun.com`)
- No es independiente del SSOO
- Comprobar la máquina virtual `java -version`

Requisitos e Instalación

Requisitos

- SunJDK 6.0 (`http://java.sun.com`)
- No es independiente del SSOO
- Comprobar la máquina virtual `java -version`

Instalación

- Eclipse Java Developers de `http://www.eclipse.org`
- Descomprimir el fichero descargado en una carpeta
- Ejecutar el fichero eclipse

Requisitos e Instalación

Requisitos

- SunJDK 6.0 (`http://java.sun.com`)
- No es independiente del SSOO
- Comprobar la máquina virtual `java -version`

Instalación

- Eclipse Java Developers de `http://www.eclipse.org`
- Descomprimir el fichero descargado en una carpeta
- Ejecutar el fichero eclipse

Workspace

- Es el directorio de trabajo de Eclipse
- Se define al arrancar
- Puede tener más de un workspace, pero solo carga uno
- En el workspace se crean todas las carpetas de los proyectos

Índice

Perspectivas y Vistas

Vistas

- Una Vista es un panel del entorno de desarrollo
- Puede ser de muchos tipos (Console, Outline, ...)
- Se pueden mover y cambiar de sitio
- Se pueden incluir nuevas vistas (Window ⇒Open View)

Perspectivas y Vistas

Vistas

- Una Vista es un panel del entorno de desarrollo
- Puede ser de muchos tipos (Console, Outline, ...)
- Se pueden mover y cambiar de sitio
- Se pueden incluir nuevas vistas (Window $\Rightarrow$Open View)

Perspectivas

- Una perspectiva es un conjunto de Vistas que Eclipse agrupa
- Depende de lo que se está haciendo se usa una u otra
- Se pueden colocar las vistas
- Se puede cambiar de perspectiva (Window $\Rightarrow$Show Perspective)
- Normalmente Eclipse sugiere automáticamente la perspectiva más adecuada

Índice

Desarrollo usando Eclipse

Creación de un Proyecto

- File ⇒New ⇒Project ⇒Java Project
- La carpeta del Proyecto se genera en el Path del Workspace

Desarrollo usando Eclipse

Creación de un Proyecto

- File ⇒New ⇒Project ⇒Java Project
- La carpeta del Proyecto se genera en el Path del Workspace

Desarrollo usando Eclipse

Creación de un Proyecto

- File ⇒New ⇒Project ⇒Java Project
- La carpeta del Proyecto se genera en el Path del Workspace

Configuración del Proyecto

- Ficheros Fuente Project ⇒Properties ⇒Java Build Path ⇒Source
- Binarios Project ⇒Properties ⇒Java Build Path ⇒Default Output Folder
- Classpath Project ⇒Properties ⇒Java Build Path ⇒Libraries

Desarrollo usando Eclipse

Creación de un Proyecto

- File ⇒New ⇒Project ⇒Java Project
- La carpeta del Proyecto se genera en el Path del Workspace

Configuración del Proyecto

- Ficheros Fuente Project ⇒Properties ⇒Java Build Path ⇒Source
- Binarios Project ⇒Properties ⇒Java Build Path ⇒Default Output Folder
- Classpath Project ⇒Properties ⇒Java Build Path ⇒Libraries

Crear Clases y Paquetes

- Crear Clase File ⇒New ⇒Class
- Crear paquete File ⇒New ⇒Package

Estructura de la Carpeta de Proyecto

Estructura

- src contiene todos los ficheros fuentes
- classes o bin contiene los ficheros compilados (`.class`)
- doc contiene la documentación `javadoc` del proyecto
- lib contiene las liberías del proyecto
- config Contendrá los ficheros de configuración de la aplicación
- No debe haber ningún fichero en la raíz del proyecto

Estructura de la Carpeta de Proyecto

Estructura

- src contiene todos los ficheros fuentes
- classes o bin contiene los ficheros compilados (`.class`)
- doc contiene la documentación `javadoc` del proyecto
- lib contiene las liberías del proyecto
- config Contendrá los ficheros de configuración de la aplicación
- No debe haber ningún fichero en la raíz del proyecto

Estructura de paquetes

- Los ficheros fuente se estructuran en paquetes (carpetas)
- Las carpetas de los paquetes surgen a partir de src
- Permiten estructurar el código de forma lógica

Desarrollo usando Eclipse

Compilación

Build Proyecto

- Eclipse por defecto tiene activo el Build Automatically
- Se compilan automáticamente indicando los errores en el fichero fuente (subrayando el error o mostrando un icono)
- Si se pasa el ratón sobre el error muestra el motivo
- Si se pincha sobre el error. Quickfix indica sugerencias para solucionar el problema

Desarrollo usando Eclipse

Compilación

Build Proyecto

- Eclipse por defecto tiene activo el Build Automatically
- Se compilan automáticamente indicando los errores en el fichero fuente (subrayando el error o mostrando un icono)
- Si se pasa el ratón sobre el error muestra el motivo
- Si se pincha sobre el error. Quickfix indica sugerencias para solucionar el problema

Warnings

- Los Warnings son avisos del compilador que entiende que pueden producir problemas
- Con Warnings se puede ejecutar el programa
- Los Problems y los Warnings de compilación salen en la misma vista Problems

Desarrollo usando Eclipse

Ejecución Programa

Ejecución de un Programa

- El método de inicio de ejecución tiene la cabecera `public static void main(String[] args)`
- En un proyecto Java puede haber más de un `main`
- Para ejecutar lo más cómodo
  - Botón derecho ⇒Run As ⇒Java Application
  - Una vez ejecutado basta con el botón "Run" de la barra de herramientas
- La salida de los `System.out.println` sale por la Vista "Consola"

Índice

Debug de Programas

Ejecución en modo Debug

Ejecutar el programa en modo Debug

- Para hacer Debug se hace de forma similar a la forma de ejecutar
  - Botón derecho ⇒Debug As ⇒Java Application
  - Una vez ejecutado basta con el botón "Debug" de la barra de herramientas
- Antes de ejecutar hay que poner Breakpoints (puntos de ruptura)

Debug de Programas

Ejecución en modo Debug

Ejecutar el programa en modo Debug

- Para hacer Debug se hace de forma similar a la forma de ejecutar
  - Botón derecho ⇒Debug As ⇒Java Application
  - Una vez ejecutado basta con el botón "Debug" de la barra de herramientas
- Antes de ejecutar hay que poner Breakpoints (puntos de ruptura)

Perspectiva de Debug

- Vista Debug ⇒Muestra los procesos que están ejecutando
- Vista Variables ⇒Muestra los valores de las variables para
- Vista Breakpoints ⇒Muestra los procesos que están ejecutando
- Vista Outline ⇒Muestra el método que se está ejecutando
- Vista Console ⇒Muestra la salida por consola que se ha ejecutado hasta el momento

Debug de Programas

Usando el Debug

Continuación de la Ejecución

- Todos los botones están en la Vista Debug
- Step Into ⇒Ejecuta la sentencia (si es un método entra)
- Step Over ⇒Ejecuta la sentencia (si es un método lo ejecuta sin entrar)
- Resume ⇒Continua la ejecución hasta el siguiente breakpoint o hasta el final del programa

Debug de Programas

Usando el Debug

Continuación de la Ejecución

- Todos los botones están en la Vista Debug
- Step Into ⇒Ejecuta la sentencia (si es un método entra)
- Step Over ⇒Ejecuta la sentencia (si es un método lo ejecuta sin entrar)
- Resume ⇒Continua la ejecución hasta el siguiente breakpoint o hasta el final del programa

Ver valores de Variables o Expresiones

- Todas las variables del ámbito se ven en la Vista Variables
- Los valores hacen referencia al momento en el que se ha detenido la ejecución

Índice

Documentación del código

Documentación Javadoc

- Muy útil para la generación de documentación del código
- Genera una página Web completa de forma automática
- El ejemplo más claro es la Web del API de Java
- Es obligatoria su entrega en el proyecto
- Permite documentar clases, métodos, atributos ...
- Dispone de tags específicos para el autor, los parámetros, las excepciones, ...
- Con /** sobre lo que se está documentando se genera casi todo

Documentación del código

Documentación Javadoc

- Muy útil para la generación de documentación del código
- Genera una página Web completa de forma automática
- El ejemplo más claro es la Web del API de Java
- Es obligatoria su entrega en el proyecto
- Permite documentar clases, métodos, atributos ...
- Dispone de tags específicos para el autor, los parámetros, las excepciones, ...
- Con /** sobre lo que se está documentando se genera casi todo

Generación Javadoc

- File ⇒Export ⇒Java ⇒Javadoc
- Establecer la carpeta doc de proyecto para generar la documentación

Índice

Otras posibilidades

Uso de TODO's y FIXME's

- TODO permite marcar dónde cosas que quedan pendientes
- FIXME permite marcar errores detectados que hay que corregir
- Ambos se pueden ver en la vista Tasks

Otras posibilidades

Uso de TODO's y FIXME's

- TODO permite marcar dónde cosas que quedan pendientes
- FIXME permite marcar errores detectados que hay que corregir
- Ambos se pueden ver en la vista Tasks

Uso de variables, atributos o métodos

- Si se selecciona una variable se sombrean todos sus accesos
- Con Ctrl + Click se va al código del método
- Con Right Click ⇒References ⇒Project se buscan todas las llamadas al método
- La vista outline nos permitirá ver los métodos de una clase

Atajos prácticos

Accesos rápidos

- Ctrl + Esp ⇒Sugerencia automática de lo que se puede hacer
- syso + Ctrl + Esp ⇒`System.out.println ()`
- Ctrl + i ⇒Indentar el código

Atajos prácticos

Accesos rápidos

- Ctrl + Esp ⇒Sugerencia automática de lo que se puede hacer
- syso + Ctrl + Esp ⇒`System.out.println ()`
- Ctrl + i ⇒Indentar el código

Botón derecho

- Source ⇒Override and Implement ⇒Sobreescribir método de la clase padre
- Source ⇒Organize Imports ⇒Organiza los imports, borra los no utilizados
- Source ⇒Generate Getter's and Seter's ⇒Genera automáticamente los métodos get y set para un atributo