Universidad de los Andes
Ingeniería de Sistemas y Computación

ISIS1205 – Algorítmica y Programación 2

Hoja de trabajo repaso APO1 (N7-EE)

Cupi2

Entender el manejo de proyectos en Eclipse	*Relacione los siguientes conceptos con la descripción más adecuada*	
1 Eclipse	Extensión de archivo con código java ejecutable	( )
2 Editor	Comprueba que el código fuente no tenga errores de sintaxis y permite la construcción del código ejecutable	( )
3 Proyecto	Archivo que contiene el modelo de un programa escrito en Racional Rose	( )
4 .class	Poner a funcionar un programa terminado	( )
5 Compilador	Ambiente de desarrollo	( )
6 .bat	Reúne los diferentes archivos que hacen parte de un programa	( )
7 Archivo fuente	Reúne un conjunto de clases con fines de organización	( )
8 Paquete	Organización del contenido de un proyecto en carpetas	( )
9 .java	Extensión de archivo texto con instrucciones ejecutables para hacer diversas tareas, entre ellas compilación y ejecución	( )
10 Estructura de directorios	Archivo con el código de un programa escrito en un lenguaje de programación	( )
11 Ejecutar	Extensión de una archivo fuente o implementación de clase en Java	( )
12 .mdl	Estándar de documentación de código fuente Java	( )
13 javadoc	Archivo que empaqueta el código ejecutable de un programa para facilitar su instalación	( )
14 .jar	Facilita la creación y modificación de código fuente Java	( )

Explorar un proyecto en Eclipse	*Siga las siguientes tareas y responda las preguntas*

1. Abrir el proyecto del curso n3_parqueadero. Puede hacerlo de dos formas:

Opción 1: Creando un proyecto sobre la estructura de directorios dada (Menú File/New/Project)

- Descomprima el archivo zip con el proyecto (por ejemplo en C:/temp/)
- Cree un proyecto Java en eclipse, con la ruta del directorio (C:/temp/n3_parqueadero) y el nombre del proyecto (n3_parqueadero)
- Puede aceptar la creación ahora (botón “Finish”), o navegar a la siguiente ventana (“Next”) para ver las propiedades del proyecto.

Opción 2: Importando el proyecto de la estructura de directorio dada (Menú File/Import)

- Descomprima el archivo zip con el proyecto (por ejemplo en C:/temp/)
- Elija la opción de “Proyecto Existente en Workspace”
- Seleccione la carpeta del proyecto (C:/temp/n3_parqueadero) y finalice

2. Explorar el proyecto desde eclipse como un sistema de archivos

- Utilice la vista llamada navegador. Si la vista no está disponible, búsquela en el menú Window/Show View/Navigator
- Revise la estructura de directorios del proyecto cupi2 y recuerde el contenido de cada carpeta (no necesariamente todas las carpetas tienen contenido en el proyecto que está explorando)

bin	
classes	
data	
docs/api	
docs/spec	
lib	
source	
test/classes	
test/data	
test/lib	
test/source	

3. Explorar el proyecto en Eclipse como un proyecto Java
   - Utilice la vista llamada Package Explorer. Si la vista anterior no está disponible, búsquela en el menú Window/Show View/Package Explorer
   - Revise las propiedades del proyecto. Puede elegir las propiedades con click derecho sobre el proyecto o en el menú Project/Propeties
   - Seleccione de las opciones a la izquierda las opciones de construcción de Java ("Java Build Path") y revise la configuración del proyecto
   - Observe la estructura de paquetes del proyecto

Los paquetes sirven para organizar las clases en un proyecto. ¿Qué tipo de organización están dando los paquetes en este ejemplo?

Crear un proyecto en Eclipse	*Siga las siguientes tareas y responda las preguntas*

1. Crear el proyecto Java
   a. Sitúese en la vista Package Explorer
   b. Cree un nuevo proyecto en el menú File/New/Project
   c. Elija un proyecto Java (Java Project)
   d. Nombre al proyecto como *nuevoProyecto* y seleccione que su locación sea en el workspace
   e. Cree la carpeta de fuentes source
   f. Defina como salida por defecto el directorio nuevoProyecto/classes
   g. Seleccione la opción de crear carpetas de salida para las carpetas fuentes
   h. Cree la carpeta de fuente test/source
   i. Selecciones su fólder de salida y edítelo para direccionarlo a test/classes
   j. Finalice la creación del proyecto
2. Completar la estructura de directorios
   a. Sitúese en la vista Navigator
   b. Sobre el proyecto cree directorios que falten de la estructura
   c. Copie del ejemplo completo que se le entregó, la librería junit.jar ubicada en la carpeta test/lib y péguela en la carpeta correspondiente en el proyecto nuevo
3. Agregue librerías para la compilación
   a. Abra el diálogo de las propiedades de nuevoProyecto
   b. Oprima el botón "Add JARs" (agregar jars)
   c. Seleccione de test/lib la librería junit.jar

Hasta este punto ya sabe cómo explorar y configurar un proyecto en Eclipse. Si va a construir un ejemplo Cupi2, ¿cuál sería el siguiente paso?

Editar código en Eclipse	*Siga las siguientes tareas y responda las preguntas*

1. Crear un nuevo paquete
   a. Explore a nuevoProyecto con la vista Package Explorer
   b. Ubíquese en la carpeta de Archivos Fuente source y cree un package de nombre *uniandes.cupi2.ejemplo* (click derecho o menú File/New/Package)
   c. Revise la estructura de directorios que se creó dentro de la carpeta source, en la vista Navigator

2. Crear una nueva clase
   a. Explore a nuevoProyecto con la vista Package Explorer
   b. Ubíquese sobre el paquete que creó anteriormente
   c. Cree una nueva clase (click derecho o menú File/New/Class) de nombre Bombillo.
   d. Al crear la clase usted se encuentra en el editor y puede completarla
   e. Escriba una descripción para la clase Bombillo, ¿Qué representa esta clase?
   f. Adiciónele un atributo *vatios* de tipo entero
   g. Adiciónele un atributo *prendido* de tipo bolean
   h. Escriba el constructor de la clase. Para crear un bombillo es necesario indicar de cuantos vatios es (parámetro). Al crear un bombillo este permanece apagado.
   i. Escriba un método *prender*. ¿Qué parámetros tiene el método? ¿Qué atributos de la clase debe afectar el método? ¿El método debe retornar algo?
   j. Documente el método prender, con su descripción, parámetros y retorno
   k. De forma similar, escriba un método *apagar* y documéntelo.
   l. Escriba además un método *estaPrendido* que me indique si el bombillo está prendido o no. ¿Qué parámetros tiene el método? ¿Algún atributo de la clase se afecta en este método? ¿Qué debe retornar el método?

Al crear la clase es probable que en su código fuente aparezca una documentación por defecto de archivo (al comienzo del archivo) y de clase (antes de la declaración de la clase).

Puede cambiar estos comentarios que se generan automáticamente en el menú Window/Preferences, en el ítem Java/Code Style/Code Templates. Se presenta una lista de templates que puede modificar. Edite solamente Comments/Types y Code/New Java Files

3. Dé formato a la clase
   a. Adjunte el profile (perfil) de formato del curso llamado cupi2-profile.xml, en el menú Window/Preferences, en el ítem Java/Code Style/Code Formater
   b. Abra la clase Bombillo en el editor
   c. Aplíquele el formato a la clase Bomibllo seleccionando la opción Source/Format en el menú emergente del clic derecho o con ctrl+shfit+F
   d. Para darle formato a una sola sección de la clase, seleccione la sección y aplíquele el formato como en el paso anterior

Buscar clases o elementos de clase en el proyecto	*Siga las siguientes tareas y responda las preguntas*

1. Localizar ágilmente el código fuente de una clase o método
   a. Explore el proyecto del curso n3_parqueadero
   b. Ubique la clase Parqueadero en el árbol del explorador
   c. Localice el atributo puestos de la clase Parqueadero y revise su definición en el explorador
   d. Ubique el cursor del ratón sobre la palabra Puesto en la declaración del atributo `puestos` y oprima la tecla ctrl. La palabra puesto aparece subrayada
   e. Oprima click sobre la palabra resaltada y podrá ver el código de la clase Puesto

2. Localizar ágilmente las referencias a una clase o método
   a. Explore el proyecto del curso n3_parqueadero
   b. Ubique la clase Carro en el árbol del explorador
   c. Encuentre las clases en las cuales se crean carros, seleccionando en el método constructor y eligiendo la opción Search/References en el menú principal o el menú emergente del clic derecho sobre el método. Existen dos alcances principales (a) project, para buscar sólo en el proyecto al que pertenece la clase y (b) workspace, para buscar en todos los proyectos del área de trabajo
   d. En la vista de búsqueda se presentan todas las clases en las que existe se crea un carro y sus ubicaciones se resaltan sobre la clase abierta
   e. Ubique la clase Parqueadero en el árbol del explorador
   f. Encuentre las clases en las que se usa el método calcularPuestosLibres(), seleccionándolo y eligiendo la opción Search/References
   g. Si hace la búsqueda sobre el nombre de una clase puede encontrar todas aquellas clases en las que se tiene una referencia al tipo o clase.

¿En qué situaciones puede serle útil encontrar ágilmente el código fuente o las referencias a una clase, o método?

Ejecutar el proyecto en Eclipse	*Siga las siguientes tareas y responda las preguntas*

1. Ejecutar la clase principal del proyecto
   a. Explore el proyecto del curso n3_parqueadero
   b. Ubique la clase InterfazParqueadero en el árbol del explorador
   c. Elija el comando Run as Java Application. Puede hacerlo desde la barra de herramientas, el menú principal o el menú emergente del clic derecho sobre la clase

2. Revisión del método de ejecución
   a. Explore el proyecto del curso n3_parqueadero
   b. Ubique el método main de la clase InterfazParqueadero en el árbol del explorador
   c. Ábralo y revíselo en el editor de código

El método main es el que permite la ejecución de un programa en Java y siempre se escribe con la misma signatura. Dicho en otras palabras es el punto de inicio de la ejecución. Describa las acciones que está haciendo este método según las instrucciones que contiene.

Eclipse facilita las tareas de edición, compilación, seguimiento y ejecución de programas. Sin embargo tareas como la compilación y la ejecución no son exclusivas del ambiente de desarrollo. Los archivos ejecutables incluidos en el directorio bin son archivos de texto que también saben hacer estas tareas

Modificaciones de los elementos del proyecto en Eclipse (Refactoring)	*Siga las siguientes tareas y responda las preguntas*

1. Cambiar el nombre de una clase
   a. Ubique la clase Carro en el explorador de paquetes
   b. Renombre la clase a Vehiculo seleccionando la opción Refactor/Rename en el menú principal, en el menú emergente del clic derecho sobre el nombre de la clase

2. Cambiar el nombre de un atributo
   a. Ubique la clase Puesto en el explorador de paquetes
   b. Renombre el atributo carro a vehiculo seleccionando la opción Refactor/Rename en el menú principal, en el menú emergente del clic derecho sobre el nombre del atributo

3. Cambiar el nombre de un método
   a. Ubique la clase Puesto en el explorador de paquetes
   b. Renombre el método parquearCarro por parquearVehiculo seleccionando la opción Refactor/Rename en el menú principal, en el menú emergente del clic derecho sobre el nombre del método

4. Cambiar el nombre de un parámetro
   a. Ubique la clase Puesto en el explorador de paquetes
   b. Abra en el editor el método parquearVehiculo
   c. Renombre el parámetro carroP por vehiculoP seleccionando sobre el editor el nombre del parámetro y luego seleccionando la opción Refactor/Rename en el menú principal, en el menú emergente del clic derecho sobre el nombre del método
   d. Otra manera de hacerlo es ubicar el método en el explorador de paquetes y seleccionando la opción Refactor/Change Method Signature

5. Ejecute el proyecto con los cambios
   a. Ejecute de nuevo el programa
   b. Observe que el programa continúa funcionando correctamente

El término *refactoring* se utiliza para indicar la reestructuración de un proyecto.

El principal cambio es el renombramiento de los elementos, pero también se incluyen otras acciones como mover los elementos. El refactoring puede aplicarse también a paquetes y carpetas.

¿Qué pasaría si tuviera que cambiar el nombre de un método de una clase sin tener esta utilidad?