

Tarea 2

Auxiliar: Juan Pablo Sandoval

Prof. Alexandre Bergel

Ayudantes: Sebastian Ferrada, Andrés Riveros, Hans Fehrmann R.

October 23, 2015

1 JROWSER: THE CC3002 CODE BROWSER

JBrowser es un programa que sirve para navegar sobre un project JAVA, resaltar y colorear código (ver figura 1). *JBrowser* toma como entrada un proyecto java (el folder donde esta el proyecto) y lo organiza en paquetes, clases y métodos.

2 TAREA 1

La tarea 2 consiste en mejorar *JBrowser*, la idea es mejorar *JBrowser* agregando un conjunto de características:

- **Sort** – JBrowser debe tener un menú que le permita ordenar los paquetes y clases con las siguientes opciones: “ordenar paquetes por nombre” , “ordenar clases por nombre” , “ordenar clases por jerarquía”.
- **Iconos** – Actualmente todos los paquetes, clases y métodos se muestran con un icono por defecto. JBrowser debe asignar los iconos a los siguientes elementos: Packages, Empty packages, Clases, Interfaces, Public methods, Protected methods y Private methods.
- **Background and Font** – Se debe asignar un background rojo a los métodos con mas de 50 lineas de código y un background amarillo a los métodos con mas de 30 lineas de código. Las letras de los nombres de las interfaces debe estar en *ITALIC*.
- **Line numbers** – Sin importar el estilo se debe mostrar los “line numbers” en el código.

La figura 2 muestra como *JBrowser* debería verse con las nuevas características.

Demo Test Para cada uno de los puntos anteriormente mencionados existen algunos test de ejemplo en el proyecto. Las imagenes, el font, el font size y toda la información necesaria ya esta en los test. **Por favor vean los test antes de preguntar en el foro**

- **Features:** modifique, re-escriba, refactorize, agregue, borre todo el código necesario para soportar las nuevas características.
- **Coverage:** cree los tests necesarios para tener un coverage de al menos 80 % por paquete, con la única excepción del paquete *parser*.
- **Javadoc:** documente todas las clases y métodos del proyecto, exceptuando el paquete *parser*.
- **Conventions:** el código y documentación deben seguir las convenciones de google *Google Java Style*¹

Todo el proyecto tiene que estar desarrollado en eclipse. **Deben subir la carpeta del proyecto eclipse en un ".zip" a u-cursos.** Al subir su tarea a u-cursos, agregue la dirección de correo electrónico que uso para testear su tarea (ver siguiente sección)

ECLIPSE Antes de empezar usted debe descargar el proyecto eclipse de u-cursos. Para importar el proyecto a eclipse, debe seleccionar el menu File>Import...>General>Existing Project into Workspace (si antes tienen que copiar el proyecto al workspace que seleccionar al abrir eclipse). Una vez importado el proyecto tienen que agregar al path los .jar de la carpeta "lib". Para lo cual tienen que hacer click derecho en el proyecto eclipse (JBrowse) seleccionar Properties>Java Build Path> Libraries> Add External JAR... **cualquier duda sobre este paso, consultar por el foro.**

2.1 EL CHANGO 0.04

Este semestre, tendremos un soporte automático para la revisión de sus tareas. El sistema se llama el "El chango", el mismo permite revisar automáticamente los puntos descritos en la anterior sección: Style, Coverage, Javadoc y Conventions. Además, permite encontrar posibles errores de diseño en su tarea.

SUBMISSION FORM El chango provee un pequeño formulario para que puedan enviar su tarea para la revisión. Para lo cual tienen que comprimir la carpeta "src" de su proyecto eclipse en un ".zip" y enviarla junto con su dirección de correo electrónico. El formulario está disponible en la siguiente página <http://users.dcc.uchile.cl/~eriveros/cc3002/elchango/>.

REPORTS Una vez enviada su tarea esta será revisada automáticamente por el chango, al finalizar usted recibirá un html que contiene un reporte completo de todos los posibles problemas en su proyecto.

¹<http://google.github.io/styleguide/javaguide.html>

EVALUATION

- **Funcionalidad (3.0)**, el cambio, tiene sus propios tests para revisar la funcionalidad del proyecto (SMegaTest). Se asignará la nota proporcionalmente al número de tests buenos.
 - **2.0** Evaluado automáticamente por los tests del cambio.
 - **1.0** Evaluado manualmente por los ayudantes. Sobre todo que se actualice la UI cuando se apliquen los ordenamientos.
- **Coverage (1.0)**, basado en el resultado del coverage por paquete (sin contar el paquete parser). De igual forma es proporcional.
- **Documentación (1.0)**, basado en el reporte de el cambio.
- **No Code Smell (1.0)**, basado en el reporte de el cambio.
- **Descuentos**, finalmente se procederá a la revisión del diseño de clases de su proyecto, se aplicará un descuento de a lo más 1 pto, según corresponda a su diseño de clases.
- **u-cursos**, no olvide subir su proyecto eclipse a u-cursos en formato .zip.

NOTA Para la tarea 2 aumentaremos los descuentos por método/clase no documentada a 0.05 y por code smell a 0.1.

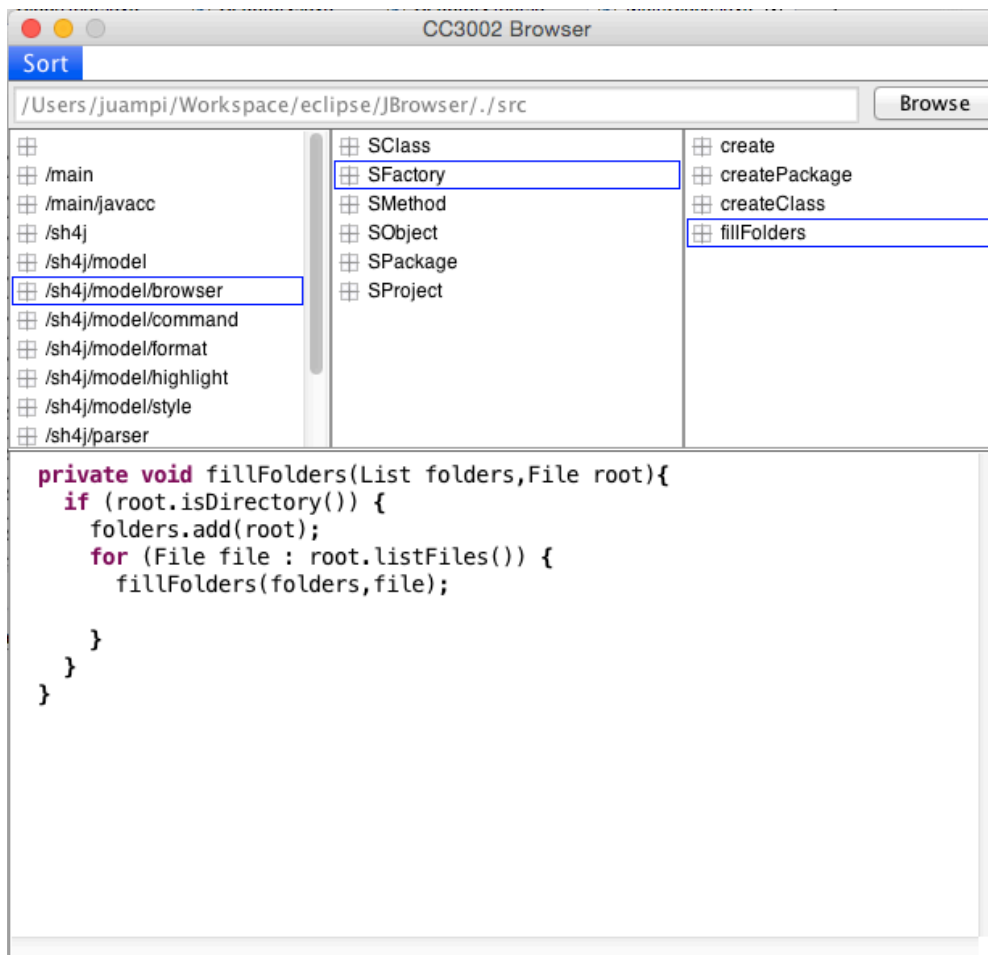


Figure 1: JBrowser Hoy

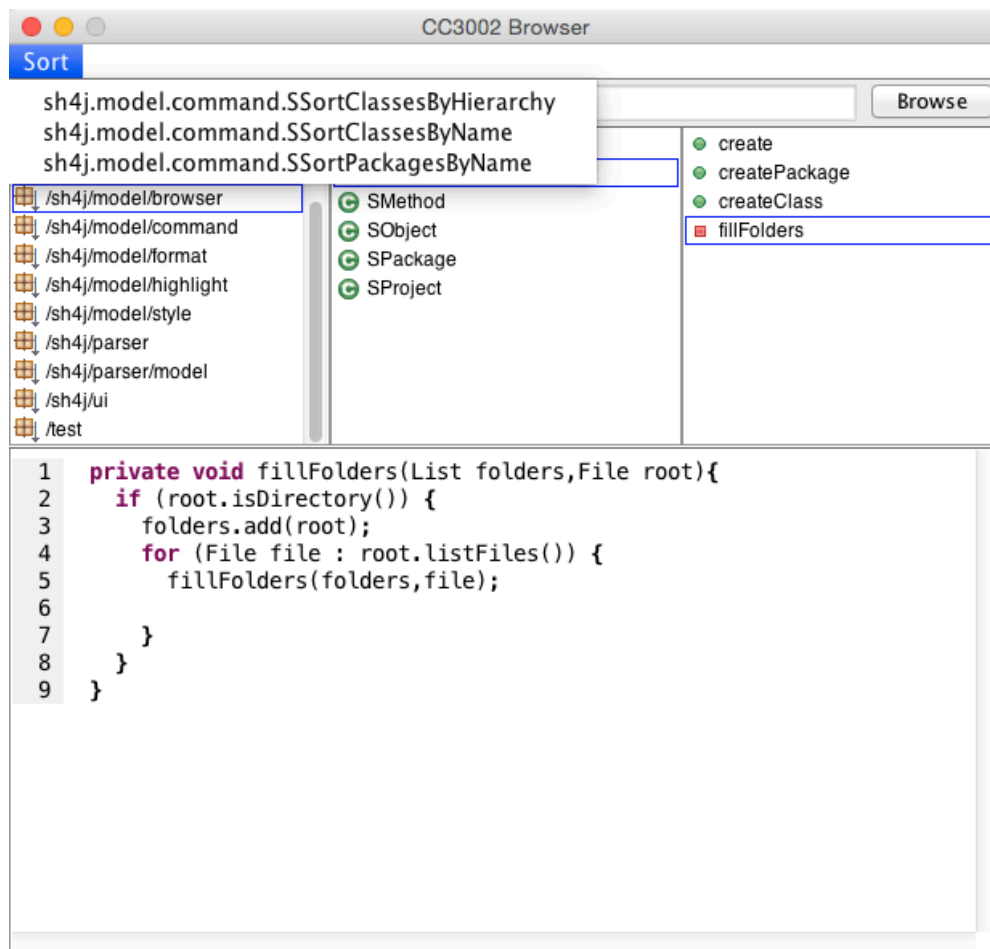


Figure 2: JBrowser Mañana