Fernando Solar Iglesias

Guion análisis estático de código Java para la detección de errores con FindBugs/SpotBugs

Este documento se trata de un anexo del documento del trabajo principal con la finalidad de que pueda ser usado como guion para el taller de la aplicación.

1. Instalación de SpotBugs como plugin de Eclipse

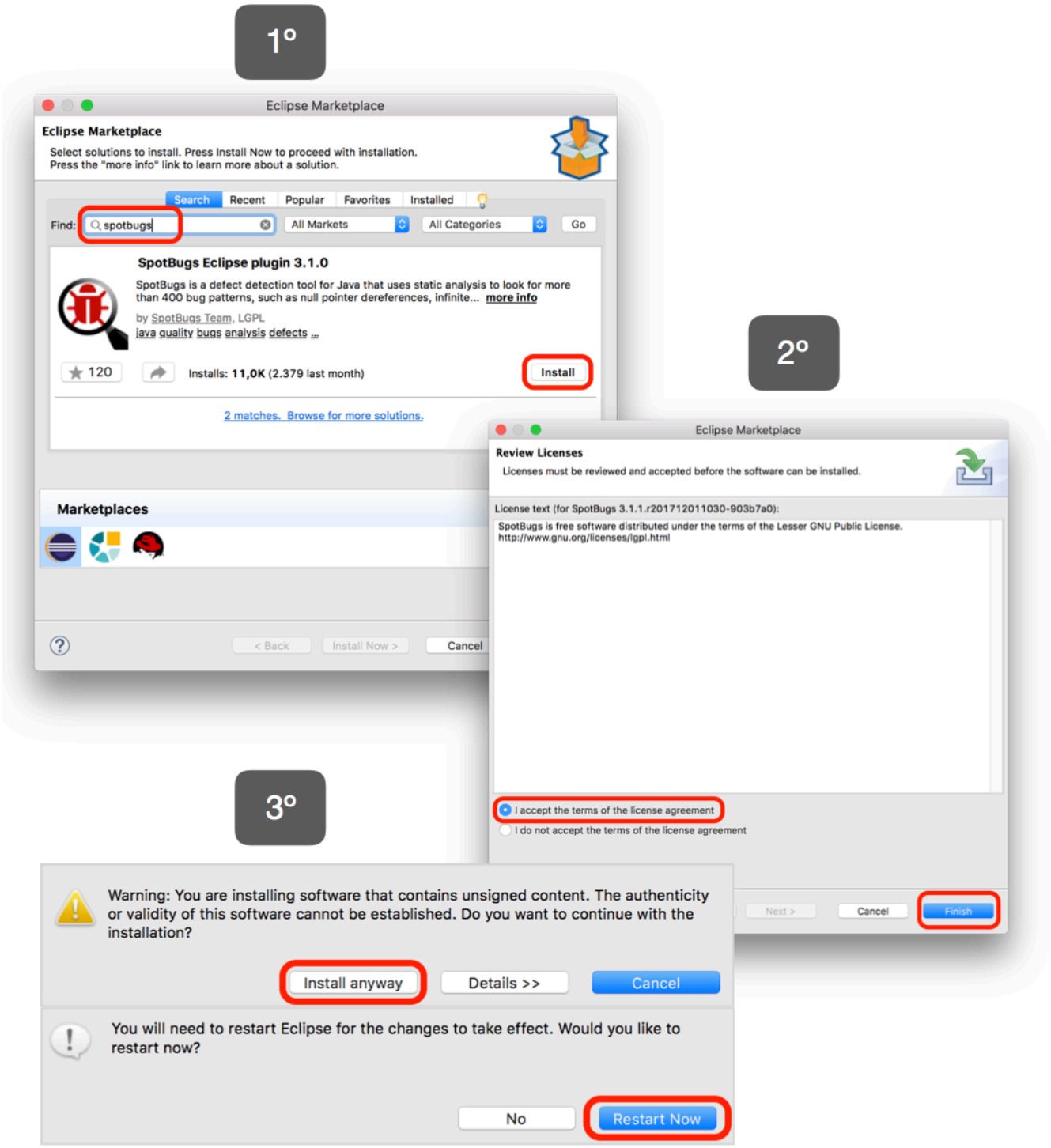
Existen varias formas de usar FindBugs/SpotBugs, y por tanto, de instalarlo. Puede ser usado como programa aislado a través de su interfaz o integrado con otros componentes: plugin de Eclipse, tarea de Ant, plugin de Maven o plugin de Gradle. Dependiendo de las necesidades o herramientas propias del proyecto, conviene realizar una instalación u otra.

Con la finalidad de establecer un entorno en común y de baja complejidad con el que comenzar, en este caso se va a hacer uso del plugin de Eclipse, editor que va a ser usado como referencia para abrir y explorar el código a analizar.

1. Una vez se cuenta con la versión Neon (4.6) o posterior del IDE, es suficiente con abrir el Marketplace (Help >> Eclipse Marketplace…) y realizar la búsqueda de “SpotBugs” para posteriormente hacer clic sobre el botón de “Install/Instalar”.

2. La interfaz nos guiará por el proceso de instalación que consiste en leer y aceptar los términos de acuerdo de licencia y posteriores advertencias.

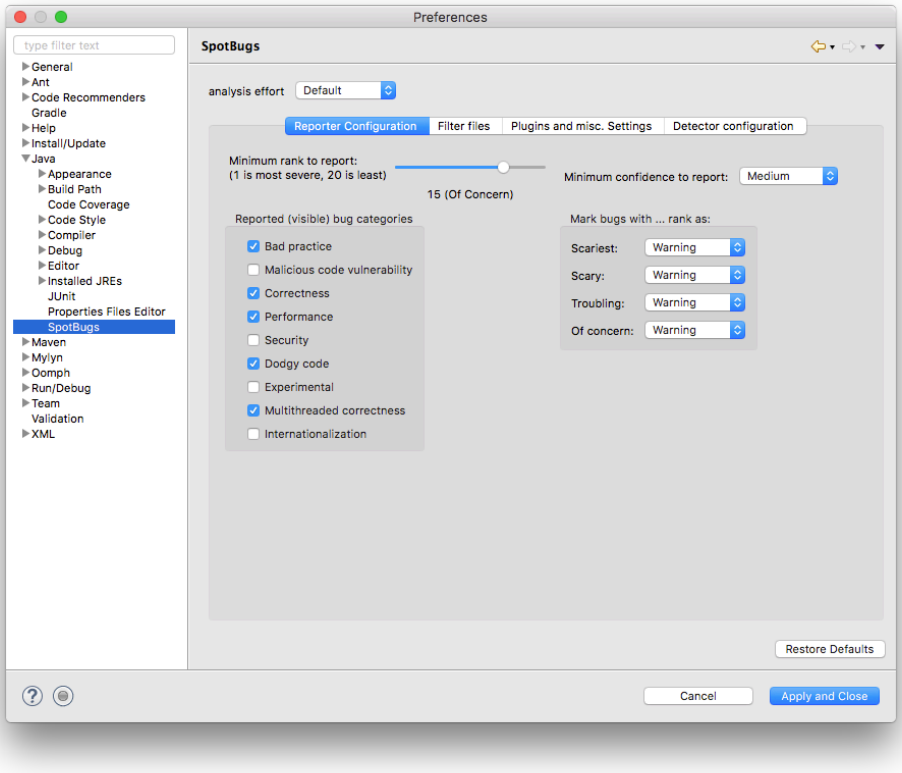
3. Una vez seguidas las indicaciones, Eclipse requerirá reiniciarse para completar el proceso. Una vez aceptado y cuando vuelva a arrancarse, el plugin ya se encontrará listo para su uso.



Resumen gráfico del proceso de instalación

2. Configuración de SpotBugs como plugin de Eclipse

Una vez instalado, es posible acceder a las opciones de configuración que permiten personalizar los parámetros de los que está compuesta la herramienta. Para ello, es suficiente con acceder a las preferencias de Eclipse, y dentro del desplegable “Java”, acceder a la sección “SpotBugs”, para seguidamente visualizar la siguiente interfaz.



La opción global “analysis effort” permite definir el esfuerzo que se le permite afrontar al programa para encontrar errores. En valores bajos, algunos tipos de errores de gran coste computacional, alta trazabilidad o amplio tamaño serán ignorados; mientras que para el valor superior, todos los tipos de casos serán analizados en el código dado.

“Reporter Configuration” permite personalizar el informe que nos proporciona el programa. Desde qué tipos de bugs nos pueden ser de interés, tanto por categoría como por importancia, así como la manera en la que deseamos que se muestre cada uno (error, advertencia, informativo). Seleccionaremos todas la categorías de bug y el máximo (20) rango de reporte.

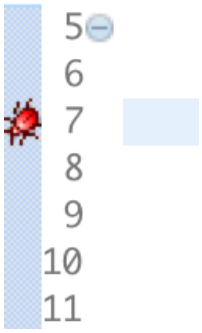
Mientras que “Filter files” da la opción incluir o excluir determinados métodos o clases del análisis, “Detector configuration” facilita contar o no contar con algunos detectores de errores.

Finalmente, “Plugins and misc. Settings” permite configurar aspectos generales del plugin, como el modo de operación y funcionamiento y aspectos relativos al rendimiento y la cache.

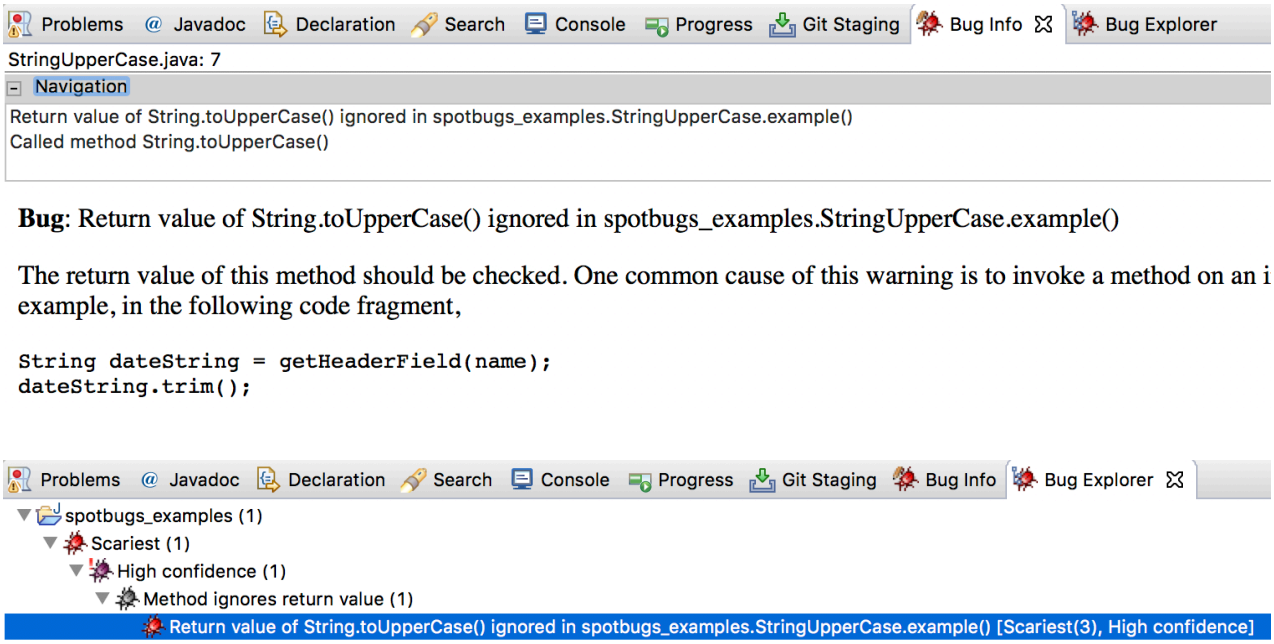
3. Ejecución y visualización de resultados

Para comenzar, basta con situarse en Eclipse sobre el proyecto, paquete o clase sobre el que queremos realizar el análisis y dar clic derecho sobre éste para seguidamente acceder al menú “SpotBugs” y seleccionar la opción “Find Bugs”. Inmediatamente después, el proceso se pondrá en marcha en base a la configuración que se haya establecido inicialmente.

Una vez termine el análisis se podrá visualizar, a la izquierda del indicador de linea, aquellas en las que se ha detectado algún error mediante el icono de un pequeño insecto rojo o verde, dependiendo de la importancia.



Haciendo clic sobre cada uno de los iconos de insecto, es posible acceder a información detallada sobre el error en las nuevas ventanas que se disponen: “Bug Info” y “Bug Explorer”.



4. Propuesta de taller

A continuación, se ofrece una taller guiado para poner en práctica el uso de FindBugs/ SpotBugs. Para ello, se utiliza el código adjunto al anexo en el que cada una de las clases corresponden a los ejemplos y Main a la principal que permite ejecutarlos.

Una vez instalada y configurada la herramienta y descargado e importado el proyecto en Eclipse, se deberán seguir los siguientes pasos **sin hacer uso de SpotBugs**. Una vez hayas razonado el ejercicio, aplica las instrucciones del paso 3 **para comprobar la solución** a través de la herramienta.

1. Abre **SwitchExample.java** para encontrar el primer código de ejemplo. Razona si la descripción de la clase corresponde con una implementación correcta en el método example. ¿Encuentras algún error sin usar la herramienta?

2. Explora **UpperCaseExample.java** para encontrar el segundo código de ejemplo. Razona si el código del método example cumple con la descripción de la clase. Antes de comprobarlo con la herramienta, ¿crees que hay algún error?

3. Abre **DivisionExample.java** para encontrar el tercer código de ejemplo. Razona si la descripción de la clase corresponde con una implementación correcta codificada en el método example. ¿Encuentras algún error sin usar la herramienta?

4. Finalmente, accede a **ValidationExample.java** para hacer uso del último código de ejemplo. Razona si la descripción de la clase corresponde con la implementación codificada en el método example. ¿Piensas que habrá algún tipo de error al comprobarlo con SpotBugs?