Code Analyzer

User Manual

Publication Date: April 2014

Version: 0.99b

©2014

Java is a registered brand of Oracle Corporation.

The UEM logos are property of “Universidad Europea de Madrid” which belongs to the “Laureate International Universities” group. All rights reserved.

This document is provided ‘As it is’. The information that this document contains including URL’s and other references to Internet websites, may change in the near future.

[Creative Commons License](http://creativecommons.org/licenses/by-nc-sa/4.0/)

Code Analyzer by David H. and bensteUEM is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-nc-sa/4.0/).  
Based on a work at <https://github.com/bensteUEM/uem.basesDeDatos.hernandezstein>

**Contents**

[1. Introduction to the program interface 4](#_Toc386307812)

[2. Program Functionalities 5](#_Toc386307813)

[3. Program Usage 6](#_Toc386307814)

[Step 1 – Selecting a Java source code file 6](#_Toc386307815)

[Step 2 – Analyzing the code 6](#_Toc386307816)

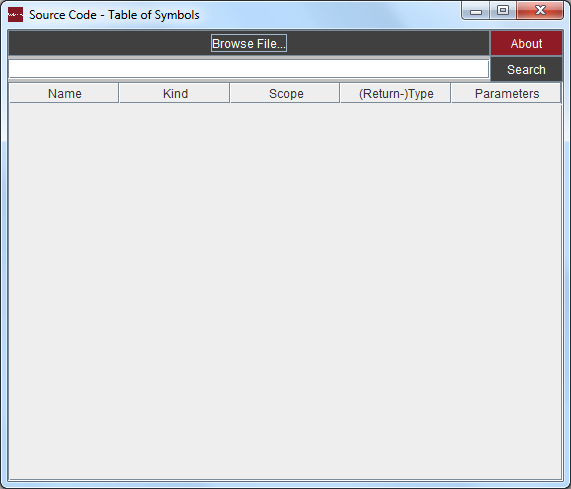
[Case 1 – Source code analysis detects compilation errors 6](#_Toc386307817)

[Case 2 – Source code analysis detects no compilation errors 7](#_Toc386307818)

[Step 3 – Searching for specific identifiers 8](#_Toc386307819)

# 1. Introduction to the program interface

When the software is executed, the following window appears:



This graphical interface was designed to be very simple. The following elements may be found:

1. “Browse File…” - Button
2. Search Textbox
3. “Search” – Button
4. Table

# 2. Program Functionalities

This section lists the program main features.

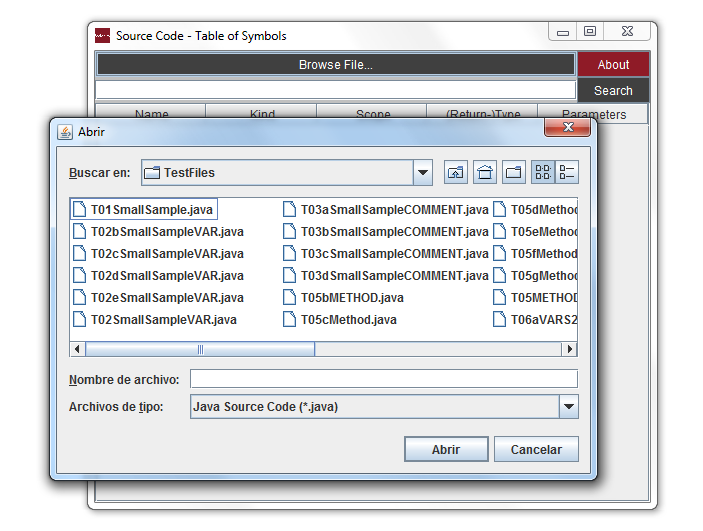
1. Analyze the code inside a java source code file and view detailed information about it:
   1. Class, methods & variables identifiers
   2. Kind of programming element
   3. Scope
   4. In the case of methods, their return type and its parameters.
2. Search for a concrete variable, method or class identifier and view detailed information about it.
3. Detect compilation errors in a java source code file such as unbalanced curly braces or variables used out of their scope. But please note that this program does not check package declaration mistakes.

# 3. Program Usage

In this section, the program operation is described step by step. After reading this section, the user will be familiarized with the interface.

## Step 1 – Selecting a Java source code file

Click the “Browse file” button. A file chooser dialog will appear:



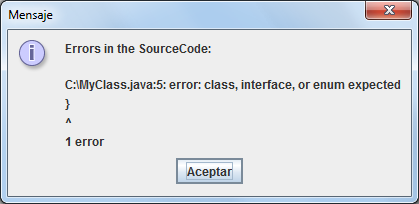
Browse to the location of the java source code to be analyzed. Select the file and click “open”.

## Step 2 – Analyzing the code

Once the java source code file has been opened, the program will start analyzing the code. When the analysis finishes, a message dialog will appear. This message dialogs notifies the user whether the source code contains compilation errors or not.

### Case 1 – Source code analysis detects compilation errors

If errors were found during the source code analysis, the program will show a message dialog indicating that issues were detected. For example, if the analysis detects unbalanced curly braces, it will show the following dialog:

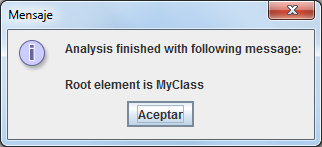


In this dialog, the user can check the location of the java source code file, followed by the line in which the compilation error has been found.

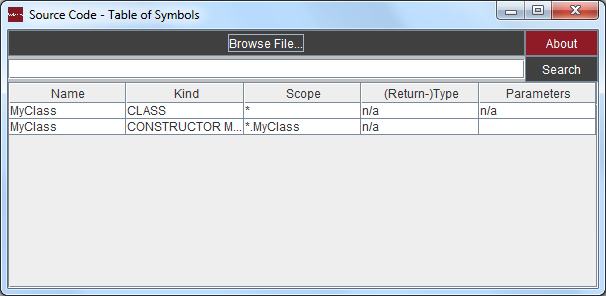
It is important to notice that when the source code analysis finds compilation errors, no information is displayed in the table of symbols. In this situation, the user should take note of the line where the error has been found and notify the programmer about it.

### Case 2 – Source code analysis detects no compilation errors

In the case that no errors were detected during the source code file analysis, the program will show the following message dialog:



In this case, we can click “OK” and view the analysis results in the table of symbols present in the graphical user interface:



The table of symbols displays detailed information about the Java source code file analyzed.

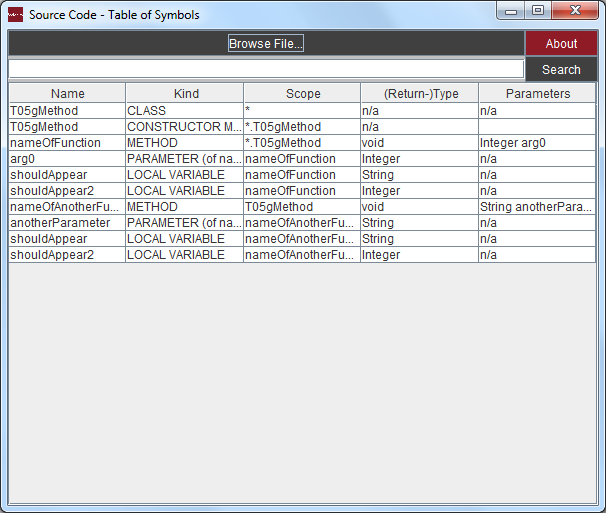
The columns show:

1. Name –The name or identifier of a Java class, method or variable.
2. Kind – The kind of programming element.
3. Scope – The scope.
4. Return Type – The return type of a method.
5. Parameters – The method parameters identifiers and their type.

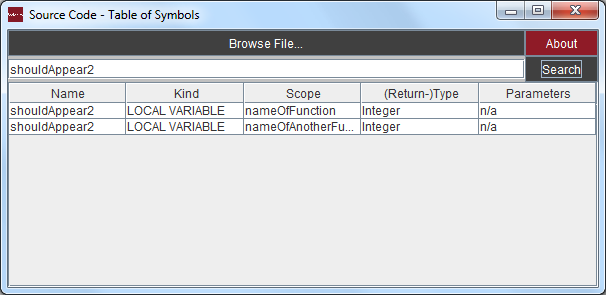
## Step 3 – Searching for specific identifiers

When analyzing large source code files, it may result difficult to find a specific variable, method, class in the table of symbols. To solve this issue, the user can type the specific identifier inside the textbox and click the “search” button. The program will search for the specific identifier in the table of symbols.

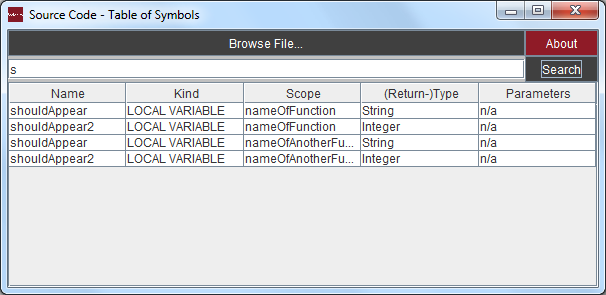
Consider the following example. The following screenshot displays the output of an analysis:



The user is able to search for any specific identifier:



There is also another way of searching for identifiers. The user may type also the first letters of the identifier and then click the “Search” button. In the previous example, the user may perform the following search:



The program displays all the identifiers that begin with those letters.