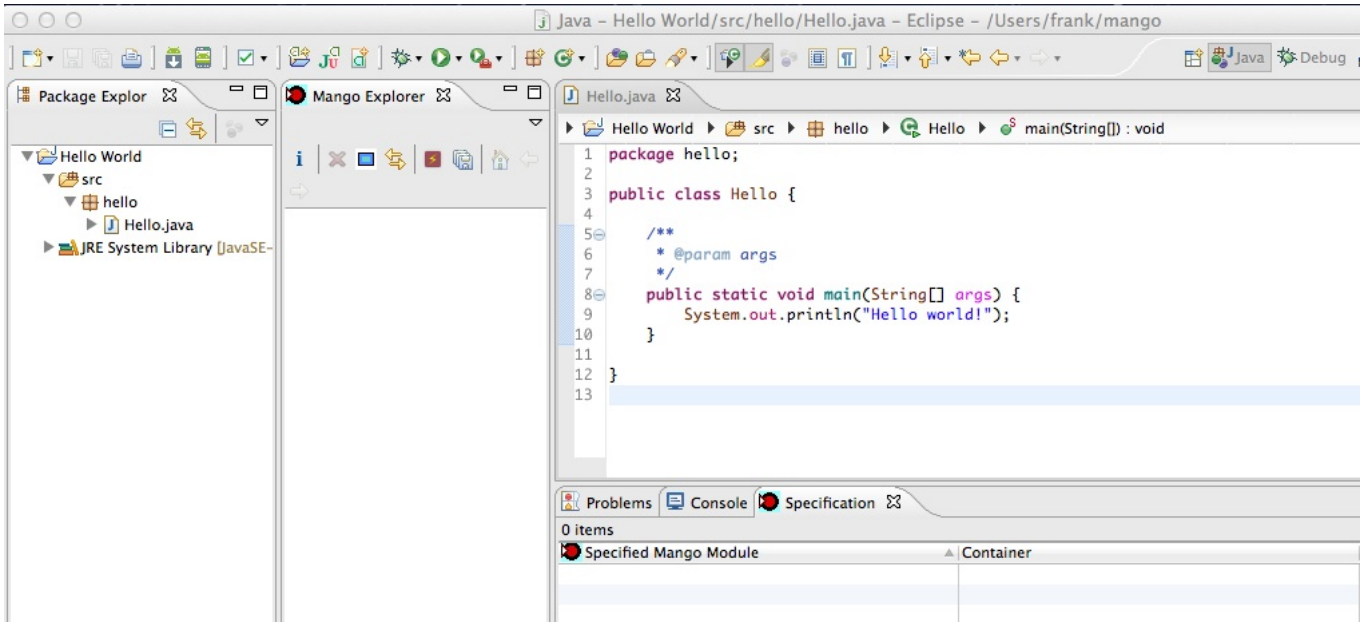
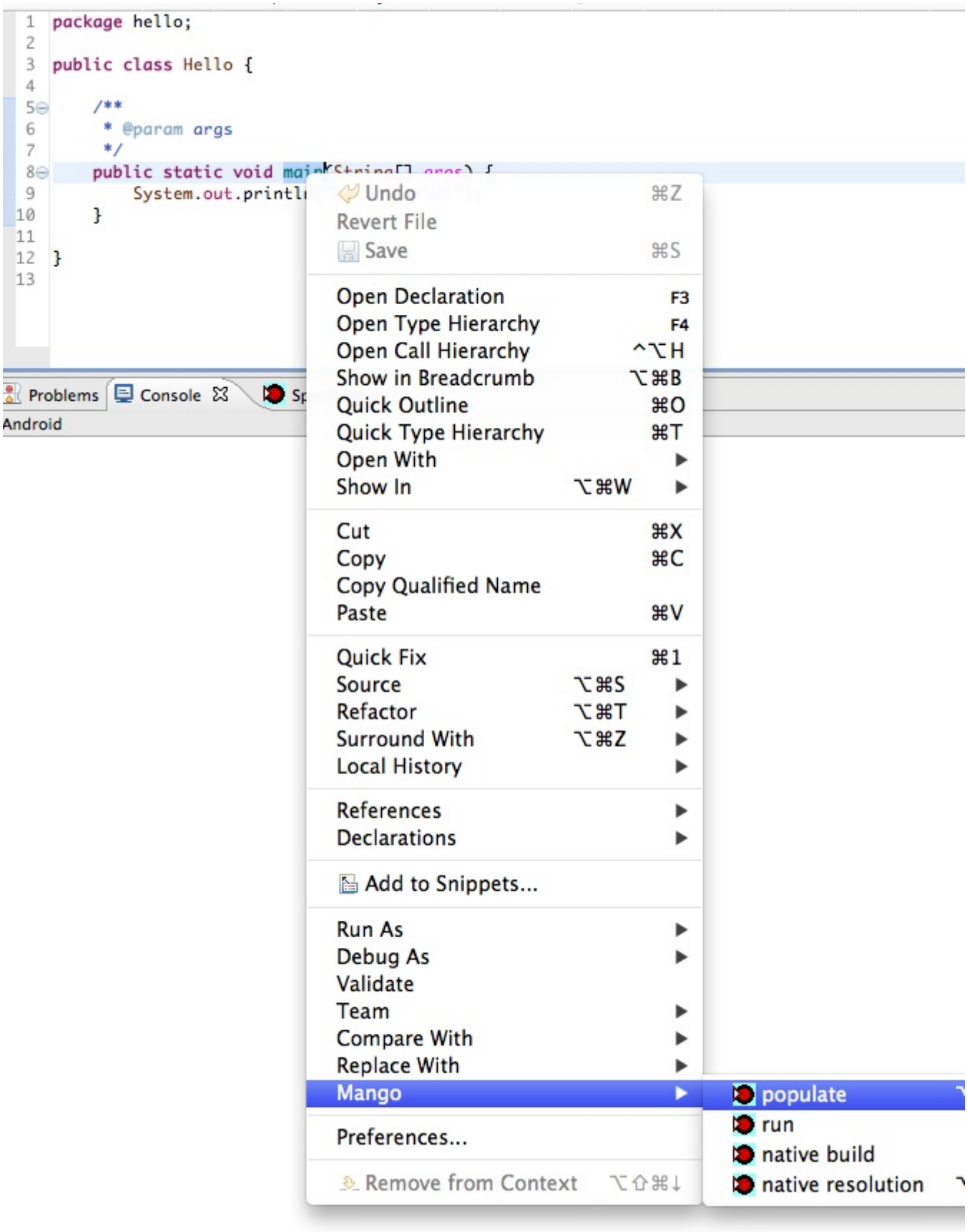
Getting Started

- Let's work through the traditional "Hello World" example.
  1. In the Java perspective, go to File>New>"Java Project".
  2. Enter "Hello World" for the project name, and hit "Finish".
  3. Right click on the "src" folder for the "Hello World" project. Add a new class to the Hello World project, checking the box to build the main routine.
  4. Add the traditional "Hello world!" println to the main routine, (see figure).



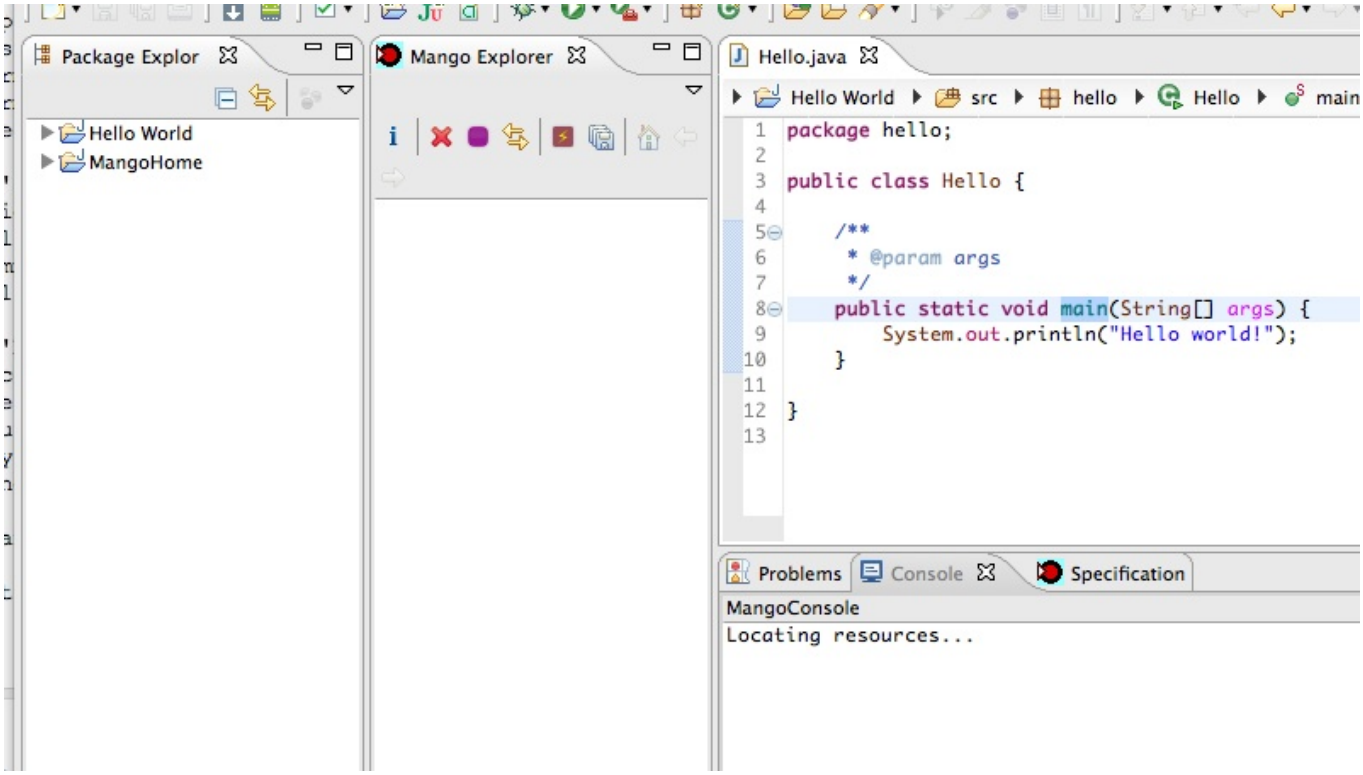
- "Hello World" continued:
  1. Build the project. This will happen automatically if Project>"Build automatically" is checked.
  2. Double-click on "main". This will ensure that both "main" and the "Hello World" tab are selected.

1. Right-click on "main", and select Mango>populate (see figure).



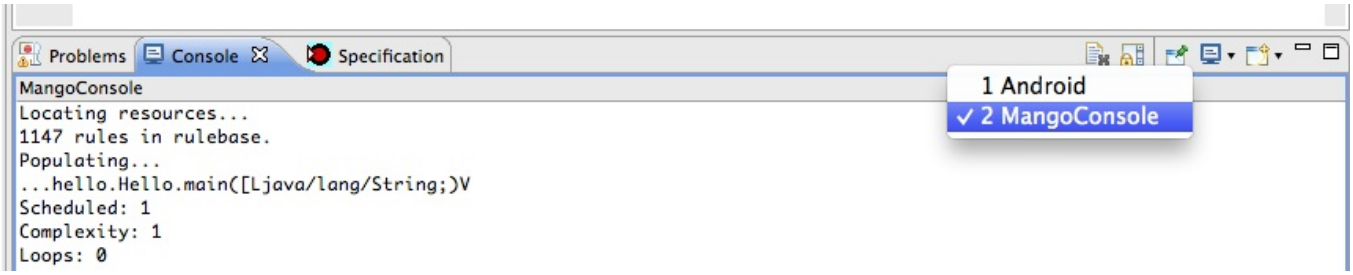
- "Hello World" continued:

1. The first time the "populate" command runs, it must install MangoHome and MangoSystem in the workspace. This takes some time. The box just below the "Mango Explorer" tab is called the Mango button. It is purple while the "populate" command is running. Also, various comments will appear in the Mango console view (see figure).



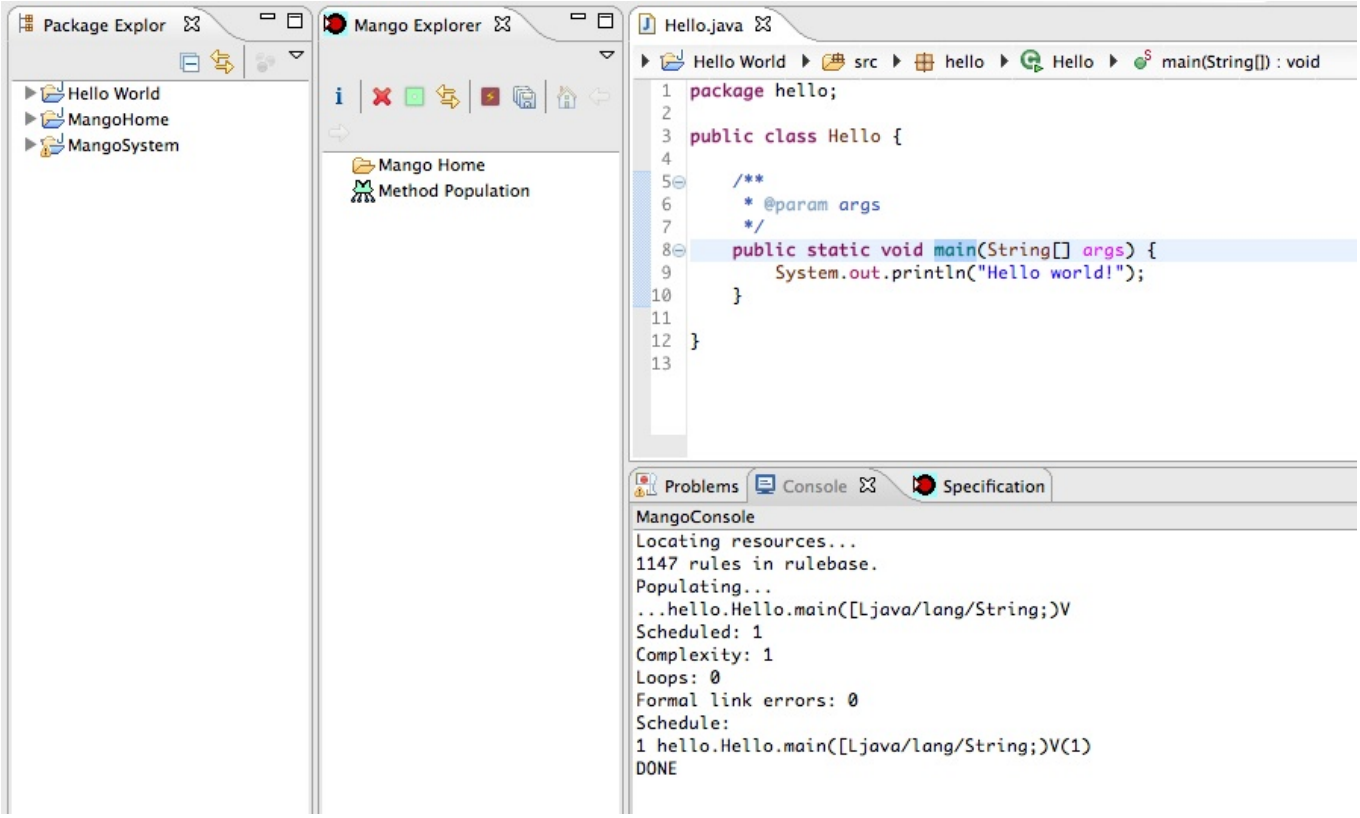
- "Hello World" continued:

1. If you have no console view, now is a good time to install it, via Window>"Show View". If you have multiple plugins installed, the console view may not be tuned to Mango. You can view the Mango console by selecting it with the Console control (see figure).



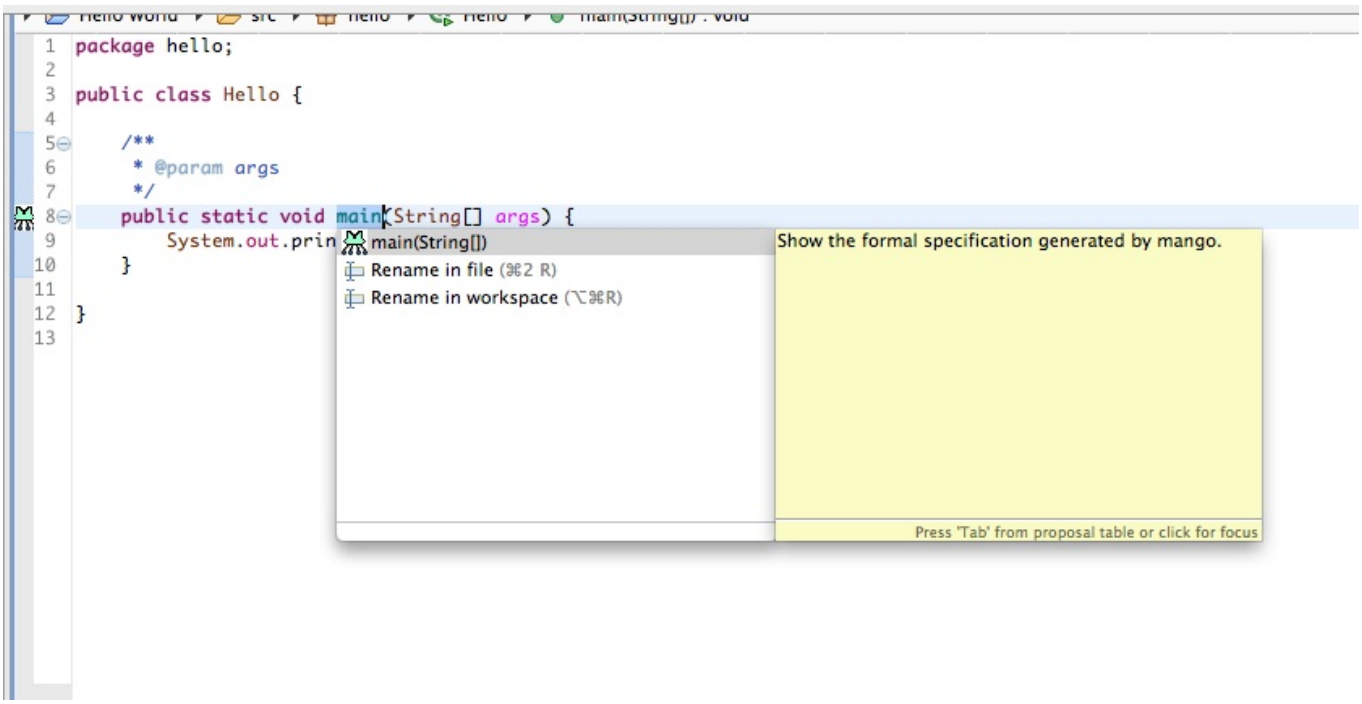
- "Hello World" continued:

  1. When the "populate" command completes, the Mango button will turn green, and both MangoHome and MangoSystem will appear as projects in the workspace (see figure). These projects are required for the proper functioning of Mango. You can learn more about them in the examples and Users Manual sections.



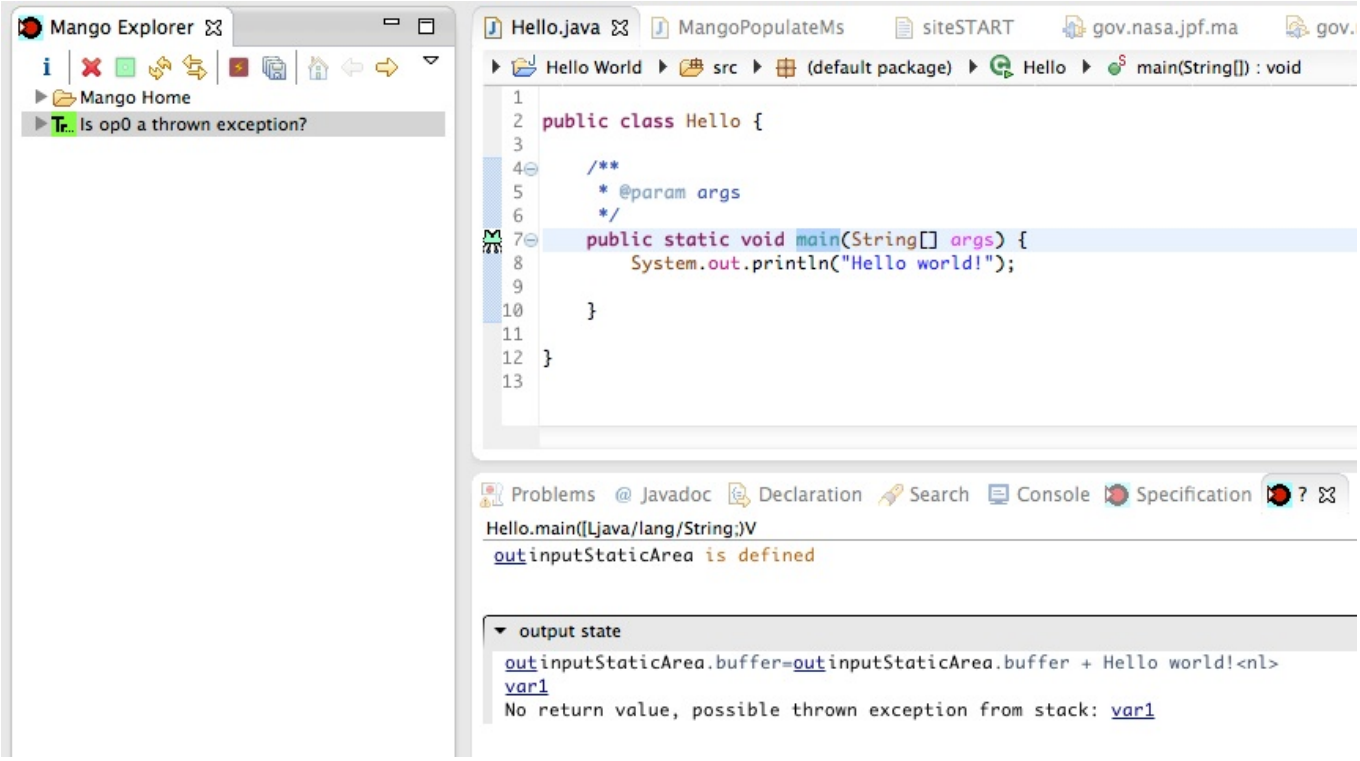
- "Hello World" continued:

1. Double-click on "main" and then right-click, selecting Mango>run. While Mango is running, the Mango button will appear red. It will turn green when Mango is done. A green "method icon" will appear in the editor view gutter next to "main". (Because of a bug, this icon may appear as a folder, but it will work just the same.) Click onthis icon to reveal the specification task. (see figure).

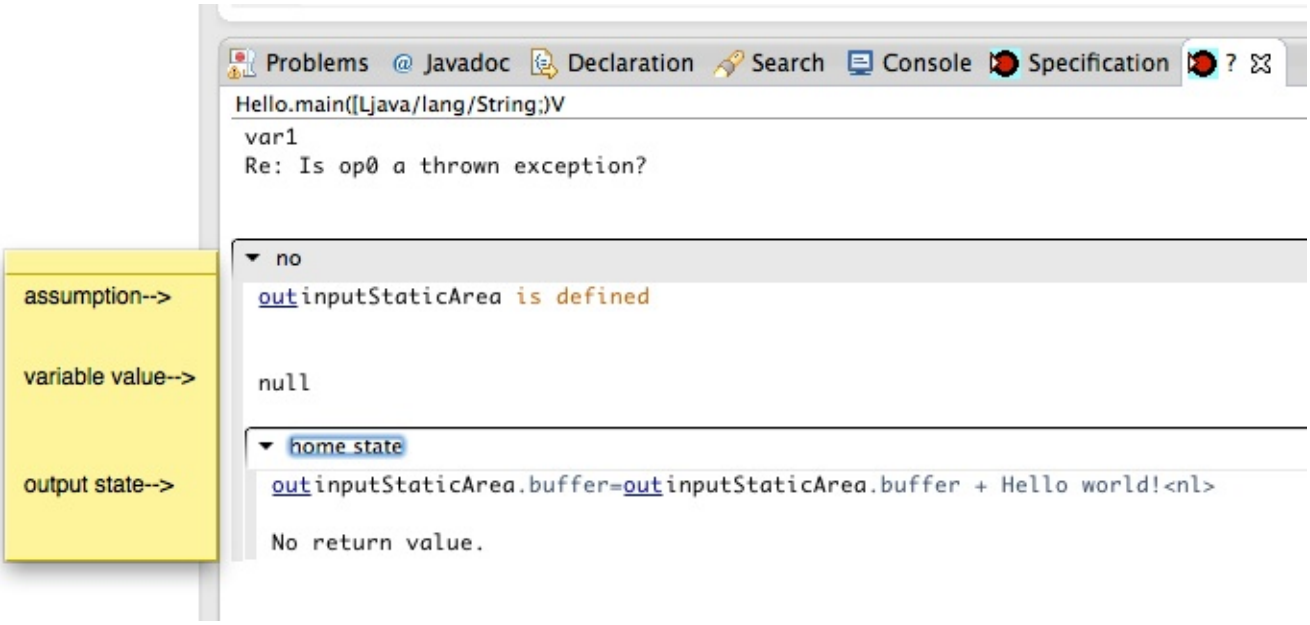


- "Hello World" continued:

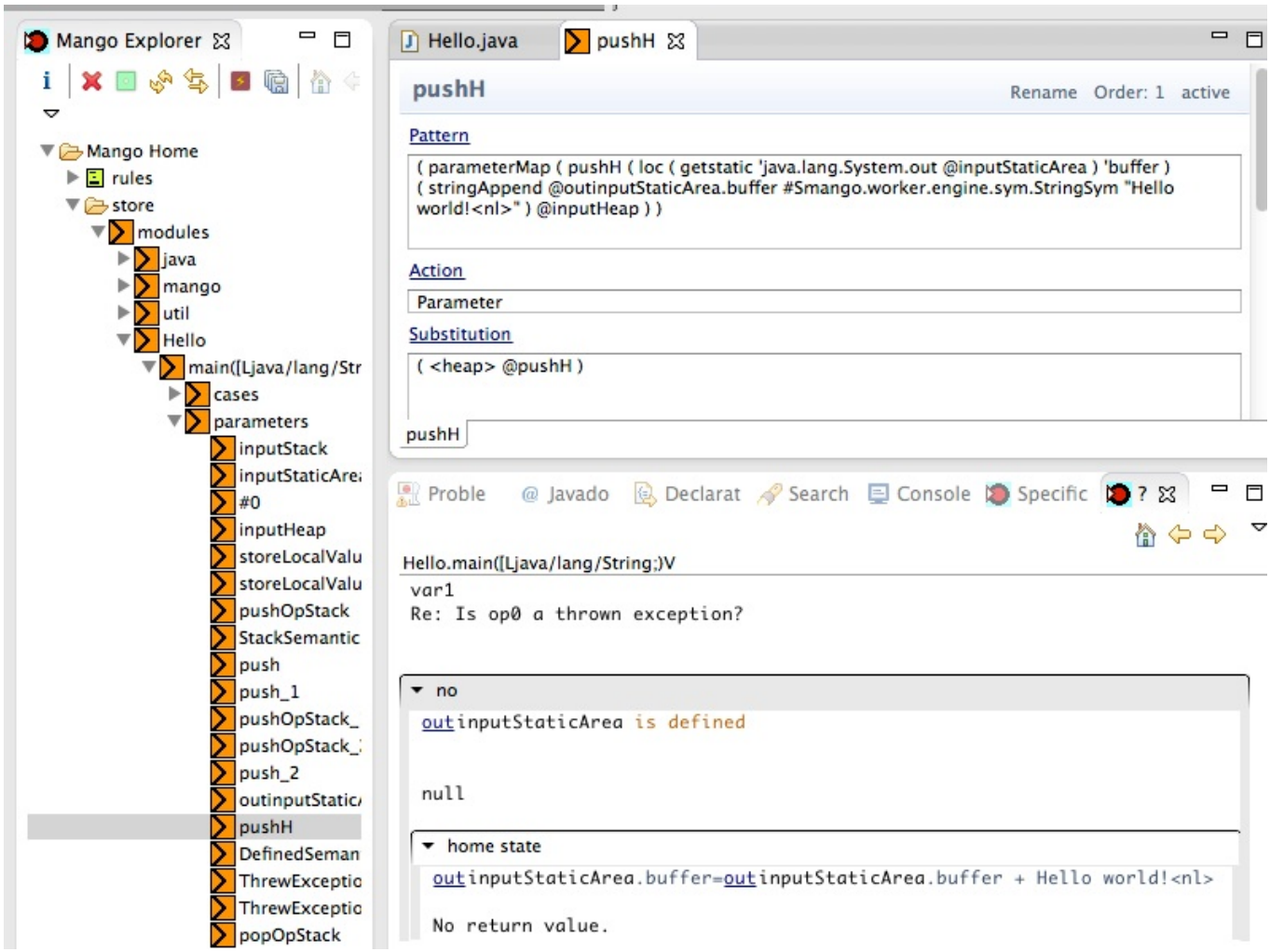
1. Select the "main(String[])" task, and hit carriage return or double-click. The method translation icon, a green "Tr" icon, will now appear in the Mango Explorer view. Finally, double-clicking on the translation icon will reveal the translation for the main method (see figure).



- "Hello World" continued:
  1. Because `println()` potentially throws an exception, there are two possible cases. Click on var1 to resolve this situation. In general, clicking on any link will expose the case values for that link, as well as the corresponding output states. In this particular case, no exception is thrown, so only one case appears. The value of var1, the putative excpetion, is accordingly just `null`.



- "Hello World" continued:
  1. The navigable translation of step 13 is a rendering of a precise specification stored persistently in the MangoHome project. You can view MangoHome content in the MangoExplorer. Try this now, navigating to the main([Ljava/lang/String;)V method folder. View the formal description of the output heap by double-clicking on the "pushH" item in the "parameters" folder. This description of the heap is in so-called Mango Formal Language (MFL). (see figure). You can also double click on the main([Ljava/lang/String;)V folder to recover its translation icon, which is not persistent.



- "Hello World" concluded:
  1. This concludes the "Hello world" exposition. Of course, there is a lot more going on. From here, you can take a look at the examples or the user manual. If you would like to become a Mango developer, please also take at the bugs.