Factorial Calculator and Debug Mode

- Slides: http://www.cs.unc.edu/~dewan/comp114/current/Lectures/Eclipse.pdf
- Import the JavaTeaching project, loading the ObjectEditor and removing invalid projects
  - http://cs.unc.edu/~dewan/comp401/current/Lectures/egit-import.pdf.

Factorial Calculator and Debug Mode

- You will need “Factorials”, “FactorialSpreadSheetDriver”, “ALoopingFactorialSpreadsheet”, “ARecursiveFactorialSpreadsheet” when using debug mode. These classes are in package “lectures.java_basics_overview”.
- Work in pairs
- Follow the quiz questions, modify the code accordingly, and test what happened.
- Test the performances of the two different implementations (recursion and loop).
- Go over slides 74-94 to get familiar with debug mode.

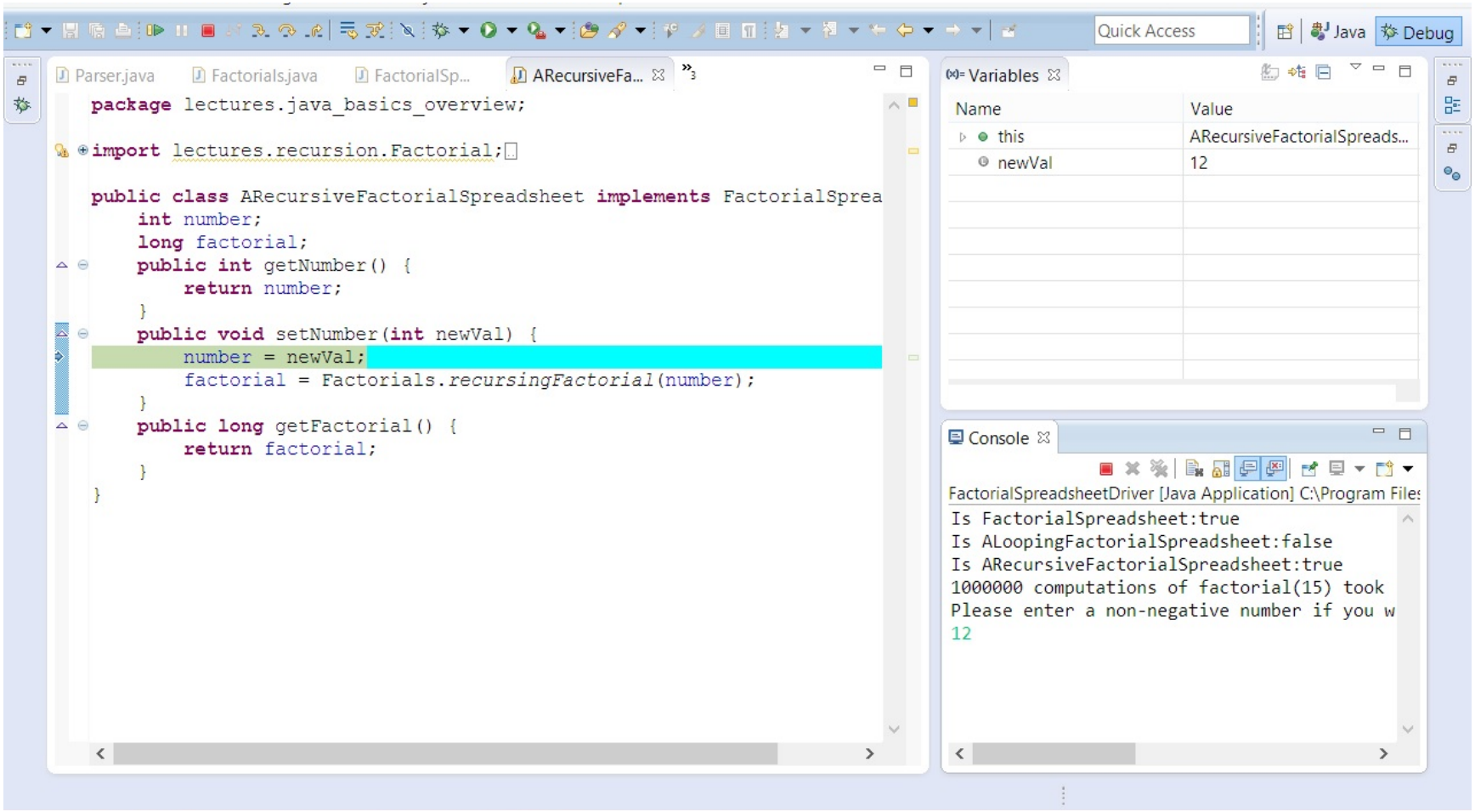
Commands in Debug Mode

- Step into (F5): A method is about to be invoked, and you want to debug into the code of that method, so the next step is to go into that method and continue debugging step-by-step.
- Step over (F6): A method is about to be invoked, but you're not interested in debugging this particular invocation, so you want the debugger to execute that method completely as one entire step.
- Step return(F7): You're done debugging this method step-by-step, and you just want the debugger to run the entire method until it returns as one entire step.
- *Cited from: http://stackoverflow.com/questions/3580715/eclipse-debug-step-into-step-over*

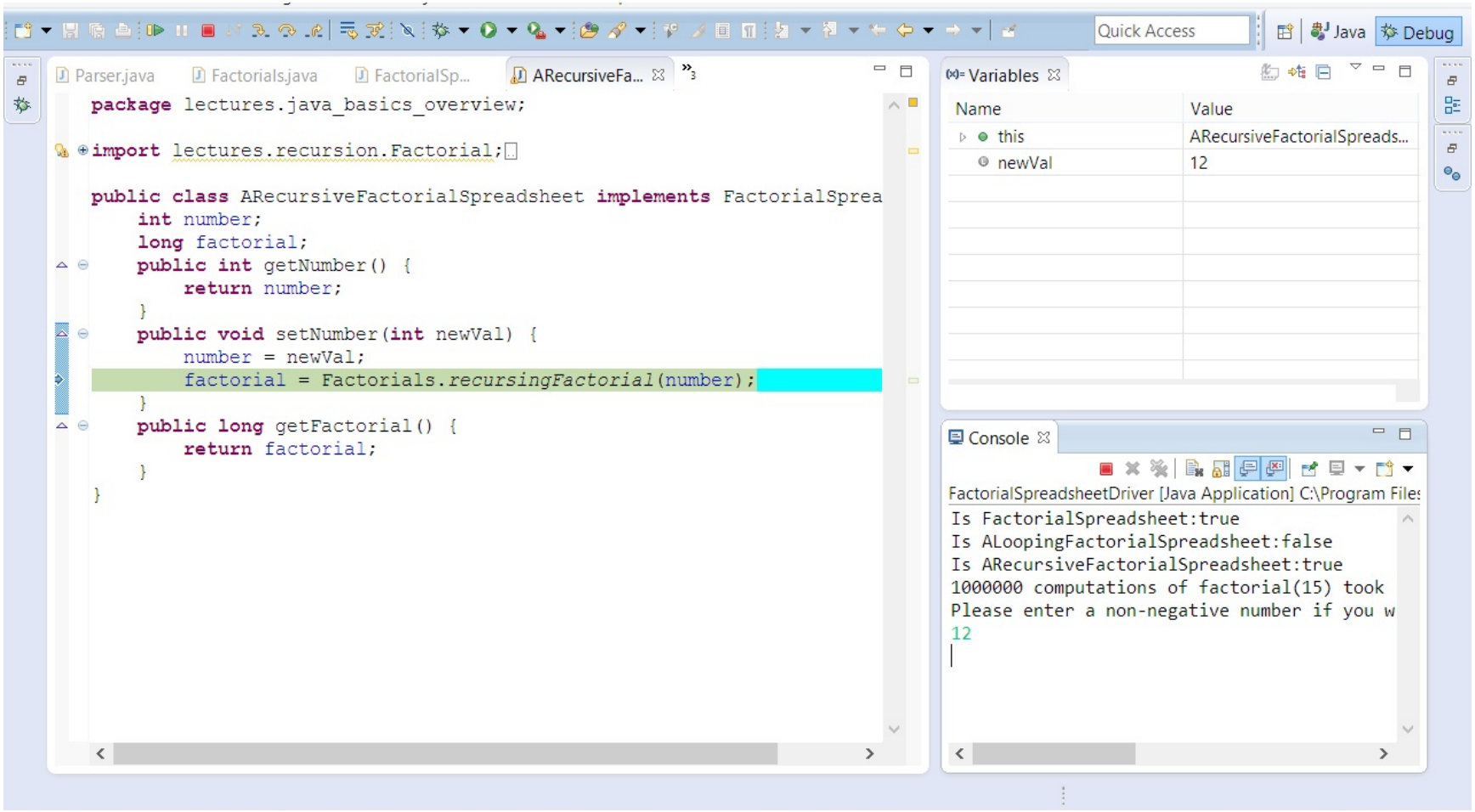
Notes

- Prof. Dewan changed the code a little bit. So when you run the program in debug mode, the process will not be identically the same as the process showed in the slides. The following slides demonstrate the real process when you run the code.
- Modification: In slides, the calculation of factorial is carried out when the method “getFactorial()” is invoked. In the code, the calculation is carried when “setNumber()” is invoked. Both of the two methods are used in classes “ARecursiveFactorialSpreadsheet” and “ALoopingFactorialSpreadsheet”

Step into setNumber()



Step into recursingFactorial()



Step into recursive call again