COM S 342

Recitation 02/24/2021

Topic

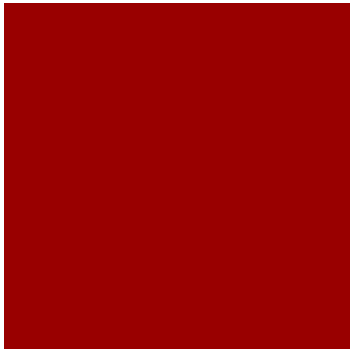
- Debugging in the IntelliJ

- Q&A

Overview

- Debugging is the routine process of locating and removing bugs, errors or abnormalities from programs.
- General Debugging Procedure:
  1. Define where the program needs to be stopped
  2. Run your program in debug mode
  3. After the program has been suspended, use the debugger to get the information about the state of the program and how it changes during running
  4. Fix changes without terminating the session

Breakpoint



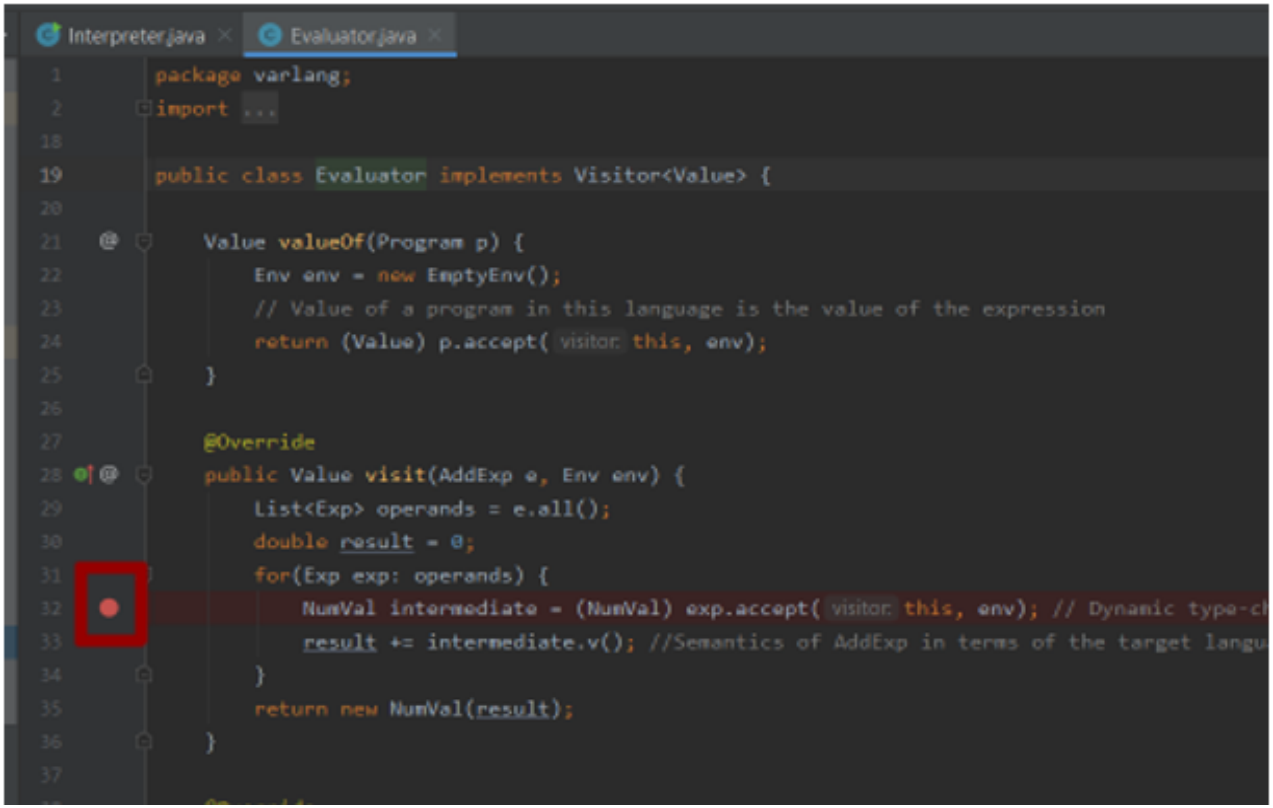
- A *breakpoint* in the source code specifies where the execution of the program should stop during debugging. Once the program is stopped you can investigate variables, change their content, etc.

Cont.

- Set Break Point

  Click beside the line number, or

  Ctrl+F8

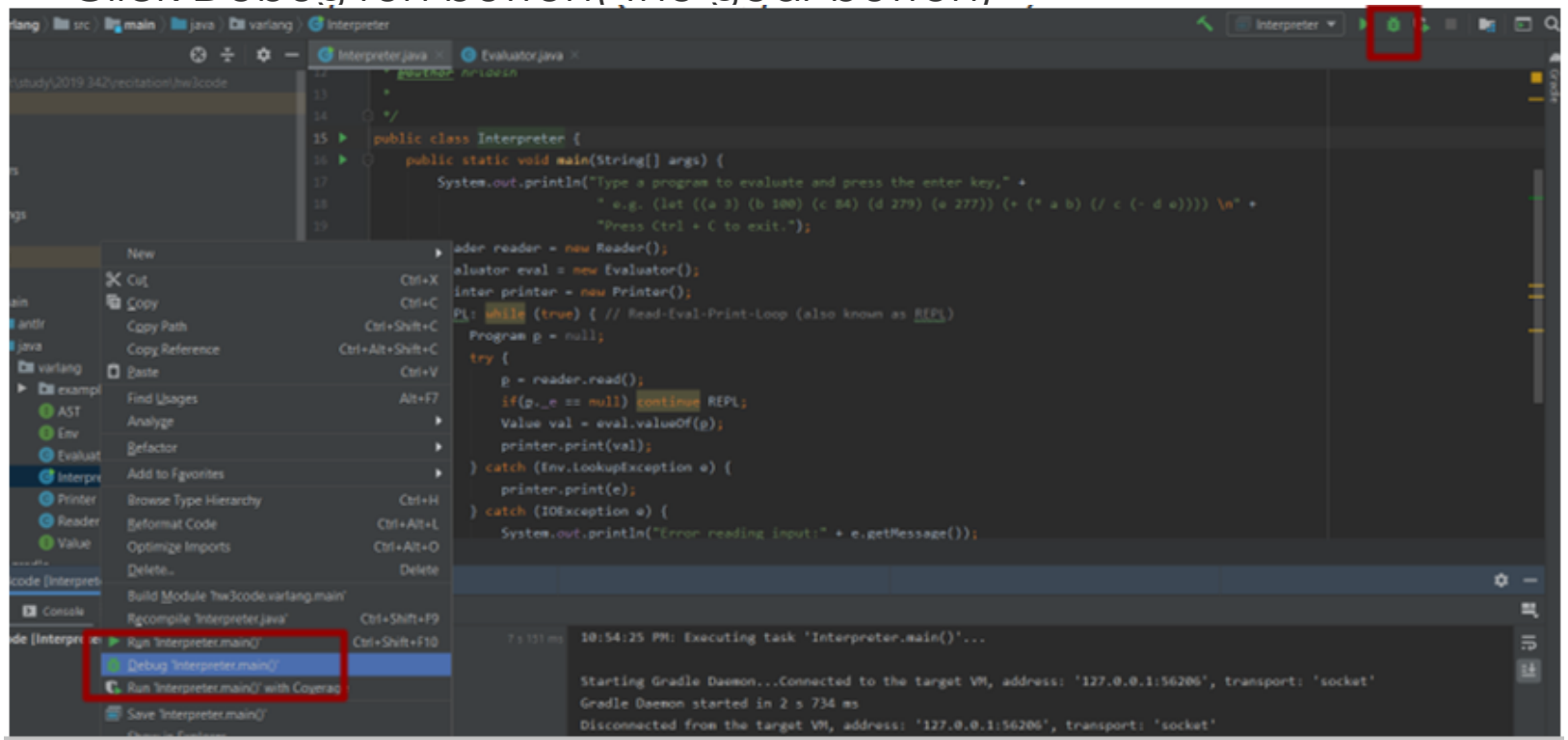


Starting the Debugger

- Run in Debug mode

  Right click Interpreter.java, select “Debug Interpreter.main()”, or

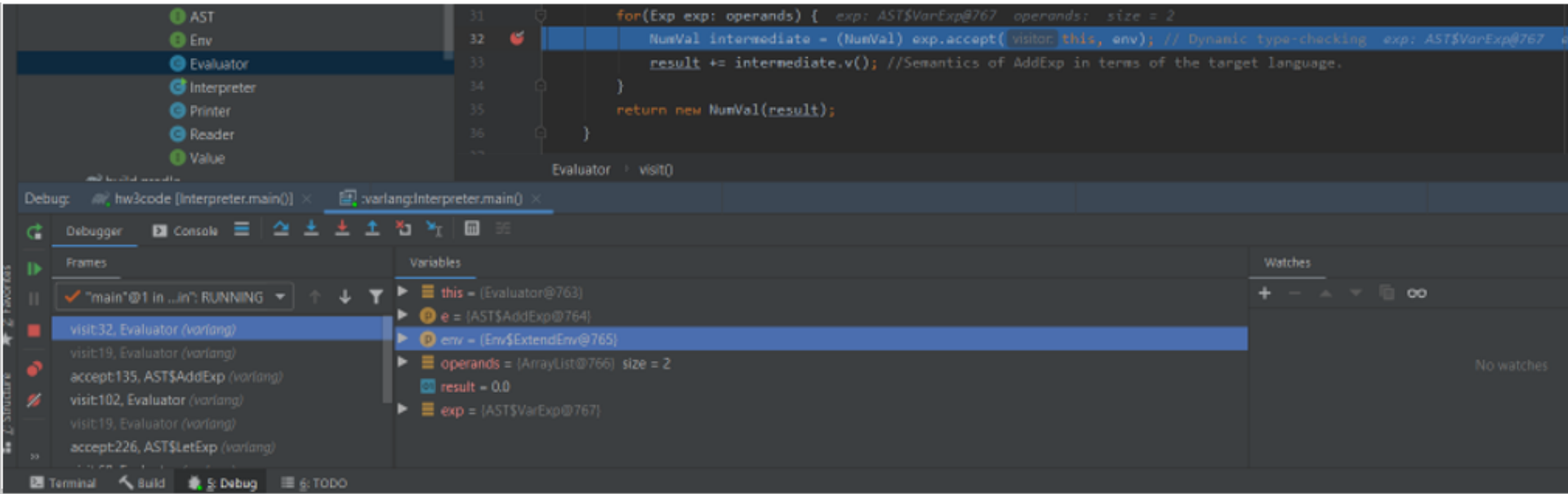
  Click Debug run button(the gear button)



Cont.

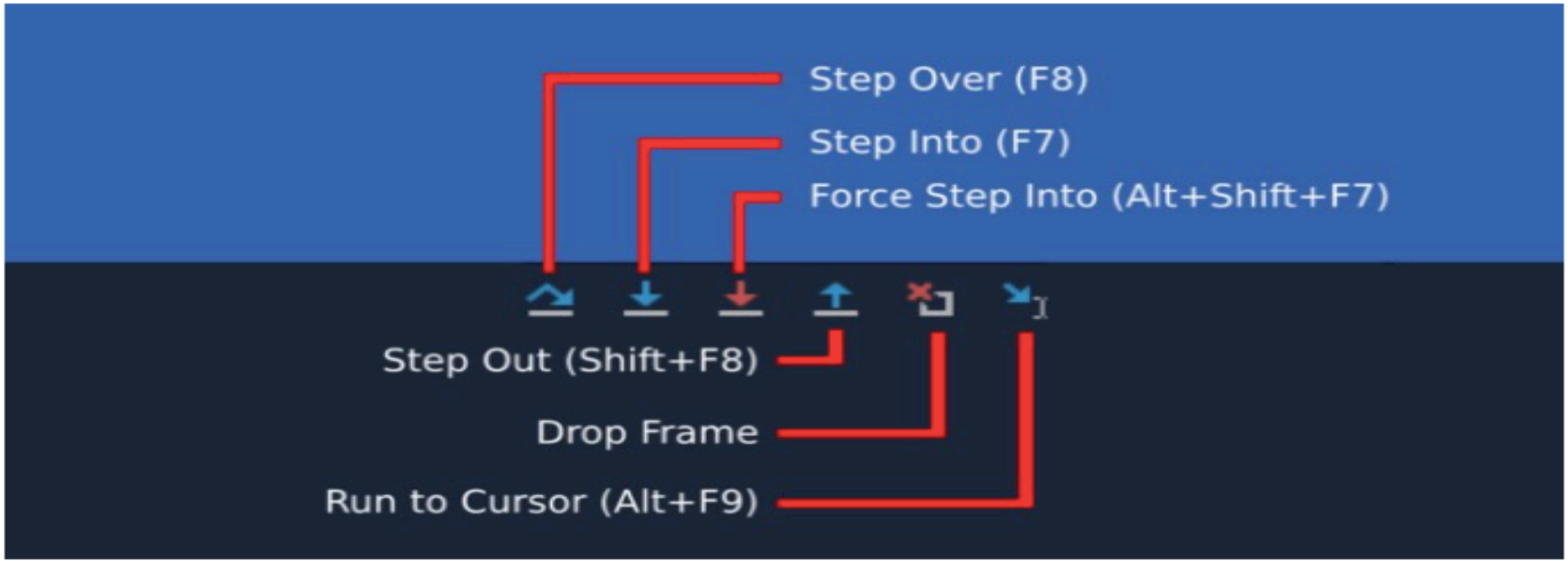
- Debug mode

  The program will pause when meet break points, debugger view will shown in the bottom

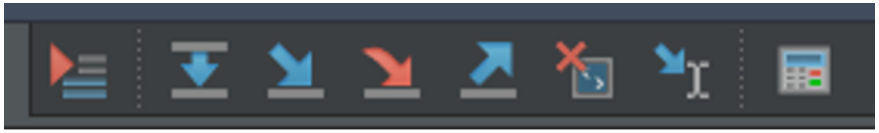


Debug Window

- After the debugger session has started, the Debug tool window will appear, and the program will run normally until one of the following happens:
    1. A breakpoint is hit
    2. You manually pause the program



Cont.



Show Execution Point: Jump to current line

Step Over: Go next line, do NOT go into method

Step Into: If the current line invokes a method, go into it; otherwise go next line

Force Step Into: Similar to Step Into, but can go into built-in library (e.g. println())

Step Out: Back to the line where invoked this method; this method has been DONE, but result is not written

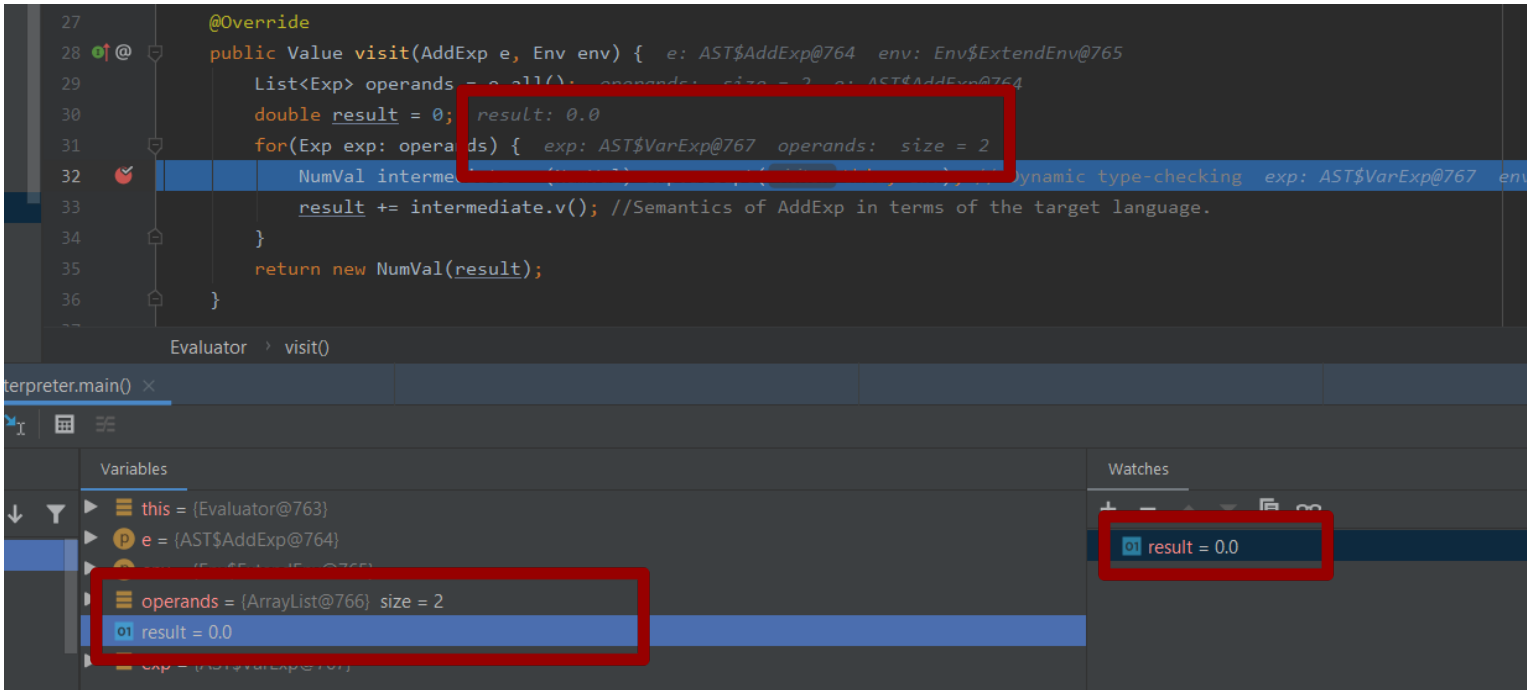
Drop Frame: Back to last method

Run to Cursor: Run the program until the line you select without using break point

Evaluate Expression: Calculate an expression (We will talk about it later)

Variables Checking

1. Current value will be shown after the line
2. Place your mouse on the variable
3. Find it in the Variables
4. Find it in Watches

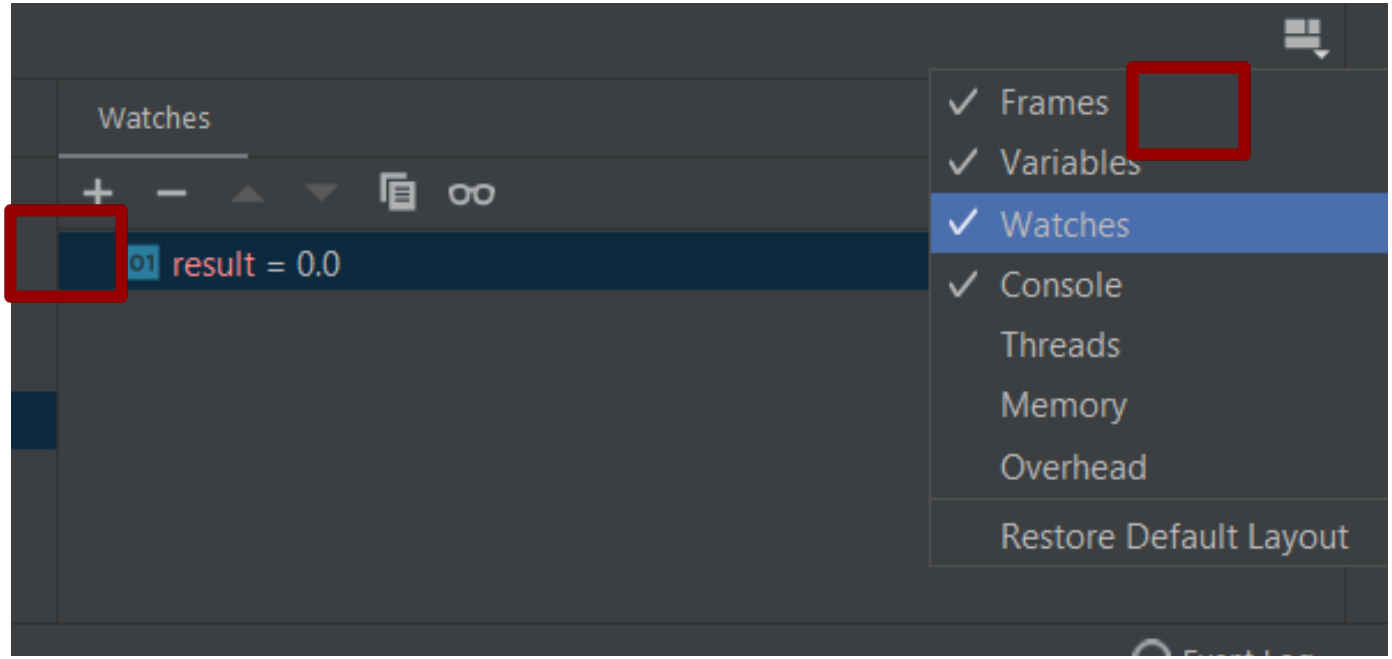


Watchpoint

- Is a breakpoint set on field

✓ Click the top-right button and select Watches if you cannot find Watches

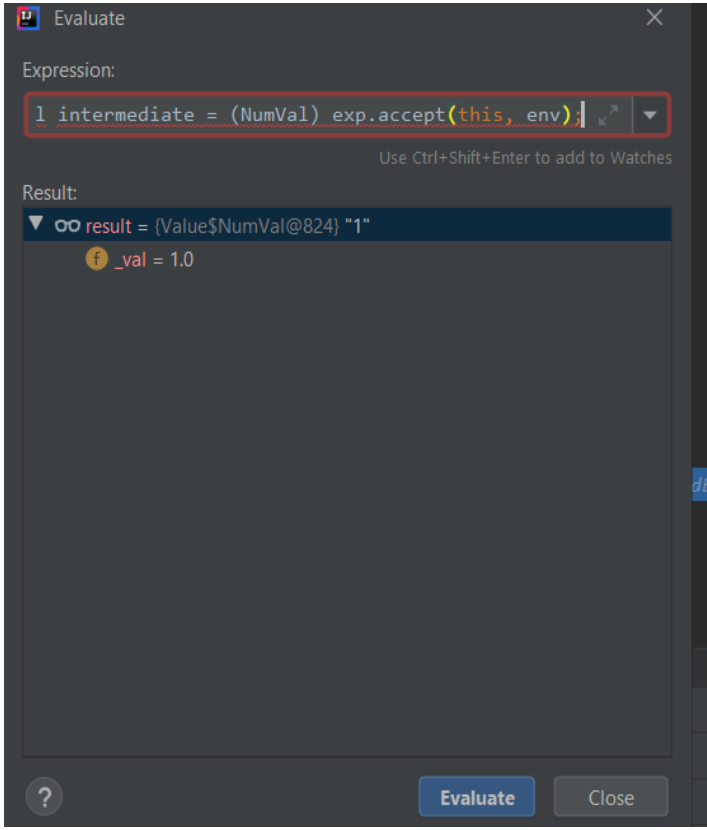
✓ Drag from Variables, or click the Plus button to add variables to Watches



Evaluate Expression

- Expressions in **Java** are used to fetch, compute, and store values.
  1. Select the expression and
  2. Right click, then click Evaluate expression

  You can get the result of one method if more than one method is invoked



Q&A