

### COM S 342

Recitation 09/23/2019 - 09/25/2019

### Topic

ODebugging in the IntelliJ

OQ&A

OSet Break Point Click beside the line number, or Ctrl+F8

```
Interpreter.java
                 © Evaluator.java
         package varlang;
         public class Evaluator implements Visitor (Value) {
             Value valueOf(Program p) {
                 Env env = new EmptyEnv();
                 return (Value) p.accept( visitor this, env);
             @Override
28 0 @
             public Value visit(AddExp e, Env env) {
                 for(Exp exp: operands) {
                      result += intermediate.v(); //Semantics of AddExp in terms of the target langu
```

ORun in Debug model

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

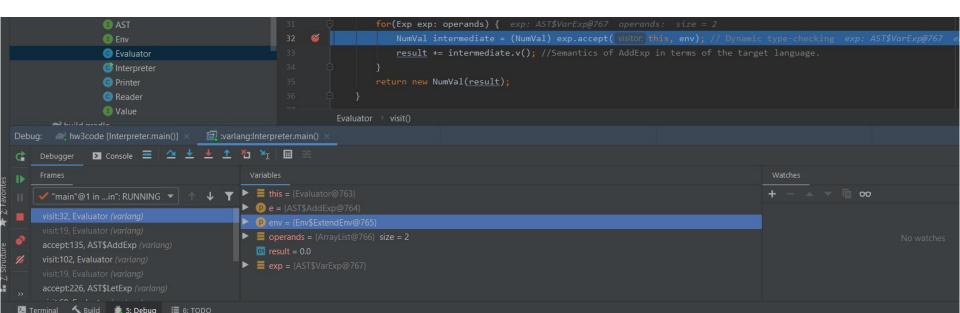
Click Debug run button( the gear button)

```
🕀 😤 💠 — 🌀 Interpreter.java 🗵 🌀 Evaluator.java
                                                   gauthor nridesh
                                              public class Interpreter {
                                                   public static void main(String[] args) {
           % Cut
                                                          inter printer = new Printer();
            <u>С</u>ору
                                                          PL: while (true) { // Read-Eval-Print-Loop (also known as REPL)
           <u>Paste</u>
                                                                if(p. e == null) continue REPL;
  ◎ Interpre
                                                            } catch (IOException e) {
  Reader
> Console
                                                                 7 s 151 ms 10:54:25 PM: Executing task 'Interpreter.main()'...
de [Interpre e | Run 'Interpreter.main()'
                                                                          Starting Gradle Daemon...Connected to the target VM, address: '127.0.0.1:56206', transport: 'socket'
          Run 'Interpreter.main()' with Coverage
                                                                          Gradle Daemon started in 2 s 734 ms
           Save 'Interpreter.main()'
```

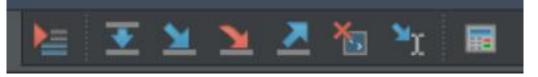


### O Debug model

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



### O Buttones



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

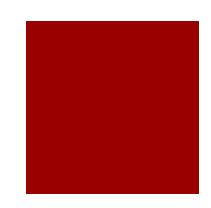
Force Step Into: Similar to Step Into, but can go into official library (e.g. println())

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)



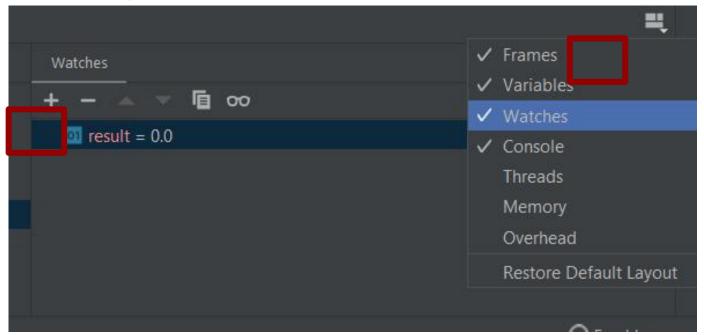
- O Variables Checking
  - 1) Current value will shoen after the line 2)Place your mouse on the variable

3) Find it in the Variables 4) Find it in Watches

### O Watches

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

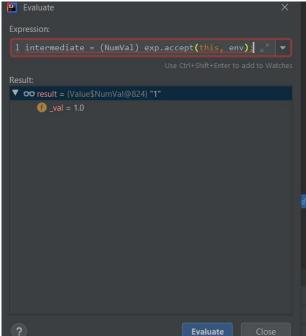
Drag from Variables, or click the Plus button to add variables to Watches



### O Evaluate Expression

Select the expression and click Evaluate to get the result

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



Q&A

