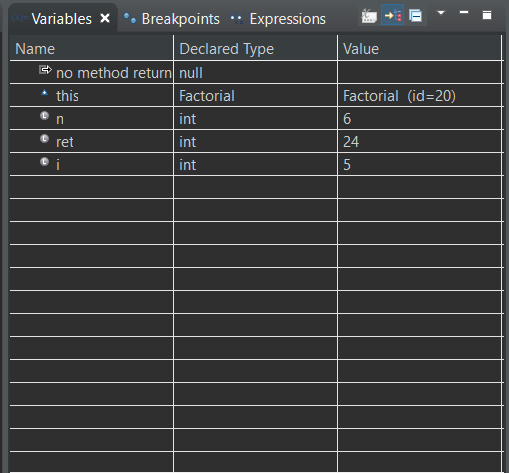
**1.8**

****

**1.9**

|  |
| --- |
| *Question: What is the Eclipse keyboard shortcut or toggling a breakpoint?* |
| CTRL + SHIFT + B |
| *Question: What is the difference between “Step-Over”, “Step-Into”, and “Step-Return”?* |
| **Step-Over:** An action that steps over a given line. If the line contains a method call, it will be executed and the result will be executed and the result is returned without debugging each line of that method.  **Step-Into:** An action which behaves the same as “step over” if the line does not contain a method call. However, if it does, the debugger will enter the called method and continue the debugging process inside of that method, going through it line-by-line.  **Step-Return:** An action that allows the user to return to the line where the current method was called. Also named “Step-Out” due to this. |
| *Task: Practice tracing through the program.*  *Based on your understanding of the program, provide a list of methods that are called when the program executes (from start to end, in order of being called). You can skip library methods like println, for example).* |
| DebugStar.main(String[]) line: 14  DebugStar.run(String, int, int) line: 32  DebugStar.getOperation(String) line: 22  Add.perform(int, int) line: 54  DebugStar.main(String[]) line: 15  DebugStar.run(String, int, int)line: 32  DebugStar.getOperatin(String) line: 22  Subtract.perform(int, int) line: 6  DebugStar.main(String[]) line: 16  DebugStar.run(final int) line: 37  Factorial.perform(final int) line: 65 |

**2.1**



**2.2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Line Number** | **Type of error (compile-time, run-time, or logical)** | **Description** | **Correction** |
| 26 | compile-time | Syntax error and n is not initialized | n = 0; |
| 27 | logical | Input validation of a while loop should have the false statement (not the true statement) as its condition | (n <= 2) |
| 40 | logical | Needs to get the previous 2 numbers of i, not just the most previous one. | Replace the first *f.get(i - 1)* with *f.get(i - 2)* |
| 47 | logical | Needs to include 0 in printList. Thus, will need to start from index 0, not 1. | int i = 1; |
| 49 | run-time | ArrayIndexOutOfBoundsException. The array has been accessed with an illegal index (in this case, the index is greater than the size of the array) | System.out.print(fiboList.get(i - 1)); |
| 49 | logical | Need to print out “, ” after each integer in the array | System.out.print(fiboList.get(i - 1) + ", "); |

**2.3**

