1. Getting Familiar with Debugging

Screencap 1

SHOW VARIABLES WINDOW:

Graphical user interface, application

Description automatically generated

9. Time to test your understanding! J

TABLE ONE

Question: What is the Eclipse keyboard shortcut for toggling a breakpoint?

Answer: Ctrl-Shift-B

Question: What is the difference between “Step-Over”, and “Step-Into”, and “Step-Return”?

Answer: **Step Into:** will move to the first line of the called function, and proceeds until it returns control back to the calling function. If you **“Step Over”** a method call then the method is run and the debugger continues on after the method call while calling the function implicitly. **Step return:** to return from a method which has been stepped into. Even though we return from the method, the remainder of the code inside the method will be executed normally.

Task: Practice tracing through the DebugStar sample program.

It is ok if you don’t understand all of the java code; but you should be able to trace the order in which statements

are executed.

Based on your best 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).

HINT: Use a combination of “Step-Into” “Step-Over” and “Step-Return”. Use the “Stack Trace” window.

*List of Methods (in order of call) below. Please use the fully qualified name, eg.*

*“DebugStar.run(String, int, int). Use the stack view to help you.*

DebugStar.main(String[])

*DebugStar.run(int)*

DebugStar.getOperation(String)

Add.perform(int, int)

DebugStar.run(String, int, int)

DebugStar.getOperation(String)

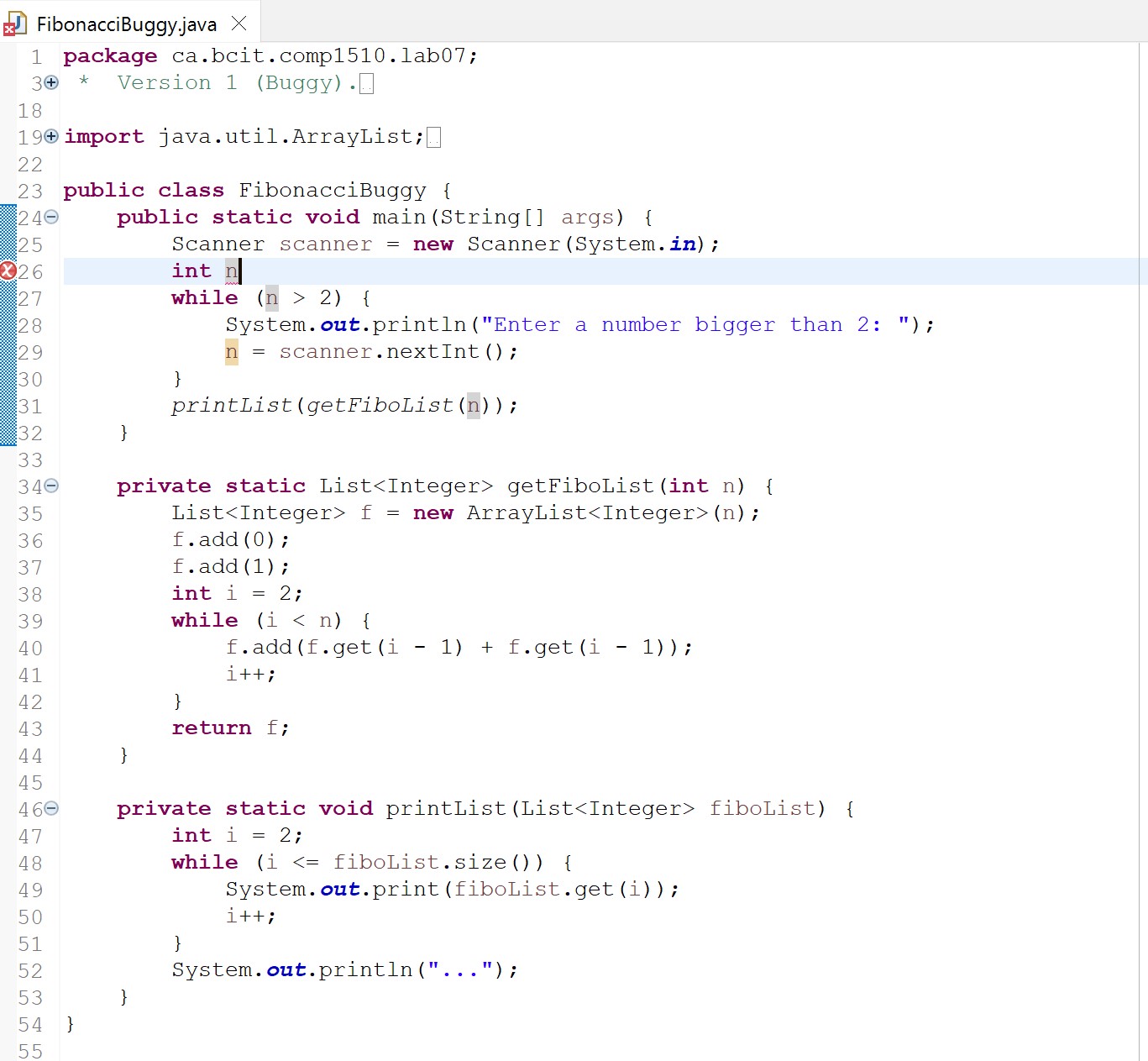
Subtract.perform(int, int)

DebugStar.run(String, int, int)

Factorial.perform(int)

1. The Debug Challenge

SCREEN CAPTURE: Original Code with line numbers



|  |  |  |  |
| --- | --- | --- | --- |
| Line Number | Type of error (compile-time,  run-time, or  logical) | Description | Correction |
| 26 | Syntax Error | Missing the “;” to end the variable declaration statement | Add “;” at the end of the line |
| 27 | Logical Error | While loop is meant to used when n < 2, not when n > 2 | Replace “>” with “<” |
| 40 | Logical Error | Each number in the sequence is twice of the number before it instead of being equal to the sum pf 2 numbers before it | f.add(f.get(i - 2) + f.get(i - 1));  Array index correction |
| 47 | Logical Error | Not printing out the first index of the list (0) | int i = 1;  also adding “, “ to enhance clarity |
| 49 | Run-time error | Array Index out of bound exception | Only print upto index of fiboList length, which is (fiboList.size() -1)  Change fiboList.get(i) to fiboList.get(i – 1) |

ScreenCapture of the fixed code and console output:

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated