|  |  |
| --- | --- |
|  | |
| Code Debugging | |
| **Updated** | **10.28.2018 2:14 PM** |

A short report on experimenting with the Debugger:

1. Insert a breakpoint at the line after where you ask the user whether a triangle or rectangle is required. How can you then step along the program line by line?

I have inserted a breakpoint at the line after where I ask the user whether a triangle or rectangle is required. When I press the Debug button, the program starts running and asks for Rectangle or Triangle as normal, but then reaches the breakpoint. The Area of Shape program is then suspended at the breakpoint I have set and Eclipse asks me to confirm, on the first occasion, that I want to switch to the debug perspective. I can then tell Eclipse to remember my decision and confirm the switch which opens the debug perspective where the current line of the program is highlighted for me. This allows me to inspect the current values of the variables and check that the values are what I expected after the breakpoint I set - which in this case is that variable shapeStr is either a 1 or 2. I can then use the Debug Toolbar 'step over' button to step along my program line by line. I could also use ‘step into’ button if I wanted to go into the methods

1. What happens with the flow of control when you reach the “if” statement?

When the program reaches the “if” statement the program will execute the next section of code only if the shape variable test evaluates to TRUE - in this case that shape = 1 which means the user selected a Rectangle as the shape to calculate the area for. If this test evaluates to FALSE (meaning that the shape is not equal to 1) the control jumps to the end of the next “else if” statement which tests that the variable shape = 2 and the user has selected a Triangle. If both of these test evaluate to FALSE the control jumps to the final “else” statement which causes the program to fail gracefully

1. Observe the values of variables in the Variables window. Are all variables always shown? When do they change?

As we step through the program each variable is shown as soon as the debugger gets to the line where the variable is assigned a value and then that variable remains visible to the developer as the program continues to execute. Variable can be changed when they are assigned new values.

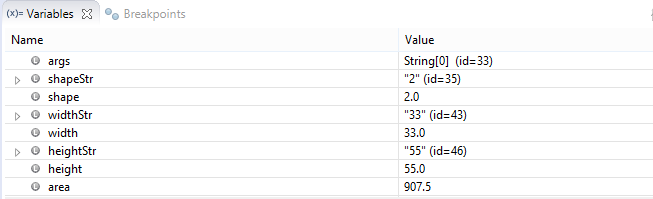


Fig 1. Debugger Variables

1. Add two bugs to your program (one in the logic of the if statement and another in one of the calculations) and describe how you might track them down with the Debugger.

I created an error in the first if statement by changing “if (shape == 1)” to “if (shape != 1)” and created a second bug in the calculations by changing the rectangle area calculation from “area = height \* width;” to “area == height \* height;”.

Both of these bugs cannot be caught by the IDE as they are syntactically correct but logically flawed so the .java file will compile without error. When I run the program and enter 2 for Triangle at the first screen everything runs without error.

However when I attempt to enter 1 for Rectangle the program accepts width and height inputs but fails gracefully at the calculation step.

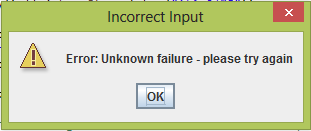


Fig2. Error Message

In order to find the error I must set a breakpoint at the start of the “if” statements and run the program in the debugger and identify the point of failure. We can then see that the variables being passed into the “if” statement are correct (Fig 3)

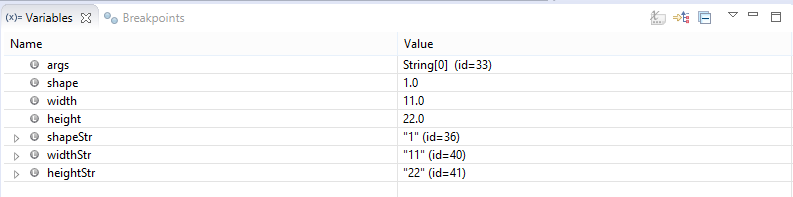


Fig 3. Debug Variables

And we can see that the Rectangle calculation is not being run because the “if” is incorrect and we immediately step to the next “if else”

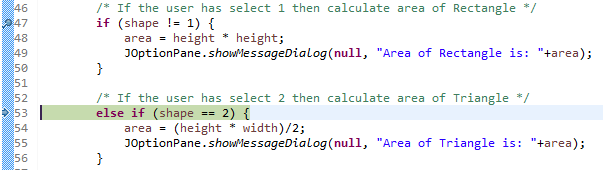


Fig 4. Step through the lines of code

So we can fix this bug and re-run the program. Now we are affected by the second bug which was hidden by the first bug we encountered. When we re-run the program it runs to completion as expected but our test pass shows that the result is incorrect because the answer is 484 when we expected it to 242. (Fig 5)

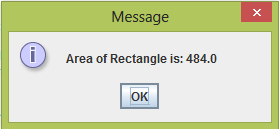
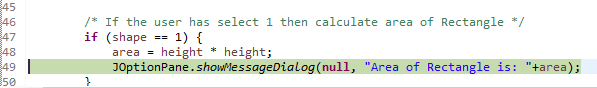
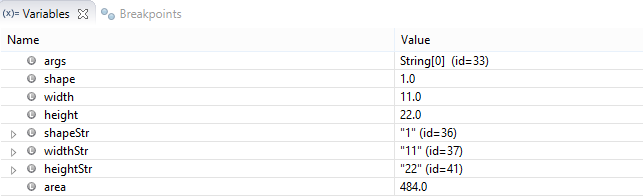


Fig 4. Incorrect Area Result

We can see from the variable output that all variable are as expected up to the calculation but the result for area is incorrect and that the error resides in the calculation which has two height variables.





We have now found both bugs in the program.

1. How many times do you have to run the program to be satisfied that no bugs remain? Explain.

The absolute minimum number of test passes is two – once for Triangle and once for Rectangle but based on my test matrix included in my code assignment document I would re-run the program a minimum of 19 times to test for both expected negative and positive test results.