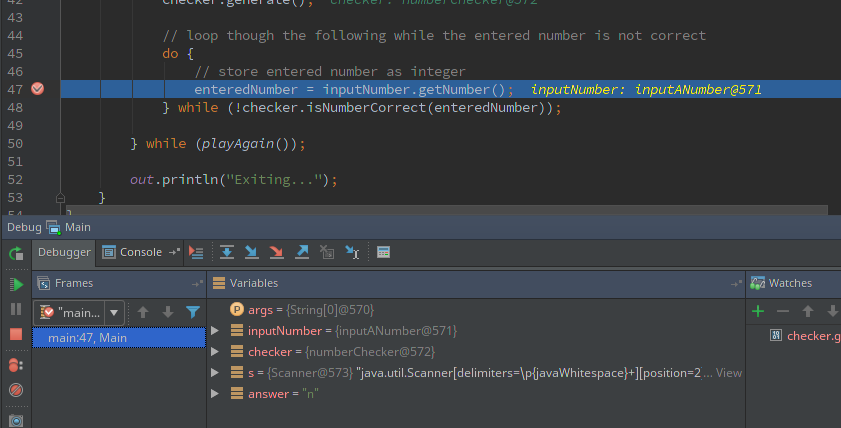
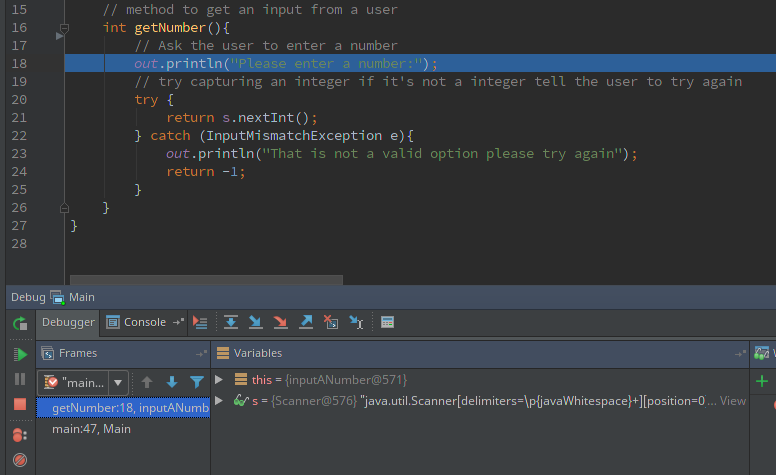
This is my test plan document within this I will demonstrate how I have tested my program and how I fixed a bug that I found

|  |  |  |  |
| --- | --- | --- | --- |
| Test step | Expected result | Actual result | Evidence |
| Open the program | The login page is shown | As expected |  |
| Click help | The help screen is shown | As expected |  |
| Enter invalid credentials | The user is told that the credentials are incorrect | As expected |  |
| Enter correct credentials | The user is taken to the search page | As expected |  |
| Click search cars | The user is taken to the search cars page and some cars are shown in the search results | As expected |  |
| Enter some search text and press search | The results are updated with different cars | As expected |  |
| Click on a car | The view button is shown | As expected |  |
| Press the back button | User is taken back to the menu | As expected |  |
| Click search customers | User is taken to the search customers page | As expected |  |
| Enter a name and click search | The name results list is updated | As expected |  |

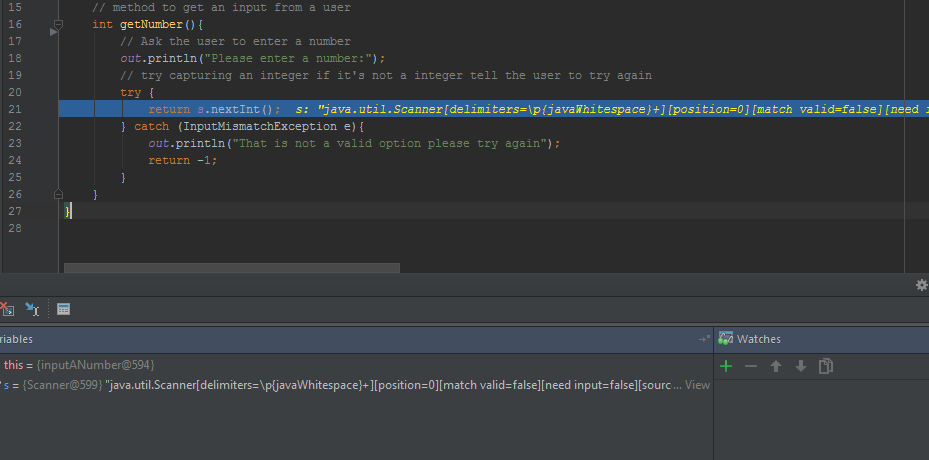
Above I showed how I found a bug in my program; I used the debugging tools within my IDE to find out why this problem was occurring. The screenshot below shows me using a breakpoint where the method is being called.



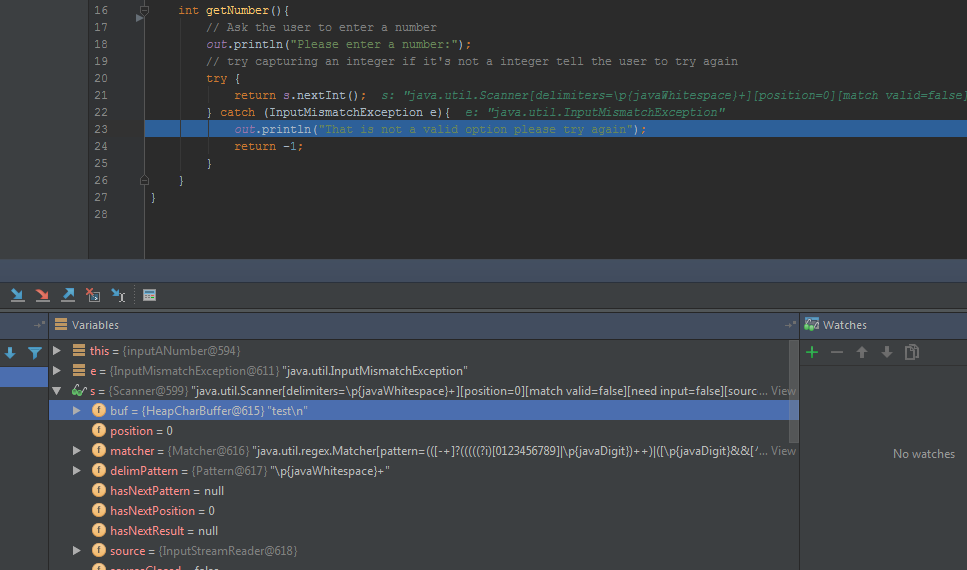
I then click “Step into” to step into the method, this is then shown below.



After stepping into the method I can then step over to the point where it captures input from the user, line 21 s.nextInt();

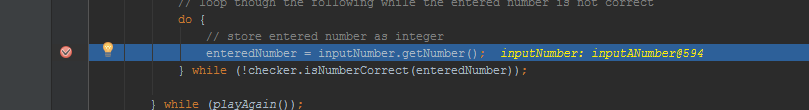


Below I entered a non-number value to see what would happen

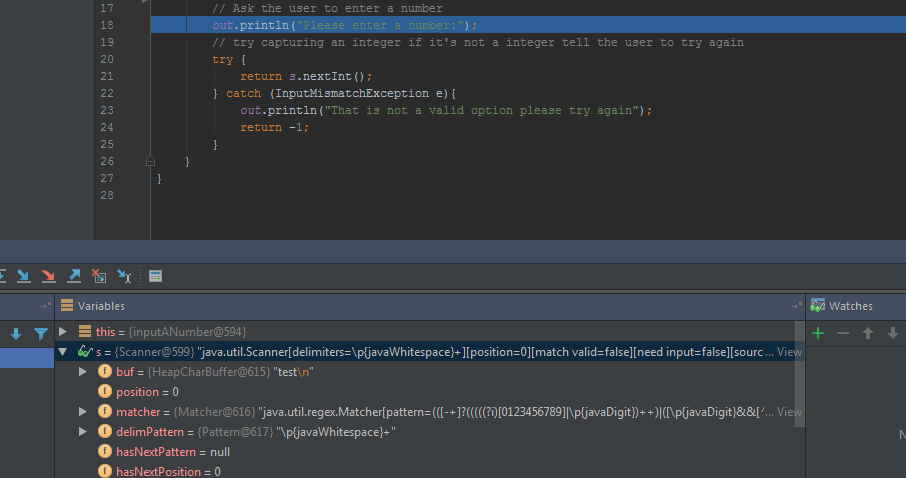


I can see that is has caught the exception and moved into the catch statement where it is about to output that that is not a valid option. Also note that you can see my entered String of “test” in the buf attribute of the s (Scanner) object in the debug console.

I will now press the continue button so the code continues up to the next breakpoint.



After pressing continue the execution has ran back to the original break point as it is surrounded by a do while, I can then step into the method again



After stepping into the method again I found that the original text that I entered was still on the buffer. This is because the scanner object only gets initialised when the constructor is called, and that constructor will only get called when an integer has been entered and not a string like I entered above. To rectify this I moved the initialisation of the Scanner object so it was recreated every time the method was called. An inline difference indicator is shown below:

