Debugging

Help with finding bugs in your code

Produced
by:

Dr. Siobhán Drohan

Mairead Meagher

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

Topic List

- What are bugs?

- What are debuggers?

- How do I use them?

What are bugs?

A software **bug** is an error, flaw, failure or fault in a **computer** program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

Software bug - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Software_**bug**

Bugs can be frustrating to find/fix

Topic List

- What are bugs?

- What are debuggers?

- How do I use them?

Help is at hand...debuggers!

A debugger can be used to fix bugs

...hence the name debugger!

Debugger

- A debugger is a software tool that helps in examining how an application executes.

- It lets programmers execute an application one statement at a time.

- It typically provides functions to stop and start a program at selected points in the source code, and to examine the values of variables.

Debugger

- Debuggers are especially useful when your program contains logical errors.
  - i.e. errors that the compiler will not pickup but that lead to incorrect results e.g. if your syntax is correct but the logic of your problem solution is faulty.

- Using the debugger, you can trace how each of the calculations and changes made to fields/variables happen and hopefully figure where the error is occurring.

Topic List

- What are bugs?

- What are debuggers?

- How do I use them?

Debugger

- Most IDEs come with a debugger; Eclipse has one.

- We are going to use the Eclipse Debugger to step through the debugging of a small application that does the following processing:

  - The program iterates over a primitive array of int and prints out the largest number in the array.

The code…

```
public class Largest {

    public static int findLargest (int[] list) {
        int index = 0;
        int max = Integer.MAX_VALUE;

        for (index = 0; index < list.length; index++) {
            if (list[index] > max) {
                max = list[index];
            }
        }
        return max;
    }

}
```

```
public class Driver {

    public static void main(String args[]){
        int list[] = {2,5,3,4};
        int largestNumber = Largest.findLargest(list);
        System.out.println("Largest number is: " + largestNumber);
    }

}
```

The code...and the bug

```java
public class Largest {

    public static int findLargest (int[] list) {
        int index = 0;
        int max = Integer.MAX_VALUE;

        for (index = 0; index < list.length; index++) {
            if (list[index] > max) {
                max = list[index];
            }
        }
        return max;
    }

}
```

We are expecting this output:
Largest number is: 5

But we get:
Largest number is: 2147483647

```java
public class Driver {

    public static void main(String args[]){
        int list[] = {2,5,3,4};
        int largestNumber = Largest.findLargest(list);
        System.out.println("Largest number is: " + largestNumber);
    }

}
```

Let's debug it
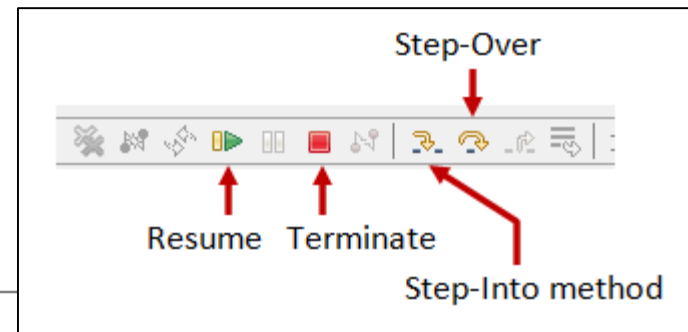
- Demo of the Debugger in Eclipse

Fixing the bug

- Instead of the line of code:

  int max = Integer.MAX_VALUE;

- We need:

  int max = list[0];

Some Debugger buttons…

- **Resume** - resume the execution of the currently suspended debug target.

- **Terminate** - to terminate the launch associated with the selected debug target.

- **Step Into** - step into the next method call at the currently executing line of code.

- **Step Over** - to step over the next method call (without entering it) at the currently executing line of code.

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/