
CSCI 3901 Lab 2: Debugging

Friday, 24.09.2021

Nadipineni Hemanth Kumar [B00899473]

Steps while finding the errors

calendar.java

- On the first run without debugging mode, days were printed from 0. So changed "`for (i = 0; i <= days_to_print; i++)`" to "`for (i = 1; i <= days_to_print; i++)`"

```
Please enter a numeric year from 1800 or later.
2020

      0  1  2  3  4
5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

              0  1
2  3  4  5  6  7  8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31 32

0  1  2  3  4  5  6  7
8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
```

- The next error I noticed was that the program is printing an incorrect number of days for each month. I looked at the month_days array and it was correct.
- Started debugging mode from "`if (year >= 1800) {`" as a **breakpoint** and started stepping over. When it reached "`days_to_print = month_days[month];`", noticed that value "28" was taken for the month "1". So, changed it to "`days_to_print = month_days[month-1];`".

```

    }
    // Figure out how many days to print for the month.
    days_to_print = month_days[ month]; month_days: [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    if (month == 2 = false ) { month: 1
        // Check for a leap year
        if ( (year % 4) == 0 ) {
            if ( ( (year % 400) == 0 ) || ( (year % 100) != 0 )

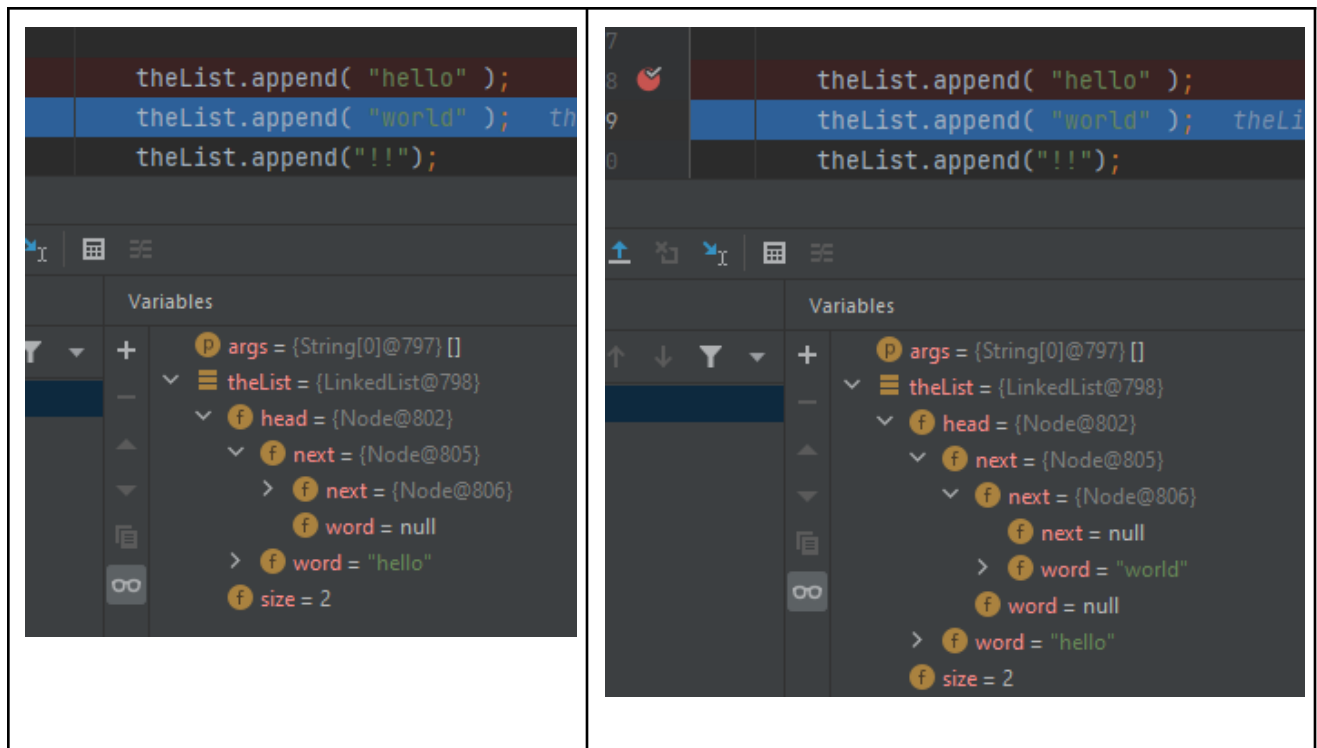
```

Variables

- > static members of calendar
- > args = {String[0]@947} []
- > month_days = {int[12]@948} [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
- > days_to_print = 28
- > userInput = {Scanner@949} "java.util.Scanner[delimiters=\p{javaWhitespace}+][pos=0]"
- > year = 2020
- > month = 1
- > day_of_week = 3
- > i = 3
- > month_days[month] = 28

LinkedList.java

- The first missing item I noticed was the package. Added package "debugger" to all the classes.
- Started running debugger from the breakpoint "`theList.append("hello");`". Stepped into the method and step over through the append method. The head was inserted fine. For the next node, it created again a new node as shown in the picture below and inserted another new node into it. So changed "`head = new Node(word, new Node());`" to "`head = new Node(word);`".



- The next one is that, even after appending an element, it's returning false. So, place `added = true;` after `location.setNext(new Node(word));`.
- The following method has been removed as it's not being used anymore and to reduce redundancy in the code.

```
public Node( String start, Node nextNode ) {
    // Create a node with a given string and connect it into the linked list.
    // It is set to be the head of a linked list, with the given node as the completion of the list.
    next = nextNode;
    word = start;
}
```

The cause of the defects & The approach

1. For calendar.java, one error is due to printing the days from '0' and the other one is due to accessing the array from the second element.
2. For LinkedList.java, the error is due to creating a new extra node while passing the next element.

'The approaches were explained above'

Time Taken:

It has taken me around 2.5 hrs to learn the basics of debugging and debug this code.

I referred to these videos:

<https://www.youtube.com/watch?v=IAWnIP1S6UA>,

<https://www.youtube.com/watch?v=1bCgzjatcr4>

Analysis

- IntelliJ has one of the best debugging methods. With breakpoints, stepping into and stepping over are very advantageous. They reduced the manual debugging using print statements after each step to look for the error.
- These debugging methods showed how data has been changed in each step. Without which it would be very time taking to track all these data changes.
- For most of the conditions, these strategies will work. These strategies might be ineffective if there are lots of methods being called from a lot of classes. Stepping into each method and again into each method will take much time.
- Debugger provided the functionalities of the breakpoint, step into, step over, step out which were very useful in debugging these programs.