Steven Thaw

sjthaw@me.com

Abstract

A Java program which takes in a file of polynomials and determines if they are sorted in strong or weak order.

CMSC 350 Data structures and analysis

Project 1

Lessons Learned

Out of two Bachelor’s of Science degrees, this assignment was by far the hardest assignment yet. The project literally gave me three nightmares! However, with that came the opportunity to learn.

The first lesson learned was implementing Iterable and Comparable. This was the first time I had implemented interfaces and it was challenging. I learned the hard way that if the required method names are not written exactly correct, the class will not compile because it does not have the methods required by the interface.

Implementing my own linked list was also a challenge. This was the first class I decided to tackle, and I spent a lot of time looking at the code examples from Chapter 5, Chapter 6, and the support.llnode package from the book. It took me two days to write this class, but now that it is done, I feel that I have a much better understanding of how linked lists actually work.

The next big challenge for me is the toString() methods. It took me a while to figure out how the toString() method from my linked list would combine with the toString method of my Polynomial class. I ended up doing all of my formatting in the linked list toString(), and then just used the Polynomial class toString() to add them together. I definitely need to work on my String manipulation. I did discover a built in String Builder class, but I just didn’t have enough time to teach it to myself for this project.

I do feel that this project was above the level that UMGC has taught us to this point. It was very complex, and I had to leave out Lambda Functions completely because it was never discussed to this point. I know most of the students taking this class are working in the field and are doing this for a degree, but I am coming from a completely different background with no experience whatsoever. I know other students had to have felt the same way that I did if they didn’t have the background.

UML Class Plan



|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Input | Expected Output | Pass/Fail |
| 1 | 5.6 3 4 1 8.3 0  4 2 54 7 5 0  1 5 2 4 5 0  3 2 1 4 7 5  1.2 5 7 8 9 2 | 5.6x^3 + 4.0x + 8.3  4.0x^2 + 54.0x^7 + 5.0  1.0x^5 + 2.0x^4 + 5.0  3.0x^2 + 1.0x^4 + 7.0x^5  1.2x^5 + 7.0x^8 + 9.0x^2  Polynomials are Weakly Ordered | Pass – mixed order |
| 2 | 1 4 2.3 2 1 0  1 3 2 2 4 0  1 2 1 0 | 1.0x^4 + 2.3x^2 + 1.0  1.0x^3 + 2.0x^2 + 4.0  1.0x^2 + 1.0  Polynomials are Weakly Ordered | Pass – weakly order |
| 3 | 4.0 3 2.5 0 8.0 0  4.0 4 5.0 0  4.5 4 5.7 2 8.6 0 | 4.0x^3 + 2.5 + 8.0  4.0x^4 + 5.0  4.5x^4 + 5.7x^2 + 8.6  Polynomials are Strongly Ordered | Pass |
| 4 | 5 2 4 -1 2 -3  5 1 4 2 4 0  1 4 7 8 9.6 2 | JOptionPane Error Message | Pass |
| 5 | Empty File | JOptionPane Error Message | Pass |

A screenshot of a computer

Description automatically generated

Figure 1 Test Case 1

A screenshot of a computer screen

Description automatically generated

Figure 2 Test Case 2

A screenshot of a computer screen

Description automatically generated

Figure 3 Test Case 3

A screenshot of a computer

Description automatically generated

Figure 4 Test Case 4

A screenshot of a computer

Description automatically generated

Figure 5 Test Case 5