Polynomial Calculator

Project 1B Report

1 February 2021 - Intro to Data Structures with Java

Github Link: https://github.com/rdmallinson7/Project_1_DS.git

ISSUED BY

St. Charles Community College

TEAM SIX

Rachael Mallinson

Sofia Nikas

Aleksis Martin

Design Explanation

List of Data Structures used in our Polynomial Calculator

1. ArrayLists

First we created a Term class defined by 2 integers (the coefficient and the exponent). Then, we split the polynomial entered by the user - isolating each string separated by the "+" sign. By doing this, we were able to identify the coefficient and exponent in each string and create Terms. The role of our first ArrayList is to hold those Terms .

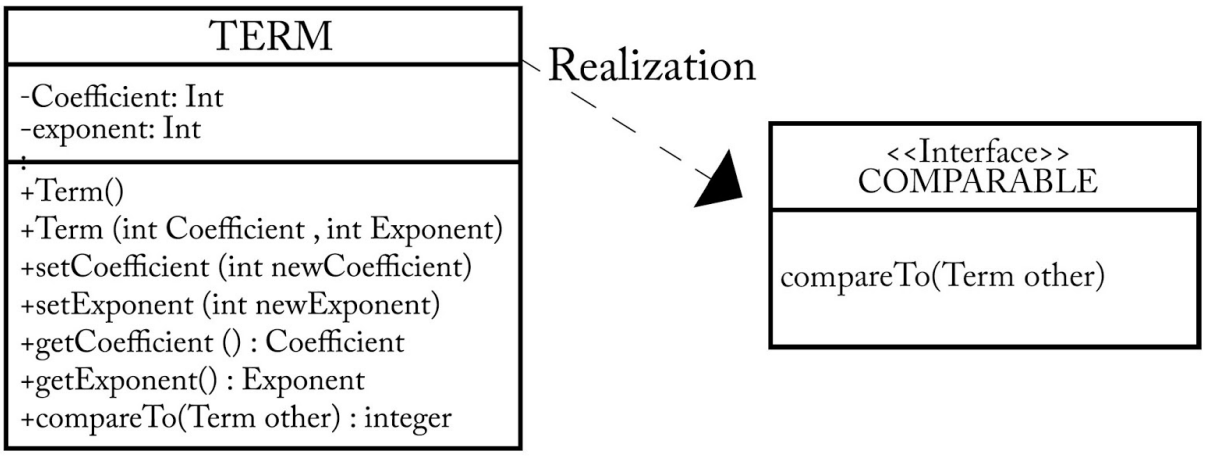
Then we sorted the ArrayList in decreasing order by exponent, and combined any coefficients with the same exponent. We accomplished this by iterating through the list and comparing each term's exponent to every other term, to see if they needed to be combined.

The last step was to output the resulting polynomial to the console as a string in a user-friendly fashion.

For additional usability we added menu options, giving the user the opportunity to continue adding polynomials or to exit the program.

UML Class Diagram

The following is a UML Diagram that outlines our Polynomial Calculator:



Test Cases

Test Case 1:

First test case is two polynomials with no duplicates but out of order:

First polynomial: 6x^2 + 3x^5 - 32 + x

Second polynomial: 3x^3 - 2x^4 + x^6

Expected result : X^6 + 3X^5 - 2X^4 + 3X^3 + 6X^2 + X - 32

Actual result:

```
Polynomial Calculator
Please Choose your Menu Option by choosing a Number to enter.
1. Enter Two Polynomials to get the product
2. Quit
Enter here:1

Please enter two polynomials with no spaces and all variables are "x".
First Polynomial: 6x^2+3x^5-32+x
Second Polynomial: 3x^3-2x^4+x^6

The sum of the two polynomials is:
X^6 + 3X^5 - 2X^4 + 3X^3 + 6X^2 + X - 32

Try again?
1. Yes
2. No
2
Thank you for using the Polynomial Calculator. Have a great day!
```

Conclusion: Expected result matches actual result. The test was a success.

Test Case 2:

Second test case is two polynomials with many duplicate exponents, some of which cancel each other out:

First polynomial: 4x^3 - 7x^2 + x^3 - x + 3 + 3x^7

Second polynomial: x^3 - 2x^7 + 2x^2 + 4x - 6x^3 + 9

Expected result: X^7 - 5X^2 + 3X + 12

Actual result:

```
Polynomial Calculator
Please Choose your Menu Option by choosing a Number to enter.
1. Enter Two Polynomials to get the product
2. Quit
Enter here:1

Please enter two polynomials with no spaces and all variables are "x".
First Polynomial: 4x^3-7x^2+x^3-x+3+3x^7
Second Polynomial: x^3-2x^7+2x^2+4x-6x^3+9

The sum of the two polynomials is:
X^7 - 5X^2 + 3X + 12

Try again?
1. Yes
2. No
2
Thank you for using the Polynomial Calculator. Have a great day!
```

Conclusion: Expected result matches actual result. The test was a success.

Test Case 3:

Third test case is two polynomials with large numbers and some negative exponents:

First polynomial: 376x^234 - 23x^-45 + 2x^-2 + 459 + 34x^-2

Second polynomial: -98x^2 + 6295x^2984 - 23487 + 6x

Expected result: 6295X^2984 + 376X^234 - 98X^2 + 6X - 23028 + 36X^-2 - 23X^-45

Actual result:

```
Polynomial Calculator
Please Choose your Menu Option by choosing a Number to enter.
1. Enter Two Polynomials to get the product
2. Quit
Enter here:1

Please enter two polynomials with no spaces and all variables are "x".
First Polynomial: 376x^234-23x^-45+2x^-2+459+34x^-2
Second Polynomial: -98x^2+6295x^2984-23487+6x

The sum of the two polynomials is:
6295X^2984 + 376X^234 - 98X^2 + 6X - 23028 + 36X^-2 - 23X^-45

Try again?
1. Yes
2. No
2
Thank you for using the Polynomial Calculator. Have a great day!
```

Conclusion: Expected result matches actual result. The test was a success.

Project Contributions

The contributions of each Team Member are as follows:

Rachael Mallinson

1. Term Class
2. Comparable Type
3. Menu
4. UML Diagram

Sofia Nikas

1. polynomialSplitter()
2. outputPolynomial()
3. sumPolynomial()
4. Test Cases

Aleksis Martin

1. findMaxTerm()
2. reduceLikeTerms()
3. sort()
4. sumPolynomial()

Ideas for Future Improvement

The following are ideas for improvements we predict we could make to our Polynomial Calculator, given an adequate amount of time.

1. Maybe implement the use of Stacks in order to convert infix polynomial operations into postfix. That might be a faster way to split up the polynomials. We'd have to experiment with it to know for sure if it would be an improvement.
2. Possibly use Stack to more efficiently split polynomials into terms.
3. Find a way to compare polynomials without rules like "must have no spaces".
4. Create a new Exception for the menu that loops the user back to the start in the case that someone accidentally types in a polynomial instead of a menu option. Right now, if the user types a polynomial right away, the program crashes.

This concludes our Project 1B Report. Thank you for this opportunity and for your attention.