**Team Project 1**

Project 1B – Addition of Two Polynomials

**Luke Janis**

**Max Hoffman**

**Montana Shaw**

**Robert Blocker**

**February 15th, 2022**

**Table of Contents**

[1. System Design 2](#_Toc95682904)

[2. UML Diagram 2](#_Toc95682905)

[3. Test Cases 2](#_Toc95682906)

[4. Team Member Contribution 2](#_Toc95682907)

[5. Possible Improvements 2](#_Toc95682908)

# System Design

Our system implemented the use of a Linked List to store the user inputted polynomials. We used the Iterator and List\_Iterator methods given by Professor Wang to iterate through the list to find like terms for the addition of the coefficients. The system also used the Iterable and the DNode created by Professor Wang, though the DNode class was placed into our Linked\_List file. The doubly-linked list was used to sort the coefficients and polynomials of each term. As the structure of the doubly-linked list, there is a “prev” Node that holds the address of the previous Node in the list, then there are the two data Nodes that hold the coefficient and exponent of each parsed-term of the polynomial, and then there is a “next” Node which contains the address of the next Node in the list.

The method called *convertString* uses nested for-loops and nested if-statements to determine whether the input values are positive or negative. Along with converting the data entered from strings to integers and storing the integers in a doubly Linked List.

While the *convertString* method was being developed the text menu for the user was being developed. The text menu uses a while-loop that continues until the user enters ‘q’, and within that while-loop there are if-statements that call the necessary methods. For example, if the user enters ‘r’ the program which will call the method *sortPolys* (which will be discussed later) to sort and combine any polynomials entered and if none were entered the method does nothing. The last options not described are ‘f’, and ‘s’  which allows the user to enter the first, and second polynomials respectively and stores the entered polynomial in a variable and calls the *convertString* method.

Next is the Term class, aside from global variables and constructor. The class contains two important methods; the *compareTo* method and the *toString* method. The *compareTo* method compares the exponent's two polynomials and returns 1 if the second exponent is greater than the first and returns -1 if the reverse is true. The *compareTo* method also contains two if-statements to test if either coefficient or exponent contains a invalid variable. One of the last methods for the Term class is the *toString* method, which formats polynomials by converting them back to strings with the methods *getNumber* and *getExponentStr*.

# UML Diagram

Diagram

Description automatically generated

# Test Cases

1. For the first test, we want to add the following polynomials:

3x3+2x2+4x

2x3+4x2+10

Let’s put this into readable syntax for the console and enter the polynomials separately:

3x^3+2x^2+4x

2x^3+4x^2+10

Text

Description automatically generated

Once we type r, we expect that the polynomials would be sorted and added together:

3x^3+2x^2+4x + 2x^3+4x^2+10 → 3x^3 + 2x^3 + 2x^2 + 4x^2 + 4x + 10

We expect the program to output the added polynomial like so:

 5x^3 + 6x^2 + 4x + 10

Text

Description automatically generated

The program outputted exactly what the expected polynomial was.

1. For the second test, we want to add the following polynomials:

3x3-2x2-4x+13x4

2x3+4x2+10-8x4

Let’s put this into readable syntax for the console and enter the polynomials separately:

3x^3-2x^2-4x+13x^4

2x^3+4x^2+10-8x^4

Text

Description automatically generated

Once we type r, we expect that the polynomials would be sorted and added together:

13x^4+3x^3-2x^2-4x + -8x^4+2x^3+4x^2+10 →

13x^4 - 8x^4 + 3x^3 + 2x^3 + 4x^2 - 2x^2 - 4x + 10

We expect the program to output the added polynomial like so:

 5x^4 + 5x^3 + 2x^2 - 4x + 10

Text

Description automatically generated

The program outputted the expected polynomial and gave the polynomials that we added.

3) For the last test, we want to add the following polynomials (using some negative exponents):

3x-3+2x2+4x

2x3+4x2+10x-4

Let’s put this into readable syntax for the console and enter the polynomials separately:

3x^-3+2x^2+4x

2x^3+4x^2+10x^-4

Text

Description automatically generated

Once we type r, we expect that the polynomials would be sorted and added together:

2x^2+4x+3x^-3 + 2x^3+4x^2+10x^-4 →

2x^3+2x^2+4x^2+4x+3x^-3+10x^-4

We expect the program to output the added polynomial like so:

 2x^3 + 6x^2 + 4x + 3x^-3 + 10x^-4

Text

Description automatically generated

The program outputted the expected polynomial and also gave the polynomials that we added.

# Team Member Contribution

Luke Janis – Luke worked on a large portion of the development of the code for the project. Through logic and testing, Luke helped make sure the code functioned properly. Luke helped develop code in the Main class and develop the logic for the doubly linked list. Also worked with the code in the Term class.

Max Hoffman – Max worked on the development of the code for the project.

Montana Shaw – Montana worked on the development of the code for the project. While working on the Term class, figuring out the logic for implementing that class, and helping with the logic of the SortPolys() method. Montana also worked on the documentation of the system, collaborating with Robert to help document the project.

Robert Blocker – Robert worked on the code development and documentation of the system. Worked on the development of the SortPolys() method and worked with implementing methods in the system.  Robert collaborated with Montana to document the project and make sure the group’s logic flowed into the document.

# Possible Improvements

* The design of the system could be improved by trying to develop a system that was more conscious of the time complexities.
* There is a flaw where if a coefficient has a (-) sign on it, the program will output for example,

5x + -4x^2

Though it is not mathematically wrong and functions the same as though it showed subtraction, it is a minor change that could be fixed.

* There is a also bug where inputting a term such as (-4x) will print out:

-4

1

This is rare and was used by us as we developed to see the separated coefficient and exponents. Though removed for every other case, we found this case has this bug. Nothing else is affected by this.