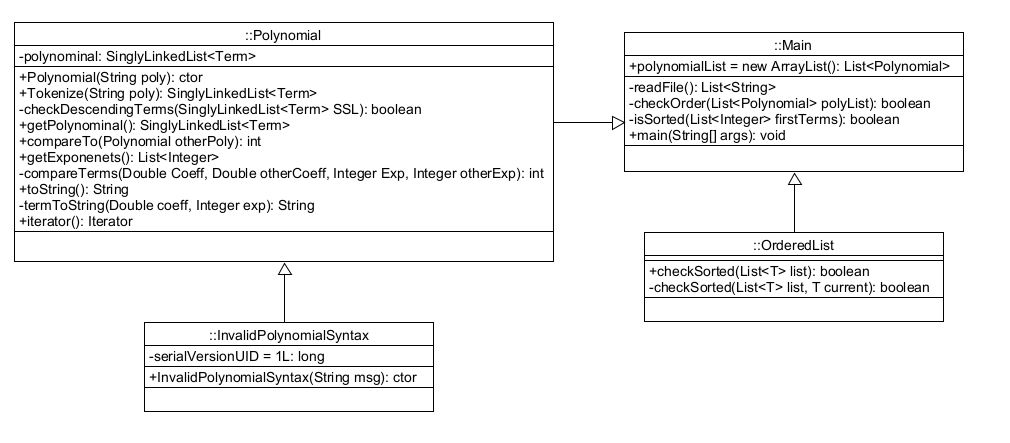
Allen Taylor

CMIS 350 6382

Project 2

2/4/2022

**UML Diagram for Project 2**



**Test Case 1**

Requirements:

1. Test cases include a file in both strong and weak sorted order.
2. Test cases include a polynomial with exponents of 0, 1 and 2 or more.

|  |  |
| --- | --- |
| Input: | 4.0 2 2.5 1 8.0 0  4.5 4 5.0 0 5.0 0  5.0 4 5.7 2 8.6 0 |
| Expected Output: | Strong Ordered: true  Weak Ordered: true |
| Actual Output: | Strong Ordered: true  Weak Ordered: true |

Graphical user interface, application

Description automatically generated with medium confidence

Graphical user interface, text, email

Description automatically generated

**Test Case 2**

Requirement:

1. Test cases include a file in weak but not strong sorted order.
2. Test cases include a polynomial with exponents of 0, 1 and 2 or more.

|  |  |
| --- | --- |
| Input: | 4.0 2 2.5 1 8.0 0  5.0 4 5.0 1 0 0  4.5 4 5.7 1 8.6 0 |
| Expected Output: | Strong Ordered: false  Weak Ordered: true |
| Actual Output: | Strong Ordered: false  Weak Ordered: true |

Graphical user interface, text, application, chat or text message

Description automatically generated

Text, email

Description automatically generated

**Test Case 3**

Requirement:

Test cases include a file in neither strong nor weak sorted order

|  |  |
| --- | --- |
| Input: | 4.5 4 5.0 0 5.0 0  4.0 2 2.5 1 8.0 0  5.0 4 5.7 2 8.6 0 |
| Expected Output: | Strong Ordered: false  Weak Ordered: false |
| Actual Output: | Strong Ordered: false  Weak Ordered: false |

Graphical user interface, text, application, chat or text message

Description automatically generated

Text

Description automatically generated

**Test Case 4**

Requirement:

InvalidPolynomialSyntax thrown by the constructor of the Polynomial class when:

1. The supplied string contains coefficients or exponents of an improper type.
2. Exponents fail to be listed in strictly descending order.

|  |  |
| --- | --- |
| Input: | -1 1 8.0 0 4.0 2  5.0 4 5.0 1 0 0  4.5 4 5.7 1 8.6 0  2.5 1 8.0 0 4.0 2  5.0 4 5.0 1 0 0  4.5 4 5.7 1 8.6 0 |
| Expected Output: | JOptionPanes display error messages. |
| Actual Output: | JOptionPanes display error messages. |

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, application

Description automatically generated

**Lessons Learned**

Graphical user interface, text, application, email

Description automatically generated

One of the hardest parts of this assignment was using the Comparable Interface. I finally found a solution in the provides coding examples in the Week 1 Generics. This provided me with the isSorted method.

To determine the Weak Order of a list of Polynomials I did the following:

1. Read all exponents from the Polynomial.
2. Take only the exponent from the FIRST term.
3. Do this for all Polynomials in the list and add them to a List<Integer>.
4. Lastly, use the isSorted to check if they are in Weak Order.