DOCUMENTATION

ASSIGNMENT 1

STUDENT NAME:Leo Mihai Ioan
GROUP: 30424

CONTENTS:

1. Assignment Objective

(i). Main objective

-the design and the implementation of a polynomial calculator which has a graphic interface

(ii).Sub-objectives

Sub-objective	Description	Section Addressed
Design graphical interface	The application needs to be user friendly and easy to understand, it will have a fields for reading inputs and writing results and buttons for operations	Dessign
Polynomial creation, reading and writing	The string value will be transformed in polynomials, and the polynomials should be able to be transformed back to strings. The Polynomials are stored in a Tree|Map	Implementation
Arithmetic operations with polynomials	The implementation of the addition, subtraction, multiplication, division, derivation and integration	Implementation
Checking Input	Check if the input string is a valid polynomial using regex pattern matching	Implementation
Testing the operations	The operations will be verified using a number of test with the help of Junit	Results

2. Problem Analysis, Modeling, Scenarios, Use Case

(i).Functional requirements

-the user needs to be able to give only a valid polynomial as an input

-if the inputs are valid the application should be able to perform the operations of addition, subtraction, multiplication, division, integration and derivation

-the results of the operations, and in the case of division the rest, will be showed on the interface

(ii).Using cases

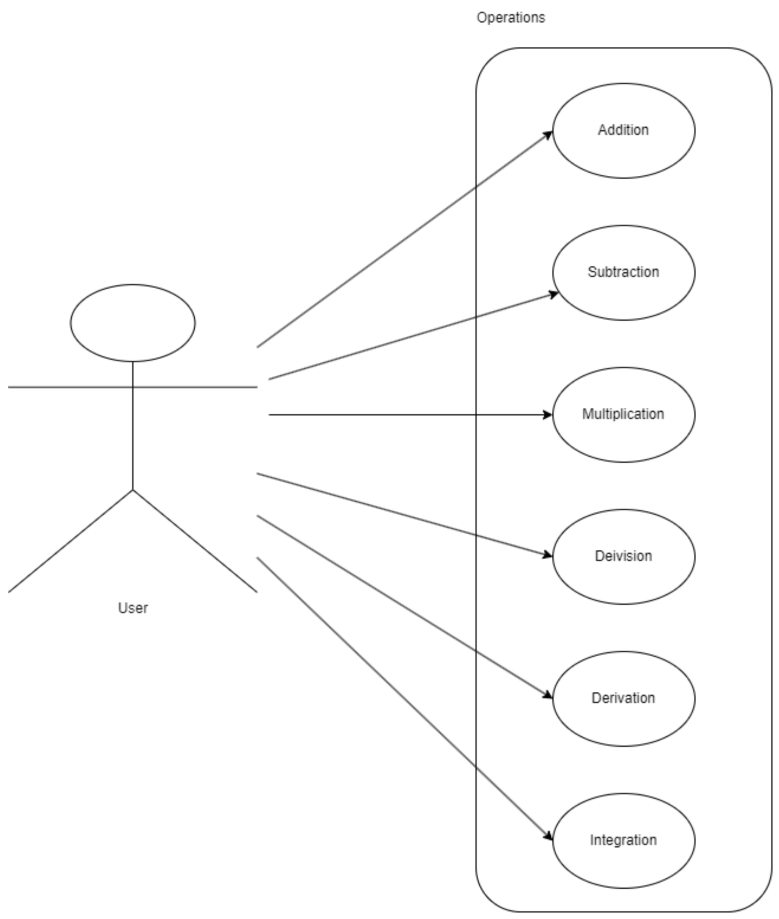
Two input operations:

-the user inserts the first input in the textbox, selects one of the operations(addition, subtraction, multiplication or division), then inserts the second input followed by the pressing equal button for the result to be showed

One input operations:

-the user can introduce a single polynomial and then press the one of the integration and derivation buttons followed by the equal button and the result will be showed
Wrong input
-if the user gives a wrong input a message will be showed

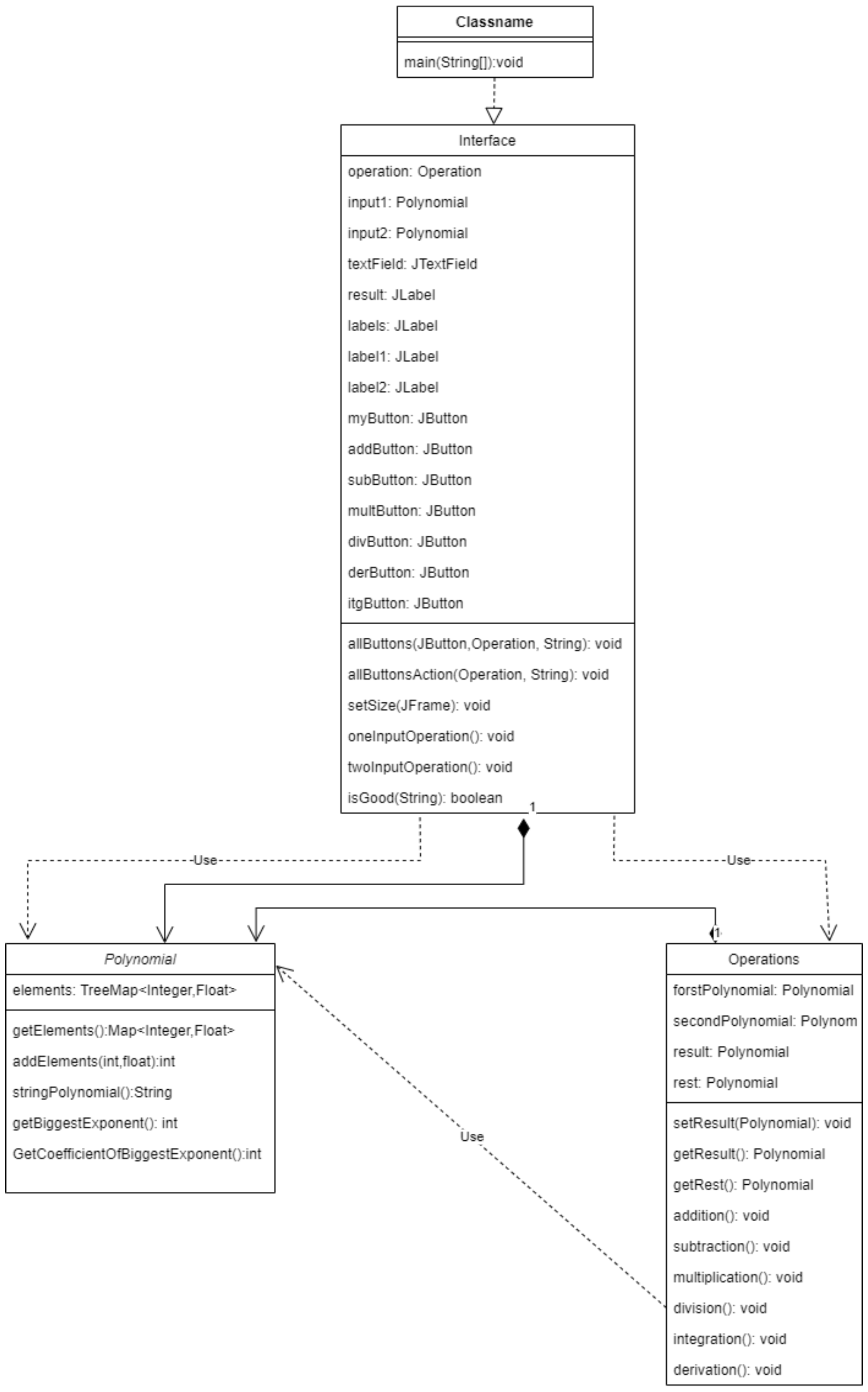
(iii).Use Case diagram



3. Design

The Polynomial Computer application adheres to Object-Oriented Programming (OOP) principles, with a focus on modularity, reusability, and encapsulation. It is organized into distinct classes, each assigned with a particular task. This approach facilitates the separation of backend and frontend components, streamlining maintenance and enabling potential expansions with ease.

The class Diagrams



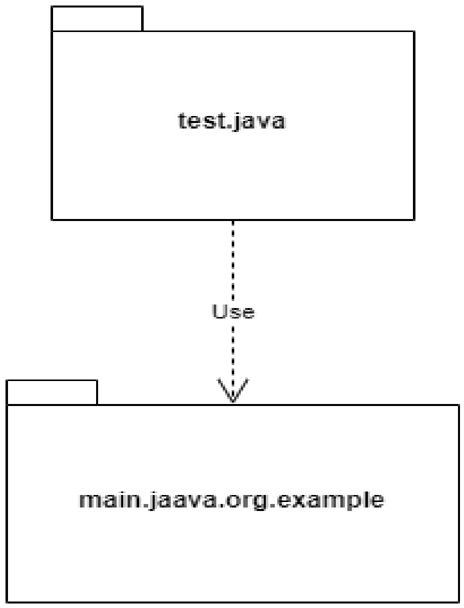
Packages

<u>src.main.java.org.example</u>

-contains the classes Main,Operations, Polynomial and Interface are located

<u>src.test.java</u>

-contains the TestingClass which is responsible for the testing



4. Implementation

Class Polynomial

<u>Fields:</u>

-TreeMap<Integer,Float> elements: Stores the coefficient as a float while its key is the exponent

<u>Constructor:</u>

-there are 3 constructors:one creates an empty TreeMap, one creates a monomial with only one exponent and coefficient, while the last one transforms a String into a Polynomial

<u>Methods:</u>

-addElements (int expo, float coefficient): Adds an monomial to an existing polynomial, returns int

-stringPolynomial(): Transforms the instance of Polynomial in a string, tha string can also be used to create an identical polynomial

-getBiggestExponent(): Returns the biggest exponent of the polynomial

-getCoefficientofBiggestExponent():

-getElements(): return elements

Class Operations

<u>Fields:</u>

-Polynomial firstPolynomial: The first polynomial in the operations of addition, subtraction, multiplication and division, and the one that will be integrated or derived if it s the case

-Polynomial secondPolynomial: The first polynomial in the operations of addition, subtraction, multiplication and division

-Polynomial result: Keeps the result if an operation is executed

-Polynomial rest: Keeps the rest for division

<u>Constructors:</u>

-there 2 constructors one has 1 input for derivation and integration, while the other one has 2 for the other operations

<u>Methods:</u>

-addition(): Adds the first and second Polynomial and then the result field of the instance will be updated

-subtraction(): Subtracts from the first the second Polynomial and then the result field of the instance will be updated

-multiplication(): multiplies the first and second Polynomial and then the result field of the instance will be updated

-division(): divides the first by the second Polynomial and then the result and rest fields of the instance will be updated

-derivation(): derive the first Polynomial and then the result field of the instance will be updated

-integration(): integrates the first Polynomial and then the result field of the instance will be updated

Class Interface

Fields:

-Operation enum: {add,sub,mul,div,der,itg,none}: Remembers what operation is being executed at the moment

-Polinomial input1: Used for operations

-Polinomial input2: Used for operations

-JButton Buttons: Those are actively more fields, a button for each of the six operations and one for equal

-JTextField textField: The field that will be used by the user to give the input

-Jlabel result/label1/label2/labels: Show the result of the operation, the inputs and the type of operation

Constructors:

-When calling an instance of the class the frame and graphic interface will be created, and the actions of the buttons will be called

Methods:

-setSizes(JFrame): It sets the sizes, positions and colors of the buttons, labels, text field and frame

-isGood(String): A boolean method that checks if the String input can be transformed in a Polynomial

-allButtons(jButton,Operation,String): Adds the actions listener for a specific Button by calling allButtonsAction()

-allButtonsAction(Operation,String): Selects the operation, reeds the first input and shows a message if the input is invalid otherwise it will show the input

-oneInputOperation(): Calculates and shows the result for derivation or integration

-twoInputOperation():Calculates and shows the result for addition, subtraction, multiplication and division, for the last one also check for the division with 0

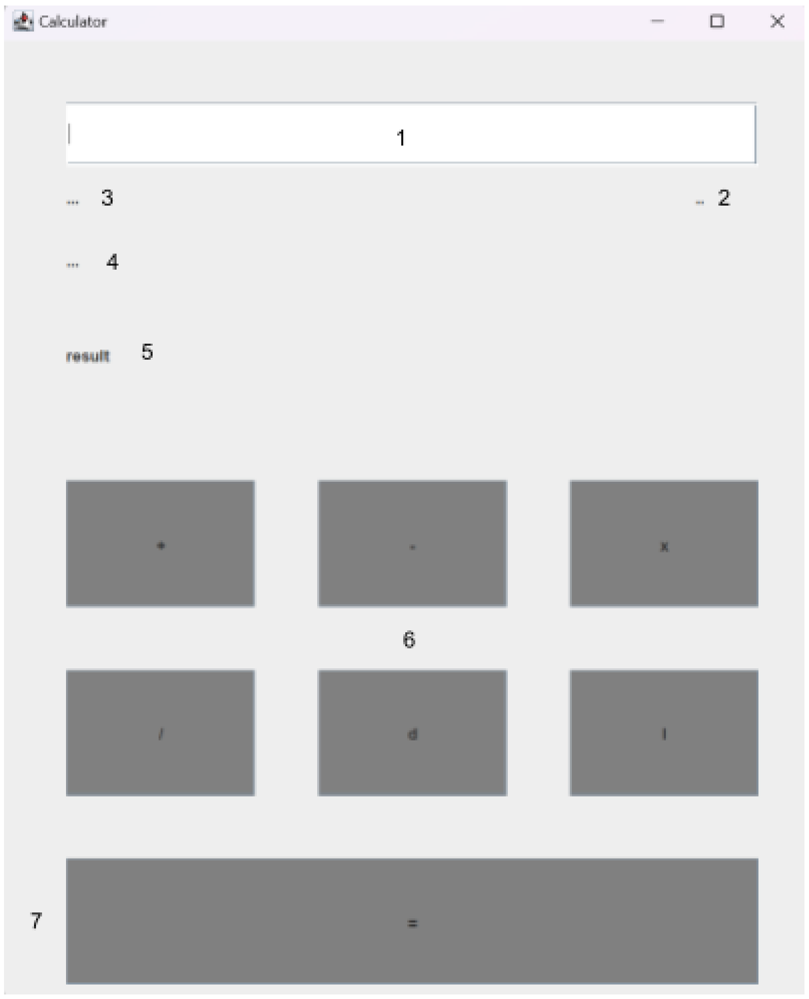
-equalPress(): calls oneInputOperation() or twoInputOperation, for the last one also checks if the second input is a valid on

Class Main

Methods:

-main(String[] args): Calls an instance of Interface

User Interface



1. *TextField for the input*
2. *Label that indicates the type of operation*
3. *Label That shows the first input after is been converted to Polynomial*
4. *Label That shows the second input after is been converted to Polynomial*
5. *Label that shows the result or an error message*
6. *The buttons for the 6 operations*
7. *The result button*

5. Results

The testing is realized with the help of JUnit, by creating a test scenario for each of the 6 operations. The comparison is made with the result and an additional input for each scenario that represents the expected output. Both the inputs and the expected results are given as Strings, which will also call the methods in the Polynomial Class

Operation	**Input1**	**Input2**	**Expected**	**Actual**	**Time**
Addition	x^2-3+x^4	-x^2+3-6x	x^4-6x	x^4-6x	1ms
Subtraction	x^2-3+x^4	-x^2+3-6x	x^4+2.0x^2+6x-6	x^4+2.0x^2+6x-6	21ms
Multiplicationion	x^2-3+x^4	-x^2+3-6x	-x^6-6x^5+2x^4-6x^3+6x^2+18x-9	-x^6-6x^5+2x^4-6x^3+6x^2+18x-9	1ms
Division	x^2-3+x^4	-x^2+3-6x	-x^2+6x-40/ -258x+117	-x^2+6x-40/ -258x+117	3ms
Derivation	x^2-3+x^4	--	4x^3+2.0x	4x^3+2.0x	1ms
Integration	x^2-3+x^4	--	0.2x^5+0.33333333333333x^3-3x	0.2x^5+0.33333333333333x^3-3x	1ms

6. Conclusion

The implementation of the polynomial calculator was a success, with each of the 6 operations also being tested and implemented. The application has a friendly interface and is easy to use.

Possible Upgrades:

-Selincting to operation from keyboard

-Keeping a history of past inputs

Thighs Learned:

-How to calculate the remainder and the result obtained from the division of two polynomials

-The testing of an algorithm using Junit

-The creation of an user interface using only JavaSwing

7.Bibliography

1. https://www.baeldung.com/junit-5
2. https://www.tutorialspoint.com/index.htm
3. https://www.geeksforgeeks.org/treemap-in-java/
4. https://en.wikipedia.org/wiki/Polynomial_long_division
5. https://www.javatpoint.com/java-swing