

DOCUMENTATION

ASSIGNMENT 1

Polynomial Calculator

CONTENTS

1. Assignment Objective 3

2. Problem Analysis, Modeling, Scenarios, Use Cases 3

3. Design 5

4. Implementation..... 6

5. Results..... 7

6. Conclusions 8

7. Bibliography 8

1. Assignment Objective

The main objective of the assignment is to design and implement a polynomial calculator with a dedicated graphical interface through which the user can insert polynomials, select the mathematical operation to be performed and view the result.

The sub-objectives are to:

- ❖ analyze the problem and identify requirements;
- ❖ design the polynomial calculator;
- ❖ implement the polynomial calculator;
- ❖ test the polynomial calculator;

2. Problem Analysis, Modeling, Scenarios, Use Cases

2.1 Problem Analysis

A polynomial is a mathematical expression that consists of variables and coefficients, which are combined using addition, subtraction, multiplication, and non-negative integer exponents. For example, the expression $3x^2+2x-1$ is a polynomial in the variable x , where 3,2 and -1 are coefficients and 2 is the highest exponent of x .

Polynomials are widely used in mathematics, physics, engineering, and other fields to represent a wide range of phenomena. They can be used to model relationships between variables, solve equations, approximate complex functions, and more. Polynomials are an important part of algebra, and they are studied extensively in both pure and applied mathematics.

Functional requirements:

- ❖ The polynomial calculator should allow users to insert polynomials
- ❖ The polynomial calculator should allow users to select the mathematical operation: addition, subtraction, multiplication, integration, derivation
- ❖ The polynomial calculator should perform the selected operation
- ❖ The calculator should be able to display polynomials in a user-friendly format, with coefficients and exponents clearly labeled
- ❖ The calculator should have a user friendly interface that is easy to navigate and use

Non-functional requirements:

- ❖ Performance: the calculator should perform the operations quickly and efficiently
- ❖ Reliability: the calculator should produce correct results for all valid polynomial inputs
- ❖ Usability: the calculator should be user-friendly with a clear and intuitive interface

2.2 Modeling the problem

The user will be able to perform operations on polynomials through the graphical interface, by introducing them and selecting the operation. The result will then be displayed in the interface.

In the model we use the following classes:

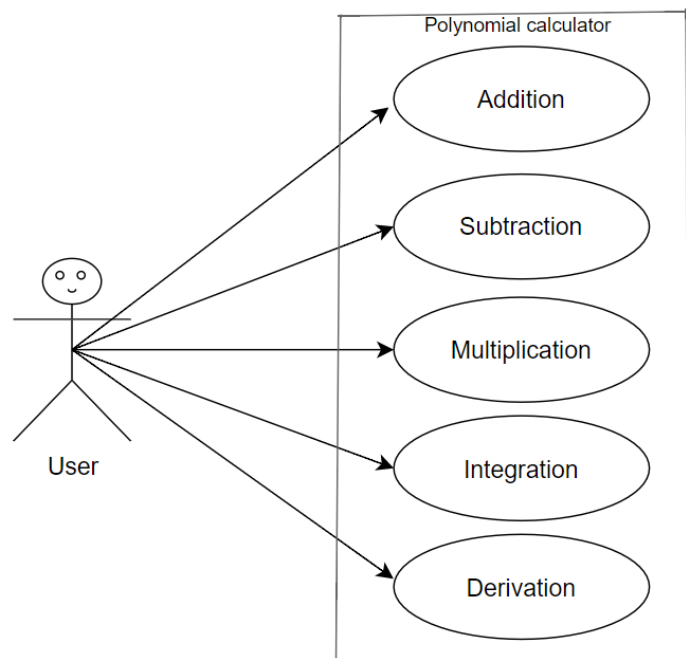
1. Polynomial class: this class define the polynomial and contains properties such as the coefficients and exponents of each term, as well as methods that allow us to perform the operations on the polynomials.
2. Operations class: this class contains methods that represent the operations that the user can perform on the polynomials, such as: addition, subtraction, multiplication, integration, derivation.
3. GUI class: is the class that define the user interface of the polynomial calculator and contains methods for handling the inputs and displaying the result.

2.3 Scenarios and use cases

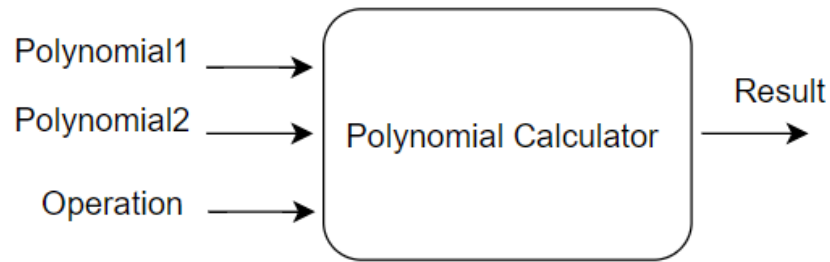
The use case presents the user that interacts with the application and performs different operations on two chosen polynomials such as addition, subtraction, multiplication, integration, derivation.

The user selects the operation, clicks on the Compute button and the result will be displayed

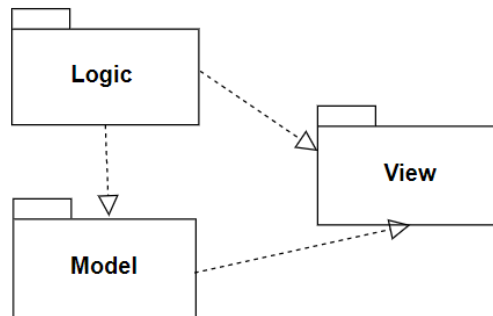
If the user introduces incorrect polynomials, with two or more variables, there will appear a message.



3. Design

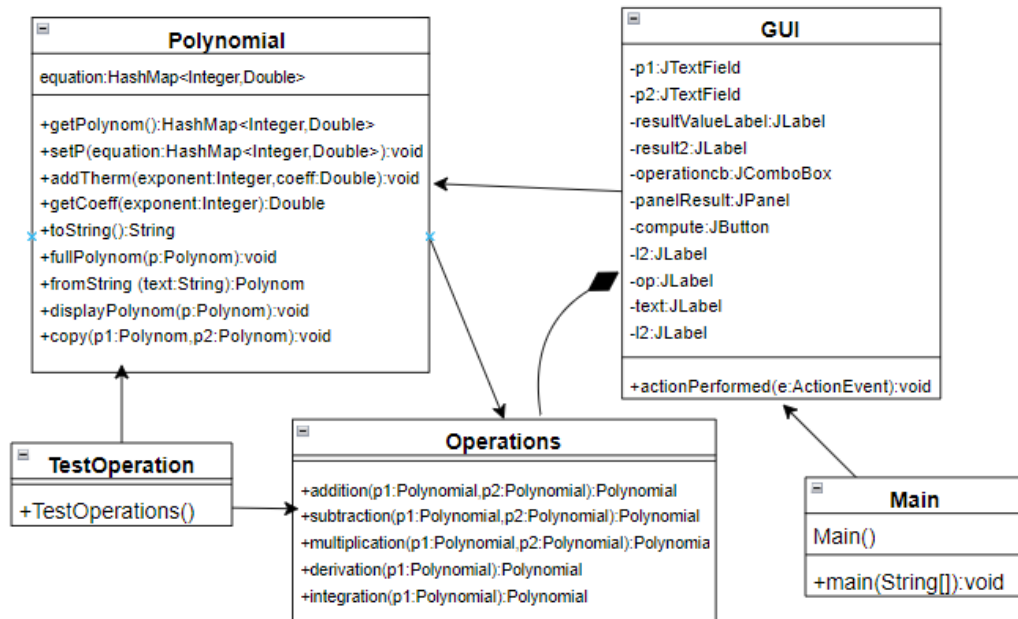


We give as inputs two polynomials and the operation we want to perform and have as output the result of the operation.



Java packages help in organizing multiple modules and group together related classes and interfaces. The projects is splitted in tree packages:

- ❖ Logic – contains the Operations class that implements the addition, subtraction, multiplication, integration, derivation.
- ❖ Model – contains the class Polynomial in which is defined the used data structure and the class TestPolynom in which we test the correctness of the Polynomial class
- ❖ View – contains a single class that implements the user interface



4. Implementation

Polynomial class

This class contains only an instance variable which consists of an `HashMap<Integer, Double>` that define the polynomial. The coefficients are of type `Double`, while the exponents are of type `Integer`. The polynomial is stored in this data structure, from the lowest to the highest exponent of the polynomial. The key of the map represents each power.

Methods:

- public `HashMap<Integer, Double>` `getPolynom()`: returns the map that contains the polynomial;
- public void `addTherm(Integer exp, Double coeff)`: add terms in the `HashMap`;
- public `Double` `getCoeff(Integer exp)`: return the coefficient of the given exponent
- public `String` `toString()`: convert the polynomial to string
- public void `fullPolynom(Polynom p)`: if we do not have all the exponents until the highest one, we add them to the polynomial with coefficient 0
- public `Polynom` `fromString(String text)`: receives as parameter a string and constructs the `HashMap` corresponding to that polynomial
- public void `displayPolynom(Polynom p)`: display the polynomial term by term from the `HashMap`
- public void `copy(Polynom p1, Polynom p2)`: copies the second polynomial in the first one

Operation class

This class contains the methods that perform the operations on the polynomials.

Methods:

-public Polynom addition(Polynom p1,Polynom p2):adds the two polynomials as such: the first polynomial p1 is copied in the sum polynomial , and then we check every monomial of the second polynomial to see if we have the exponent in the HashMap; if not, we create a new monomial, otherwise we add it to the existing one.

-public Polynom subtraction(Polynom p1,Polynom p2):perform the subtraction of the two polynomials with the same method as addition

-public Polynom multiplication(Polynom p1,Polynom p2): multiply the two polynomials

-public Polynom integration(Polynom p1):integrate the polynomial

-public Polynom derivation(Polynom p1):differentiate the polynomial

Gui class

Polynomial Equation Calculator

Polynom1:

Polynom2:

Operation:

For the derivation and integration please enter the first polynom

Result: + 3.0 - 1.0x¹ + 3.8x² - 9.0x³

The user introduces the polynomials, selects the preferred operation and clicks on the Compute button. Each term must respect the format [sign][coefficient][x[^]][exponent] ,ignoring the brackets.

5. Results

In order to test the application, the user has to enter two polynomials and to select the operations.

Polynomial Equation Calculator

Polynom1:

Polynom2:

Operation:

Result: + 1.0 + 3.0x² + 3.0x³ + 1.0x⁵

Polynomial Equation Calculator

Polynom1

Polynom2

Operation

Substraction

Compute

Result: - 1.0 - 1.0x^2 + 1.0x^3 - 1.0x^5

Polynomial Equation Calculator

Polynom1

Polynom2

Operation

Multiplication

Compute

Result: - 1.0 - 1.0x^2 + 1.0x^3 - 1.0x^5

Polynomial Equation Calculator

Polynom1

Polynom2

Operation

Derivation

Compute

Result: + 2.0x^1 + 6.0x^2

Polynomial Equation Calculator

Polynom1

Polynom2

Operation

Integration

Compute

Result: + 0.3333333333333333x^3 + 0.5x^4

6. Conclusions

This application is a user friendly polynomial calculator that performs basic operations such as addition, subtraction, multiplication, integration, derivation. It was an helpful exercise to remember OOP concepts and GUI. As future developments, it could be improved by adding more operations such as computing the result with a certain value, make the plot of the polynomial, extract the roots and many others.

7. Bibliography

- ❖ <https://dsrl.eu/courses/pt/#>
- ❖ <https://www.geeksforgeeks.org/java-util-hashmap-in-java-with-examples/>
- ❖ <https://stackoverflow.com/questions/36490757/regex-for-polynomial-expression>

