

DOCUMENTATION

ASSIGNMENT 1

STUDENT NAME: Doia Bogdan-Mihai
GROUP: 30421

CONTENTS

1.	Assignment Objective.....	3
2.	Problem Analysis, Modeling, Scenarios, Use Cases	3
3.	Design	5
4.	Implementation.....	5
5.	Results.....	6
6.	Conclusions	10
7.	Bibliography.....	10

1. Assignment Objective

Main objective: Design and implement a polynomial calculator with a dedicated graphical interface through which the user can insert polynomials, select the mathematical operation (i.e., addition, subtraction, multiplication, division, derivative, integration) to be performed and view the result.

Sub-objectives:

- Analyze the problem and identify requirements
- Design the polynomial calculator
- Implement the polynomial calculator
- Test the polynomial calculator
- Design a dedicated GUI

2. Problem Analysis, Modeling, Scenarios, Use Cases

Problem Analysis: polynomials are composed of one or more monomials, that have a coefficient, a variable and a degree. The mathematical operations must take into consideration the structure of the polynomial and the graphic user interface must take care of the input and output format. So the functional requirements are as follows: the user should be able to insert polynomials, select the operation they want to perform and read the results, and the calculator should be able to correctly compute all the operation results and handle input and output correctness.

The screenshot shows a graphical user interface for a polynomial calculator. The window is titled "Polynomial Calculator" and has standard window controls (minimize, maximize, close). The interface includes four input fields for polynomials and results, and six buttons for mathematical operations.

Input/Label	Field
Polynomial 1	<input type="text"/>
Polynomial 2	<input type="text"/>
Result 1 (quotient)	<input type="text"/>
Result 2 (remainder)	<input type="text"/>

Sum	Difference
Multiply	Divide
Derivation	Integration

Modeling: for implementing the polynomials we use HashMaps, the keys being the powers of the monomials and the values, the coefficients. The operations that can be computed are addition, subtraction, multiplication, division, differentiation and integration. Selecting the operation and providing the polynomials to be computed is done through the GUI, which will also display the results.

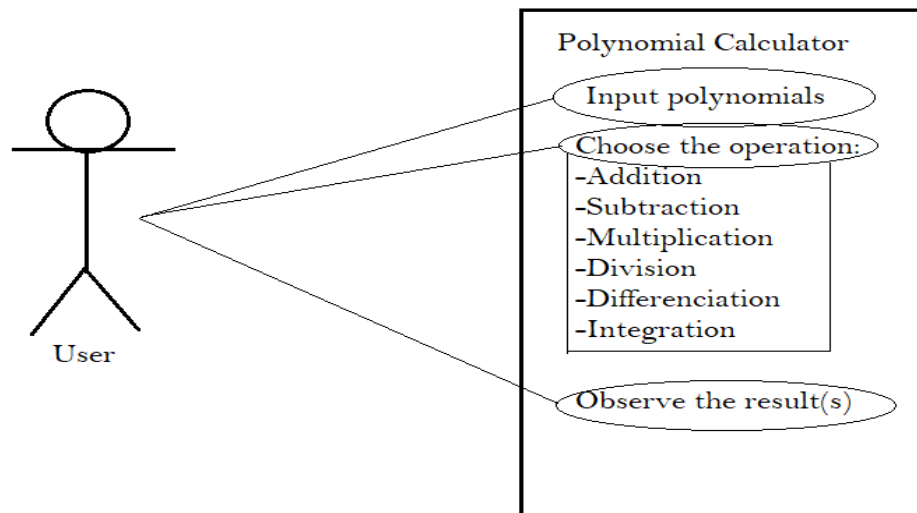
Scenarios and use cases: The use case of the calculator if the computation of an operation on one or two polynomials, depending on the operation. The primary actor is the User.

Main Success Scenario:

1. The user inputs 2/1 polynomials depending on the operation he wants to perform
2. The user presses the button corresponding to that operation
3. The calculator computes the results(s) and the user can read them

Alternative Sequences: Incorrect polynomials/Division by zero

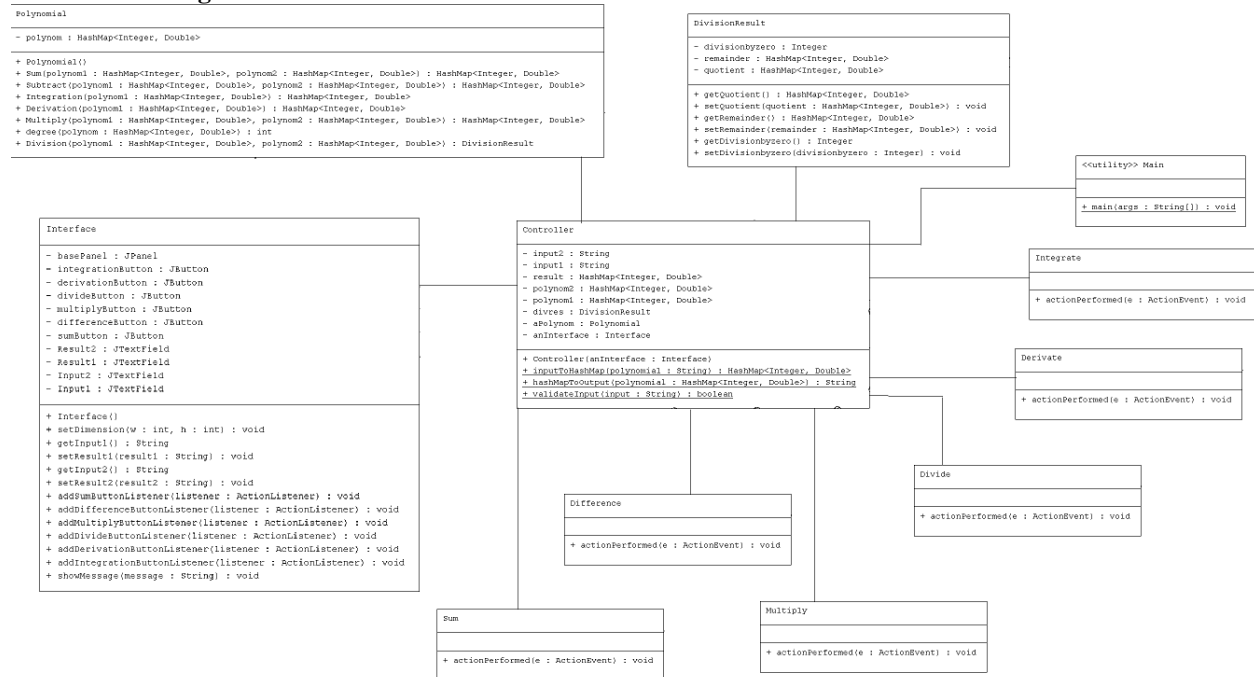
- The user inputs polynomials in a wrong way, for example inserting letters other than x.
- The user attempts to divide a polynomial by 0.
- In both cases he will be prompted with an error message.



3. Design

OOP design of the application: The polynomial calculator respects the OOP concepts, and contains 3 packages model, view and controller, each with their respective classes, some being Polynomial, DivisionResult, Controller, Interface.

UML classes diagram:



Polynomial is the class that contains all the operations that are to be used on the polynomials, DivisionResult is a class used in order to return both the quotient and the remainder after the Division operation, Interface is the class that implements the GUI and Controller is the class controlling the computations and transformations of the input and output behind the GUI.

Data structures: the main data structure used for the calculator is the HashMap, in which the keys represent the powers in the polynomial and their respective values the coefficients.

4. Implementation

Main: this is the class that calls the interface and the controller

Polynomial: this is the class that contains the computations on the polynomials. It contains methods for each operation and also methods to aid in that. Sum, Subtract, Multiply, Integration and Derivation work monomial by monomial and build the result in a separate HashMap, and Division works based on the algorithm provided in the laboratory resources and returns the result using the next class. Another method is degree, which helps the main computation methods.

DivisionResult: this class helps to return the results of the Division method in the Polynomial class.

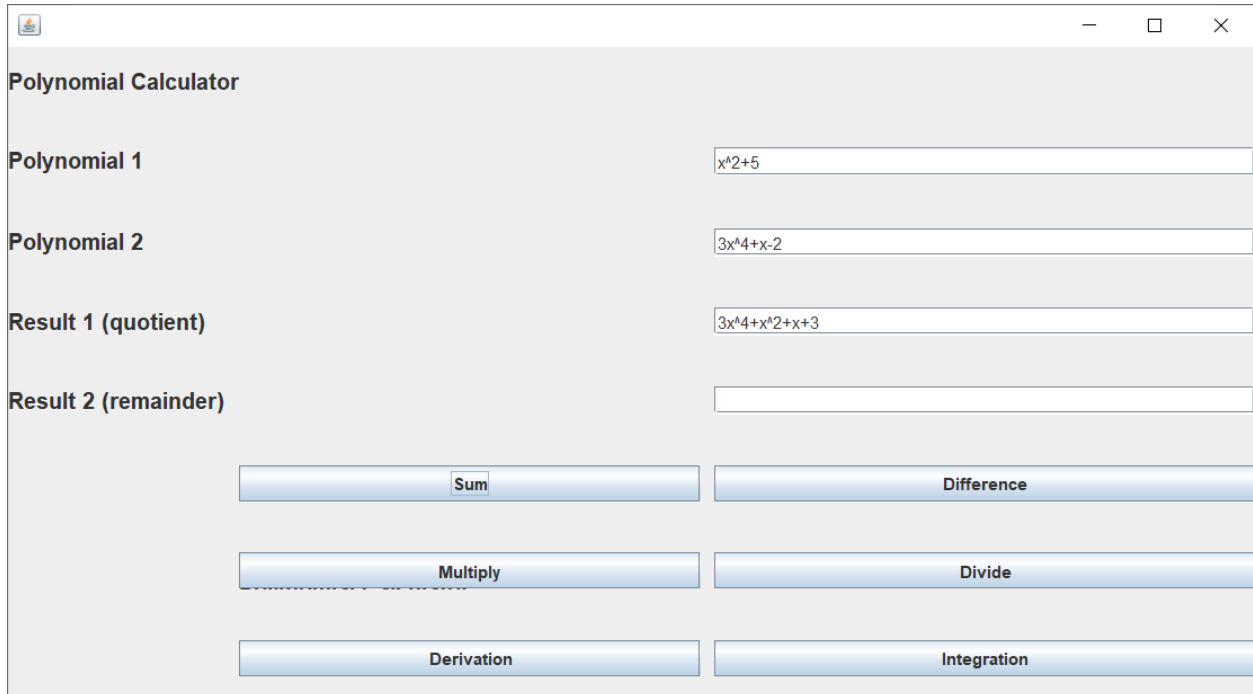
Interface: this is the class that builds the visual side of the class.

Controller: this is the class that “controls” the connection between the interface and the operations, it contains methods that validate the inputs, convert the input strings into HashMaps and also converts the HashMaps into strings. InputToHashMap uses RegEx and pattern matchings to turn the strings provided by the user into HashMaps, and hashMapToOutput method transforms the HashMap by iterating through it, into a string to be displayed, and the validateInput method aids in checking if the provided input is a valid one that can be operated on and if not it is used to print an Error Message.

5. Results

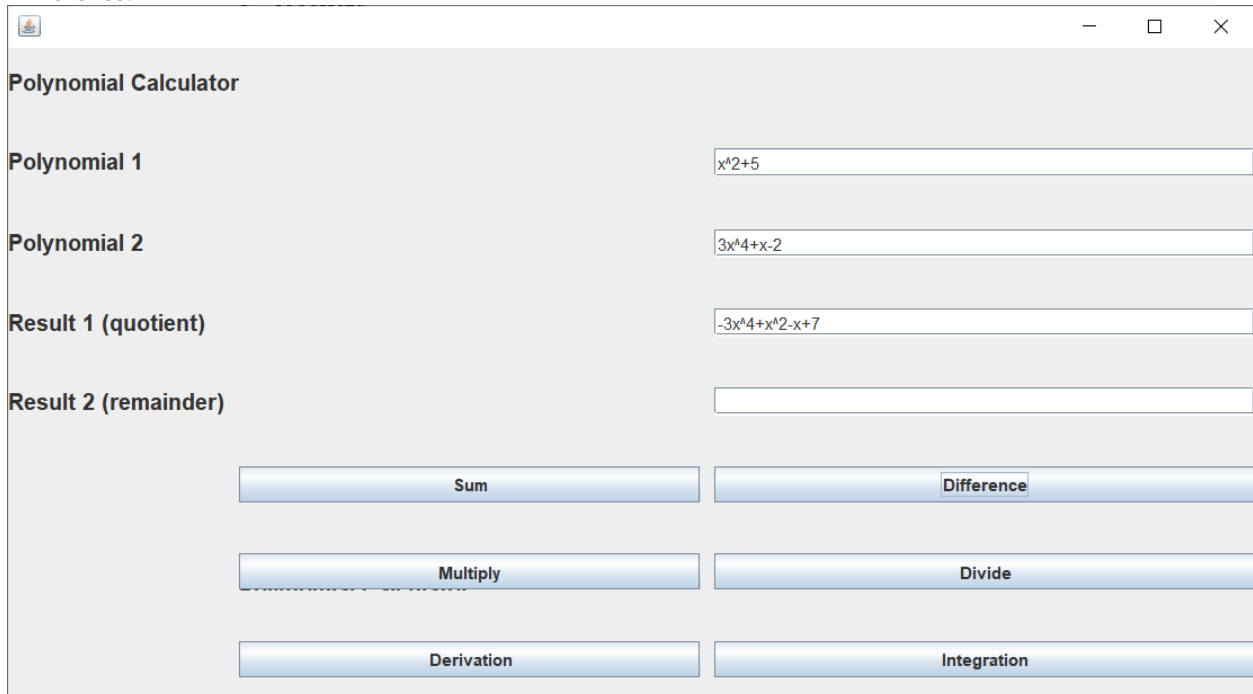
The input is first validated by the methods in the controller and then the polynomials are used to compute the operation specified with the press of a button. Here are some use cases:

Sum:




The screenshot shows a window titled "Polynomial Calculator" with a standard Windows title bar (minimize, maximize, close buttons). The interface contains four input fields and a set of operation buttons. The first two fields are labeled "Polynomial 1" and "Polynomial 2". The next two fields are labeled "Result 1 (quotient)" and "Result 2 (remainder)". The "Polynomial 1" field contains the text x^2+5 . The "Polynomial 2" field contains the text $3x^4+x-2$. The "Result 1 (quotient)" field contains the text $3x^4+x^2+x+3$. The "Result 2 (remainder)" field is empty. Below the input fields is a grid of six buttons: "Sum", "Difference", "Multiply", "Divide", "Derivation", and "Integration". The "Sum" button is highlighted with a blue gradient.

Difference:



The screenshot shows the same "Polynomial Calculator" window as above, but with the "Difference" button highlighted. The input fields contain the same values: "Polynomial 1" is x^2+5 , "Polynomial 2" is $3x^4+x-2$, "Result 1 (quotient)" is $-3x^4+x^2-x+7$, and "Result 2 (remainder)" is empty. The "Difference" button is highlighted with a blue gradient.

Multiply:

 — □ ×

Polynomial Calculator

Polynomial 1

Polynomial 2

Result 1 (quotient)

Result 2 (remainder)

Sum

Difference


Multiply

Divide

Derivation

Integration

Divide:

 — □ ×

Polynomial Calculator

Polynomial 1

Polynomial 2

Result 1 (quotient)

Result 2 (remainder)

Sum


Difference

Multiply

Divide

Derivation

Integration



—

□

×

Polynomial Calculator

Polynomial 1

Polynomial 2

Result 1 (quotient)

Result 2 (remainder)

Sum

Difference


Multiply

Divide

Derivation

Integration

Derivation:



—

□

×

Polynomial Calculator

Polynomial 1

Polynomial 2

Result 1 (quotient)

Result 2 (remainder)

Sum

Difference


Multiply

Divide

Derivation

Integration

Integration:

 — □ ×

Polynomial Calculator

Polynomial 1


Polynomial 2

Result 1 (quotient)

Result 2 (remainder)

Sum	Difference
Multiply	Divide
Derivation	Integration

Invalid input:

 — □ ×

Polynomial Calculator

Polynomial 1


Polynomial 2

Result 1 (quotient)

Result 2 (remainder)

Sum	Difference
Multiply	Divide
Derivation	Integration

Message

 Invalid input!

OK

Junit testing:

▼	✓ Default Suite	26 ms
▼	✓ PT2023_30421_Doia_Bogdan_Assignm	26 ms
▼	✓ OperationsTest	26 ms
	✓ DerivationTest1	17 ms
	✓ DerivationTest2	0 ms
	✓ DiffTest1	1 ms
	✓ DiffTest2	1 ms
	✓ DivisionTest1	1 ms
	✓ DivisionTest2	1 ms
	✓ IntegrationTest1	1 ms
	✓ IntegrationTest2	1 ms
	✓ MultiplyTest1	1 ms
	✓ MultiplyTest2	1 ms
	✓ SumTest1	1 ms
	✓ SumTest2	0 ms

6. Conclusions

While working on this assignment I had the opportunity to consolidate my Java and OOP knowledge and also discover new features that IntelliJ and the Java programming language provide the user. It has also taught me how to better manage my time and how to approach a project, things very valuable in any field or circumstance not just programming.

Future development of the polynomial calculator can include adding the option to perform operations on more polynomials at the same time or to take the present result as the future input.

7. Bibliography

1. <https://regex101.com/>.
2. <https://www.geeksforgeeks.org/convert-string-or-string-array-to-hashmap-in-java/>
3. <https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Format-double-Java-printf-example>
4. <https://www.digitalocean.com/community/tutorials/java-convert-string-to-double>
5. <https://netbeans.apache.org/kb/docs/java/gui-functionality.html>