DOCUMENTATION

ASSIGNMENT 1

STUDENT NAME: Orian Teodor Octavian

GROUP:30423\_2

CONTENTS

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

**1. Assignment Objective**

Main Objective:

Design and implement a polynomial calculator with a dedicated graphical interface.

Sub-Objective:

-Define polynomial representation and operations, and integrate them into the interface:

-Define polynomial representation and basic operations such as addition, subtraction, multiplication, division, derivative, and integration.

-Implement graphical user interface (GUI) for user interaction.

-Develop input interfaces for polynomial insertion.

-Incorporate mathematical operations (addition, subtraction, multiplication, division, derivative, integration) into the GUI.

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

Functional Requirements:

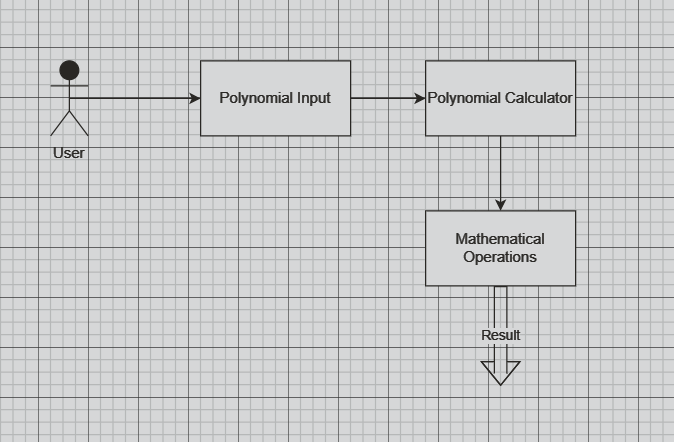
**Polynomial Input:** Users can input polynomials with one variable and integer coefficients.

**Mathematical Operations:** Addition, subtraction, multiplication, division, derivative, and integration of polynomials are supported.

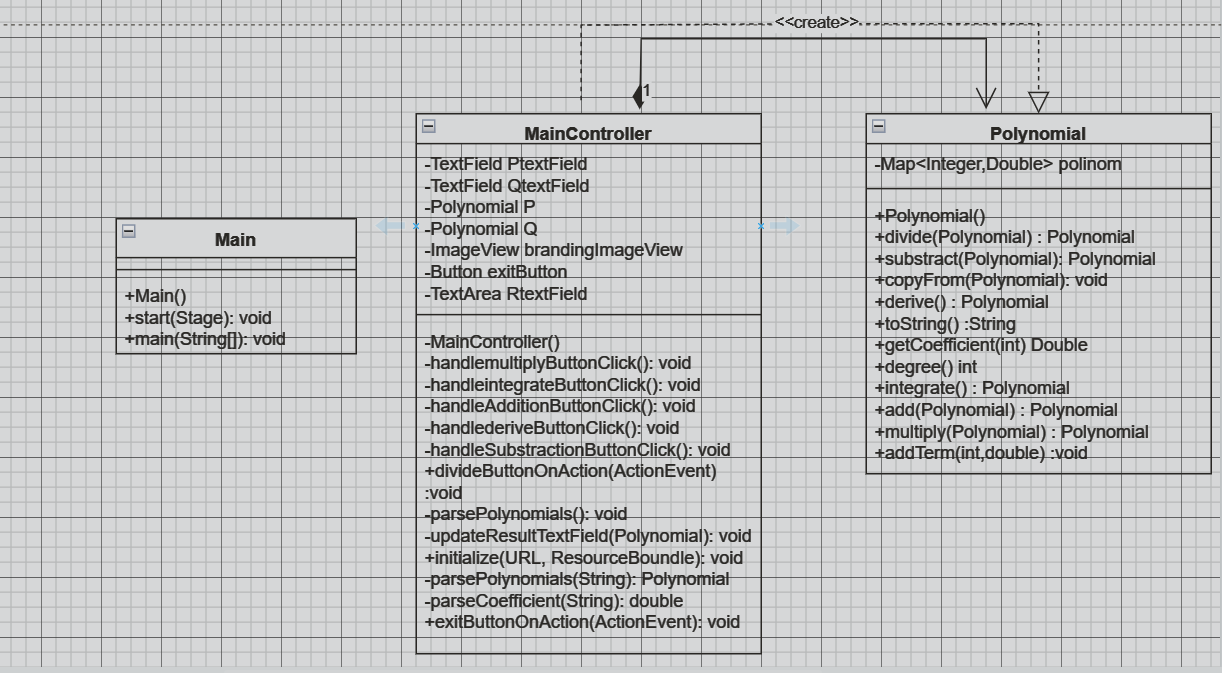
**Graphical User Interface (GUI):** Users interact with a user-friendly GUI for polynomial input and operation selection.

Use Case Description:

**Polynomial Input:** User inputs polynomial, system validates and displays it. **Mathematical Operations:** User selects operation and the system performs operation and displays result.



**3. Designs**



**Main Class (Main.java):**

-Responsible for starting the JavaFX application.

-Loads the FXML file for the main view.

-Initializes the stage and scene.

**Polynomial Class (Polynomial.java):**

-Represents a polynomial and provides methods to perform polynomial operations like addition, subtraction, multiplication, division, differentiation, and integration.

-Uses a TreeMap to store polynomial terms with exponents as keys and coefficients as values.

-Implements various polynomial operations such as addition, subtraction, multiplication, division, differentiation, and integration.

**MainController Class (MainController.java):**

-Acts as the controller for the JavaFX UI.

-Handles user interactions and calls appropriate methods of the Polynomial class to perform polynomial operations.

**Data Structures:**

-TreeMap<Integer, Double>: Used in the Polynomial class to store polynomial terms sorted by exponents.

**Algorithms:**

-Addition, Subtraction, Multiplication, Division, Differentiation, and Integration algorithms are implemented in the Polynomial class to perform respective operations on polynomials.

**4. Implementation**

1.Main (JavaFX Application):

Methods:

-start(Stage stage): Initializes the JavaFX application, loads the FXML file, sets up the stage, and displays it.

-main(String[] args): Entry point of the application, launches the JavaFX application.

2.Polynomial:

Fields: polinom: TreeMap to store polynomial terms with exponents as keys and coefficients as values.

Methods:

-getCoefficient(int exponent) -> Double: Retrieves the coefficient of the term with the specified exponent.

-degree() -> int: Computes the degree of the polynomial.

-addTerm(int exponent, double coefficient): Adds a new term to the polynomial.

-add(Polynomial other) -> Polynomial: Performs polynomial addition.

-subtract(Polynomial other) -> Polynomial: Performs polynomial subtraction.

-multiply(Polynomial other) -> Polynomial: Performs polynomial multiplication.

-derive() -> Polynomial: Computes the derivative of the polynomial.

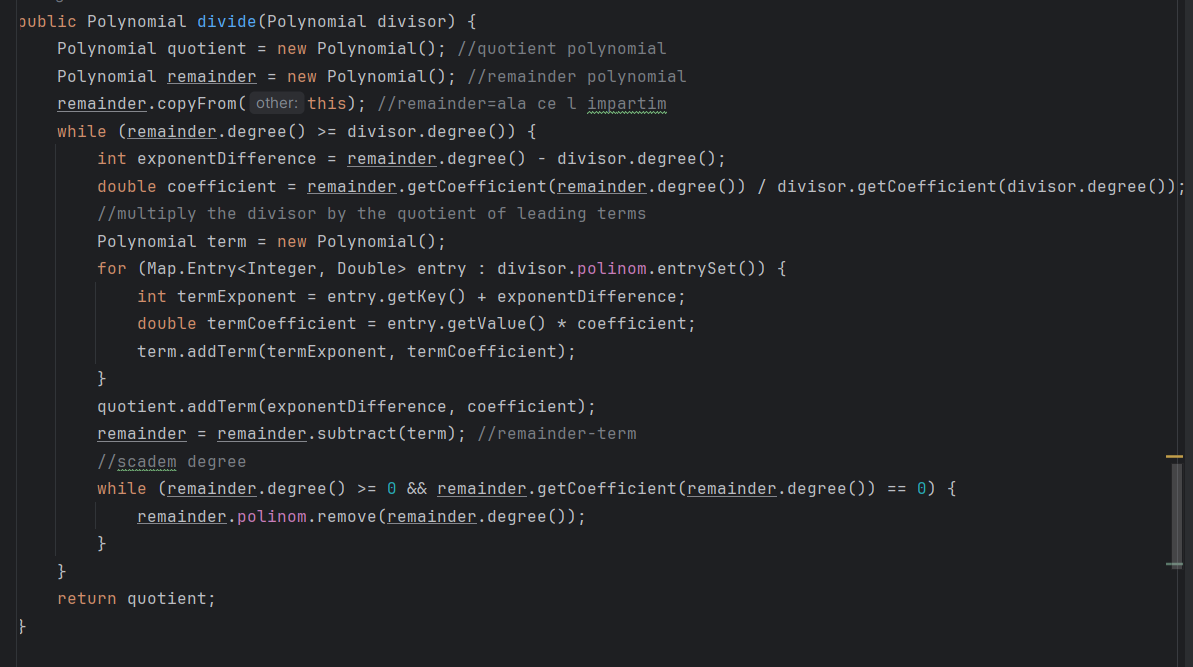
-integrate() -> Polynomial: Computes the integral of the polynomial.

-copyForm(Polynomial other): Copies the terms from another polynomial.

-divide(Polynomial divisor) -> Polynomial: Performs polynomial division.

-toString() -> String: Returns the string representation of the polynomial.

Divide function implementation:



3.MainController (Controller for GUI):

Fields:-Ptextfield: TextField for entering the first polynomial.

-Qtextfield: TextField for entering the second polynomial.

-Rtextfield: TextArea for displaying the result.

-P: Polynomial object representing the first polynomial.

-Q: Polynomial object representing the second polynomial.

Methods:

-initialize(URL url, ResourceBundle resourceBundle): Initializes the controller, sets up the GUI elements.

-handleAdditionButtonClick(): Event handler for addition button, performs polynomial addition.

-handleSubtractionButtonClick(): Event handler for subtraction button, performs polynomial subtraction.

-handleMultiplyButtonClick(): Event handler for multiplication button, performs polynomial multiplication.

-handleDeriveButtonClick(): Event handler for derivative button, computes the derivative of the polynomial.

-handleIntegrateButtonClick(): Event handler for integration button, computes the integral of the polynomial.

-exitButtonOnAction(ActionEvent event): Event handler for exit button, closes the application.

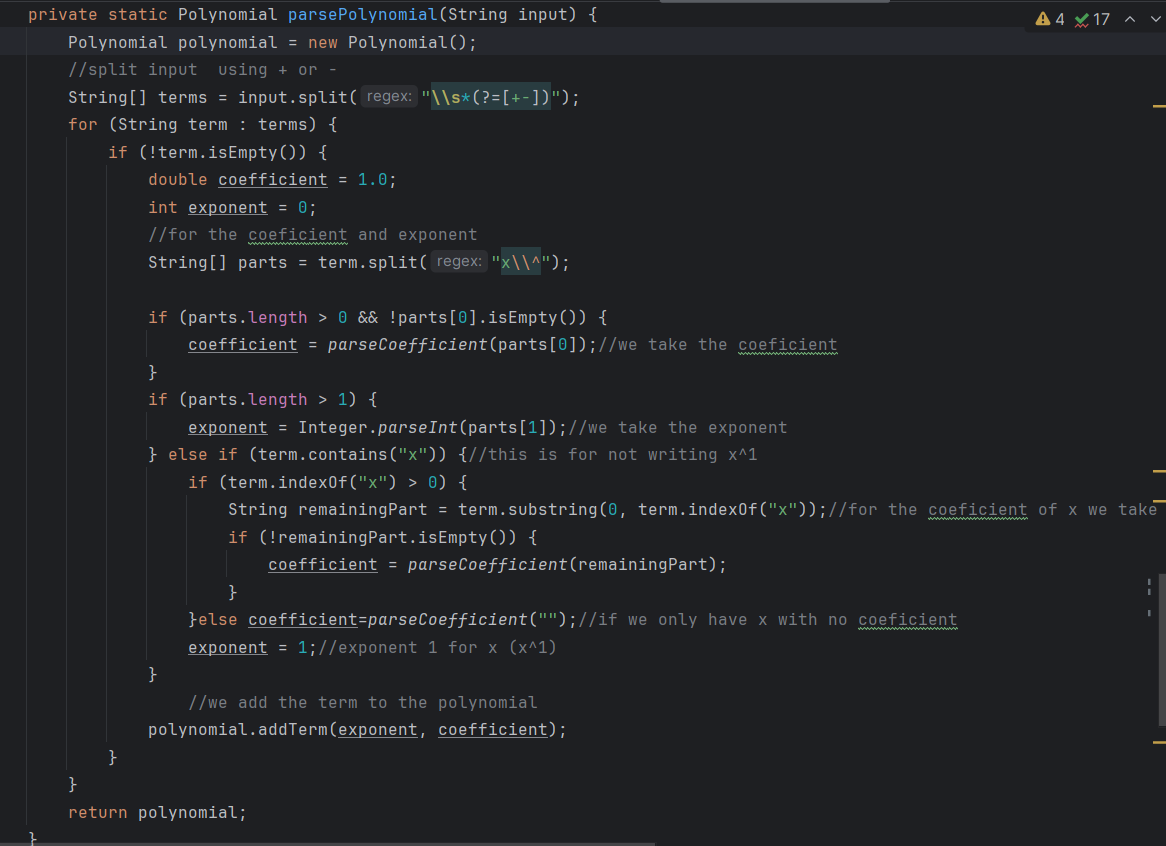
-divideButtonOnAction(ActionEvent event): Event handler for division button, performs polynomial division.

-parsePolynomials(): Parses the polynomial input from text fields.

-parsePolynomial(String input) -> Polynomial: Parses a single polynomial input string.

-parseCoefficient(String coefficient) -> double: Parses the coefficient string into a double.

-updateResultTextField(Polynomial result): Updates the result text field with the computed polynomial result.

Parse Polynomial implementatio , we get rid of the terms with coefficient 0.0, or print x instead of x^1. 

**5. Results**

-Polynomial Addition (testAddTerm()): Ensure proper addition of terms to a polynomial, merging terms with the same exponent.

-Polynomial Addition (testAdd()): Add two polynomials and verify the correctness of the sum.

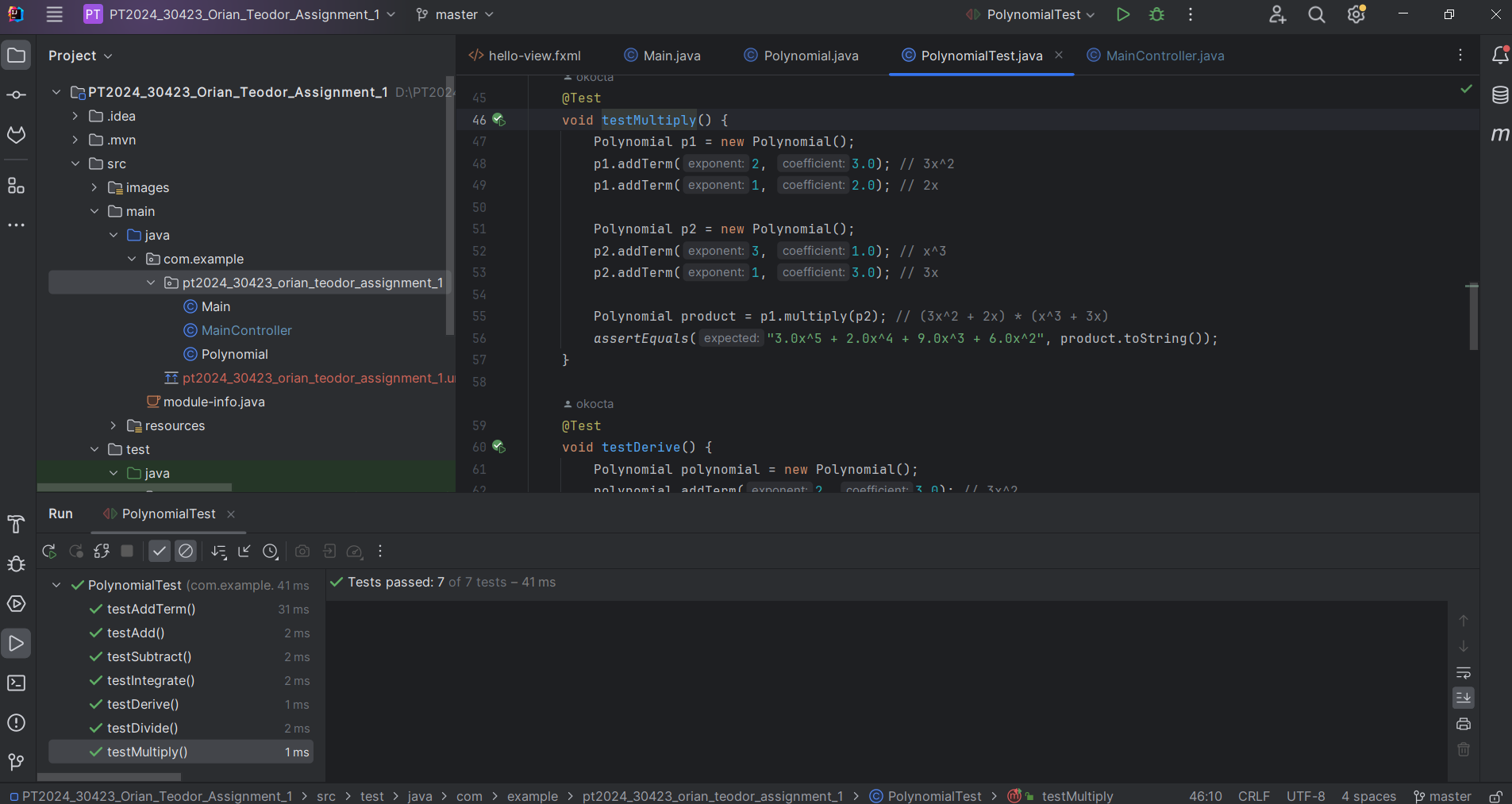
-Polynomial Subtraction (testSubtract()): Subtract one polynomial from another and validate the result.

-Polynomial Multiplication (testMultiply()): Multiply two polynomials and check the correctness of the product.

-Polynomial Differentiation (testDerive()): Compute the derivative of a polynomial and verify the result.

-Polynomial Integration (testIntegrate()): Compute the indefinite integral of a polynomial and verify the result.

-Polynomial Division (testDivide()): Divide one polynomial by another and validate the quotient.



**6. Conclusions**

*  Conclusion: The polynomial application supports addition, subtraction, multiplication, division, differentiation, and integration, all functioning as intended based on testing.

What have I learned: Class handling, Polynomial Operations (didn’t study them in HighSchool), using JavaFX to develop GUI’s, JUnit Testing (didn’t use them before).

Future Developments: better looking UI and handle the wrong inputs better, optimize the algorithms.

**7. Bibliography**

<https://www.jackrutorial.com/2020/04/how-to-highlight-text-in-javafx-textarea.html>

<https://www.geeksforgeeks.org/treemap-in-java/>

<https://www.cuemath.com/algebra/polynomials/>

<https://stackoverflow.com/questions/24260019/javafx-image-not-showing-in-stage>

<https://socratic.org/calculus/introduction-to-integration-/integrals-of-polynomial-functions>

<https://math.libretexts.org/Bookshelves/Algebra/Beginning_Algebra/05%3A_Polynomials_and_Their_Operations/5.04%3A_Multiplying_Polynomials>

<https://courses.lumenlearning.com/waymakercollegealgebra/chapter/polynomial-long-division/>

https://chat.openai.com/