DOCUMENTATION

ASSIGNMENT *1*

STUDENT NAME: Gengiu Robert - Constantin
GROUP: 30421

CONTENTS

1. Assignment Objective 3

2. Problem Analysis, Modeling, Scenarios, Use Cases 4

3. Design 4

4. Implementation 5

5. Results 6

6. Conclusions 6

7. Bibliography 6

1. Main Objective

Develop a polynomial calculator application to perform various operations on polynomials, including addition, subtraction, multiplication, division, differentiation, and integration.

Problem Analysis and Modeling (Section 2):

I) Identify the functional requirements of the polynomial calculator.

II) Develop use case diagrams and describe use cases for the identified requirements.

III) Analyze scenarios and create a detailed problem analysis.

Design (Section 3):

I) Design the object-oriented structure of the application.

II) Create UML package and class diagrams to illustrate the relationships and interactions between classes.

III) Define the necessary interfaces and algorithms for polynomial operations.

Implementation (Section 4):

I) Implement the polynomial calculator using Java.

II) Describe each class, including fields and important methods.

III) Implement the graphical user interface for user interaction.

Testing and Results (Section 5):

I) Define testing scenarios for each polynomial operation.

II) Integrate JUnit tests to verify the correctness of implemented functionalities.

III) Present the results of the testing phase, including successful and failed test cases.

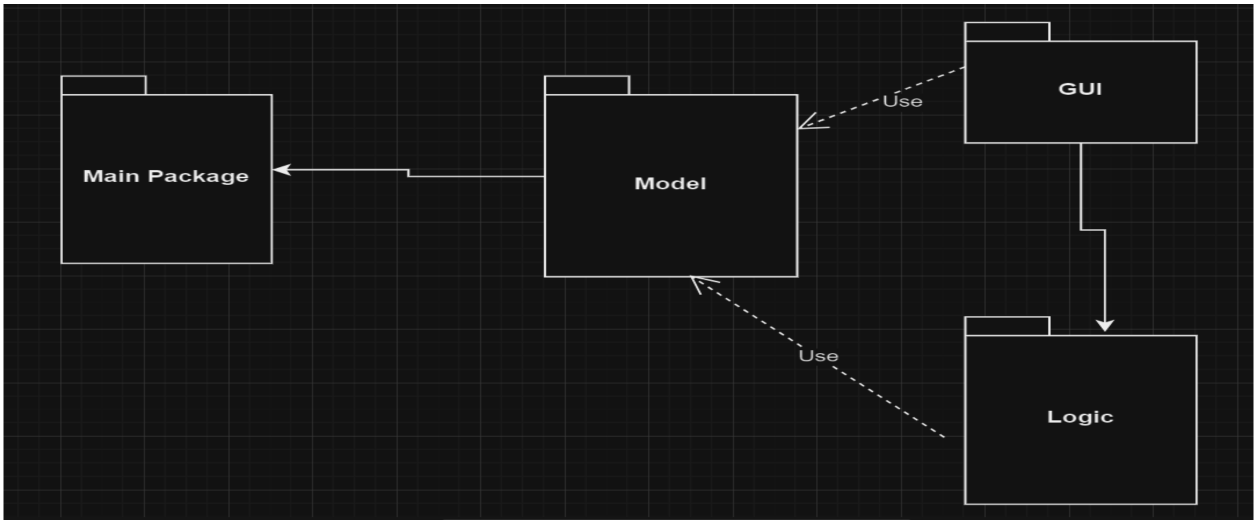
2. Problem Analysis, Modeling, Scenarios, Use Cases

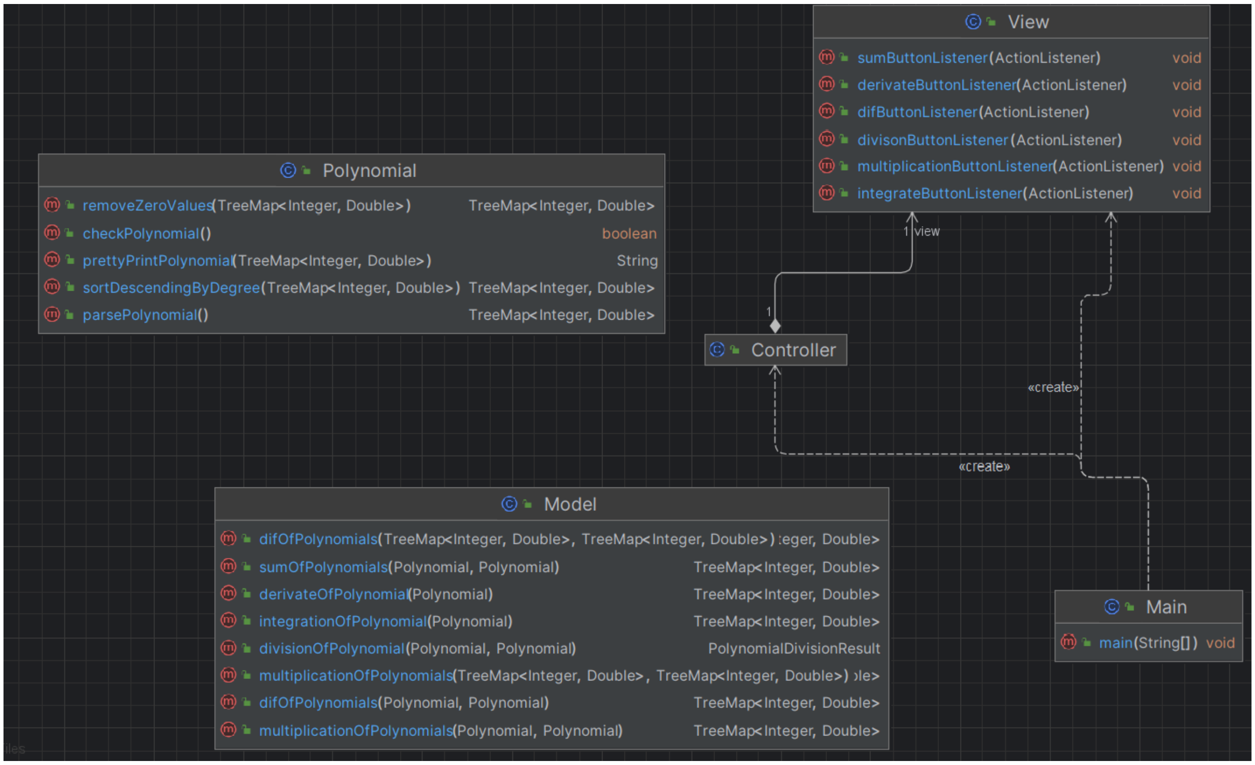
The use of the app resumes in introducing two polynomials in the two TextFields that are going to pop up when you run the app.
The user introduces the two polynomials in the respective fields for each other and after that selects one of the six operations based on them: sum, difference, multiplication, division, derivation, integration. Based on the way the user introduced the polynomials, the operation selected may compute or not. If the operation succeeded in computing, the result will be shown on the TextField that is related to it. If the operation fails to compute, the user will see some error messages on the screen related to the error that the app encounters. Some examples may be: "More input needed", "Use only x as variable".

3. Design

The design part follows a very simple method called MVC (Model,View,Controller). In the Model part, are present the six operations that occur on the polynomials, everyone being represented by a final method. In the View part are found all the things related to the GUI of the app, like TextFields, Labels, Buttons or the principal frame. The Controller part represents the tool that makes the connection between the logic of the app and the frame via the buttons which are very suggestive for each operation related to polynomials. The data structure chosen is represented by the TreeMap. Via TreeMap, all the monomials that are representing a polynomial are stored, forming a key -> value set that is represented by the degree of a monomial and its coefficient.

Below are exposed the UML Package Diagram and UML Class Diagram




View
sumButtonListener(ActionListener) void
derivateButtonListener(ActionListener) void
difButtonListener(ActionListener) void
divisonButtonListener(ActionListener) void
multiplicationButtonListener(ActionListener) void
integrateButtonListener(ActionListener) void
Polynomial
removeZeroValues(TreeMap<Integer, Double>) TreeMap<Integer, Double>
checkPolynomial() boolean
prettyPrintPolynomial(TreeMap<Integer, Double>) String
sortDescendingByDegree(TreeMap<Integer, Double>) TreeMap<Integer, Double>
parsePolynomial() TreeMap<Integer, Double>
1
view
1
Controller
«create»
«create»
Model
difOfPolynomials(TreeMap<Integer, Double>, TreeMap<Integer, Double>) :eger, Double>
sumOfPolynomials(Polynomial, Polynomial) TreeMap<Integer, Double>
derivateOfPolynomial(Polynomial) TreeMap<Integer, Double>
integrationOfPolynomial(Polynomial) TreeMap<Integer, Double>
divisionOfPolynomial(Polynomial, Polynomial) PolynomialDivisionResult
multiplicationOfPolynomials(TreeMap<Integer, Double>, TreeMap<Integer, Double>) ›le>
difOfPolynomials(Polynomial, Polynomial) TreeMap<Integer, Double>
multiplicationOfPolynomials(Polynomial, Polynomial) TreeMap<Integer, Double>
Main
main(String[]) void

4. Implementation

Main Class -> the place where all it the View and the Controller are linked, has a View object and a Controller object.
Controller Class -> the constructor contains a View object and an action listener for every button related for every operation. Also, there is an inner class for every button that controls the operation, implementing ActionListener interface and overriding the actionPerformed method.

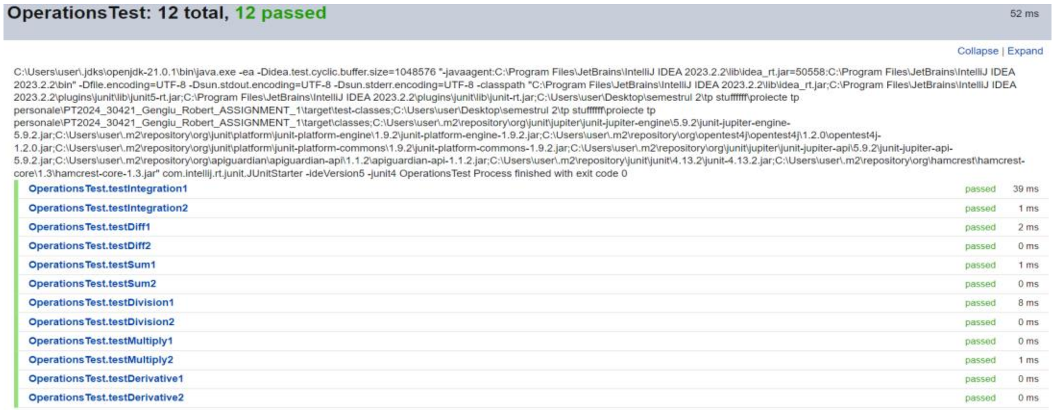
View Class -> extends the JFrame class. The constructor contains all the GUI objects that are used to implement the interface (JPanel, JTextField, JLabel, JButton). Regarding the methods, they have a suggestive name for every operation, an ActionListener parameter that is used to make the buttons work.

Polynomial Class -> the constructor has a String parameter which represents the polynomial as a String. Regarding the most important methods, we found: parsePolynomial, has a TreeMap argument and transforms the String into a TreeMap<Integer,Double> which holds every monomial, key = degree and value = coefficient; sortDescendingByDegree which has a TreeMap as argument and sorts it descending by the key to each set which represents the degree of every monomial; removeZeroValues which has a TreeMap argument and eliminates all the sets which have the coefficient = value equal to 0.

Model Class ->has an empty constructor and contains a method for every operation related to the polynomials. Every operation is implemented in a suggestive and easy way to understand, nothing complicated.

5. Results

For the testing part, JUnit is used. Every operation has two tests, whose results are shown below



```
OperationsTest: 12 total, 12 passed                                                    52 ms

                                                                          Collapse | Expand

C:\Users\user\.jdks\openjdk-21.0.1\bin\java.exe -ea -Didea.test.cyclic.buffer.size=1048576 "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.2\lib\idea_rt.jar=50558:C:\Program Files\JetBrains\IntelliJ IDEA
2023.2.2\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.2\lib\idea_rt.jar;C:\Program Files\JetBrains\IntelliJ IDEA
2023.2.2\plugins\junit\lib\junit5-rt.jar;C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.2\plugins\junit\lib\junit-rt.jar;C:\Users\user\Desktop\semestrul 2\tp stufffff\proiecte tp
personale\PT2024_30421_Gengiu_Robert_ASSIGNMENT_1\target\test-classes;C:\Users\user\Desktop\semestrul 2\tp stufffff\proiecte tp
personale\PT2024_30421_Gengiu_Robert_ASSIGNMENT_1\target\classes;C:\Users\user\.m2\repository\org\junit\jupiter\junit-jupiter-engine\5.9.2\junit-jupiter-engine-
5.9.2.jar;C:\Users\user\.m2\repository\org\junit\platform\junit-platform-engine\1.9.2\junit-platform-engine-1.9.2.jar;C:\Users\user\.m2\repository\org\opentest4j\opentest4j\1.2.0\opentest4j-
1.2.0.jar;C:\Users\user\.m2\repository\org\junit\platform\junit-platform-commons\1.9.2\junit-platform-commons-1.9.2.jar;C:\Users\user\.m2\repository\org\junit\jupiter\junit-jupiter-api\5.9.2\junit-jupiter-api-
5.9.2.jar;C:\Users\user\.m2\repository\org\apiguardian\apiguardian-api\1.1.2\apiguardian-api-1.1.2.jar;C:\Users\user\.m2\repository\junit\junit\4.13.2\junit-4.13.2.jar;C:\Users\user\.m2\repository\org\hamcrest\hamcrest-
core\1.3\hamcrest-core-1.3.jar" com.intellij.rt.junit.JUnitStarter -ideVersion5 -junit4 OperationsTest Process finished with exit code 0
  OperationsTest.testIntegration1                                               passed   39 ms
  OperationsTest.testIntegration2                                               passed    1 ms
  OperationsTest.testDiff1                                                      passed    2 ms
  OperationsTest.testDiff2                                                      passed    0 ms
  OperationsTest.testSum1                                                       passed    1 ms
  OperationsTest.testSum2                                                       passed    0 ms
  OperationsTest.testDivision1                                                  passed    8 ms
  OperationsTest.testDivision2                                                  passed    0 ms
  OperationsTest.testMultiply1                                                  passed    0 ms
  OperationsTest.testMultiply2                                                  passed    1 ms
  OperationsTest.testDerivative1                                                passed    0 ms
  OperationsTest.testDerivative2                                                passed    0 ms
```

6. Conclusions

In conclusion, within this assignment, we have developed a polynomial calculator in Java capable of performing operations such as addition, subtraction, multiplication, division, differentiation, and integration. We have learned about the problem analysis process, design, and software implementation in the context of mathematics and object-oriented programming.

For the future, there is the possibility of extending the functionalities of the polynomial calculator by adding additional features, such as support for polynomials with multiple variables, implementing a polynomial evaluation system, optimizations of existing algorithms, and improvements to the user interface to make the application more intuitive and user-friendly.

7. Bibliography

1. *Bruce Eckel, Thinking in Java (4th Edition), Publisher: Prentice Hall PTRUpper Saddle River, NJUnited States, ISBN:978-0-13-187248-6 Published:01 December 2005.*
2. *What are Java classes? - www.tutorialspoint.com*

3. *Java GUI Full Course - https://www.youtube.com/watch?v=Kmgo00avvEw*
4. *PT site - https://dsrl.eu/courses/pt/*
5. *Reggex for polynomials - https://topic.alibabacloud.com/a/use-a-regular-expression-regex-to-match-polynomials-polynomial-regexpolynomial_1_31_32668196.html*