Documentation

Assignment 1

Cot Mihaita Marian

Group: 30421

1. Assignment objective
2. Problem analysing, Modeling
3. Design
4. Implementation
5. Results
6. Conclusions
7. Bibliography

Assignment Objective

The objective of this assignment is to create a working polynomial calculator, using the data received from the assignment sheet. This calculator should be able to perform the following operations: addition, subtraction, multiplication, division, integration and division.

The software development process consists of understanding the requirements which is part of the initial sequence.

Constructing the software is the second section of this approach and consist of: the design, which is related to thinking and choosing appropriate strategies for implementing all the requirements; the unit-coding and testing which refers to the actual implementation of the code, the coding style applied which is chosen as an oop approach.

Integration refers to constructing the functionality of the system by the usage of all the components created, this approach thinks of every component being separate and used for the main structure. Last but not least it refers to testing the entire system and putting it to work-

The following represent the sub-objectives:

Table

Description automatically generated

# Problem Analysis, Modeling, Scenarios, Use Cases

Chart of the system:

Diagram

Description automatically generated

The chart from above shows how the problem works.

At first to describe the problem: it reduces to an oop approach where every operation is a separate class, every polynomial is parsed and send towards the calculator model, which in it’s turn takes the inputs from the controller meaning it works like a condition checker for the button pressing.

The model then gives the results to the calculator views which then displays everything on screen.

# Design

A screenshot of a computer

Description automatically generated with low confidence

On the right side of the screen I put a picture of

the organization of the program which is sectioned as

it follows: controller, model, view, operations and the

main app.

The whole project is designed using an oop strategy

where all operations are contained within a separate

class, the polynomial functions are taken from within

the view file which passes them towards the polynomial

class, parsing the strings and creating the necessary

objects. The controller is used to access the required

operation after pressing the buttons. The operation is accessed as an object using as parameters the two polynomials gives / or the first one in case of the integration and derivation.

The gui is implemented using swing and has a simple design:

Graphical user interface

Description automatically generated with medium confidence

Implementation

App: main class which creates the gui, controller and model for the calculator

Polynomial: class used to parse the strings who come from the gui, a regex is used to parse the strings and create the monoms using a map of integer, double type, the key is represented by the exponent and value by the coefficient. The methods presented are one to return the string, one to return the map, one to return the degree of the polynomial.

Operations:

1. Addition: receives two polynomials and the constructor creates the addition already, by key search using for each on both the polynomials and the extra components are added to the polynomial-
2. Subtraction: receives two polynomials and the constructor creates the subtraction instantly by key searching, the extra terms are either subtracted by 0 if on first polynomial or reverse the sign if on second
3. Multiplication: receives two polynomials and the constructor uses two nested for each structures to multiply the maps term by term
4. Division: gets two polynomials, checks if the second one is 0, if it is throws exception, if not then passes on to the algorithm used for division, the polynomial short division algorithm, in the end the class has two methods for returning both the q and r
5. Integration: uses the first text field and a for each structure to integrate each monom separately, then it adds at the end the constant “c”
6. Derivation: uses the first text firld and a for each structure to derivate each monom separately

Gui: used javaswing and manually placed all the elements inside the frame, the views contains all details for design, the controller had the on click events and the model has the operations used by the click events

# Results

The results are tested using the Junit installed by the maven project, every operation has a set of separate and fitting tests, which show the correctness of the system. The files can be found in the tests directory.

# Conclusions

This assignment, thought me a better way to “think oop”, as well as making gui by hand and working with maps. Until this point I’ve yet to use maps and working with them has been a pleasant experience. I’m not a big fan of java but this assignment was pretty fun and a good time investment considering the things I was able to learn

As for future development could create a better gui and sort out some errors on the division process, also could slightly modify the regex

# Bibliography

1.Wiki polynomial long division: <https://en.wikipedia.org/wiki/Polynomial_long_division>

2. Geeks for Geeks, for tips on maps:

<https://www.geeksforgeeks.org>

3. Stack Overflow, place you can get answer to any problem:

<https://stackoverflow.com>