

Technical University of Cluj-Napoca

Laboratory Work – Assignment 1
Polynomial Processing

Name: Florea Razvan

Group ID: 30425

Table of Contents

1. Introduction.....	page 3
1.1 Task objective	page 3
1.2 Personal Approach	page 3
2. Problem Description	page 3
2.1 Problem Analysis	page 3
2.2 Modeling.....	page 3
2.3 Scenarios	page 4
3. Projection	page 4
3.1 UML Diagrams	page 4
3.1.1 Use Case Diagram	page 4
3.1.2 Class Diagram	page 5
3.2 Data Structures	page 6
3.3 Class Projection	page 6
3.4 User Interface	page 8
4. Implementing and Testing	page 9
5. Results	page 9
6. Conclusions	page 10
7. Bibliography	page 10

1. Introduction

1.1 Task objective

The objective of this task is rather permissive and is defined as follows: “Propose, design and implement a system for polynomial processing. Consider the polynomials of one variable and integer coefficients”. Due to this reason, the students have full flexibility concerning the implementation and use of resources offered by the Java programming language. It is up to everyone’s decision rather to develop the algorithms in a particular way or another. Same for using a Graphic User Interface or not, or modeling in a specific way or not.

1.2 Personal approach

Personally I developed a Java application for processing polynomials where the user can have full control over the input data, operations he or she wants to perform and I also provided a user-friendly Graphic User Interface. The most used arithmetical operations which involve one or two polynomials have been implemented, and also other helpful functions for the user like clearing the screen or changing the polynomial that is supposed to be the second operand .

2. Problem Description

2.1 Problem analysis

The problem domain can be divided in smaller problems. In this way we only have to deal with problems of a smaller complexity that together form the main problem, the main objective of our application.

The problem can be seen from different perspectives and different points of view. The modeling of the polynomial itself as an entity recognized as a structure, as an independent class, is very important because it is the core of almost every operation and every function that we want our application to perform. The user must not feel neglected and therefore the graphic user interface must not be forgotten and even more, it should be user friendly and efficient. The functionality is the back end “thinking process” of our application and is the most important aspect that we have to take into consideration.

2.2 Modeling

Problem modeling is done by means of object oriented programming and establishing classes, objects and relationships between the components of our problem.

2.3 Scenarios

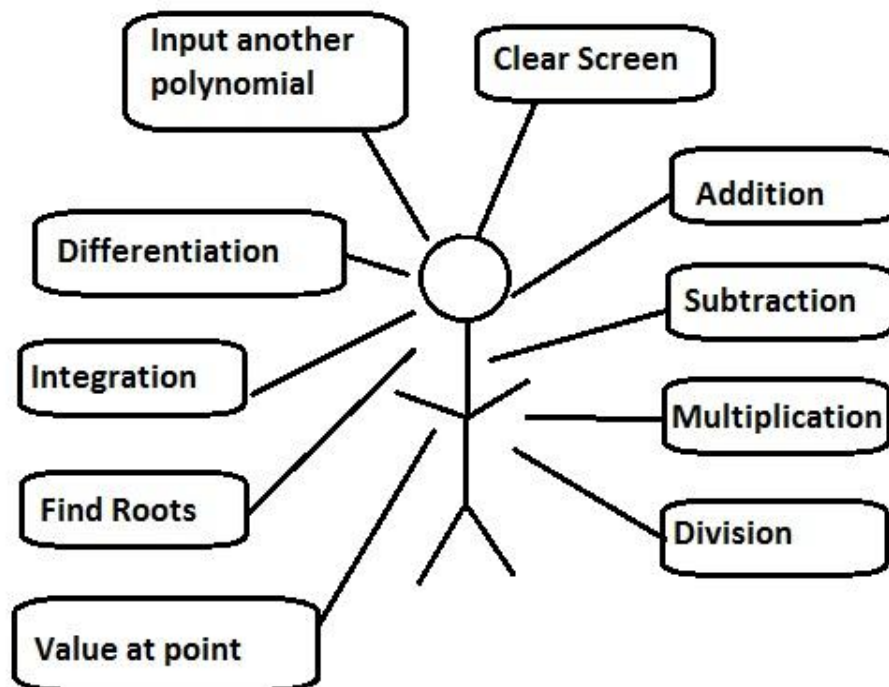
A variety of scenarios have to be taken into consideration. That is, we must be able to predict what the user wants to do once the application is finished and ready to be used, so it should behave properly of any types of input and any operation that he or she may want to perform.

That's why I implemented a variety of user options and operations in this applications, such as: polynomial addition, polynomial subtraction, polynomial multiplication, polynomial division, polynomial differentiation, polynomial integration, finding the roots of a given polynomial, computing the value of a polynomial in a certain point specified by the user, clearing data from the output screen for readability, inserting a different polynomial as the second operand whenever we want, and also others features.

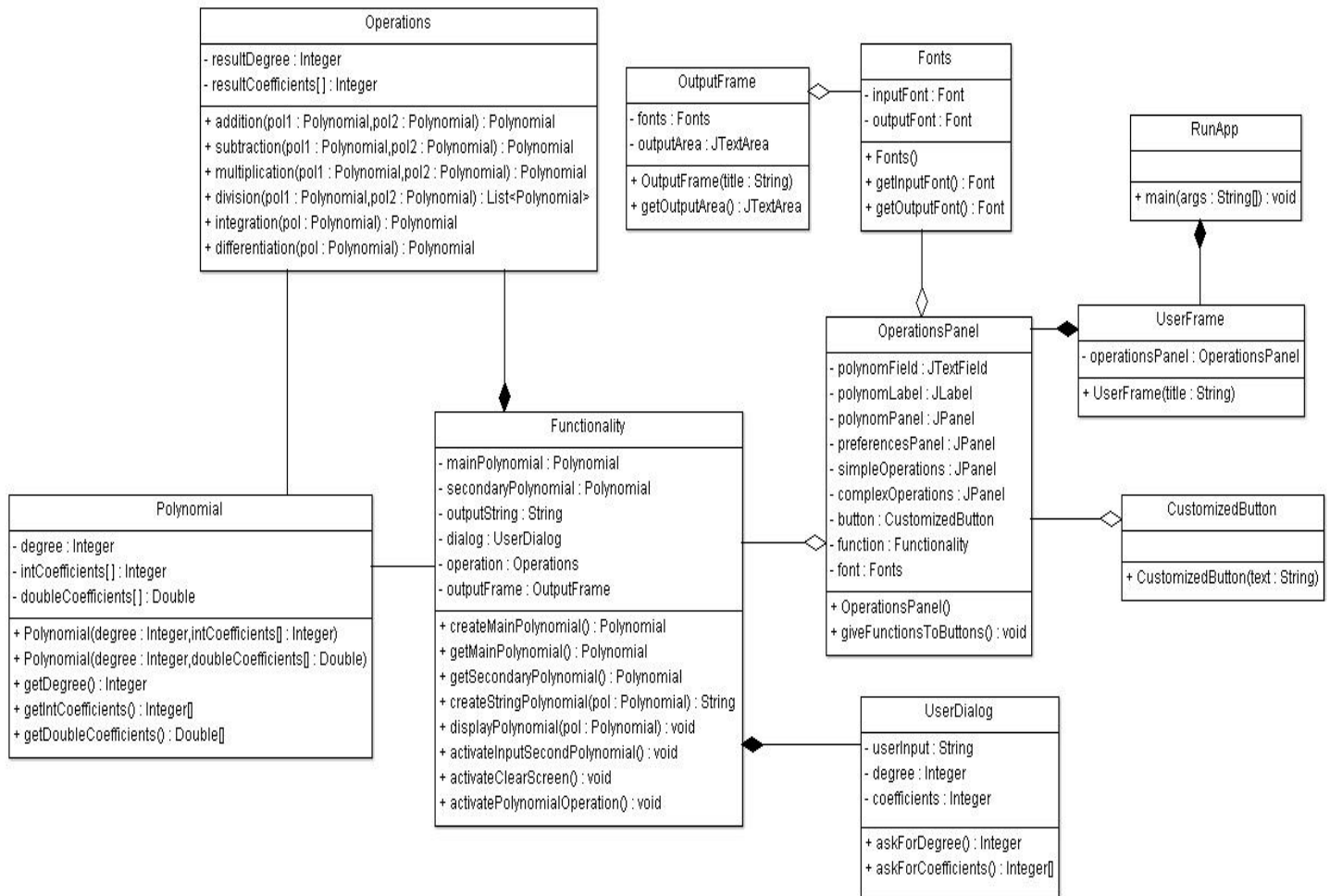
3. Projection

3.1 UML Diagrams

3.1.1 Use Case Diagram



3.1.2 Class Diagram



The diagrams show how the problem is divided in smaller problems and how the modeling was done. The classes communicate between each other and therefore different types of relationships are established.

There are associations between the classes sharing the Polynomial model, aggregations between the classes that use mutual objects, compositions between classes that are absolutely necessary for other classes to work properly and extensions from classes provided by the java language(for example I extended th JButton class to create the CustomizedButton class, I extended the JFrame class in both cases when I created the UserFrame and then the OutputFrame).

For the class diagrams I have used the ArgoUML environment.

3.2 Data Structures

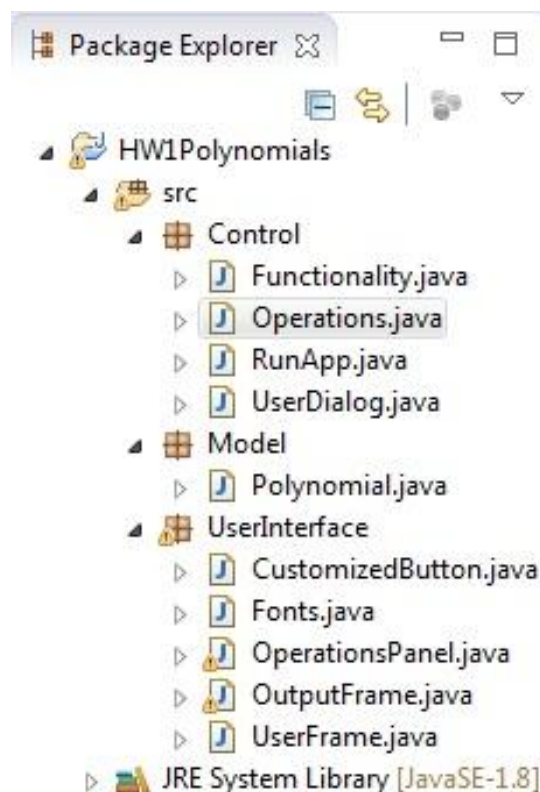
The data structures used are the primitive data types provided by the Java language such as String, integer, double, boolean, font, or other data types such as the customized button, JLabels, JTextFields, JTextAreas, fonts, JFrames.

I also used the List, respectively the ArrayList which I created in order to store and transmit polynomials, so I made them of type Polynomial(ArrayList<Polynomial>). It is useful to have this list because for example in the case of the division of two polynomials, often two polynomials will have to be returned and by means of this List, it is easier to implement this behavior.

3.3 Class Projection

The modeling and partitioning of the problem domain is done by means of dividing the problem and data types needed into several classes from which I instantiate objects of that type and use it in another class. This connection between object of one class and methods of another class is done by means of relationships.

These classes are separated regarding their purpose into different packages. These packages are as follows:



1. Control Package:

The Control package contains the “thinking” done by the applications. That is the functionality of the application and the mathematical operations as well as the management of the user input and user dialogs.

It contains the following classes:

- Functionality.java;
- Operations.java;
- RunApp.java;
- UserDialog.java

2. Model Package:

The Model package contains the class responsible for modeling the polynomials and making a recognizable data structure for the application to work with. The class contained by it is the Polynomial.java class.

3. UserInterface Package:

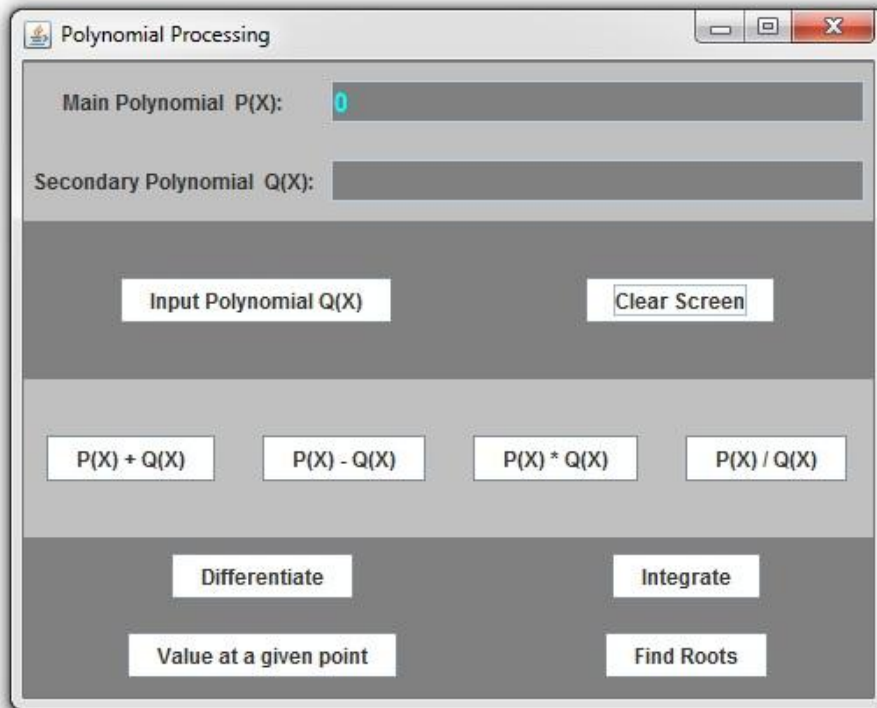
The UserInterface package contains all the classes responsible for managing a nice display for the user and creating the graphic user interface. The classes that this package contains are the following:

- UserFrame.java;
- OutputFrame.java;
- OperationsPanel.java;
- CustomizedButton.java;
- Fonts.java

The packages communicate and everything work together in order to satisfy the user and all of his or her requirements. The algorithms are quite simple due to the good managements of resources, data available and modeling the problem domain.

3.4 User Interface

The user interface that I created for this application is the following one:



The frame is divided in four panels for a more pleasant arrangement of the elements in the frame. The first one is the Polynomial Panel, followed by the Preferences Panel, the arithmetical operations panel and then the “complex” operations panel.

I have used two JLabels , two JTextFields, and ten CustomizedButtons (CustomizedButtons is a class that extends the JButton class).

Mainly I have used javax.swing and java.awt Libraries for creating all the elements for the graphic user interface. Different colors and different fonts have been used for a much nicer appearance and also for making the interface more suggestive.

4. Implementing and Testing

Implementation of the algorithms that I have used and modeling the problem was done by means of the Java programming language and the Integrated Development Environment (IDE) that I used is Eclipse. Even though I have not tested the application on another IDE, it should maintain its portability.

I have covered all the possible cases for user input for the dialogs with the one using the application so that no errors should occur to stop the running time of the applications. For this I used a lot of try - catch blocks to handle different exceptions that may occur.

First I have tested on paper all the algorithms for performing operations and then I run multiple tests on a variety of different possible cases that an actual user might introduce or want to compute.

5. Results

The Application is user friendly and very helpful in computing polynomials and different operations on polynomials. Being developed in the Java programming language the final product is highly portable and is able to run on different operating systems as long a java development kit is installed. The application is very straightforward and instructions are always given to the user whenever some wrong data is send as an input and is considered as not being valid.

6. Conclusions

This application meets all of its requirements but it can also be enhanced and developed even more. I have learned some new things and also I increased my java knowledge and experience with this programming language.

7. Bibliography

https://en.wikipedia.org/wiki/Polynomial_greatest_common_divisor#Euclidean_division
<http://stackoverflow.com/>