CS.TUCN / JAVA LABS

Java project:
Polynomial Calculator^{alfa}

Group 3024
Stas SUSHKOV
stas@net.utcluj.ro
http://stas.nerd.ro

Project homepage
http://code.nerd.ro/jpolcalc.git

March 2009, Cluj-Napoca, TUCN, Romania

Description

Current project had the meaning to teach and familiarize student with Java programming language development tools and technologies.

On building this project it was asked to follow the following requirements:
- design the UML of the application
- write a good description of the required classes with all the attributes and operations it must contain
- implement the project in Java
- use Java Swing technologies to design a GUI
- comment your code in order to generate a comprehensive javadoc documentation
- test your code and provide data for testing and results for your tests
- write a document explaining everything (used ideas, implementation, ideas, libraries). (← this document)
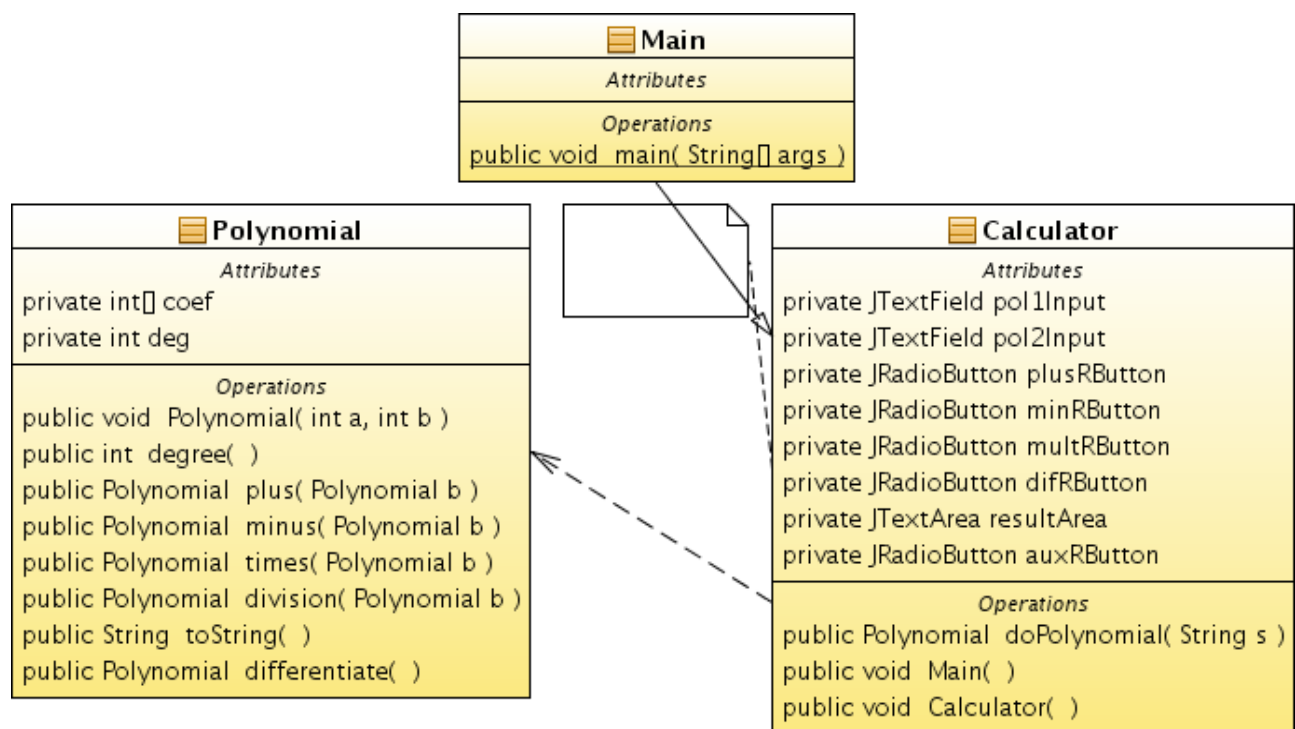
The project theme and features

The theme of the project was to implement a polynomial calculator able to calculate for two polynomials the following operations:

- polynomials summing
- polynomials decrease
- polynomials multiplication
- polynomials division
- one of the additional tasks:
  - polynomial derivative
  - polynomial integral
  - polynomial plot

The UML design

Below is the UML diagram of the project. You can find it inside the distribution package in the *docs/uml* directory.

Main

Attributes
Operations
public void main(String[] args)

Polynomial

Attributes
private int[] coef
private int deg

Operations
public void Polynomial(int a, int b)
public int degree( )
public Polynomial plus(Polynomial b)
public Polynomial minus(Polynomial b)
public Polynomial times(Polynomial b)
public Polynomial division(Polynomial b)
public String toString( )
public Polynomial differentiate( )

Calculator

Attributes
private JTextField pol1Input
private JTextField pol2Input
private JRadioButton plusRButton
private JRadioButton minRButton
private JRadioButton multRButton
private JRadioButton difRButton
private JTextArea resultArea
private JRadioButton auxRButton

Operations
public Polynomial doPolynomial(String s)
public void Main( )
public void Calculator( )

Project implementation

The application uses three classes. Two of those are crucially important, the last one is used just as a launcher for our application.

`Calculator.class`, is the the class used to build the GUI of the application.

It's attributes serve for the inputs and choice selections which are coming directly from the end user:

`pol1Input`	– Input for our input for polinom #1
`pol2Input`	– Input for our input for polinom #2
`plusRButton`	– radio button for summing selection
`minRButton`	– radio button for decrease selection
`multRButton`	– radio button for multiplication selection
`difRButton`	– radio button for division selection
`auxRButton`	– radio button for auxiliary selection
`resultArea`	– for output

It's most important operation is the conversion method:

`doPolynomial`	– converts input (String s) to a Polynomial object
`s`	– is the String input which has to be converted

It's constructor is:

`Calculator()`	– constructor of the class. Builds the whole GUI.
`Main`	– main method, this starts and runs the calculator

`Main()` is attached as an action to the `startButton` listener.

`Polynomial.class`, is the class which offers us the polynomial abstraction object and the methods (operations) we can use with it.

`Polynomial` object is represented as an array for the polynomial coefficients (`int[] coef`), and an integer variable used for defining polynomial grade (`int deg`).

`Polynomial(int a, int b)`	– class constructor. Method returns `a * x^b`
`degree()`	– calculates and returns the degree
`differentiate()`	– differentiates this polynomial and returns it
`division(Polynomial b)`	– calculates and returns (a / b), incomplete yet
`minus(Polynomial b)`	– calculates and returns (a − b)
`plus(Polynomial b)`	– calculates and returns c = a + b
`times(Polynomial b)`	– calculates and returns (a * b)
`toString()`	– converts to string representation

`Main.class`, is the starter for our application. It's method `Main()` calls the `Calculator` constructor to build the whole GUI. It's single method also sets the look and feel of the application to get a better integration cross WM.

Final result and testing

Here's an instance of the running application.
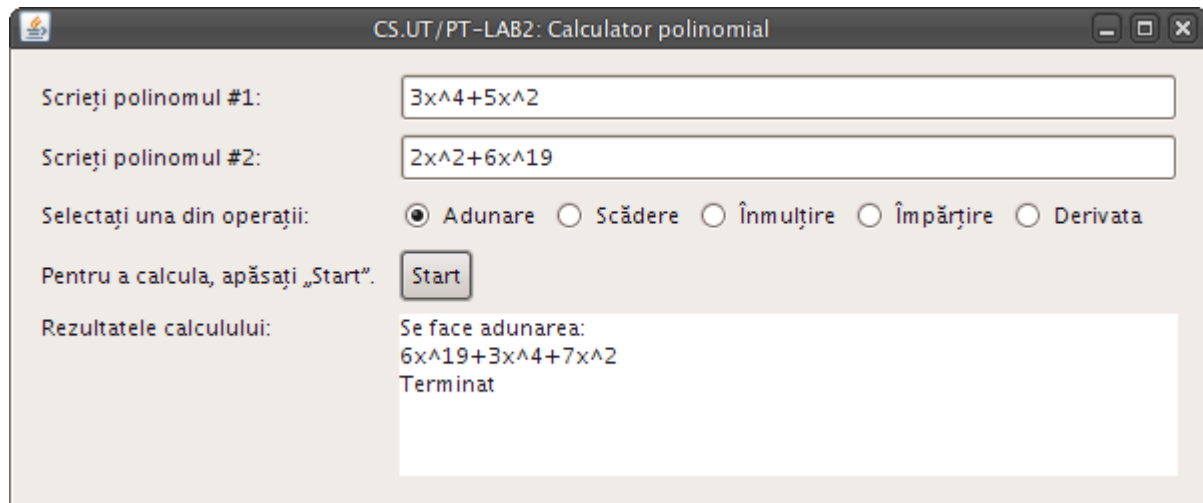
As testing data one can use:

3x^4+5x^2	– as for the first polynomial input
2x^2+6x^19	– as for the second polynomial input

the results should look like this:

6x^19+3x^4+7x^2	– as for summing
-6x^19+3x^4+3x^2	– as for decrease
18x^23+30x^21+6x^6+10x^4	– as for multiplication
4x^19+1x^2+2	– as for division, probably incorrect
12x^3+10x	– as for derivative for the first polynomial
114x^18+4x	– as for derivative for the second polynomial

Final note

The project is currently in alfa, I can't guarantee it's ready for massive stress testing. Project currently is held in my own Git code repository at http://code.nerd.ro/jpolcalc.git so I invite anyone who wants to continue working on it.

Current status: in abandoned alfa.