Authors: Ben Gendler, Roy Mash and Tal Ben Gozi

General:

The project aims to assist the customer to create and modify Monomials and/or Polynomials and in addition implement Mathematical functions to them such as: Area calculation, Intermediate Value Theorem, derivative and more.

The main components of the project are:

Monom.class which intends to create and modify Monomials, an object which has 2 fields (Coefficient, and Exponent).

A Monom can be created either from a string or with these two variables discussed above (double,int)

Polynom.class which intends to create and modify Polynomials, where most of the methods are based on Monom type methods.

Each polynom is implemented via a singly-linked-list whose Nodes.data contain Monom type

The **Linked-List** has Insert,Delete by monom methods as well as an insertion sort implementation based on their monomial Exponent value (High to low).

Polynom_able.class which implements a Polynom.class interface.

Functions_GUI.class which has an ArrayList based, which represents a collection of functions.

The class has a Collection methods in addition to various methods (written below).

ComplexFunction.class which has 3 constructors, and represents a Complex Function which includes an Operation and up to 2 functions. It has an f function which is very important for drawing the function. (Array of X values and Y values whose obtained by f function.

In addition to those noted above, it has some helper functions which are straight forward (Such as GetOpTo/From String) and toString.

Junit.*classes

A variety of Junit 5 tests for the classes.

Notes and function explanations (for those which are not straight forward):

Polynom.Root: Root function finds by recursion the first X value of a function where it's Y value is smaller than Epsilon. it will throw exception when the X values are both positive or negative.

Polynom.Area: Area function calculates the area between a function to axis x from x0 to x1.

Epsilon divides the calculation to epsilon-size areas.

Monom(String) constructor: a method which creates a Monomial from an inputted string, It checks the validity of the string via regex and then uses the helper methods to parse the coefficient and the exponents.

Polynom(String) constructor: a method which creates a Polynomial from an inputted string,

It checks the validity of the string via regex, and then splits the String into substrings which represent Monomial and then calls the Monomial(String) constructor for each of them.

Polynom.subtract(Polynom_able) : A method for subtracting a Polynom_able type from a Polynom. It simply multiplies the Polynom by (-1) Monom and then adds each of the Monos of the Polynom returned.

DrawFunctions(Integers,Ranges): which sets the general Draw parameters and iterating over the collection.

Later on it calls the **DrawFunction** helper method which actually draws each of the functions with randomized color from the Colors array. (Assisted by getRGB helper function which gets the actual values of the color RGB parameters)

DrawFunctions(String json_file):

Gets GUI parameters from JSON file using **Gson** lib. Uses the **SetFromJson** helper setter method.

initFromFile(String file):

Gets function Strings from a txt file, converting those using **FileInputStream and DataInputStream, splits the txt file into lines and using the LineChecker helper method for each of the Lines.**

saveToFile(String file):

Writing all collection data (Using ToString method for each Function from the collection)

Using Files and Paths libs