Project 0:

From:
Bar Goldshtein: 308417989

Hai Hatan: 312368749

The entirety project is to be able to handle the mathematical function of Polynom and Monom.

The function and variable of Monom:

Monom is a mathematical function represent by a*x^b.

int power: save the power of the Monom.

double coefficient: save the coefficient of the Monom.

Public Monom(double a, int b): a constructor that get a double and set it as the coefficient of the Monom than get a int and set it as the power of the Monom.

Public Monom(Monom of): a constructor thet get a Monom and make a deep copy of it.

Public Monom(): A default contractor of Monom, just set a place in the memory to a Monom.

Public Boolean Add(Monom toAdd): a function that get a Monom and add it to the original (this.) Monom.

Public void Multiply(Monom multIn): a function that get a Monom and multiply it
and the original Monom.

Public double f(double x): get a X value and return $f(x)$.

Public Monom derivative(): do a derivative function on the original Monom and return it, the function doesn't change the original Monom.

Public String toString(): "convert" the Monom to String.

Public double get_coefficient(): return the coefficient of the original Monom.

Public int get_power(): return the power of the original Monom.

Public Boolean isEquals(Monom m1): get a Monom and return true if and only if that Monom and the original are equals.

Public void set_coefficient(double a): get a double and set it as the original Monom coefficient.

Public void set_power(int a): get a int and set it as the original Monom power.

The functions of Polynom

Polynom is a collection of Monom in order of power from largest of smallest.

All the Monoms are held in a ArrayList named Mtp.

Public Polynom(): A default contractor of Polynom, just set a place in the memory for the Polynom.

Public Polynom(String Poly): a contractor that get a String and convert it to a Polynom.

Public double f(double x): get a X value and return $f(x)$.

Public void add(Polynom_able p1): get a Polynom and add him to the original Polynom.

Public void add (Monom m1): get a Monom and add if to the original Polynom.

Public void substract(Polynom_able p1): get a polynom and subtract it from the original Polynom.

Public void multiply(Polynom_able p1): get a Polynom and Multiply it with the original Polynom.

Public boolean equals(Polynom_able p1): get a Polynom and return true if and only if the Polynom and the original one are equals.

Public boolean isZero(): return true if the Polynom is equal to 0.

Public double root(double x0, double x1, double eps): get x0 and x1 and return the intercept with the X axle by a offset of eps.

Public Polynom_able copy(): return a deep copy of the original Polynom.

Public Polynom_able derivative(): return a derivative Polynom of the original Polynom, the function doesn't change the original Polynom.

Public double area(double x0, double x1, double eps): get x1 and x0 and return the area between them, the Polynom and the X axle. Via Riemann Integral.

Public String toString(): “convert” the Polynom to String.

Public Iterator<Monom> iterator(): this function will be a pointer to the Mtp ArrayList in the Polynom via Polynom_able p1

Private void multiply(Monom m1): get a Monom and multiply the original Polynom by it, it’s a help function for multiply by Polynom.

Private void cleanUp(): a function that clean all Monom with a 0 coefficient.

Private String [ ] cleanEmptySpaces(String [ ] str): clean all empty spaces in the array of the string a help us use the String constructor.