MyMath

Monom class-

Constructors-
There are 3 constructors in monom class, one is a constructor how gets 2 variables (double variable for coefficient and an integer variable for the power) and defines them. (the power number cannot be a negative number). The second constructor is a copy constructor how gets a monon and copy his coefficient and his power to the other monon.
The third constructor gets a string variable and the first thing is to change all capital letter to a small letter, if there is a "*" in the string its save its location in the string in the variable index_pi, if there is a "x" its saves its location in the variable index_x and if there is a "^" its saves its location in the variable index_pow (if there isn't "x" or "*" or "^" so in each of the variables its saves "-1").
Is there isn't an "x" in the string it means that the string is only a number without a power so its change the string to a double variable, otherwise its checks if there is a "*" character, if there aren't its checks it there is a coefficient to the "x" character then the coefficient will be 1 or if the coefficient is a negative number then the coefficient will be with "-" at the beginning and then change it into a double variable.
If there is a "*" so its cut the string from the beginning to the "*" and checks if its empty, if its empty its change the coefficient to 1 and then change it to a double variable, else its change it directly.
Additionally, its checks if there is a "^" in the string (meaning if there is a power to the number), if there isn't its checks if after the "x" there is a number because if there is it's not allowed, otherwise its checks if the power number is empty or a negative number and if it is, its return an exception, it there is an "x" bet the power is empty its adding 1 to the power, when all the tests are done its change the string into a double variable.

Functions-

1. The f(x) function gets a number and calculate the monon according to the number.

2. The add function gets a monom and if the power of the number is the same power as the other monoms power, the function adds them together.

3. The multiplying function gets a monom and multiply the to two coefficients of the monom and adds its two powers together.

4. The equals function checks between two monoms if its coefficient and its power is equal.

5. The derivative function gets a monom and multiply the coefficient by its power, and subtract from the power 1.

6. The toString function printing the monom.

polynom class-

Constructors-

There are 3 constructors in this class. The first is default constructor, the second is a copy constructor how gets a polynom and copy each of its elements to the other polynom, the last one is a constructor how gets a string and change it into a double variable by split the string by the character "+" or "-" and put it into the array, then its send each of its element to the string constructor in the monom class so at the end we get from the string a double type polynomial.

Functions-

1. The f(x) function gets a number and calculate each of the monoms at the polynom according to the number.

2. There are 2 adding functions, the first gets a monom and adding it to the polynom by checking if there is another monom with the same power, if there is its add them together, otherwise its add the monom at the end.

   The second one gets a polynom and adding the two together by checking each of the monoms in both of them and adding them as required.

3. The subtraction function takes two polynomials and subtracts them from each other

4. The multiply function gets a polynom and multiply each of the monom in each of the polynom by the other one by multiply each of the coefficients of the monom by the other coefficients and adds its two powers together.

5. The derivative function gets a polynom and multiply each of the coefficient by its power, and subtract from the power 1.

6. The root function checks when the polynom cut the X-axis (meaning when the polynom is equal to zero.) so its checks if the function is continuous and then me use the bisection method and cut in half until we getting to x_0.

7. At the area function we uses the Riemann method to calculate the positive area of the function by dividing the area into small squares whose base is epsilon and their length is y (our polynom) and calculate the amount of the squares.

8. We created a new function –negativeArea- because we cannot use the first area function because it cannot calculate the negative area of the function.

9. The equal function gets a polynom and checks between two of the two polynoms if each of theirs monoms the coefficient and its power is equal.

10. The isZero function checks whether the polynomial is the zero polynomial.

Monom_Comperator class-

Here we arrange the polynom according to the power of each monom, each time we get two monoms and compare their power.

Print polynom-

We use Julien code to print the graph of the polynomial.
We added a few function for his code to match our code.

1. We created the Changesyntax function so we can adjust our polynom string to his, the function gets polynom and change it to a string variable by checking for each monom at the polynom a few things:
   if the coefficient of the monom is zero so its add to the string the number zero, otherwise if the power of the monom is zero its check if the coefficient is bigger than zero, if it is bigger its add to the string the character "+" else its add the character "+(" and then change the coefficient to a string variable and add it too. if the power of the coefficient is not zero its check the same but additionally to the "+" or "+(" character its add to the string, the character "X" and the power of the coefficient.

2. The foundMinMax function gets the variables: polynom and 3 doubles (two is for the range in which we will look for extreme points and one for the Epsilon), its sent the polynom to the derivative function and then we go over the derivative in the range by adding to the range epsilon on each test and check if when we set the range in the derivative once just the range number and once we add epsilon to it and if the calculation is smaller than zero and the calculation with the epsilon is bigger than zero or that the calculation without the eps is bigger than zero and the calculation with eps is smaller than zero we found an extreme points and we add it to the points array.

3. To print the graph, we use Julien function and added that for each extreme points we found we would mark it.

Or Shemesh and Yaara Goldenberg.