

## Task number 0

At this task we implement 3 classes:

1. Monom
2. Monom Comparator
3. Polynom

### The class Monom:

This class represents a simple "Monom" of shape  $a \cdot x^b$  while  $a$  is a real number and  $b$  is an integer none negative.

The class implements function and support simple operations as: construction, value at  $x$ , derivative, add and multiply.

The variables of the this class:

```
private double _coefficient;  
private int _power;
```

### The functions:

1. `public double f(double x) {...}`

This function calculates the value of  $Y$  at the point  $X$ .  
The function received parameter  $X$  from the user.  
The function returns the value of  $Y$  at  $X$  point.

2. `public Monom(double a, int b) {...}`

This function initializes the values of Monom's coefficient and power.  
The function received parameters:  $a$ ,  $b$  from the user.

3. `public Monom(Monom ot) {...}`

The copy constructor.  
This function copy the values of the default Monom to the new Monom.  
The function received Monom from the user which called `ot`.

4. `public Monom (String str) {...}`

This function received a String and change it to Monom  
The function received string from the user.  
If the string is invalid, for examples:  $2x^$  ,  $3 \cdot x^{-6}$  the user will get an exception.

5. `public Monom derivative() {...}`

This function calculates the derivative of the Monom and return it.  
If after the derivative the power become negative the user will get an exception.

6. `public void add(Monom m) {...}`

This function adds any Monom that it gets from the user to another Monom. If the powers are not the same the user will get an exception.

7. `public void sub(Monom m) {...}`

This function subtracts any Monom to another Monom. If the powers are not the same the user will get an exception.

8. `Public void multiply(Monom m){...}`

This function multiplies any Monom with another Monom.

9. `public String toString() {...}`

This function returns string of Monom.

10. `public double get_coefficient() {...}`

This function gets the value of the coefficient.

11. `public int get_power() {...}`

This function gets the value of the power.

12. `private void set_coefficient(double a){...}`

This function sets the value of the coefficient.

13. `private void set_power(int p) {...}`

This function sets the value of the power.

## The class Polynom:

This class represents a Polynom with add, multiply functionality, it also should support the following: integral of Riman, finding a numerical value between two values (currently support root only  $f(x)=0$ ), and Derivative.

1. `public double f(double x) {...}`

This function calculates the value of Y at the point X which received by the user.

This function return the value of Y at the point X.

2. `public Polynom () {...}`

The Polynom constructor.

3. `public Polynom_able copy() {...}`

The copy constructor.

The function return new Polynom which received the values of the default Polynom.

4. `public Polynom(String p) {...}`

This function received a string and change it to Polynom.

5. `public void add(Polynom_able p1) {...}`

This function adds any Polynom\_able to another Polynom.

6. `public void add(Monom m1){...}`

This function adds any Monom to Polynom.

After each addition we perform a sort on the polynom.

7. `public void sub(Monom m1){...}`

This function subtracts any Monom from Polynom.

After the operation we perform a sort on the polynom.

8. `public void subtract(Polynom_able p1){...}`

This function subtracts some Polynom\_able from Polynom.

9. `public void multiply(Polynom_able p1){...}`

This function multiplies any Polynom\_able with another Polynom.

```
10. public boolean equals(Polynom_able p1){...}
```

This function checks if the Polynom\_able which received from the user equal to another Polynom\_able.

```
11. public boolean isZero(){...}
```

This function checks if the Polynom contains just zero.  
The function return true or false.

```
12. public double root(double x0, double x1, double eps){...}
```

This function finds the point of intersection with the x axis.  
x0 - the first point which received from the user.  
x1 - the second point which received from the user.  
eps - the point range which received from the user.  
The point of intersection with the x axis.

```
13. public String toString (){...}
```

This function returns string of Polynom.

```
14. public Polynom_able derivative(){...}
```

This function calculates the derivative of the Polynom.

```
15. public double area(double x0, double x1, double eps) {...}
```

This function assuming  $(f(x_0) * f(x_1)) \leq 0$ .

```
16. public Iterator<Monom> iteretor(){...}
```

This function permit use with Iterator.

### [The class Monom\\_Comperator:](#)

```
1. public class Monom_Comperator implements Comparator<Monom> {...}
```

This function compare between two Monoms.

If the first Monom smaller then the second Monom, the function return -1,  
if it bigger it return 1, else they equal ant the function return 0.

Authors : Karin Aharon .Id :312502537.

Mor Danino. Id :205443583.