**OOP Course - Ex0**

In this assignment we were asked to write a class of polynom, and all sorts of important functions to perform on it. Writing the assignment was divided into two classes: Monom and Polynom.

**Authors: Meir Nizri and Avihay Bernholtz**

Monom

The Monom class represents a simple monom of shape ax^b, where a is a real number and b is an integer (summed a none negative), see: <https://en.wikipedia.org/wiki/Monomial>. We defined monom so that there can be spaces everywhere (e,g: 2 x ^ 3), as long as the main structure we described is preserved. Writing capital 'X' instead of 'x' will cause an exception.

There are some special monoms:

* Real number followed by x - the power of the monom is 1.
* Real number only - the power of the monom is 0.

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

* **Field Summary**

| **Fields** | | |
| --- | --- | --- |
| **Modifier and Type** | **Field** | **Description** |
| static java.util.Comparator<[**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html)> | [**\_Comp**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#_Comp) | The Constant \_Comp. |
| static double | [**EPSILON**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#EPSILON) | The Constant EPSILON. |
| static [**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html) | [**MINUS1**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#MINUS1) | The Constant MINUS1. |
| static [**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html) | [**ZERO**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#ZERO) | The Constant ZERO. |

* **Constructor Summary**

| **Constructors** | |
| --- | --- |
| **Constructor** | **Description** |
| [**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#%3Cinit%3E(double,int))​(double a, int b) | Constructs and initializes a monom. a - coefficient, b - power. |
| [**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#%3Cinit%3E(java.lang.String))​(String s) | Constructs and initializes a monom from String. |
| [**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#%3Cinit%3E(myMath.Monom))​([**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html) ot) | Constructs a deep copy of the monom. |

* **Method Summary**

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| void | [**add**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#add(myMath.Monom))​([**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html) m) | Add monom to this monom. |
| [**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html) | [**derivative**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#derivative())() | this method returns the derivative of the monom. |
| boolean | [**equals**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#equals(myMath.Monom))​([**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html) d) | check if equals. |
| double | [**f**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#f(double))​(double x) | This method calculate the value of the monom for given x. |
| double | [**get\_coefficient**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#get_coefficient())() | Gets the coefficient. |
| int | [**get\_power**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#get_power())() | Gets the power. |
| static java.util.Comparator<[**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html)> | [**getComp**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#getComp())() | Gets the comp. |
| boolean | [**isZero**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#isZero())() | Checks if is zero. |
| void | [**multiply**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#multiply(myMath.Monom))​([**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html) d) | Multiply monom to this monom. |
| java.lang.String | [**toString**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html#toString())() | Print the monom to the screen, in format a\_x^\_b. |

**Methods inherited from class java.lang.Object**

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Polynom

This class represents a general Polynom: f(x) = a\_1X^b\_1 + a\_2\*X^b\_2 +...+ a\_n\*Xb\_n, where: a\_1, a\_2 ... a\_n are real numbers and b\_1, b\_2 ... b\_n are integer (summed a none negative). all the elements in the polynom are standard monoms as we described in the monom class. We defined Polynom so that there can be spaces everywhere (e,g: 2x^ 3 – 5 x), as long as the main structure we described is preserved.

This Polynom can get different monoms with equal powers, The constructor will add all monoms of the same power and order all the elements from highest power to the lowest.

Polynom implements function and support simple operations as: construction, value at x, add, subtract, multiply functionality, it also should support the following:

1. Riemann's Integral: <https://en.wikipedia.org/wiki/Riemann_integral>

2. Finding a numerical value between two values (currently support root only f(x)=0).

3. Derivative

* **Constructor Summary**

| **Constructors** | |
| --- | --- |
| **Constructor** | **Description** |
| [**Polynom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#%3Cinit%3E())() | Constructs the zero polynom. |
| [**Polynom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#%3Cinit%3E(java.lang.String))​(String s) | init a Polynom from a String such as: {"x", "3+1.4X^3-34x", "(2x^2-4)\*(-1.2x-7.1)", "(3-3.4x+1)\*((3.1x-1.2)-(3X^2-3.1))"}; |

* **Method Summary**

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| void | [**add**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#add(myMath.Monom))​([**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html) m1) | Add m1 to this Polynom. |
| void | [**add**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#add(myMath.Polynom_able))​([**Polynom\_able**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom_able.html) p1) | Add p1 to this Polynom. |
| double | [**area**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#area(double,double,double))​(double x0, double x1, double eps) | Compute a Riman's integral from x0 to x1 in eps steps. |
| [**Polynom\_able**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom_able.html) | [**copy**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#copy())() | create a deep copy of this Polynom. |
| [**Polynom\_able**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom_able.html) | [**derivative**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#derivative())() | Compute a new Polynom which is the derivative of this Polynom. |
| boolean | [**equals**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#equals(myMath.Polynom_able))​([**Polynom\_able**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom_able.html) p1) | Test if this Polynom is logically equals to p1. |
| double | [**f**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#f(double))​(double x) | calculate the value of the polynom when inserted value of x |
| boolean | [**isZero**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#isZero())() | Test if this is the Zero Polynom. |
| java.util.Iterator<[**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html)> | [**iteretor**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#iteretor())() | find the iterator of the polynom, which is a pointer to the start of the polynom |
| void | [**multiply**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#multiply(myMath.Monom))​([**Monom**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Monom.html) m1) | Multiply this Polynom by Monom m1. |
| void | [**multiply**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#multiply(myMath.Polynom_able))​([**Polynom\_able**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom_able.html) p1) | Multiply this Polynom by p1. |
| double | [**root**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#root(double,double,double))​(double x0, double x1, double eps) | Compute a value x' (x0<=x'<=x1) for with |f(x')| < eps assuming (f(x0)\*f(x1)<=0, else should throws runtimeException computes f(x') such that: (i) x0<=x'<=x1 && (ii) |f(x')|<eps< div=""> </eps<> |
| void | [**sortAndSumEqualDegree**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#sortAndSumEqualDegree())() | Sort the polynom and sum monoms of equal degree. |
| void | [**substract**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#substract(myMath.Polynom_able))​([**Polynom\_able**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom_able.html) p1) | Subtract p1 from this Polynom. |
| java.lang.String | [**toString**](file:///C:\Users\Hagit-PC\Desktop\לימודים\Eclipse\Exo\myMath\Polynom.html#toString())() | print the polynom |

**Methods inherited from class java.lang.Object**

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait