

Readme.pdf file - EX0

This is a Java Package for working with polynomials and monomials, with this library package you can create polynomials using human-readable string or as a java's collections behavior, add monomials and polynomials, multiply and much more, see documentation for further info and technical details.

Introduction to Monomials (for this library)

see: <https://en.wikipedia.org/wiki/Monomial>

What are monomials? Monomials in algebra are terms with the representation of $a \cdot X^b$ or $a \cdot X^b$, where 'a' is real number (hence double) and b is natural number (hence integer ≥ 0)

Examples:

1). $3 \cdot X^2$ or $3 \cdot X^2$.

2). $8 \cdot X^0$ or 8.

Introduction to Polynomials (for this library)

see: <https://en.wikipedia.org/wiki/Polynomial>

What are polynomials? Polynomials in algebra are single or multiple terms of monomials added together (as the word "poly" = many)

Examples:

1). $3 \cdot X^2 + 2 \cdot X^1 + X^0$ **or** $3 \cdot X^2 + 2 \cdot X^1 + 1$

2). $2X^3 - x^1 + 8 \cdot X^0$ **or** $2 \cdot X^3 - X + 8$.

Using the library to create Polynomials

As polynomials are just monomials added together, we can first create monomials objects and then add them to a polynomial.

To create a monomial we do:

Using manual configuration of coefficient 'a' and power degree 'b'

```
Monom myMonom = new Monom(a, b);
```

Using human-readable string of monom

```
Monom myMonom = new Monom("3x^25");
```

```
Monom myMonom = new Monom("-3*x^25");
```

```
Monom myMonom = new Monom("+3x");
```

Using a copy constructor and copy from another monom

```
Monom myMonom1 = new Monom("-3*x^25");
```

```
Monom myMonom2 = new Monom(myMonom1);
```

or

```
Monom myMonom2 = new Monom(myMonom1.toString());
```

Add your monomial to polynomial

```
Polynom myPolynom = new Polynom();
```

```
myPolynom.add(myMonom1);
```

```
myPolynom.add(myMonom2);
```

To create a polynomial at time of initialization

using multiple monoms splitter by ','

```
Polynom myPolynom = new Polynom(myMonom1, myMonom2);
```

or using string representation

```
Polynom myPolynom = new Polynom("5x^2 + 3x + 1");
```

or using a copy

```
Polynom myPolynom = new Polynom(myPolynom2);
```

Note: The Polynom class implements Polynom_Able and cont_function interfaces.

Operations on polynomials

You can add polynomials together and even add monomial to polynomial or monomial to another monomial, multiply polynomial by another polynomial or monomial, get the derivative polynomial of certain polynomial and more.

There are few special methods you can use with polynoms

Polynom.area(double x0, double x1, double epsilon)

Returns: a Double type, represents the area of the polynomial in given range x0 and x1.

The "area" function calculates the Riemann sum integral above the x-axis with approximate of epsilon, see:

https://en.wikipedia.org/wiki/Riemann_integral.

Polynom.root(double x0, double x1, double epsilon)

Returns: a Double type, represents one of the roots of the Polynomial in given range x0 and x1, (where $f(\text{root}) = 0$) in approximation of epsilon.

The "root" function finds a root of the polynomial in the closed area of x0 and x1, you must deliver correct x0 and x1 in a way that $f(x_0) * f(x_1) \leq 0$, see:

https://en.wikipedia.org/wiki/Properties_of_polynomial_roots

Polynom.derivative()

Returns: a Polynom object represents the derivative of current Polynomial.

The "derivative" function returns a new Polynomial object from current Polynomial, see:

<https://en.wikipedia.org/wiki/Derivative>

Example code

Example 1

```
Polynom myPolynom = new Polynom();
myPolynom.add(new Monom("x^2"));
myPolynom.add(new Monom(5, 3));
Polynom derPolynom = myPolynom.derivative();
System.out.println(derPolynom);
```

Will Print:

$15*x^2 + 2*x^1$

Example 2

```
Polynom myPolynom = new Polynom("x^2 - 1");
double myRoot = myPolynom.root(-3, 0, 0.001);
System.out.println("root is: " + myRoot);
```

Will Print:

root is: -0.9998