

Package `myMath`

Class **Polynom**

`java.lang.Object`
`myMath.Polynom`

All Implemented Interfaces:

`java.io.Serializable`, `cont_function`, `function`, `Polynom_able`

```
public class Polynom
extends java.lang.Object
implements Polynom_able
```

This class represents a Polynom with add, 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

Author:

Boaz

See Also:

[Serialized Form](#)

Constructor Summary

Constructors

Constructor	Description
-------------	-------------

Polynom() Zero (empty polynom) constructing an polynom of the form $0.0x^0$

Polynom(java.lang.String s) init a Polynom from an String in a hashtable with the power as key and monom as value ";

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description	
void	add (Monom m1)	Add m1 to this Polynom	
void	add (Polynom_able other)	Add p1 to this Polynom	
double	area (double x0, double x1, double eps)	Compute a Riman's integral from x0 to x1 in eps steps.	
Polynom_able	copy ()		
Polynom_able	derivative ()	Compute a new Polynom which is the derivative of this Polynom	
boolean	equals (java.lang.Object other)	function to multiply a Polynom_able from the Polynom onject by multiply all the Monom's from the Polynom object (Hashtable)	
double	f (double x)	function to substitute to a Polynom an double value by replacing the x with the user parameter The function returning a double value	
int	hashCode ()	function to multiply a Polynom_able from the Polynom onject by multiply all the Monom's from the Polynom object (Hashtable)	
function	initFromString (java.lang.String s)		
boolean	isZero ()	Test if this is the Zero Polynom	

java.util.Iterator< Monom >	iteretor ()	
static void	main (java.lang.String[] args)	
void	multiply (Monom m1)	Multiply this Polynom by Monom m1
void	multiply (Polynom_able other)	Multiply this Polynom by p1
double	root (double x0, double x1, double eps)	Compute a value x' ($x_0 \leq x' \leq x_1$) for with $ f(x') < \text{eps}$ assuming $(f(x_0) * f(x_1)) \leq 0$, else should throws runtimeException computes f(x') such that: (i) $x_0 \leq x' \leq x_1$ && (ii) $ f(x') $
void	subtract (Polynom_able other)	Subtract p1 from this Polynom
java.lang.String	toString ()	return a String representing this complex function

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Constructor Detail

Polynom

public Polynom()

Zero (empty polynom) constructing an polynom of the form $0.0x^0$

Polynom

```
public Polynom(java.lang.String s)
```

init a Polynom from an String in a hashtable with the power as key and monom as value "

Parameters:

String - of a simply Monoms of the type $a \cdot x^b$, where a is double and b is a positive integer. The Monoms can only have a + or - sine between them such as: {"x", "3+1.4X^3-34x"} a unvalid Polynom example : {(2x^2-4)*(-1.2x-7.1)}, \$x^2, 5x^b, 6x^7.7, 5*x^2+1}

Method Detail

f

```
public double f(double x)
```

function to substitute to a Polynom an double value by replacing the x with the user parameter The function returning a double value

Specified by:

f in interface `function`

Parameters:

an - double value such as {4.34324,1948}

add

```
public void add(Polynom_able other)
```

Description copied from interface: Polynom_able

Add p1 to this Polynom

Specified by:

add in interface Polynom_able

add

```
public void add(Monom m1)
```

Description copied from interface: Polynom_able

Add m1 to this Polynom

Specified by:

add in interface Polynom_able

Parameters:

m1 - Monom

subtract

```
public void subtract(Polynom_able other)
```

Description copied from interface: Polynom_able

Subtract p1 from this Polynom

Specified by:

subtract in interface Polynom_able

multiply

```
public void multiply(Polynom_able other)
```

Description copied from interface: Polynom_able

Multiply this Polynom by p1

Specified by:

multiply in interface Polynom_able

hashCode

```
public int hashCode()
```

function to multiply a Polynom_able from the Polynom onject by multiply all the Monom's from the Polynom object (Hashtable)

Overrides:

hashCode in class java.lang.Object

Parameters:

an - valid Polynom_able that is instance of Polynom

equals

```
public boolean equals(java.lang.Object other)
```

function to multiply a Polynom_able from the Polynom onject by multiply all the Monom's from the Polynom object (Hashtable)

Specified by:

`equals` in interface `function`

Specified by:

`equals` in interface `Polynom_able`

Overrides:

`equals` in class `java.lang.Object`

Parameters:

an - valid `Polynom_able` that is instance of `Polynom`

Returns:

true iff this `polynom` represents the same function as `p1`

isZero

```
public boolean isZero()
```

Description copied from interface: `Polynom_able`

Test if this is the Zero `Polynom`

Specified by:

`isZero` in interface `Polynom_able`

Returns:

root

```
public double root(double x0, double x1, double eps)
```

Description copied from interface: `cont_function`

Compute a value x' ($x_0 \leq x' \leq x_1$) for with $|f(x')| < \epsilon$ assuming $(f(x_0) * f(x_1)) \leq 0$, else should throws runtimeException computes $f(x')$ such that: (i) $x_0 \leq x' \leq x_1$ && (ii) $|f(x')|$

Specified by:

`root` in interface `cont_function`

Parameters:

`x0` - starting point

`x1` - end point

`eps` - > 0 (positive) representing the epsilon range the solution should be within.

Returns:

an approximated value (root) for this (cont.) function

copy

```
public Polynom_able copy()
```

Specified by:

`copy` in interface `function`

derivative

```
public Polynom_able derivative()
```

Description copied from interface: `Polynom_able`

Compute a new Polynom which is the derivative of this Polynom

Specified by:

`derivative` in interface `Polynom_able`

Returns:

area

```
public double area(double x0, double x1, double eps)
```

Description copied from interface: [cont_function](#)

Compute a Riman's integral from x0 to x1 in eps steps.

Specified by:

[area](#) in interface [cont_function](#)

Parameters:

x0 - starting point

x1 - end point

eps - positive step value

Returns:

the approximated area above X-axis below this function bounded in the range of [x0,x1]

iteretor

```
public java.util.Iterator<Monom> iteretor()
```

Specified by:

[iteretor](#) in interface [Polynom_able](#)

Returns:

an Iterator (of Monoms) over this Polynom

multiply

```
public void multiply(Monom m1)
```

Description copied from interface: [Polynom_able](#)

Multiply this Polynom by Monom m1

Specified by:

[multiply](#) in interface [Polynom_able](#)

toString

```
public java.lang.String toString()
```

Description copied from interface: [function](#)

return a String representing this complex function

Specified by:

[toString](#) in interface [function](#)

Overrides:

[toString](#) in class [java.lang.Object](#)

main

```
public static void main(java.lang.String[] args)
```

initFromString

```
public function initFromString(java.lang.String s)
```

Specified by:

`initFromString` in interface `function`

[PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

ALL CLASSES

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)