Package myMath

Class Monom

java.lang.Object
    myMath.Monom

All Implemented Interfaces:

`java.io.Serializable`, `function`

```
public class Monom
extends java.lang.Object
implements function
```

This class represents a simple "Monom" of shape a*x^b, where a is a real number and a is an integer (summed a none negative), see: https://en.wikipedia.org/wiki/Monomial The class implements function and support simple operations as: construction, value at x, derivative, add and multiply.

Author:

Boaz

See Also:

Serialized Form

Field Summary

Fields

Modifier and Type	Field	Description
`static java.util.Comparator<Monom>`	`_Comp`	
`static double`	`EPSILON`	
`static Monom`	`MINUS1`	
`static Monom`	`ZERO`	

Constructor Summary

Constructors

Constructor	Description
`Monom()`	
`Monom (double a, int b)`	
`Monom (java.lang.String s)`	The monom constractor.
`Monom (Monom ot)`	

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
void	**add** (**Monom** toAdd)	function to add to monoms with the same power
function	**copy**()	
Monom	**derivative**()	The derivative of a function of a real variable measures the sensitivity to change of the function value (output value) with respect to a change in its argument (input value).
boolean	**equals** (java.lang.Object other)	
double	**f** (double x)	
double	**get_coefficient**()	
int	**get_power**()	
static java.util.Comparator<**Monom**>	**getComp**()	
function	**initFromString** (java.lang.String s)	
boolean	**isZero**()	
static void	**main** (java.lang.String[] args)	
void	**multipy** (**Monom** other)	function to multipy to monoms with the same power
java.lang.String	**toString**()	creat a String out of the monom

Methods inherited from class java.lang.Object

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

Field Detail

ZERO

public static final Monom ZERO

MINUS1

public static final Monom MINUS1

EPSILON

public static final double EPSILON

See Also:

Constant Field Values

_Comp

```
public static final java.util.Comparator<Monom> _Comp
```

Constructor Detail

Monom

```
public Monom (double a,
              int b)
```

Monom

```
public Monom (Monom ot)
```

Monom

```
public Monom()
```

Monom

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

The monom constractor. make a monom out of string.

Parameters:

`String` - of a simply of the type a*x^b , where a is is double and b is a positive integer.} a unvalid Polynom example : {4*x^4,2x^3.0,4x^^4,$4x^4}

Method Detail

getComp

```
public static java.util.Comparator<Monom> getComp()
```

get_coefficient

```
public double get_coefficient()
```

get_power

```
public int get_power()
```

derivative

```
public Monom derivative()
```

The derivative of a function of a real variable measures the sensitivity to change of the function value (output value) with respect to a change in its argument (input value). Derivatives are a fundamental tool of calculus. this function is doing derivative by the fprmula "xn=nxn–1"

f

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

Specified by:

`f` in interface `function`

isZero

```
public boolean isZero()
```

add

```
public void add (Monom toAdd)
```

function to add to monoms with the same power

Parameters:

an - Monom with the same Power as the main Monom

multipy

```
public void multipy (Monom other)
```

function to multipy to monoms with the same power

Parameters:

an - Monom with the same Power as the main Monom

toString

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

creat a String out of the monom

Specified by:

`toString` in interface `function`

Overrides:

`toString` in class `java.lang.Object`

equals

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

Specified by:

`equals` in interface `function`

Overrides:

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

initFromString

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

Specified by:

`initFromString` in interface `function`

copy

```
public function copy()
```

Specified by:

`copy` in interface `function`

main

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