Package myMath

Class ComplexFunction

java.lang.Object
    myMath.ComplexFunction

All Implemented Interfaces:

`java.io.Serializable`, `complex_function`, `function`

```
public class ComplexFunction
extends java.lang.Object
implements complex_function
```

ComplexFunction is a function made of connecting simple (Monom or Polynom) functions with an Operation with the tree database. altroght ComplexFunction can input an Monom it will change it to Polynom because Polynom contains all the MOnom methods you wont feel the difference as user but it's still important to know . Plus: plus(f1(x), f2(x)), Times: mul(f1(x), f2(x)), Divid: div(f1(x), f2(x)), Max: max(f1(x), f2(x)), Min: min(f1(x), f2(x)), Comp: comp(f1(x), f2(x)) == f1(f2(x)) this class contains multiple methods on this function such as f , equals , and more... the ComplexFunction is the type function and have all the function interface methods included. fore more information read the wiki.

Author:

Simon Piklov and Ron Leib

See Also:

Serialized Form

Constructor Summary

Constructors

Constructor	Description
ComplexFunction()	ComplexFunction defualt constractor all the tree params as null
ComplexFunction (java.lang.String operation, function left, function right)	ComplexFunction constractor for string and left and right function when the operation can be one of the params listed bellow or null , if it us a nuul only one function can be in the CompleFunction
ComplexFunction(function func)	ComplexFunction constractor of onre (Monom or Polynom or ComplexFunction) functions with an null as operation
ComplexFunction(Operation symbol, function left, function right)	ComplexFunction contractor for Operation and left and right function when the operation can be one of the params listed bellow or null , if it us a nuul only one function can be in the CompleFunction

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
void	comp(function f1)	function to change the cuurent coplxfunction symbol to Comp , move current node left and f1 right
function	copy()	a deep copy function that return a coplex function.
void	div(function f1)	function to change the cuurent coplxfunction symbol to Divid , move current node left and f1 right
boolean	equals(java.lang.Object obj)	this function returns true or false with defualt range and step .
boolean	equals(java.lang.Object obj, double accurcy, double width)	this function returns true or false but because coplex functions havw

		operation like max and min that can return right or left depending on f(x) so we can not know if two functions are alwaws equal thats why we the function check a lot of point's as the user input but stiil the answer my not alwas be true , if you want a reeal eccurcy take a as much wide range as you can.
`double`	`f(double x)`	take the x and calculating f(x) at the inputed x,the fanction
`Operation`	`getOp()`	The complex_function oparation: plus, mul, div, max, min, comp
`static Operation`	`ifOperation(java.lang.String checker)`	
`function`	`initFromString(java.lang.String complexString)`	recursive function to make a Complexfunction out of string example for a good String :"plus(div(+1.0x +1.0,mul(mul(+1.0x +3.0,+1.0x -2.0),+1.0+ 4.0)),2.0)"; it's imortant that the operation is one of the suppurted one you can read the building counstractor ComplexFunction (String operation , function left , function right) for more information or the wiki.
`function`	`left()`	returns the left side of the complex function - this side should always exists (should NOT be null).
`static void`	`main(java.lang.String[] args)`	
`void`	`max(function f1)`	function to change the cuurent coplxfunction symbol to Max , move current node left and f1 right
`void`	`min(function f1)`	function to change the cuurent coplxfunction symbol to Min , move current node left and f1 right
`void`	`mul(function f1)`	function to change the cuurent coplxfunction symbol to Times , move current node left and f1 right
`java.lang.String`	`nameOperation(Operation checker)`	a helping function to cunvert Operation to string
`void`	`plus(function f1)`	function to change the cuurent coplxfunction symbol to plus , move current node left and f1 right

`function`	`right()`	returns the right side of the complex function - this side might not exists (aka equals null).
`java.lang.String`	`toString()`	return a String representing this complex function

Methods inherited from class java.lang.Object

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

Constructor Detail

ComplexFunction

```
public ComplexFunction(function func)
```

ComplexFunction constractor of onre (Monom or Polynom or ComplexFunction) functions with an null as operation

Parameters:

`func` -

ComplexFunction

```
public ComplexFunction()
```

ComplexFunction defualt constractor all the tree params as null

ComplexFunction

```
public ComplexFunction(java.lang.String operation,
                       function left,
                       function right)
```

ComplexFunction constractor for string and left and right function when the operation can be one of the params listed bellow or null , if it us a nuul only one function can be in the CompleFunction

Parameters:

`operation` - - string of the type : Plus, Times, Divid, Max, Min, Comp ,times , div ,mul. there is no importance for small or big letters

`function` - of the the type Monom or Polynom or ComplexFunction

`function` - of the the type Monom or Polynom or ComplexFunction

ComplexFunction

```
public ComplexFunction(Operation symbol,
                       function left,
                       function right)
```

ComplexFunction contractor for Operation and left and right function when the operation can be one of the params listed bellow or null , if it us a nuul only one function can be in the CompleFunction

Parameters:

`symbol` - Operation : Plus, Times, Divid, Max, Min, Comp important ! None and Error are Not supported and if you will put them you will get an exception

`left` - - function of the the type Monom or Polynom or ComplexFunction

`right` - - function of the the type Monom or Polynom or ComplexFunction

Method Detail

f

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

take the x and calculating f(x) at the inputed x,the fanction

Specified by:

`f` in interface `function`

Parameters:

`x` - : an double to put in the domain function

ifOperation

```
public static Operation ifOperation(java.lang.String checker)
```

Parameters:

`checker` - - a string that represent a Operation

Returns:

Operation of the string

initFromString

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

recursive function to make a Complexfunction out of string example for a good String :"plus(div(+1.0x +1.0,mul(mul(+1.0x +3.0,+1.0x -2.0),+1.0+ 4.0)),2.0)"; it's imortant that the operation is one of the suppurted one you can read the building counstractor ComplexFunction (String operation , function left , function right) for more information or the wiki.

Specified by:

`initFromString` in interface `function`

Parameters:

`complexString` - - that represent the function "string_Operation(f1,f2)"

copy

```
public function copy()
```

a deep copy function that return a coplex function. should be use as ComplexFunction aCopy = (ComplexFunction) a.copy();

Specified by:

`copy` in interface `function`

plus

```
public void plus(function f1)
```

function to change the cuurent coplxfunction symbol to plus , move current node left and f1 right

Specified by:

`plus` in interface `complex_function`

Parameters:

`f1` - - the functiob input can be of the the type Monom or Polynom or ComplexFunction

mul

```
public void mul(function f1)
```

function to change the cuurent coplxfunction symbol to Times , move current node left and f1 right

Specified by:

`mul` in interface `complex_function`

Parameters:

`f1` - - the functiob input can be of the the type Monom or Polynom or ComplexFunction

div

```
public void div(function f1)
```

function to change the cuurent coplxfunction symbol to Divid , move current node left and f1 right

Specified by:

`div` in interface `complex_function`

Parameters:

`f1` - - the functiob input can be of the the type Monom or Polynom or ComplexFunction

max

```
public void max(function f1)
```

function to change the cuurent coplxfunction symbol to Max , move current node left and f1 right

Specified by:

`max` in interface `complex_function`

Parameters:

`f1` - - the functiob input can be of the the type Monom or Polynom or ComplexFunction

min

```
public void min(function f1)
```

function to change the cuurent coplxfunction symbol to Min , move current node left and f1 right

Specified by:

`min` in interface `complex_function`

Parameters:

`f1` - - the functiob input can be of the the type Monom or Polynom or ComplexFunction

comp

```
public void comp(function f1)
```

function to change the cuurent coplxfunction symbol to Comp , move current node left and f1 right

Specified by:

`comp` in interface `complex_function`

Parameters:

`f1` - - the functiob input can be of the the type Monom or Polynom or ComplexFunction

nameOperation

```
public java.lang.String nameOperation(Operation checker)
```

a helping function to cunvert Operation to string

Parameters:

`checker` -

Returns:

the sting of the operation

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`

equals

```
public boolean equals(java.lang.Object obj, double accurcy, double width)
```

this function returns true or false but because coplex functions havw operation like max and min that can return right or left depending on f(x) so we can not know if two functions are alwaws equal thats why we the function check a lot of point's as the user input but stiil the answer my not alwas be true , if you want a reeal eccurcy take a as much wide range as you can.

Parameters:

`obj` - of the the type Monom or Polynom or ComplexFunction

`accurcy` - the steps of the function

`from` - what point to what point to check the equals

equals

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

this function returns true or false with defualt range and step . because complex functions have operation like max and min that can return right or left depending on f(x) so we can not know if two functions are always equal thats why we the function check a lot of point's as the user input but stiil the answer my not alwas be true , if you want a reeal eccurcy take a as much wide range as you can with the custum equals function .

Specified by:

`equals` in interface `function`

Overrides:

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

Parameters:

`obj` - of the the type Monom or Polynom or ComplexFunction accurcy the steps of the function defualt (0.5) from what point to what point to check the equals defualt 1000

left

```
public function left()
```

Description copied from interface: `complex_function`
returns the left side of the complex function - this side should always exists (should NOT be null).

Specified by:
`left` in interface `complex_function`

Returns:
a function representing the left side of this complex funcation

right

```
public function right()
```

Description copied from interface: `complex_function`
returns the right side of the complex function - this side might not exists (aka equals null).

Specified by:
`right` in interface `complex_function`

Returns:
a function representing the left side of this complex funcation

getOp

```
public Operation getOp()
```

Description copied from interface: `complex_function`
The complex_function oparation: plus, mul, div, max, min, comp

Specified by:

`getOp` in interface `complex_function`

Returns:

main

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