



Escuela De Computación

Programación Orientada a Objetos

Informe de Proyecto:

PROYECTO 1: Cálculo infinitesimal de 1 variable usando métodos numéricos con graficación

Elaborado por:

Jorge Barquero Villalobos - 2015019476

Allan Mauricio Castillo Vega - 2015043074

Juan Diego Escobar Sánchez - 2015110428

Índice

1. Enunciado	2
2. Detalles de Implementación	10
3. Repositorio	515
4. Conclusiones	515

1. Enunciado



Proyecto 1: Munchkin, Cálculo infinitesimal.

M. Sc. Ricardo Román Brenes - `rroman@itcr.ac.cr`

I-2017

Tabla de contenidos

1. Enunciado	1
1.1. Cálculo infinitesimal de 1 variable usando métodos numéricos con graficación	1
1.2. Munchkin	3
2. Evaluación	7
2.1. Implementación	7
2.2. Manual de Usuario	7
3. Consideraciones	7

1. Enunciado

El objetivo de este proyecto es brindarle al estudiante la capacidad de modelar e implementar problemas utilizando la programación orientada a objetos, incluida la herencia, el polimorfismo y algunos patrones de diseño de software, usando el lenguaje de programación Java. Éste proyecto tiene un aporte para la nota final del curso de 20 %.

Se debe programar y documentar **1 de las 2** propuestas descritas a continuación.

1.1. Cálculo infinitesimal de 1 variable usando métodos numéricos con graficación

Desarrolle una aplicación que dada una función matemática permita calcular su integral definida, su derivada en un punto arbitrario y su limite en un punto arbitrario; todo para funciones de una variable. Su programa debe también poder graficar dichas funciones.

Para facilitar la implementación, utilice la biblioteca **mXparser** <http://mathparser.org/> que permite transformar Strings en un objeto tipo función que se puede evaluar en diferentes puntos. Se adjunta el código de un ejemplo pequeño (recuerde en NetBeans agregar el .jar al proyecto en Propiedades>Bibliotecas>Archivo JAR o directorio).

Implemente al menos las siguientes clases:

- **CalculoIntegral**: interfaz para las clases de calculo de integrales.
- **MetodoDelTrapezio**: implementación del método del trapezio para integrales.
- **MetodoDeSimpson**: implementación del método de Simpson de 1/3 para integrales.
- **CalculoDerivada**: interfaz para las clases de calculo de derivadas.
- **MetodoDiferenciasCentrales**: implementación del método de las diferencias centrales para derivadas.
- **CalculoLimite**: interfaz para las clases de calculo de límites.
- **MetodoAproximacion**: implementación del método de aproximación numérica de límites.

Establezca una herencia de clases como se muestra en la figura 1. Los métodos que se muestran son lo mínimo que se debe implementar, posiblemente se deberán crear más métodos y atributos.

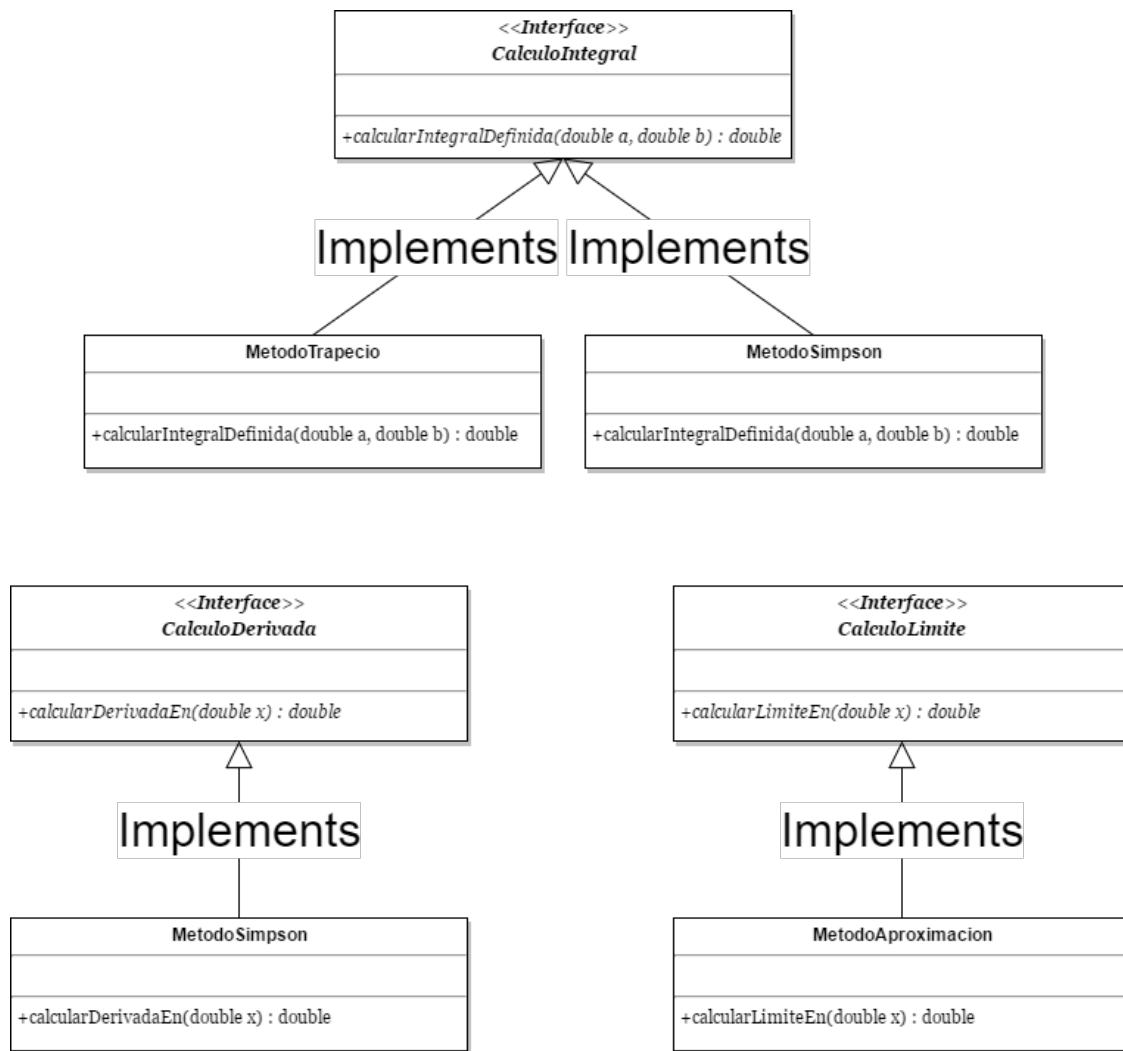


Figura 1: Diagrama de clases para opción de cálculo infinitesimal.

Además deberá utilizar alguna biblioteca para graficar las funciones, como **JFreeChart** <http://www.jfree.org/jfreechart/>..

1.2. Munchkin

Desarrolle una aplicación para jugar una versión reducida de **Munchkin**.

La baraja de cartas para Munchkin consiste de:

- 40 × cartas *treasure*:
 - 3 × cartas *trinket +1*
 - 3 × cartas *armor +1*
 - 3 × cartas *feet +1*
 - 3 × cartas *1-hand +1*
 - 3 × cartas *2-hands +1*
 - 2 × cartas *headgear +2*
 - 2 × cartas *armor +2 BIG*
 - 2 × cartas *feet +2*
 - 2 × cartas *1-hand +2*
 - 2 × cartas *2-hands +2 BIG*
 - 1 × cartas *headgear +4 BIG*
 - 1 × cartas *armor +4 BIG*
 - 1 × cartas *feet +3*
 - 1 × cartas *1-hand +4 BIG*
 - 1 × cartas *2-hands +5 BIG*
 - 2 × cartas *curse lose a level*
 - 2 × cartas *curse lose an item*
 - 4 × cartas *blessing go up a level*
- 40 × cartas *door*:
 - 3 × cartas *monster level 1; bad stuff: lose an item; gain 1 level*
 - 3 × cartas *monster level 2; bad stuff: lose an item; gain 1 level*
 - 3 × cartas *monster level 3; bad stuff: lose an item; gain 1 level*
 - 3 × cartas *monster level 4; bad stuff: lose an item; gain 1 level*
 - 3 × cartas *monster level 5; bad stuff: lose an item; gain 2 levels*
 - 2 × cartas *monster level 6; bad stuff: lose a level; gain 1 level*
 - 2 × cartas *monster level 7; bad stuff: lose a level; gain 1 level*
 - 2 × cartas *monster level 8; bad stuff: lose a level; gain 1 level*
 - 2 × cartas *monster level 9; bad stuff: lose a level; gain 1 level*
 - 2 × cartas *monster level 10; bad stuff: lose a level; gain 2 levels*
 - 1 × cartas *monster level 12; bad stuff: lose a level and item; gain 2 levels*

- 1 × cartas *monster level 14*; bad stuff: lose a level and item; gain 2 levels
- 1 × cartas *monster level 16*; bad stuff: die; gain 2 levels
- 1 × cartas *monster level 18*; bad stuff: die; gain 2 levels
- 1 × cartas *monster level 20*; bad stuff: lose 2 levels, 2 items and die; gain 2 levels
- 2 × cartas *curse lose a level*
- 2 × cartas *curse lose an item*
- 4 × cartas *blessing go up a level*

Las reglas de Munchkin que se deben implementar para esta opción son (adaptado de <https://github.com/andersfischernielsen/Munchkin-Short-Rules/blob/master/RULES.md>):

Simplified Munchkin Rules

Setup

- Divide the cards into a treasure stack and a door (AKA room) stack.
- Give each player four cards from each stack. Give each player a level token. Everyone starts at level 1.

Cards

- Item cards: Items have a price at the bottom on the card and - some-number.^{at} the top of the card.
- Monster cards: Monster cards have a level and name at the top of the card and "some-number Treasures.^{at} the bottom.
- Curse cards: Curse cards have Curse!^{at} the top of the card.

Starting the Game

- If you have any item cards and want to equip them you can also do so now.
- Roll the dice. Decide who starts from the die rolls.

Kick Open the Door

- Draw a card from the door deck.
- If it's a monster - fight it. Then your turn ends.
- If it's a curse - it hits you. Then Look For Trouble or Loot the Room.
- If it's any other card - put it in your hand. Look For Trouble or Loot the Room.

Look For Trouble or Loot the Room

- If you did not fight a monster, choose one of these options:
 1. Look For Trouble: Choose to play a monster you have in your hand and fight it for levels and treasure.

2. Loot the Room: Choose to loot the room you entered and draw another card from the door pile. Keep it in your hand.

Charity

- When your turn ends, you must have no more than five cards in your hand. If you have more, choose to:
 - Play cards until you're down to five (curses, items etc.)
 - Give cards to the lowest level player until you're down to five cards. If you're the lowest level player, discard cards until you're down to five.

Combat

- If your level plus bonuses add up to more than the monster, you win. You only get level(s) from killing monster(s), not making them disappear etc. Collect the indicated treasures and gain the indicated level(s).
- If you and the monster are tied, or the monster has more than you, you lose. Try to run away.

Running Away

- Roll the dice. If you get a 5 or more, you escape. If not, read the monster card and do what the "Bad Stuff" says.

Items

- Items are not equipped when turned sideways on the table. You can only equip when not in combat.
- You can only carry one type of item (headgear, armor, feet, 1-hand, 2-hands).
- You can equip any amount of items that doesn't not have a type.
- You can only carry one Big item.

Curses

- Curses in your hand can be played at the start of your turn, they apply to any player, including yourself.

Death

- If you die, you lose everything except for your level, curses on you.
- Put every other card you have on the table face up.
- The other players each pick items, until everything is gone. The highest level players pick first.
- On your next turn, draw four cards from each deck, like when the game began.

General Rules

- Cards in front of you are in play. When a card has been played it cannot be taken back into your hand.
- First player to reach level 10 wins.
- You can only reach level 10 by killing a monster.
- You can never have a lower level than 1. On your turn you can: Play items, send curses, discard items, kick the door.
- “Go Up A Level” cards can be used on any player.

Dele nombre a los monstruos y a los ítemes, use su imaginación. Utilice algún tipo de interfaz que haga que el juego sea fácil de jugar (gráfica o de texto)

Cree clases para modelar las cartas, los ítemes y los jugadores (al menos). Implemente una herarquía de clases para las cartas y los ítemes, identificando componentes comunes en clases abstractas.

2. Evaluación

La fecha de entrega es el **domingo 9 de abril**. La evaluación se desglosa de la siguiente manera:

1. Implementación	50 %
2. Informe	25 %
3. Manual de Usuario	25 %
Total	100 %

2.1. Implementación

El equipo deberá implementar la especificación provista anteriormente en Java utilizando. Se evaluará completitud, efectividad, eficiencia y elegancia del código.

2.2. Manual de Usuario

El equipo deberá redactar un informe en \LaTeX sobre el trabajo realizado, se recomienda utilizar el formato de los laboratorios. Este debe incluir al menos:

- Requerimientos mínimos para la ejecución.
- Proceso de instalación de su programa.
- Como ejecutar el programa.
- Como se juega el programa.

El proyecto debe ser defendido por el equipo frente al profesor bajo previa cita.

3. Consideraciones

- Haga grupos de hasta 3 personas.
- Suba su código y documentación (informe, javadoc y manual de usuario) al GitLab respectivo de su grupo y el directorio del proyecto.
- Todos los estudiantes del grupo deben subir el reporte a Schoology. (<https://app.schoology.com/assignment/1071024870/>).
- Recuerde que por cada día tardío de entrega **se le rebajaran puntos** de acuerdo con la formula: 4^d , donde $d > 1$ es la cantidad de días tardíos.

2. Detalles de Implementación

■ Diagrama de clases



Javadoc

Package calculo

Interface Summary

[CalculoDerivada](#)

[CalculoIntegral](#)

[CalculoLimite](#)

Class Summary

[DerivadaGUI](#)

[FunctionVisualizer](#)

[GraficoGUI](#)

[InitGUI](#)

[IntegralGUI](#)

[LimGUI](#)

[MetodoAproximacion](#)

[MetodoDiferenciasCentrales](#)

[MetodoSimpson](#)

[MetodoTrapecio](#)

calculo

Interface CalculoDerivada

< [Methods](#) >

public interface **CalculoDerivada**

Author:

Mauricio Castillo

Methods

calcularDerivadaEn

```
public double calcularDerivadaEn(double x)
```

Parameters:

x -

Returns:

calculo

Interface CalculoIntegral

< [Methods](#) >

```
public interface CalculoIntegral
```

Author:

Mauricio Castillo

Methods

calcularIntegralDefinida

```
public double calcularIntegralDefinida(double a,  
                                       double b)
```

Parameters:

a -

b -

Returns:

calculo

Interface CalculoLimite

< [Methods](#) >

```
public interface CalculoLimite
```

Author:

Mauricio Castillo

Methods

LimiteEn

```
public java.lang.Double LimiteEn(double x)
```

Parameters:

x -

Returns:

calculo

Class DerivadaGUI

```
java.lang.Object
|
+-- java.awt.Component
|   |
|   +-- java.awt.Container
|       |
|       +-- java.awt.Window
|           |
|           +-- java.awt.Frame
|               |
|               +-- javax.swing.JFrame
|                   |
|                   +-- calculo.DerivadaGUI
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.WindowConstants

< [Constructors](#) >

```
public class DerivadaGUI
extends javax.swing.JFrame
```

Author:

Mauricio Castillo

Constructors

DerivadaGUI

```
public DerivadaGUI(InitGUI ventanaPrincipal,  
                  Function funcion)
```

Parameters:

ventanaPrincipal -
funcion -

calculo

Class FunctionVisualizer

```
java.lang.Object  
|  
+--calculo.FunctionVisualizer
```

< [Constructors](#) > < [Methods](#) >

```
public class FunctionVisualizer  
extends java.lang.Object
```

Author:

Mauricio Castillo

Constructors

FunctionVisualizer

```
public FunctionVisualizer(Function funcion,  
                          double valorX)
```

Parameters:

funcion -
valorX -

FunctionVisualizer

```
public FunctionVisualizer(Function funcion,  
                          double valorA,  
                          double valorB)
```

Parameters:

funcion -
valorA -
valorB -

FunctionVisualizer

```
public FunctionVisualizer(Function pFunction,  
                          double[] pPuntosCentrales,  
                          double pValorX)
```

Parameters:

pFunction -
pPuntosCentrales -
pValorX -

Methods

creacionGrafico

```
public ChartPanel creacionGrafico()
```

Returns:

panel

calculo

Class GraficoGUI

```
java.lang.Object
|
+-- java.awt.Component
|   |
|   +-- java.awt.Container
|       |
|       +-- java.awt.Window
|           |
|           +-- java.awt.Frame
|               |
|               +-- javax.swing.JFrame
|                   |
|                   +-- calculo.GraficoGUI
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.TransferHandler, javax.swing.WindowConstants

< [Constructors](#) > < [Methods](#) >

```
public class GraficoGUI
extends javax.swing.JFrame
```

Author:

Mauricio Castillo

Constructors

GraficoGUI

```
public GraficoGUI(InitGUI ventanaPrincipal,
                  Function funcion,
                  int indicadorMetodo)
```

Parameters:

ventanaPrincipal -
funcion -
indicadorMetodo -

Methods

setDerivateValues

```
public void setDerivateValues(double pValorX,  
                             double[] pPuntos,  
                             java.lang.String pMetodo)
```

setGrafica

```
public void setGrafica()
```

setIntegralValues

```
public void setIntegralValues(double valorA,  
                              double valorB,  
                              double resultado,  
                              java.lang.String metodo)
```

Parameters:

- valorA -
- valorB -
- resultado -
- metodo -

setLimitValues

```
public void setLimitValues(double valorX,  
                           java.lang.Double resultado,  
                           java.lang.String metodo)
```

Parameters:

- valorX -
- resultado -
- metodo -

setVentana

```
public void setVentana(IntegralGUI gui)
```

Parameters:

- gui -

calculo

Class InitGUI

```
java.lang.Object
|
+-- java.awt.Component
|   |
|   +-- java.awt.Container
|       |
|       +-- java.awt.Window
|           |
|           +-- java.awt.Frame
|               |
|               +-- javax.swing.JFrame
|                   |
|                   +-- calculo.InitGUI
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.WindowConstants

< [Constructors](#) > < [Methods](#) >

```
public class InitGUI
extends javax.swing.JFrame
```

Author:

Mauricio Castillo

Constructors

InitGUI

```
public InitGUI()
```

Methods

checkValue

```
public boolean checkValue(java.lang.String texto)
```

Parameters:

texto -

Returns:

main

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

Parameters:

args - the command line arguments

calculo

Class IntegralGUI

```
java.lang.Object
|
+-- java.awt.Component
|   |
|   +-- java.awt.Container
|       |
|       +-- java.awt.Window
|           |
|           +-- java.awt.Frame
|               |
|               +-- javax.swing.JFrame
|                   |
|                   +-- calculo.IntegralGUI
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.WindowConstants

< [Constructors](#) > < [Methods](#) >

```
public class IntegralGUI
extends javax.swing.JFrame
```

Author:

Mauricio Castillo

Constructors

IntegralGUI

```
public IntegralGUI(InitGUI ventanaPrincipal,  
                  Function funcion)
```

Parameters:

ventanaPrincipal -
funcion -

Methods

setStateButtons

```
public void setStateButtons()
```

calculo

Class LimGUI

```
java.lang.Object
|
+-- java.awt.Component
|   |
|   +-- java.awt.Container
|       |
|       +-- java.awt.Window
|           |
|           +-- java.awt.Frame
|               |
|               +-- javax.swing.JFrame
|                   |
|                   +-- calculo.LimGUI
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
 javax.accessibility.Accessible, javax.swing.RootPaneContainer,
 javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.WindowConstants

< [Constructors](#) >

```
public class LimGUI
extends javax.swing.JFrame
```

Author:

Mauricio Castillo

Constructors

LimGUI

```
public LimGUI(InitGUI ventanaPrincipal,  

               Function funcion)
```

Parameters:

ventanaPrincipal -
 funcion -

calculo

Class MetodoAproximacion

```
java.lang.Object
|
+--calculo.MetodoAproximacion
```

All Implemented Interfaces:

[CalculoLmite](#)

< [Constructors](#) > < [Methods](#) >

```
public class MetodoAproximacion
extends java.lang.Object
implements CalculoLmite
```

Author:

Mauricio Castillo

Constructors

MetodoAproximacion

```
public MetodoAproximacion(Function funcion)
```

Parameters:

funcion -

Methods

LmiteEn

```
public java.lang.Double LmiteEn(double x)
```

Parameters:

x -

Returns:

getTendenciaInferior

```
public java.util.ArrayList getTendenciaInferior()
```

Returns:

getTendenciaSuperior

```
public java.util.ArrayList getTendenciaSuperior()
```

Returns:

calculo

Class MetodoDiferenciasCentrales

```
java.lang.Object
|
+--calculo.MetodoDiferenciasCentrales
```

All Implemented Interfaces:

[CalculoDerivada](#)

< [Constructors](#) > < [Methods](#) >

```
public class MetodoDiferenciasCentrales
  extends java.lang.Object
  implements CalculoDerivada
```

Author:

Mauricio Castillo

Constructors

MetodoDiferenciasCentrales

```
public MetodoDiferenciasCentrales(Function pFuncion)
```

Parameters:

pFuncion -
h -

Methods

calcularDerivadaEn

```
public double calcularDerivadaEn(double x)
```

Parameters:

x -

Returns:

valorF(x)

getListaCentrada

```
public double[] getListaCentrada()
```

calculo

Class MetodoSimpson

```
java.lang.Object
|
+--calculo.MetodoSimpson
```

All Implemented Interfaces:

[CalculoIntegral](#)

< [Constructors](#) > < [Methods](#) >

```
public class MetodoSimpson
extends java.lang.Object
implements CalculoIntegral
```

Author:

Mauricio Castillo

Constructors

MetodoSimpson

```
public MetodoSimpson(Function funcion,
                    int valorN)
```

Parameters:

funcion -

valorN -

Methods

calcularIntegralDefinida

```
public double calcularIntegralDefinida(double a,  
                                         double b)
```

Parameters:

a -
b -

Returns:

calculo

Class MetodoTrapecio

```
java.lang.Object  
|  
+--calculo.MetodoTrapecio
```

All Implemented Interfaces:

[CalculoIntegral](#)

< [Constructors](#) > < [Methods](#) >

```
public class MetodoTrapecio  
extends java.lang.Object  
implements CalculoIntegral
```

Author:

Mauricio Castillo

Constructors

MetodoTrapecio

```
public MetodoTrapecio(Function funcion,  
                      int valorN)
```

Parameters:

funcion -
valorN -

Methods

calcularIntegralDefinida

```
public double calcularIntegralDefinida(double a,  
                                         double b)
```

Parameters:

a -
b -

Returns:

Package org.mariuszgromada.math.mxparser

Interface Summary

[FunctionExtension](#)

FunctionExtension provides interface for function algorithm definition.

Class Summary

[Argument](#)

Argument class enables to declare the argument (variable) which can be used in further processing (in expressions, functions and dependent / recursive arguments).

[ArgumentParameter](#)

Handling argument parameters

[Constant](#)

Constant class provides ability to declare constants.

[DescKwLenComparator](#)

Comparator for key word list sorting by descending key word length .

[Expression](#)

Expression - base class for real expressions definition.

[Function](#)

Function class provides possibility to define user functions.

[FunctionParameter](#)

Package level class for handling function parameters.

[HeadEqBody](#)

[IterativeOperatorParameters](#)

Package level class for generating iterative operator parameters

[KwStrComparator](#)

Comparator for key word list sorting by key word string.

[KwTypeComparator](#)

Comparator for key word list sorting by type of the key word

[PrimitiveElement](#)

Class used for connecting all basic elements such as: Argument, Constant, Function.

[RecursiveArgument](#)

RecursiveArgument class enables to declare the argument (variable) which is defined in a recursive way.

[SyntaxStackElement](#)

[TokenModification](#)

Data structure used internally for token to be modified list

[TokenStackElement](#)

Internal token class which is used with stack while evaluation of tokens levels

[Tutorial](#)

Tutorial class.

[mXparser](#)

mXparser class provides usefull methods when parsing, calculating or parameters transforming.

org.mariuszgromada.math.mxparser

Class Argument

```
java.lang.Object
|
+--PrimitiveElement
    |
    +--org.mariuszgromada.math.mxparser.Argument
```

Direct Known Subclasses:

[RecursiveArgument](#)

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class Argument
extends PrimitiveElement
```

Argument class enables to declare the argument (variable) which can be used in further processing (in expressions, functions and dependent / recursive arguments).

For example:

- 'x' - argument in expression 'sin(x)'
- 'x' and 'y' - arguments in expression 'sin(x)+cos(y)'.
- 'x=2*t' - dependent argument (dependent from 't') in expression 'cos(x)'

Using Argument class you can define two argument types:

- **free argument** - when value of argument 'x' is directly given by a number (for example 'x=5')
- **dependent argument** - when value of argument 'x' is given by expression (for example: 'x=2*a+b' - argument 'x' depends from argument/constant 'a' and argument/constant 'b' or any other possible option like function, etc...)

When creating an argument you should avoid names reserved as parser keywords, in general words known in mathematical language as function names, operators (for example: sin, cos, +, -, etc...). Please be informed that after associating the argument with the expression, function or dependent/recursive argument its name will be recognized by the parser as reserved key word. It means that it could not be the same as any other key word known by the parser for this particular expression. Parser is case sensitive.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](https://bitbucket.org/mariuszgromada/mxparser/)

[mXparser on CodePlex](https://codeplex.com/mxparser/)

[Janet Sudoku - project web page](http://janet.sudoku.pl)

[Janet Sudoku on GitHub](https://github.com/mariuszgromada/Janet-Sudoku)

[Janet Sudoku on CodePlex](https://codeplex.com/Janet-Sudoku/)

[Janet Sudoku on SourceForge](https://sourceforge.net/projects/janetsudoku/)

[Janet Sudoku on BitBucket](https://bitbucket.org/mariuszgromada/janet-sudoku/)

Version:

4.0.0

RecursiveArgument

Expression

Function

Constant

Fields

ARGUMENT_INITIAL_VALUE

```
public static final double ARGUMENT_INITIAL_VALUE  
    Double.NaN as initial value of the argument.
```

DEPENDENT_ARGUMENT

```
public static final int DEPENDENT_ARGUMENT  
    Type indicator for dependent argument.
```

FREE_ARGUMENT

```
public static final int FREE_ARGUMENT  
    Type indicator for free argument.
```

NOT_FOUND

```
public static final int NOT_FOUND
    When argument was not not found
```

NO_SYNTAX_ERRORS

```
public static final boolean NO_SYNTAX_ERRORS
    No syntax errors in the dependent argument definition.
```

RECURSIVE_ARGUMENT

```
public static final int RECURSIVE_ARGUMENT
    Type indicator for recursive argument.
```

SYNTAX_ERROR_OR_STATUS_UNKNOWN

```
public static final boolean SYNTAX_ERROR_OR_STATUS_UNKNOWN
    Syntax error in the dependent argument definition.
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
    Argument type id for the definition of key words known by the parser.
```

argumentExpression

[Expression](#) **argumentExpression**
Argument expression for dependent and recursive arguments.

argumentType

```
int argumentType
    Argument type (free, dependent)
```

argumentValue

double **argumentValue**
Argument value (for free arguments).

n

protected [Argument](#) **n**
Index argument.

Constructors

Argument

```
public Argument(java.lang.String argumentName,  
                double argumentValue)
```

Constructor - creates free argument.

Parameters:

argumentName - the argument name
argumentValue - the argument value

Argument

```
public Argument(java.lang.String argumentName,  
                java.lang.String argumentExpressionString,  
                org.mariuszgromada.math.mxparser.PrimitiveElement[] elements)
```

Constructor - creates dependent argument(with hidden argument expression).

Parameters:

argumentName - the argument name
argumentExpressionString - the argument expression string
elements - Optional parameters (comma separated) such as Arguments, Constants,
Functions

Argument

```
public Argument(java.lang.String argumentDefinitionString,  
                org.mariuszgromada.math.mxparser.PrimitiveElement[] elements)
```

Default constructor - creates argument based on the argument definition string.

Parameters:

argumentDefinitionString - Argument definition string, i.e.:

- 'x' - only argument name
- 'x=5' - argument name and argument value
- 'x=2*5' - argument name and argument value given as simple expression
- 'x=2*y' - argument name and argument expression (dependent argument 'x' on argument 'y')

elements - Optional parameters (comma separated) such as Arguments, Constants, Functions

Methods

addArguments

```
public void addArguments(org.mariuszgromada.math.mxparser.Argument[]  
arguments)
```

Adds arguments (variadic) to the argument expression definition.

Parameters:

arguments - the arguments list (comma separated list)

addConstants

```
public void addConstants(java.util.List constantsList)
```

Adds constants to the argument expression definition.

Parameters:

constantsList - the list of constants

addConstants

```
public void addConstants(org.mariuszgromada.math.mxparser.Constant[]  
constants)
```

Adds constants (variadic parameters) to the argument expression definition.

Parameters:

constants - the constants (comma separated list)

addDefinitions

```
public void addDefinitions(org.mariuszgromada.math.mxparser.PrimitiveElement[]  
elements)
```

Adds user defined elements (such as: Arguments, Constants, Functions) to the argument expressions.

Parameters:

elements - Elements list (variadic - comma separated) of types: Argument, Constant, Function

addFunctions

```
public void addFunctions(org.mariuszgromada.math.mxparser.Function[]  
functions)
```

Adds functions (variadic parameters) to the argument expression definition.

Parameters:

functions - the functions (variadic parameters) comma separated list

addRelatedExpression

```
void addRelatedExpression(Expression expression)
```

Adds related expression to the argumentExpression

Parameters:

expression - the related expression

checkSyntax

```
public boolean checkSyntax()
```

Checks argument syntax

Returns:

syntax status: Argument.NO_SYNTAX_ERRORS,
Argument.SYNTAX_ERROR_OR_STATUS_UNKNOWN

clone

```
public Argument clone()
```

Creates cloned object of the this argument."

Returns:

clone of the argument.

Overrides:

clone in class java.lang.Object

defineArgument

```
public void defineArgument(java.lang.String argumentName,  
                             double argumentValue)
```

Enables to define the argument (associated with the argument expression) based on the argument name and the argument value.

Parameters:

argumentName - the argument name

argumentValue - the the argument value

defineArguments

```
public void defineArguments(java.lang.String[] argumentsNames)
```

Enables to define the arguments (associated with the argument expression) based on the given arguments names.

Parameters:

argumentsNames - the arguments names (variadic) comma separated list

defineConstant

```
public void defineConstant(java.lang.String constantName,  
                             double constantValue)
```

Enables to define the constant (associated with the argument expression) based on the constant name and constant value.

Parameters:

constantName - the constant name

constantValue - the constant value

defineFunction

```
public void defineFunction(java.lang.String functionName,  
                           java.lang.String functionExpressionString,  
                           java.lang.String[] argumentsNames)
```

Enables to define the function (associated with the argument expression) based on the function name, function expression string and arguments names (variadic parameters).

Parameters:

functionName - the function name

functionExpressionString - the expression string

argumentsNames - the function arguments names (variadic parameters) comma separated list

getArgument

```
public Argument getArgument(int argumentIndex)
```

Gets argument from the argument expression.

Parameters:

argumentIndex - the argument index

Returns:

Argument if the argument index is between 0 and the last available argument index (getArgumentsNumber()-1), otherwise returns null.

getArgument

```
public Argument getArgument(java.lang.String argumentName)
```

Gets argument from the argument expression.

Parameters:

argumentName - the argument name

Returns:

The argument if the argument name was found, otherwise returns null.

getArgumentExpressionString

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

Gets argument expression string

Returns:

the argument expression string

getArgumentIndex

```
public int getArgumentIndex(java.lang.String argumentName)
```

Gets argument index from the argument expression.

Parameters:

argumentName - the argument name

Returns:

The argument index if the argument name was found, otherwise returns
Argument.NOT_FOUND

getArgumentName

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

Gets argument name

Returns:

the argument name as string

getArgumentType

```
public int getArgumentType()
```

Gets argument type

Returns:

Argument type: Argument.FREE_ARGUMENT, Argument.DEPENDENT_ARGUMENT,
Argument.RECURSIVE_ARGUMENT

getArgumentValue

```
public double getArgumentValue()
```

Gets argument value.

Returns:

direct argument value for free argument, otherwise returns calculated argument value
based on the argument expression.

getArgumentsNumber

```
public int getArgumentsNumber()
```

Gets number of arguments associated with the argument expression.

Returns:

The number of arguments (int >= 0)

getComputingTime

```
public double getComputingTime()
```

Gets computing time

Returns:

Computing time in seconds.

getConstant

```
public Constant getConstant(int constantIndex)
```

Gets constant associated with the argument expression.

Parameters:

constantIndex - the constant index

Returns:

Constant if the constantIndex is between 0 and the last available constant index (getConstantsNumber() - 1), otherwise it returns null.

getConstant

```
public Constant getConstant(java.lang.String constantName)
```

Gets constant associated with the argument expression.

Parameters:

constantName - the constant name

Returns:

Constant if constant name was found, otherwise return null.

getConstantIndex

```
public int getConstantIndex(java.lang.String constantName)
```

Gets constant index associated with the argument expression.

Parameters:

constantName - the constant name

Returns:

Constant index if constant name was found, otherwise return Constant.NOT_FOUND.

getConstantsNumber

```
public int getConstantsNumber()
```

Gets number of constants associated with the argument expression.

Returns:

number of constants (int >= 0)

getDescription

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

Gets argument description.

Returns:

The argument description string.

getErrorMessage

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

Returns error message after checking the syntax

Returns:

Error message as string.

getFunction

```
public Function getFunction(int functionIndex)
```

Gets function associated with the argument expression.

Parameters:

functionIndex - the function index

Returns:

Function if function index is between 0 and the last available function index (getFunctionsNumber()-1), otherwise returns null.

getFunction

```
public Function getFunction(java.lang.String functionName)
```

Gets function associated with the argument expression.

Parameters:

functionName - the function name

Returns:

Function if function name was found, otherwise returns null.

getFunctionIndex

```
public int getFunctionIndex(java.lang.String functionName)
```

Gets index of function associated with the argument expression.

Parameters:

functionName - the function name

Returns:

Function index if function name was found, otherwise returns Function.NOT_FOUND

getFunctionsNumber

```
public int getFunctionsNumber()
```

Gets number of functions associated with the argument expression.

Returns:

number of functions (int >= 0)

getRecursiveMode

```
public boolean getRecursiveMode()
```

Gets recursive mode status

Returns:

true if recursive mode is enabled, otherwise returns false

getVerboseMode

```
public boolean getVerboseMode()
```

Returns verbose mode status

Returns:

true if verbose mode is on, otherwise returns false.

removeAllArguments

```
public void removeAllArguments()
```

Removes all arguments associated with the argument expression.

removeAllConstants

```
public void removeAllConstants()
```

Removes all constants associated with the argument expression

removeAllFunctions

```
public void removeAllFunctions()
```

Removes all functions associated with the argument expression.

removeArguments

```
public void removeArguments(java.lang.String[] argumentsNames)
```

Removes first occurrences of the arguments associated with the argument expression.

Parameters:

argumentsNames - the arguments names (variadic parameters) comma separated list

removeArguments

```
public void removeArguments(org.mariuszgromada.math.mxparser.Argument[] arguments)
```

Removes first occurrences of the arguments associated with the argument expression.

Parameters:

arguments - the arguments (variadic parameters) comma separated list

removeConstants

```
public void removeConstants(java.lang.String[] constantsNames)
```

Removes first occurrences of the constants associated with the argument expression.

Parameters:

constantsNames - the constants names (variadic parameters) comma separated list

removeConstants

```
public void removeConstants(org.mariuszgromada.math.mxparser.Constant[] constants)
```

Removes first occurrences of the constants associated with the argument expression

Parameters:

constants - the constants (variadic parameters) comma separated list

removeDefinitions

```
public void removeDefinitions(org.mariuszgromada.math.mxparser.PrimitiveElement[] elements)
```

Removes user defined elements (such as: Arguments, Constants, Functions) from the argument expressions.

Parameters:

elements - Elements list (variadic - comma separated) of types: Argument, Constant, Function

removeFunctions

```
public void removeFunctions(java.lang.String[] functionsNames)
```

Removes first occurrences of the functions associated with the argument expression.

Parameters:

functionsNames - the functions names (variadic parameters) comma separated list

removeFunctions

```
public void removeFunctions(org.mariuszgromada.math.mxparser.Function[]  
functions)
```

Removes first occurrences of the functions associated with the argument expression.

Parameters:

functions - the functions (variadic parameters) comma separated list.

removeRelatedExpression

```
void removeRelatedExpression(Expression expression)
```

Adds related expression form the argumentExpression

Parameters:

expression - related expression

setArgumentExpressionString

```
public void setArgumentExpressionString(java.lang.String  
argumentExpressionString)
```

Sets argument expression string. Each expression / function / dependent argument associated with this argument will be marked as modified (requires new syntax checking).

Parameters:

argumentExpressionString - the argument expression string

setArgumentName

```
public void setArgumentName(java.lang.String argumentName)
```

Sets (modifies) argument name. Each expression / function / dependent argument associated with this argument will be marked as modified (requires new syntax checking).

Parameters:

argumentName - the argument name

setArgumentValue

```
public void setArgumentValue(double argumentValue)
```

Sets argument value

Parameters:

argumentValue - the value of argument

setDescription

```
public void setDescription(java.lang.String description)
```

Sets argument description.

Parameters:

description - the argument description.

setExpressionModifiedFlags

```
void setExpressionModifiedFlags()
```

Sets expression was modified flag to all related expressions to the argumentExpression.

setSilentMode

```
public void setSilentMode()
```

Disables argument verbose mode (sets default silent mode)

setVerboseMode

```
public void setVerboseMode()
```

Enables argument verbose mode

org.mariuszgromada.math.mxparser

Class ArgumentParameter

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.ArgumentParameter
```

< [Fields](#) > < [Constructors](#) >

class **ArgumentParameter**
extends java.lang.Object

Handling argument parameters

Fields

argument

[Argument](#) argument

index

int index

initialType

int initialType

initialValue

double initialValue

presence

int presence

Constructors

ArgumentParameter

`ArgumentParameter()`

org.mariuszgromada.math.mxparser

Class Constant

```
java.lang.Object
|
+-- PrimitiveElement
|
+-- org.mariuszgromada.math.mxparser.Constant
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

public class **Constant**
 extends [PrimitiveElement](#)

Constant class provides ability to declare constants. Constants can be used in further processing by any expression, dependent or recursive argument, function, etc...

When creating a constant you should avoid names reserved as parser keywords, in general words known in mathematical language as function names, operators (for example: sin, cos, +, -, pi, e, etc...). Please be informed that after associating the constant with the expression, function or dependent/recursive argument its name will be recognized by the parser as reserved key word. It means that it could not be the same as any other key word known by the parser for this particular expression.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.0.0

RecursiveArgument

Expression

Function

Argument

Fields

NOT_FOUND

```
public static final int NOT_FOUND
    When constant could not be found
```

NO_SYNTAX_ERRORS

```
public static final boolean NO_SYNTAX_ERRORS
    Status of the Expression syntax
```

SYNTAX_ERROR_OR_STATUS_UNKNOWN

```
public static final boolean SYNTAX_ERROR_OR_STATUS_UNKNOWN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
    Type identifier for constants
```

Constructors

Constant

```
public Constant(java.lang.String constantName,
                double constantValue)
```

Constructor - creates constant with a given name and given value

Parameters:

constantName - the constant name
constantValue - the constant value

Constant

```
public Constant(java.lang.String constantName,
                 double constantValue,
                 java.lang.String description)
```

Constructor - creates constant with a given name and given value. Additionally description is being set.

Parameters:

constantName - the constant name
 constantValue - the constant value
 description - the constant description

Constant

```
public Constant(java.lang.String constantDefinitionString,
                 org.mariuszgromada.math.mxparser.PrimitiveElement[] elements)
```

Constructor for function definition in natural math language, for instance providing on string "f(x,y) = sin(x) + cos(x)" is enough to define function "f" with parameters "x and y" and function body "sin(x) + cos(x)".

Parameters:

constantDefinitionString - Constant definition in the form of one String, ie "c = 2" or "c = 2*sin(pi/3)"
 elements - Optional parameters (comma separated) such as Arguments, Constants, Functions

Methods

addRelatedExpression

```
void addRelatedExpression(Expression expression)
```

Adds related expression.

Parameters:

expression - the related expression.

getConstantName

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

Gets constant name

Returns:

the constant name as string.

getConstantValue

```
public double getConstantValue()
```

Gets constant value.

Returns:

constant value as double

getDescription

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

Gets constant description.

Returns:

constant description as string.

getErrorMessage

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

Method return error message after

Returns:

Error message as string.

getSyntaxStatus

```
public boolean getSyntaxStatus()
```

Gets syntax status of the expression.

Returns:

Constant.NO_SYNTAX_ERRORS if there are no syntax errors,
Const.SYNTAX_ERROR_OR_STATUS_UNKNOWN when syntax error was found or
syntax status is unknown

removeRelatedExpression

```
void removeRelatedExpression(Expression expression)
```

Removes related expression.

Parameters:

expression - the related expression.

setConstantName

```
public void setConstantName(java.lang.String constantName)
```

Sets constant name. If constant is associated with any expression then this operation will set modified flag to each related expression.

Parameters:

constantName - the constant name

setDescription

```
public void setDescription(java.lang.String description)
```

Sets constant description.

Parameters:

description - the constant description

setExpressionModifiedFlags

```
void setExpressionModifiedFlags()
```

Sets expression modified flag to each related expression.

org.mariuszgromada.math.mxparser

Class DescKwLenComparator

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.DescKwLenComparator
```

All Implemented Interfaces:

java.util.Comparator

< [Constructors](#) > < [Methods](#) >

```
class DescKwLenComparator
extends java.lang.Object
implements java.util.Comparator
```

Comparator for key word list sorting by descending key word length . This kind of sorting is used while tokenizing (best match)

Constructors

DescKwLenComparator

`DescKwLenComparator()`

Methods

compare

```
public int compare(Keyword kw1,  
                  Keyword kw2)
```

`org.mariuszgromada.math.mxparser`

Class Expression

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.Expression
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class Expression  
extends java.lang.Object
```

Expression - base class for real expressions definition. Examples:

- '1+2'
- 'sin(x)+1'
- 'asin(3*x)^10-log(4,8)'
- in general 'f(x1,x2,...,xn)' where x1,...,xn are real arguments

Class provides easy way to define multivariate arithmetic expression.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](https://bitbucket.org/mariuszgromada/mxparser/)

[mXparser on CodePlex](https://codeplex.com/mxparser/)

[Janet Sudoku - project web page](http://janet.sudoku.pl)

[Janet Sudoku on GitHub](https://github.com/mariuszgromada/janet-sudoku)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Argument

RecursiveArgument

Constant

Function

Fields

DISABLE_ULP_ROUNDING

```
static final boolean DISABLE_ULP_ROUNDING
```

FOUND

```
static final int FOUND
```

INTERNAL

```
static final boolean INTERNAL  
Marker for internal processing
```

KEEP_ULP_ROUNDING_SETTINGS

```
static final boolean KEEP_ULP_ROUNDING_SETTINGS
```

NOT_FOUND

```
static final int NOT_FOUND  
FOUND / NOT_FOUND used for matching purposes
```

NO_SYNTAX_ERRORS

```
public static final boolean NO_SYNTAX_ERRORS  
Status of the Expression syntax
```

SYNTAX_ERROR_OR_STATUS_UNKNOWN

public static final boolean **SYNTAX_ERROR_OR_STATUS_UNKNOWN**

argumentsList

java.util.List **argumentsList**
List of arguments

constantsList

java.util.List **constantsList**
List of user defined constants

disableUlpRounding

boolean **disableUlpRounding**
Internal parameter for calculus expressions to avoid decrease in accuracy.

expressionString

java.lang.String **expressionString**
Expression string (for example: "sin(x)+cos(y)")

functionsList

java.util.List **functionsList**
List of user defined functions

recursiveMode

boolean **recursiveMode**
If recursive mode is on the recursive calls are permitted. It means there will be no null pointer exceptions due to expression, and functions cloning.

relatedExpressionsList

java.util.List **relatedExpressionsList**
List of related expressions, for example when user defined function is used in the expression or dependent argument was defined. Modification of function expression calls the method expression

modified flag method to all related expressions. Related expression usually are used for -
dependent arguments - recursive arguments - user functions

Constructors

Expression

```
Expression(java.lang.String expressionString,  
            boolean parserKeyWordsOnly)
```

Constructor - creates new expression from expression string.

Parameters:

expressionString - definition of the expression

parserKeyWordsOnly - if true then all keywords such as functions, constants, arguments
will not be recognized.

Expression

```
Expression(java.lang.String expressionString,  
            java.util.List argumentsList,  
            java.util.List functionsList,  
            java.util.List constantsList,  
            boolean internal)
```

Package level constructor - creates new expression from expression string, arguments list,
functions list and constants list (used by the RecursiveArgument class). No related expressions at
the beginning.

Parameters:

expressionString - the expression string

argumentsList - the arguments list

functionsList - the functions list

constantsList - the constants list

internal - the marker for internal processing

Expression

```
Expression(java.lang.String expressionString,  
            java.util.List initialTokens,  
            java.util.List argumentsList,  
            java.util.List functionsList,  
            java.util.List constantsList,  
            boolean disableUlpRounding)
```

Package level constructor - creates new expression from subexpression (sublist of the tokens list), arguments list, functions list and constants list (used by the internal calculus operations, etc...).

Parameters:

expressionString - the expression string
initialTokens - the tokens list (starting point - no tokenizing, no syntax checking)
argumentsList - the arguments list
functionsList - the functions list
constantsList - the constants list

Expression

```
public Expression(java.lang.String expressionString,  
                  org.mariuszgromada.math.mxparser.PrimitiveElement[]  
elements)
```

Constructor - creates new expression from expression string.

Parameters:

expressionString - definition of the expression
elements - Optional elements list (variadic - comma separated) of types: Argument, Constant, Function

Expression

```
public Expression(org.mariuszgromada.math.mxparser.PrimitiveElement[]  
elements)
```

Default constructor - empty expression

Parameters:

elements - Optional elements list (variadic - comma separated) of types: Argument, Constant, Function

Methods

addArguments

```
public void addArguments(org.mariuszgromada.math.mxparser.Argument[] arguments)
```

Adds arguments (variadic) to the expression definition.

Parameters:

arguments - the arguments list (comma separated list)

addConstants

```
public void addConstants(java.util.List constantsList)
```

Adds constants to the expression definition.

Parameters:

constantsList - the list of constants

addConstants

```
public void addConstants(org.mariuszgromada.math.mxparser.Constant[] constants)
```

Adds constants (variadic parameters) to the expression definition.

Parameters:

constants - the constants (comma separated list)

addDefinitions

```
public void addDefinitions(org.mariuszgromada.math.mxparser.PrimitiveElement[] elements)
```

Adds user defined elements (such as: Arguments, Constants, Functions) to the expressions.

Parameters:

elements - Elements list (variadic), where Argument, Constant, Function extend the same class PrimitiveElement

addFunctions

```
public void addFunctions(org.mariuszgromada.math.mxparser.Function[] functions)
```

Adds functions (variadic parameters) to the expression definition.

Parameters:

functions - the functions (variadic parameters) comma separated list

addRelatedExpression

```
void addRelatedExpression(Expression expression)
```

Adds related expression The same expression could be added more than once For example when

Parameters:

expression - the expression

calculate

```
public double calculate()
```

Calculates the expression value

Returns:

The expression value if syntax was ok, otherwise returns Double.NaN.

checkLexSyntax

```
public boolean checkLexSyntax()
```

Checks syntax of the expression string.

Returns:

true if syntax is ok

checkSyntax

```
public boolean checkSyntax()
```

Checks syntax of the expression string.

Returns:

true if syntax is ok

clearDescription

```
public void clearDescription()
```

Clears expression description

clearExpressionString

```
public void clearExpressionString()
```

Clears expression string

clone

```
protected Expression clone()
```

Expression cloning.

Overrides:

clone in class java.lang.Object

defineArgument

```
public void defineArgument(java.lang.String argumentName,  
                           double argumentValue)
```

Enables to define the argument (associated with the expression) based on the argument name and the argument value.

Parameters:

argumentName - the argument name

argumentValue - the the argument value

defineArguments

```
public void defineArguments(java.lang.String[] argumentsNames)
```

Enables to define the arguments (associated with the expression) based on the given arguments names.

Parameters:

argumentsNames - the arguments names (variadic) comma separated list

defineConstant

```
public void defineConstant(java.lang.String constantName,  
                           double constantValue)
```

Enables to define the constant (associated with the expression) based on the constant name and constant value.

Parameters:

constantName - the constant name
constantValue - the constant value

defineFunction

```
public void defineFunction(java.lang.String functionName,  
                           java.lang.String functionExpressionString,  
                           java.lang.String[] argumentsNames)
```

Enables to define the function (associated with the expression) based on the function name, function expression string and arguments names (variadic parameters).

Parameters:

functionName - the function name
functionExpressionString - the expression string
argumentsNames - the function arguments names (variadic parameters) comma separated list

disableRecursiveMode

```
void disableRecursiveMode()
```

Disables recursive mode

getArgument

```
public Argument getArgument(int argumentIndex)
```

Gets argument from the expression.

Parameters:

argumentIndex - the argument index

Returns:

Argument if the argument index is between 0 and the last available argument index (getArgumentsNumber()-1), otherwise returns null.

getArgument

```
public Argument getArgument(java.lang.String argumentName)
```

Gets argument from the expression.

Parameters:

argumentName - the argument name

Returns:

The argument if the argument name was found, otherwise returns null.

getArgumentIndex

```
public int getArgumentIndex(java.lang.String argumentName)
```

Gets argument index from the expression.

Parameters:

argumentName - the argument name

Returns:

The argument index if the argument name was found, otherwise returns
Argument.NOT_FOUND

getArgumentValue

```
public double getArgumentValue(java.lang.String argumentName)
```

Gets argument value.

Parameters:

argumentName - the argument name

Returns:

Argument value if argument name was found, otherwise return Double.NaN.

getArgumentsNumber

```
public int getArgumentsNumber()
```

Gets number of arguments associated with the expression.

Returns:

The number of arguments (int >= 0)

getComputingTime

```
public double getComputingTime()
```

Gets computing time.

Returns:

computing time in seconds.

getConstant

```
public Constant getConstant(int constantIndex)
```

Gets constant associated with the expression.

Parameters:

constantIndex - the constant index

Returns:

Constant if the constantIndex is between 0 and the last available constant index (getConstantsNumber() - 1), otherwise it returns null.

getConstant

```
public Constant getConstant(java.lang.String constantName)
```

Gets constant associated with the expression.

Parameters:

constantName - the constant name

Returns:

Constant if constant name was found, otherwise return null.

getConstantIndex

```
public int getConstantIndex(java.lang.String constantName)
```

Gets constant index associated with the expression.

Parameters:

constantName - the constant name

Returns:

Constant index if constant name was found, otherwise return Constant.NOT_FOUND.

getConstantsNumber

```
public int getConstantsNumber()
```

Gets number of constants associated with the expression.

Returns:

number of constants (int >= 0)

getCopyOfInitialTokens

```
public java.util.List getCopyOfInitialTokens()
```

Tokenizes expression string and returns tokens list, including: string, type, level.

Returns:

Copy of initial tokens.

getDescription

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

Gets expression description.

Returns:

String description.

getErrorMessage

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

Method return error message after calling checkSyntax() method or calculate().

Returns:

Error message as string.

getExpressionString

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

Returns expression string

Returns:

Expression string definition.

getFunction

```
public Function getFunction(int functionIndex)
```

Gets function associated with the expression.

Parameters:

functionIndex - the function index

Returns:

Function if function index is between 0 and the last available function index (getFunctionsNumber()-1), otherwise returns null.

getFunction

```
public Function getFunction(java.lang.String functionName)
```

Gets function associated with the expression.

Parameters:

functionName - the function name

Returns:

Function if function name was found, otherwise returns null.

getFunctionIndex

```
public int getFunctionIndex(java.lang.String functionName)
```

Gets index of function associated with the expression.

Parameters:

functionName - the function name

Returns:

Function index if function name was found, otherwise returns Function.NOT_FOUND

getFunctionsNumber

```
public int getFunctionsNumber()
```

Gets number of functions associated with the expression.

Returns:

number of functions (int >= 0)

getHelp

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

Gets help content.

Returns:

The help content.

getHelp

```
public java.lang.String getHelp(java.lang.String word)
```

Searching help content.

Parameters:

word - searching key word

Returns:

The help content.

getInitialTokens

```
java.util.List getInitialTokens()
```

Gets initial tokens and returns copied list

getKeyWords

```
public java.util.List getKeyWords()
```

Returns list of key words known to the parser

Returns:

List of keywords known to the parser.

getKeyWords

```
public java.util.List getKeyWords(java.lang.String query)
```

Returns list of key words known to the parser

Parameters:

query - Give any string to filter list of key words against this string. User more precise syntax: str=tokenString, desc=tokenDescription, syn=TokenSyntax, sin=tokenSince, wid=wordId, tid=wordTypeId to narrow the result.

Returns:

List of keywords known to the parser filter against query string.

getRecursiveMode

```
public boolean getRecursiveMode()
```

Gets recursive mode status

Returns:

true if recursive mode is enabled, otherwise returns false.

getSyntaxStatus

```
public boolean getSyntaxStatus()
```

Gets syntax status of the expression.

Returns:

true if there are no syntax errors, false when syntax error was found or syntax status is unknown

getVerboseMode

```
public boolean getVerboseMode()
```

Returns verbose mode status.

Returns:

true if verbose mode is on, otherwise returns false.

removeAllArguments

```
public void removeAllArguments()
```

Removes all arguments associated with the expression.

removeAllConstants

```
public void removeAllConstants()
```

Removes all constants associated with the expression

removeAllFunctions

```
public void removeAllFunctions()
```

Removes all functions associated with the expression.

removeArguments

```
public void removeArguments(java.lang.String[] argumentsNames)
```

Removes first occurrences of the arguments associated with the expression.

Parameters:

argumentsNames - the arguments names (variadic parameters) comma separated list

removeArguments

```
public void removeArguments(org.mariuszgromada.math.mxparser.Argument[] arguments)
```

Removes first occurrences of the arguments associated with the expression.

Parameters:

arguments - the arguments (variadic parameters) comma separated list

removeConstants

```
public void removeConstants(java.lang.String[] constantsNames)
```

Removes first occurrences of the constants associated with the expression.

Parameters:

constantsNames - the constants names (variadic parameters) comma separated list

removeConstants

```
public void removeConstants(org.mariuszgromada.math.mxparser.Constant[] constants)
```

Removes first occurrences of the constants associated with the expression

Parameters:

constants - the constants (variadic parameters) comma separated list

removeDefinitions

```
public void removeDefinitions(org.mariuszgromada.math.mxparser.PrimitiveElement[] elements)
```

Removes user defined elements (such as: Arguments, Constants, Functions) to the expressions.

Parameters:

elements - Elements list (variadic), where Argument, Constant, Function extend the same class PrimitiveElement

removeFunctions

```
public void removeFunctions(java.lang.String[] functionsNames)
```

Removes first occurrences of the functions associated with the expression.

Parameters:

functionsNames - the functions names (variadic parameters) comma separated list

removeFunctions

```
public void removeFunctions(org.mariuszgromada.math.mxparser.Function[] functions)
```

Removes first occurrences of the functions associated with the expression.

Parameters:

functions - the functions (variadic parameters) comma separated list.

removeRelatedExpression

```
void removeRelatedExpression(Expression expression)
```

Removes related expression

Parameters:

expression - the expression

setArgumentValue

```
public void setArgumentValue(java.lang.String argumentName,  
                             double argumentValue)
```

Sets argument value.

Parameters:

argumentName - the argument name

argumentValue - the argument value

setDescription

```
public void setDescription(java.lang.String description)
```

Sets expression description.

Parameters:

description - the description string

setExpressionModifiedFlag

```
void setExpressionModifiedFlag()
```

Sets expression status to modified Calls setExpressionModifiedFlag() method to all related expressions.

setExpressionString

```
public void setExpressionString(java.lang.String expressionString)
```

Sets (modifies expression) expression string.

Parameters:

expressionString - the expression string

setRecursiveMode

```
void setRecursiveMode()
```

Sets recursive mode

setSilentMode

```
public void setSilentMode()
```

Disables verbose mode (default silent mode).

setSyntaxStatus

```
void setSyntaxStatus(boolean syntaxStatus,  
                     java.lang.String errorMessage)
```

Package level method for passing information about errors identified on the constructors level

Parameters:

syntaxStatus - Syntax status

errorMessage - Error message

setVerboseMode

```
public void setVerboseMode()
```

Enables verbose mode.

showInitialTokens

```
void showInitialTokens()
```

shows initial tokens

showKeywords

```
void showKeywords()
```

shows known keywords

showRelatedExpressions

```
void showRelatedExpressions()
```

Prints related expression list

showTokens

```
void showTokens()
```

showTokens

```
static final void showTokens(java.util.List tokensList)
```

org.mariuszgromada.math.mxparser

Class Function

```
java.lang.Object
|
+-- PrimitiveElement
|
+-- org.mariuszgromada.math.mxparser.Function
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class Function
extends PrimitiveElement
```

Function class provides possibility to define user functions. Functions can be used in further processing by any expression, dependent or recursive argument, function, etc... For example:

- 'f(x) = sin(x)'
- 'g(x,y) = sin(x)+cos(y)'
- 'h(x,y = f(x)+g(y,x)'
- in general 'f(x1,x2,...,xn)' where x1,...,xn are arguments

When creating a function you should avoid names reserved as parser keywords, in general words known in mathematical language as function names, operators (for example: sin, cos, +, -, pi, e, etc...). Please be informed that after associating the constant with the expression, function or dependent/recursive argument its name will be recognized by the parser as reserved key word. It means that it could not be the same as any other key word known by the parser for this particular expression.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

[MathSpace.pl](#)

[MathParser.org - mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

RecursiveArgument

Expression

Argument

Constant

FunctionExtension

Fields

BODY_EXTENDED

```
public static final int BODY_EXTENDED
    Function with body based on the extended code.
```

BODY_RUNTIME

```
public static final int BODY_RUNTIME
    Function with body based on the expression string.
```

NOT_FOUND

```
public static final int NOT_FOUND
    When function was not found
```

NO_SYNTAX_ERRORS

```
public static final boolean NO_SYNTAX_ERRORS
    No syntax errors in the function.
```

SYNTAX_ERROR_OR_STATUS_UNKNOWN

```
public static final boolean SYNTAX_ERROR_OR_STATUS_UNKNOWN
    Syntax error in the function or syntax status unknown.
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
    Function type id identifier
```

functionExpression

[Expression](#) **functionExpression**
function expression

Constructors

Function

```
public Function(java.lang.String functionName,
                java.lang.String functionExpressionString,
                java.lang.String[] argumentsNames)
```

Constructor - creates function from function name, function expression string and argument names.

Parameters:

functionName - the function name

functionExpressionString - the function expression string

argumentsNames - the arguments names (variadic parameters) comma separated list

Function

```
public Function(java.lang.String functionName,
                java.lang.String functionExpressionString,
                org.mariuszgromada.math.mxparser.PrimitiveElement[] elements)
```

Constructor - creates function from function name and function expression string.

Parameters:

functionName - the function name
 functionExpressionString - the function expression string
 elements - Optional elements list (variadic - comma separated) of types: Argument, Constant, Function

Function

```
public Function(java.lang.String functionName,
                FunctionExtension functionExtension)
```

Constructor for function definition based on your own source code - this is via implementation of FunctionExtension interface.

Parameters:

functionName - Function name
 functionExtension - Your own source code

Function

```
public Function(java.lang.String functionDefinitionString,
                org.mariuszgromada.math.mxparser.PrimitiveElement[] elements)
```

Constructor for function definition in natural math language, for instance providing on string "f(x,y) = sin(x) + cos(x)" is enough to define function "f" with parameters "x and y" and function body "sin(x) + cos(x)".

Parameters:

functionDefinitionString - Function definition in the form of one String, ie "f(x,y) = sin(x) + cos(x)"
 elements - Optional elements list (variadic - comma separated) of types: Argument, Constant, Function

Methods

addArguments

```
public void addArguments(org.mariuszgromada.math.mxpaser.Argument[] arguments)
```

Adds arguments (variadic) to the function expression definition.

Parameters:

arguments - the arguments list (comma separated list)

addConstants

```
public void addConstants(java.util.List constantsList)
```

Adds constants to the function expression definition.

Parameters:

constantsList - the list of constants

addConstants

```
public void addConstants(org.mariuszgromada.math.mxpaser.Constant[] constants)
```

Adds constants (variadic parameters) to the function expression definition.

Parameters:

constants - the constants (comma separated list)

addDefinitions

```
public void addDefinitions(org.mariuszgromada.math.mxpaser.PrimitiveElement[] elements)
```

Adds user defined elements (such as: Arguments, Constants, Functions) to the function expressions.

Parameters:

elements - Elements list (variadic), where Argument, Constant, Function extend the same class PrimitiveElement

addFunctions

```
public void addFunctions(org.mariuszgromada.math.mxparser.Function[] functions)
```

Adds functions (variadic parameters) to the function expression definition.

Parameters:

functions - the functions (variadic parameters) comma separated list

addRelatedExpression

```
void addRelatedExpression(Expression expression)
```

Adds related expression.

Parameters:

expression - the related expression

calculate

```
public double calculate()
```

Calculates function value

Returns:

Function value as double.

calculate

```
public double calculate(double[] params)
```

Calculates function value

Parameters:

params - the function parameters values (as doubles)

Returns:

function value as double.

calculate

```
public double calculate(org.mariuszgromada.math.mxparser.Argument[] arguments)
```

Calculates function value

Parameters:

arguments - function parameters (as Arguments)

Returns:

function value as double

checkRecursiveMode

```
void checkRecursiveMode()
```

Checks whether function name appears in function body if yes the recursive mode is being set

checkSyntax

```
public boolean checkSyntax()
```

Checks function syntax

Returns:

syntax status: Function.NO_SYNTAX_ERRORS,
Function.SYNTAX_ERROR_OR_STATUS_UNKNOWN

clone

```
protected Function clone()
```

clone method

Overrides:

clone in class java.lang.Object

defineArgument

```
public void defineArgument(java.lang.String argumentName,  
                           double argumentValue)
```

Enables to define the argument (associated with the function expression) based on the argument name and the argument value.

Parameters:

argumentName - the argument name
argumentValue - the the argument value

defineArguments

```
public void defineArguments(java.lang.String[] argumentsNames)
```

Enables to define the arguments (associated with the function expression) based on the given arguments names.

Parameters:

argumentsNames - the arguments names (variadic) comma separated list

defineConstant

```
public void defineConstant(java.lang.String constantName,  
                             double constantValue)
```

Enables to define the constant (associated with the function expression) based on the constant name and constant value.

Parameters:

constantName - the constant name

constantValue - the constant value

defineFunction

```
public void defineFunction(java.lang.String functionName,  
                             java.lang.String functionExpressionString,  
                             java.lang.String[] argumentsNames)
```

Enables to define the function (associated with the function expression) based on the function name, function expression string and arguments names (variadic parameters).

Parameters:

functionName - the function name

functionExpressionString - the expression string

argumentsNames - the function arguments names (variadic parameters) comma separated list

getArgument

```
public Argument getArgument(int argumentIndex)
```

Gets argument from the function expression.

Parameters:

argumentIndex - the argument index

Returns:

Argument if the argument index is between 0 and the last available argument index (getArgumentsNumber()-1), otherwise returns null.

getArgument

```
public Argument getArgument(java.lang.String argumentName)
```

Gets argument from the function expression.

Parameters:

argumentName - the argument name

Returns:

The argument if the argument name was found, otherwise returns null.

getArgumentIndex

```
public int getArgumentIndex(java.lang.String argumentName)
```

Gets argument index from the function expression.

Parameters:

argumentName - the argument name

Returns:

The argument index if the argument name was found, otherwise returns
Argument.NOT_FOUND

getArgumentsNumber

```
public int getArgumentsNumber()
```

Gets number of arguments associated with the function expression.

Returns:

The number of arguments (int >= 0)

getComputingTime

```
public double getComputingTime()
```

Gets computing time

Returns:

computing time in seconds.

getConstant

```
public Constant getConstant(int constantIndex)
```

Gets constant associated with the function expression.

Parameters:

constantIndex - the constant index

Returns:

Constant if the constantIndex is between 0 and the last available constant index (getConstantsNumber() - 1), otherwise it returns null.

getConstant

```
public Constant getConstant(java.lang.String constantName)
```

Gets constant associated with the function expression.

Parameters:

constantName - the constant name

Returns:

Constant if constant name was found, otherwise return null.

getConstantIndex

```
public int getConstantIndex(java.lang.String constantName)
```

Gets constant index associated with the function expression.

Parameters:

constantName - the constant name

Returns:

Constant index if constant name was found, otherwise return Constant.NOT_FOUND.

getConstantsNumber

```
public int getConstantsNumber()
```

Gets number of constants associated with the function expression.

Returns:

number of constants (int >= 0)

getDescription

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

Gets function description

Returns:

Function description as string.

getErrorMessage

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

Returns error message after checking the syntax.

Returns:

Error message as string.

getFunction

```
public Function getFunction(int functionIndex)
```

Gets function associated with the function expression.

Parameters:

functionIndex - the function index

Returns:

Function if function index is between 0 and the last available function index (getFunctionsNumber()-1), otherwise returns null.

getFunction

```
public Function getFunction(java.lang.String functionName)
```

Gets function associated with the function expression.

Parameters:

functionName - the function name

Returns:

Function if function name was found, otherwise returns null.

getFunctionBodyType

```
public int getFunctionBodyType()
```

Returns function body type: {@link Function#BODY_RUNTIME} {@link Function#BODY_EXTENDED}

Returns:

Returns function body type: {@link Function#BODY_RUNTIME} {@link Function#BODY_EXTENDED}

getFunctionExpressionString

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

Gets function expression string

Returns:

Function expression as string.

getFunctionIndex

```
public int getFunctionIndex(java.lang.String functionName)
```

Gets index of function associated with the function expression.

Parameters:

functionName - the function name

Returns:

Function index if function name was found, otherwise returns Function.NOT_FOUND

getFunctionName

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

Gets function name.

Returns:

Function name as string.

getFunctionsNumber

```
public int getFunctionsNumber()
```

Gets number of functions associated with the function expression.

Returns:

number of functions (int >= 0)

getParameterName

```
public java.lang.String getParameterName(int parameterIndex)
```

Gets user defined function parameter name

Parameters:

parameterIndex - Parameter index between 0 and n-1

Returns:

If parameter exists returns parameters name, otherwise empty string is returned.

getParametersNumber

```
public int getParametersNumber()
```

Gets number of parameters associated with the function expression.

Returns:

The number of function parameters (int >= 0)

getRecursiveMode

```
public boolean getRecursiveMode()
```

Gets recursive mode status

Returns:

true if recursive mode is enabled, otherwise returns false

getVerboseMode

```
public boolean getVerboseMode()
```

Returns verbose mode status

Returns:

true if verbose mode is on, otherwise returns false

removeAllArguments

```
public void removeAllArguments()
```

Removes all arguments associated with the function expression.

removeAllConstants

```
public void removeAllConstants()
```

Removes all constants associated with the function expression

removeAllFunctions

```
public void removeAllFunctions()
```

Removes all functions associated with the function expression.

removeArguments

```
public void removeArguments(java.lang.String[] argumentsNames)
```

Removes first occurrences of the arguments associated with the function expression.

Parameters:

argumentsNames - the arguments names (variadic parameters) comma separated list

removeArguments

```
public void removeArguments(org.mariuszgromada.math.mxparser.Argument[] arguments)
```

Removes first occurrences of the arguments associated with the function expression.

Parameters:

arguments - the arguments (variadic parameters) comma separated list

removeConstants

```
public void removeConstants(java.lang.String[] constantsNames)
```

Removes first occurrences of the constants associated with the function expression.

Parameters:

constantsNames - the constants names (variadic parameters) comma separated list

removeConstants

```
public void removeConstants(org.mariuszgromada.math.mxparser.Constant[] constants)
```

Removes first occurrences of the constants associated with the function expression

Parameters:

constants - the constants (variadic parameters) comma separated list

removeDefinitions

```
public void removeDefinitions(org.mariuszgromada.math.mxparser.PrimitiveElement[] elements)
```

Removes user defined elements (such as: Arguments, Constants, Functions) from the function expressions.

Parameters:

elements - Elements list (variadic), where Argument, Constant, Function extend the same class PrimitiveElement

removeFunctions

```
public void removeFunctions(java.lang.String[] functionsNames)
```

Removes first occurrences of the functions associated with the function expression.

Parameters:

functionsNames - the functions names (variadic parameters) comma separated list

removeFunctions

```
public void removeFunctions(org.mariuszgromada.math.mxparser.Function[] functions)
```

Removes first occurrences of the functions associated with the function expression.

Parameters:

functions - the functions (variadic parameters) comma separated list.

removeRelatedExpression

```
void removeRelatedExpression(Expression expression)
```

Removes related expression.

Parameters:

expression - the related expression

setArgumentValue

```
public void setArgumentValue(int argumentIndex,  
                             double argumentValue)
```

Sets value of function argument (function parameter).

Parameters:

argumentIndex - the argument index (in accordance to arguments declaration sequence)
argumentValue - the argument value

setDescription

```
public void setDescription(java.lang.String description)
```

Sets function description.

Parameters:

description - the function description

setExpressionModifiedFlags

```
void setExpressionModifiedFlags()
```

Set expression modified flags in the related expressions.

setFunctionName

```
public void setFunctionName(java.lang.String functionName)
```

Sets function name.

Parameters:

functionName - the function name

setParametersNumber

```
public void setParametersNumber(int parametersNumber)
```

Set parameters number.

Parameters:

parametersNumber - the number of function parameters (default = number of arguments (less number might be specified)).

setSilentMode

```
public void setSilentMode()
```

Disables function verbose mode (sets default silent mode)

setVerboseMode

```
public void setVerboseMode()
```

Enables verbose function mode

org.mariuszgromada.math.mxparser

Interface FunctionExtension

< [Methods](#) >

```
public interface FunctionExtension
```

FunctionExtension provides interface for function algorithm definition. In this case algorithm definition is based on source code using JAVA (for JAVA / Android) or .NET. If implemented Function Extension object can be further used while Function object construction, which means it can extend mXparser math collection. mXparser extension with your own implementation can be achieved by implementing FunctionExtension interface, creating an FunctionExtension object, creating Function object based on FunctionExtension, adding Function object to Expression / mXparser definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](https://bitbucket.org/mariuszgromada/mxparser/)

[mXparser on CodePlex](https://www.codeplex.com/mxparser)

[Janet Sudoku - project web page](http://janet.sudoku.pl)

[Janet Sudoku on GitHub](https://github.com/mariuszgromada/JanetSudoku)

[Janet Sudoku on CodePlex](https://www.codeplex.com/JanetSudoku)

[Janet Sudoku on SourceForge](https://sourceforge.net/projects/janetsudoku/)

[Janet Sudoku on BitBucket](https://bitbucket.org/mariuszgromada/janetsudoku/)

Version:

4.1.0

Function

Methods

calculate

```
public double calculate()
```

Actual algorithm implementation.

Parameters:

parameters - Function parameters.

Returns:

Function Extension value.

clone

```
public FunctionExtension clone()
```

Cloning in case of usage in Expression with recursive statements.

Returns:

Returns FunctionExtension object that was cloned.

getParameterName

```
public java.lang.String getParameterName(int parameterIndex)
```

Gets parameter name

Parameters:

parameterIndex - - parameter index (from 0 to n-1)

Returns:

Returns parameter name

getParametersNumber

```
public int getParametersNumber()
```

Gets parameters number.

Returns:

Returns parameters number.

setParameterValue

```
public void setParameterValue(int parameterIndex,  
                                double parameterValue)
```

Sets value of function parameter

Parameters:

parameterIndex - - parameter index (from 0 to n-1)

parameterValue - - parameter value

org.mariuszgromada.math.mxparser

Class FunctionParameter

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.FunctionParameter
```

< [Fields](#) > < [Constructors](#) >

```
class FunctionParameter  
extends java.lang.Object
```

Package level class for handling function parameters.

Fields

fromIndex

```
int fromIndex
```

paramStr

```
java.lang.String paramStr
```

toIndex

```
int toIndex
```

tokens

```
java.util.List tokens
```

Constructors

FunctionParameter

```
FunctionParameter(java.util.List tokens,  
                  java.lang.String paramStr,  
                  int fromIndex,  
                  int toIndex)
```

org.mariuszgromada.math.mxparser

Class HeadEqBody

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.HeadEqBody
```

[< Fields](#) > [< Constructors](#) >

```
class HeadEqBody  
extends java.lang.Object
```

Fields

bodyStr

java.lang.String **bodyStr**

definitionError

boolean **definitionError**

eqPos

int **eqPos**

headStr

java.lang.String **headStr**

headTokens

java.util.List **headTokens**

Constructors

HeadEqBody

HeadEqBody(java.lang.String definitionString)

org.mariuszgromada.math.mxparser

Class IterativeOperatorParameters

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.IterativeOperatorParameters
```

< [Fields](#) > < [Constructors](#) >

class **IterativeOperatorParameters**
extends java.lang.Object

Package level class for generating iterative operator parameters

Fields

delta

double **delta**

deltaExp

[Expression](#) **deltaExp**

deltaParam

[FunctionParameter](#) **deltaParam**

from

double **from**

fromExp

[Expression](#) **fromExp**

fromParam

[FunctionParameter](#) **fromParam**

funExp

[Expression](#) **funExp**

funParam

[FunctionParameter](#) **funParam**

indexParam

[FunctionParameter](#) `indexParam`

to

double `to`

toExp

[Expression](#) `toExp`

toParam

[FunctionParameter](#) `toParam`

withDelta

boolean `withDelta`

Constructors

IterativeOperatorParameters

```
IterativeOperatorParameters(java.util.List functionParameters)
```

org.mariuszgromada.math.mxparser

Class KwStrComparator

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.KwStrComparator
```

All Implemented Interfaces:

java.util.Comparator

< [Constructors](#) > < [Methods](#) >

```
class KwStrComparator
extends java.lang.Object
implements java.util.Comparator
```

Comparator for key word list sorting by key word string. This kind of sorting is used while checking the syntax (duplicated key word error)

Constructors

KwStrComparator

```
KwStrComparator()
```

Methods

compare

```
public int compare(Keyword kw1,  
                  Keyword kw2)
```

org.mariuszgromada.math.mxparser

Class KwTypeComparator

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.KwTypeComparator
```

All Implemented Interfaces:

java.util.Comparator

< [Constructors](#) > < [Methods](#) >

```
class KwTypeComparator  
extends java.lang.Object  
implements java.util.Comparator
```

Comparator for key word list sorting by type of the key word

Constructors

KwTypeComparator

```
KwTypeComparator()
```

Methods

compare

```
public int compare(Keyword kw1,  
                  Keyword kw2)
```

org.mariuszgromada.math.mxparser

Class PrimitiveElement

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.PrimitiveElement
```

Direct Known Subclasses:

[Argument](#), [Constant](#), [Function](#)

< [Constructors](#) > < [Methods](#) >

```
public class PrimitiveElement  
extends java.lang.Object
```

Class used for connecting all basic elements such as: Argument, Constant, Function. Class not used by the end user.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

[MathSpace.pl](#)

[MathParser.org - mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Argument

Constant

Function

RecursiveArgument

Expression#addDefinitions(PrimitiveElement...)

Expression#removeDefinitions(PrimitiveElement...)

Constructors

PrimitiveElement

```
public PrimitiveElement(int typeId)
```

Default constructor setting element type id

Parameters:

typeId - Element type id

Methods

getMyTypeId

```
public int getMyTypeId()
```

Returns element type id

Returns:

Element type id as int Function.TYPE_ID, Argument.TYPE_ID, Function.TYPE_ID

org.mariuszgromada.math.mxparser

Class RecursiveArgument

```
java.lang.Object
|
+-- PrimitiveElement
|   |
|   +-- Argument
|       |
|       +-- org.mariuszgromada.math.mxparser.RecursiveArgument
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class RecursiveArgument
extends Argument
```

RecursiveArgument class enables to declare the argument (variable) which is defined in a recursive way. Such an argument can be used in further processing in expressions, functions and dependent or recursive arguments.

For example:

- 'fib(n) = fib(n-1)+fib(n-2), fib(0) = 0, fib(1) = 1'
- 'factorial(n) = n*factorial(n-1), factorial(0) = 1'

When creating an argument you should avoid:

- names reserved as parser keywords, in general words known in mathematical language as function names, operators (for example: sin, cos, +, -, etc...). Please be informed that after associating the argument with the expression, function or dependent/recursive argument its name will be recognized by the parser as reserved key word. It means that it could not be the same as any other key word known by the parser for this particular expression.
- defining statements with increasing index: 'a(n) = a(n+1) + ... ', otherwise you will get Double.NaN
- if recursion is not properly defined you will get Double.NaN in the result. This is due to the recursion counter inside of the recursive argument. Calculating n-th element requires no more than n recursion steps (usually less than n).
- For negative 'n' you will get Double.NaN.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Argument

Expression

Function

Constant

Fields

TYPE_DESC_RECURSIVE

```
public static final java.lang.String TYPE_DESC_RECURSIVE
```

TYPE_ID_RECURSIVE

```
public static final int TYPE_ID_RECURSIVE
    Type identifier for recursive arguments.
```

Constructors

RecursiveArgument

```
public RecursiveArgument(java.lang.String argumentName,
    java.lang.String recursiveExpressionString,
    java.lang.String indexName)
```

Constructor - creates recursive argument.

Parameters:

argumentName - the argument name
recursiveExpressionString - the recursive expression string
indexName - index argument name

RecursiveArgument

```
public RecursiveArgument(java.lang.String argumentName,
    java.lang.String recursiveExpressionString,
    Argument n,
    org.mariuszgromada.math.mxparser.PrimitiveElement[]
elements)
```

Constructor - creates recursive argument.

Parameters:

argumentName - the argument name
recursiveExpressionString - the recursive expression string
n - the index argument
elements - Optional elements list (variadic - comma separated) of types: Argument, Constant, Function

RecursiveArgument

```
public RecursiveArgument(java.lang.String argumentDefinitionString,  
                        org.mariuszgromada.math.mxparser.PrimitiveElement[]  
elements)
```

Constructor - creates argument based on the argument definition string.

Parameters:

argumentDefinitionString - Argument definition string, i.e.:

- 'x' - only argument name
- 'x=5' - argument name and argument value
- 'x=2*5' - argument name and argument value given as simple expression
- 'x=2*y' - argument name and argument expression (dependent argument 'x' on argument 'y')
- 'x(n)=x(n-1)+x(n-2)' - for recursive arguments)

elements - Optional elements list (variadic - comma separated) of types: Argument, Constant, Function

Methods

addBaseCase

```
public void addBaseCase(int index,  
                        double value)
```

Adds base case

Parameters:

index - the base case index

value - the base case value

getArgumentValue

```
public double getArgumentValue(double index)
```

Gets recursive argument value

Parameters:

index - the index

Returns:

value as double

resetAllCases

```
public void resetAllCases()
```

Clears all based cases and stored calculated values

org.mariuszgromada.math.mxpaser

Class SyntaxStackElement

```
java.lang.Object
|
+--org.mariuszgromada.math.mxpaser.SyntaxStackElement
```

< [Fields](#) > < [Constructors](#) >

class **SyntaxStackElement**
extends java.lang.Object

Fields

tokenLevel

int tokenLevel

tokenStr

java.lang.String tokenStr

Constructors

SyntaxStackElement

```
SyntaxStackElement(java.lang.String tokenStr,  
                    int tokenLevel)
```

org.mariuszgromada.math.mxpaser

Class TokenModification

```
java.lang.Object
|
+--org.mariuszgromada.math.mxpaser.TokenModification
```

< [Fields](#) > < [Constructors](#) >

class **TokenModification**
extends java.lang.Object

Data structure used internally for token to be modified list

Fields

currentToken

`java.lang.String currentToken`

newToken

`java.lang.String newToken`

newTokenDescription

`java.lang.String newTokenDescription`

Constructors

TokenModification

`TokenModification()`

`org.mariuszgromada.math.mxpaser`

Class TokenStackElement

```
java.lang.Object
|
+--org.mariuszgromada.math.mxpaser.TokenStackElement
```

[< Fields](#) > [< Constructors](#) >

class **TokenStackElement**
extends `java.lang.Object`

Internal token class which is used with stack while evaluation of tokens levels

Fields

precedingFunction

boolean **precedingFunction**

tokenId

int tokenId

tokenIndex

int tokenIndex

tokenLevel

int tokenLevel

tokenTypeId

int typeId

Constructors

TokenStackElement

TokenStackElement()

org.mariuszgromada.math.mxparser

Class Tutorial

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.Tutorial
```

< [Constructors](#) > < [Methods](#) >

```
public class Tutorial
    extends java.lang.Object
```

Tutorial class.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

[MathSpace.pl](#)

[MathParser.org - mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

RecursiveArgument

Expression

Function

Constant

Constructors

Tutorial

```
public Tutorial()
```

Methods

main

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

org.mariuszgromada.math.mxparser

Class mXparser

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.mXparser
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public final class mXparser
extends java.lang.Object
```

mXparser class provides usefull methods when parsing, calculating or parameters transforming.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

RecursiveArgument

Expression

Function

Constant

Fields

FOUND

```
static final int FOUND
```

LICENSE

```
public static final java.lang.String LICENSE
    License info.
```

MAX_RECURSION_CALLS

```
static int MAX_RECURSION_CALLS
    Internal limit for counter to avoid infinite loops while calculating expression defined in the way
    shown by below examples Argument x = new Argument("x = 2*y"); Argument y = new Argument("y
    = 2*x"); x.addDefinitions(y); y.addDefinitions(x); Function f = new Function("f(x) = 2*g(x)");
    Function g = new Function("g(x) = 2*f(x)"); f.addDefinitions(g); g.addDefinitions(f);
```

NAMEv10

```
public static final java.lang.String NAMEv10
```

NAMEv20

```
public static final java.lang.String NAMEv20
```

NAMEv23

```
public static final java.lang.String NAMEv23
```

NAMEv24

```
public static final java.lang.String NAMEv24
```

NAMEv30

```
public static final java.lang.String NAMEv30
```

NAMEv40

```
public static final java.lang.String NAMEv40
```

NAMEv41


```
public static final java.lang.String NAMEv41
```

NOT_FOUND

```
static final int NOT_FOUND  
    FOUND / NOT_FOUND used for matching purposes
```

PRIMES_CACHE_NOT_INITIALIZED

```
public static final int PRIMES_CACHE_NOT_INITIALIZED
```

VERSION

```
static final java.lang.String VERSION  
    mXparser version
```

mXparserExp

```
static final Expression mXparserExp  
    Empty expression for general help purposes.
```

overrideBuiltinTokens

```
static boolean overrideBuiltinTokens  
    Indicator whether user defined tokens should override built-in tokens.
```

primesCache

```
public static PrimesCache primesCache  
    Prime numbers cache
```

tokensToModify

```
static final java.util.List tokensToModify  
    List of built-in tokens to modify
```

tokensToRemove

```
static final java.util.List tokensToRemove  
    List of built-in tokens to remove.
```

ulpRounding

`static boolean ulpRounding`

Double floating-point precision arithmetic causes rounding problems, i.e. $0.1 + 0.1 + 0.1$ is different than 0.3 mXparser provides intelligent ULP rounding to avoid this type of errors.

Constructors

mXparser

`public mXparser()`

Methods

arrayList2double

`public static final double[] arrayList2double(java.util.List numbers)`

Converts List of double to double[]

Parameters:

numbers - the numbers list

Returns:

numbers array

checkIfEpsilonMode

`public static final boolean checkIfEpsilonMode()`

Checks if epsilon comparison mode is active;

Returns:

True if epsilon mode is active, otherwise returns false.

checkIfExactMode

`public static final boolean checkIfExactMode()`

Checks if exact comparison mode is active;

Returns:

True if exact mode is active, otherwise returns false.

checkIfUlpRounding

```
public static final boolean checkIfUlpRounding()
```

Double floating-point precision arithmetic causes rounding problems, i.e. $0.1 + 0.1 + 0.1$ is slightly different than 0.3, additionally doubles are having a lot of advantages providing flexible number representation regardless of number size. mXparser is fully based on double numbers and that is why is providing intelligent ULP rounding to minimize misleading results. By default this option is enabled resulting in automatic rounding only in some cases. Using this mode $0.1 + 0.1 + 0.1 = 0.3$

Returns:

True if ULP rounding is enabled, otherwise false.

checkIfsetToOverrideBuiltinTokens

```
public static final boolean checkIfsetToOverrideBuiltinTokens()
```

Checks whether mXparser is set to override built-in tokens.

Returns:

True if mXparser is set to override built-in tokens by user defined tokens, otherwise false.

consolePrint

```
public static final void consolePrint(java.lang.Object o)
```

Prints object.toString to the Console

Parameters:

o - Object to print

consolePrintHelp

```
public static final void consolePrintHelp()
```

Prints all help content.

consolePrintHelp

```
public static final void consolePrintHelp(java.lang.String word)
```

Prints filtered help content.

Parameters:

word - Key word.

consolePrintTokens

```
public static final void consolePrintTokens(java.util.List tokens)
```

Prints tokens to the console.

Parameters:

tokens - Tokens list.

consolePrintln

```
public static final void consolePrintln()
```

Prints new line to the Console, no new line

consolePrintln

```
public static final void consolePrintln(java.lang.Object o)
```

Prints object.toString to the Console + new line

Parameters:

o - Object to print

disableUlpRounding

```
public static final void disableUlpRounding()
```

Double floating-point precision arithmetic causes rounding problems, i.e. $0.1 + 0.1 + 0.1$ is slightly different than 0.3, additionally doubles are having a lot of advantages providing flexible number representation regardless of number size. mXparser is fully based on double numbers and that is why is providing intelligent ULP rounding to minimize misleading results. By default this option is enabled resulting in automatic rounding only in some cases. Disabling this mode $0.1 + 0.1 + 0.1$ will be slightly different than 0.3.

enableUlpRounding

```
public static final void enableUlpRounding()
```

Double floating-point precision arithmetic causes rounding problems, i.e. $0.1 + 0.1 + 0.1$ is slightly different than 0.3, additionally doubles are having a lot of advantages providing flexible number representation regardless of number size. mXparser is fully based on double numbers and that is why is providing intelligent ULP rounding to minimize misleading results. By default this option is enabled resulting in automatic rounding only in some cases. Using this mode $0.1 + 0.1 + 0.1 = 0.3$

getBuiltinTokensToModify

```
public static final java.lang.String[][] getBuiltinTokensToModify()
```

Return details on tokens marked to be modified.

Returns:

String[i][0] - current token, String[i][1] - new token, String[i][2] - new token description.

getBuiltinTokensToRemove

```
public static final java.lang.String[] getBuiltinTokensToRemove()
```

Returns current list of tokens marked to be removed.

Returns:

Current list of tokens marked to be removed

getConsoleOutput

```
public static final java.lang.String getConsoleOutput()
```

Returns console output string, console output string is being built by consolePrintln(), consolePrint().

Returns:

Console output string

getEpsilon

```
public static final double getEpsilon()
```

Returns current epsilon value.

Returns:

Returns current epsilon value.

getFunctionValue

```
public static final double getFunctionValue(Expression f,  
                                             Argument x,  
                                             double x0)
```

Calculates function $f(x_0)$ (given as expression) assigning Argument $x = x_0$;

Parameters:

f - the expression
x - the argument
x0 - the argument value

Returns:

f.calculate()

getFunctionValues

```
public static final double[] getFunctionValues(Expression f,  
                                                Argument index,  
                                                double from,  
                                                double to,  
                                                double delta)
```

Returns array of double values of the function $f(i)$ calculated on the range: $i = \text{from}$ to $i = \text{to}$ by step $= \text{delta}$

Parameters:

f - Function expression
index - Index argument
from - 'from' value
to - 'to' value
delta - 'delta' step definition

Returns:

Array of function values

getHelp

```
public static final java.lang.String getHelp()
```

General mXparser expression help

Returns:

String with all general help content

getHelp

```
public static final java.lang.String getHelp(java.lang.String word)
```

General mXparser expression help - in-line key word searching

Parameters:

word - Key word to be searched

Returns:

String with all help content lines containing given keyword

getKeyWords

```
public static final java.util.List getKeyWords()
```

Returns list of key words known to the parser

Returns:

List of keywords known to the parser.

getKeyWords

```
public static final java.util.List getKeyWords(java.lang.String query)
```

Returns list of key words known to the parser

Parameters:

query - Give any string to filter list of key words against this string. User more precise syntax: str=tokenString, desc=tokenDescription, syn=TokenSyntax, sin=tokenSince, wid=wordId, tid=wordTypeId to narrow the result.

Returns:

List of keywords known to the parser filter against query string.

getLicense

```
public static java.lang.String getLicense()
```

Gets license info

Returns:

license info as string

getMaxAllowedRecursionDepth

```
public static final int getMaxAllowedRecursionDepth()
```

Internal limit to avoid infinite loops while calculating expression defined in the way shown by below examples. Argument x = new Argument("x = 2*y"); Argument y = new Argument("y = 2*x"); x.addDefinitions(y); y.addDefinitions(x); Function f = new Function("f(x) = 2*g(x)"); Function g = new Function("g(x) = 2*f(x)"); f.addDefinitions(g); g.addDefinitions(f); Currently does not affect properly defined recursive mode.

getMaxNumInPrimesCache

```
public static final int getMaxNumInPrimesCache()
```

Returns maximum integer number in primes cache

Returns:

If primes cache was initialized then maximum number in primes cache, otherwise {@link mXparser#PRIMES_CACHE_NOT_INITIALIZED}

getThreadsNumber

```
public static final int getThreadsNumber()
```

Gets maximum threads number

Returns:

Threads number.

getTokenTypeDescription

```
public static final java.lang.String getTokenTypeDescription(int tokenId)
```

Returns token type description.

Parameters:

tokenId - Token type id

Returns:

String representing token type description.

hexString2AsciiString

```
public static final java.lang.String hexString2AsciiString(java.lang.String  
hexString)
```

Converts hex string into ASCII string, where each letter is represented by two hex digits (byte) from the hex string.

Parameters:

hexString - Hex string (i.e. 48656C6C6F)

Returns:

ASCII string (i.e. '48656C6C6F' = 'Hello')

initPrimesCache

```
public static final void initPrimesCache()
```

Initialization of prime numbers cache. Cache size according to {@link PrimesCache#DEFAULT_MAX_NUM_IN_CACHE}

initPrimesCache

```
public static final void initPrimesCache(int mximumNumberInCache)
```

Initialization of prime numbers cache.

Parameters:

mximumNumberInCache - The maximum integer number that will be stored in cache.

initPrimesCache

```
public static final void initPrimesCache(PrimesCache primesCache)
```

Initialization of prime numbers cache.

Parameters:

primesCache - The primes cache object

modifyBuiltinToken

```
public static final void modifyBuiltinToken(java.lang.String currentToken,
                                             java.lang.String newToken)
```

Method to change definition of built-in token - more precisely using this method allows to modify token string recognized by the parser (i.e. $\sin(x)$ -> $\sinus(x)$). Procedure affects only tokens classified to built-in functions, built-in constants, built-in units, built-in random variables.

Parameters:

currentToken - Current token name
newToken - New token name

modifyBuiltinToken

```
public static final void modifyBuiltinToken(java.lang.String currentToken,
                                             java.lang.String newToken,
                                             java.lang.String
newTokenDescription)
```

Method to change definition of built-in token - more precisely using this method allows to modify token string recognized by the parser (i.e. $\sin(x)$ -> $\sinus(x)$). Procedure affects only tokens classified to built-in functions, built-in constants, built-in units, built-in random variables.

Parameters:

currentToken - Current token name
newToken - New token name
newTokenDescription - New token description (if null the previous one will be used)

numberToAsciiString

```
public static final java.lang.String numberToAsciiString(double number)
```

Converts (long)double number into ASCII string, where each letter is represented by two hex digits (byte) from the hex representation of the original number casted to long type.

Parameters:

number - Double number (i.e. 310939249775 = '48656C6C6F')

Returns:

ASCII string (i.e. '48656C6C6F' = 'Hello')

numberToAsciiString

```
public static final java.lang.String numberToAsciiString(int number)
```

Converts number into ASCII string, where each letter is represented by two hex digits (byte) from the hex representation of the original number

Parameters:

number - Integer number (i.e. 310939249775 = '48656C6C6F')

Returns:

ASCII string (i.e. '48656C6C6F' = 'Hello')

numberToAsciiString

```
public static final java.lang.String numberToAsciiString(long number)
```

Converts number into ASCII string, where each letter is represented by two hex digits (byte) from the hex representation of the original number

Parameters:

number - Long number (i.e. 310939249775 = '48656C6C6F')

Returns:

ASCII string (i.e. '48656C6C6F' = 'Hello')

numberToHexString

```
public static final java.lang.String numberToHexString(double number)
```

Converts (long)double number to hex string (plain text)

Parameters:

number - Double number

Returns:

Hex string (i.e. FF23)

numberToHexString

```
public static final java.lang.String numberToHexString(int number)
```

Converts integer number to hex string (plain text)

Parameters:

number - Integer number

Returns:

Hex string (i.e. FF23)

numberToHexString

```
public static final java.lang.String numberToHexString(long number)
```

Converts long number to hex string (plain text)

Parameters:

number - Long number

Returns:

Hex string (i.e. FF23)

regexMatch

```
static final boolean regexMatch(java.lang.String str,  
                                java.lang.String pattern)
```

Function used to introduce some compatibility between JAVA and C# while regexp matching.

Parameters:

str - String

pattern - Pattern (regexp)

Returns:

True if pattern matches entirely, False otherwise

removeBuiltinTokens

```
public static final void removeBuiltinTokens(java.lang.String[] tokens)
```

Removes built-in tokens from the list of tokens recognized by the parsers. Procedure affects only tokens classified to built-in functions, built-in constants, built-in units, built-in random variables.

Parameters:

tokens - List of tokens to remove.

resetConsoleOutput

```
public static final void resetConsoleOutput()
```

Resets console output string, console output string is being built by `consolePrintln()`, `consolePrint()`.

setConsoleOutputPrefix

```
public static void setConsoleOutputPrefix(java.lang.String consoleOutputPrefix)
```

Sets console output string prefix.

Parameters:

consoleOutputPrefix - String containing console output prefix definition.

setConsolePrefix

```
public static void setConsolePrefix(java.lang.String consolePrefix)
```

Sets console prefix.

Parameters:

consolePrefix - String containing console prefix definition.

setDefaultConsoleOutputPrefix

```
public static void setDefaultConsoleOutputPrefix()
```

Sets default console output string prefix.

setDefaultConsolePrefix

```
public static void setDefaultConsolePrefix()
```

Sets default console prefix.

setDefaultEpsilon

```
public static final void setDefaultEpsilon()
```

Sets default epsilon value.

setDefaultThreadsNumber

```
public static final void setDefaultThreadsNumber()
```

Sets default threads number

setEpsilon

```
public static final void setEpsilon(double epsilon)
```

Sets epsilon value.

Parameters:

epsilon - Epsilon value (grater than 0).

setEpsilonComparison

```
public static final void setEpsilonComparison()
```

Sets comparison mode to EPSILON.

setExactComparison

```
public static final void setExactComparison()
```

Sets comparison mode to EXACT.

setMaxAllowedRecursionDepth

```
public static final void setMaxAllowedRecursionDepth(int  
maxAllowedRecursionDepth)
```

Internal limit to avoid infinite loops while calculating expression defined in the way shown by below examples. Argument x = new Argument("x = 2*y"); Argument y = new Argument("y = 2*x"); x.addDefinitions(y); y.addDefinitions(x); Function f = new Function("f(x) = 2*g(x)"); Function g = new Function("g(x) = 2*f(x)"); f.addDefinitions(g); g.addDefinitions(f); Currently does not affect properly defined recursive mode.

Parameters:

maxAllowedRecursionDepth -

setNoPrimesCache

```
public static void setNoPrimesCache()
```

Sets {@link mXparser#primesCache} to null

setNotToOverrideBuiltinTokens

```
public static final void setNotToOverrideBuiltinTokens()
```

Sets mXparser not to override built-in tokens by user defined tokens.

setRandomGenerator

```
public static final void setRandomGenerator(java.util.Random randomGenerator)
```

Modifies random generator used by the ProbabilityDistributions class.

Parameters:

randomGenerator - Random generator.

setThreadsNumber

```
public static final void setThreadsNumber(int threadsNumber)
```

Sets threads number

Parameters:

threadsNumber - Thread number.

setToOverrideBuiltinTokens

```
public static final void setToOverrideBuiltinTokens()
```

Sets mXparser to override built-in tokens by user defined tokens.

unmodifyAllBuiltinTokens

```
public static final void unmodifyAllBuiltinTokens()
```

Un-marks all tokens previously marked to be modified.

unmodifyBuiltinTokens

```
public static final void unmodifyBuiltinTokens(java.lang.String[]  
currentOrNewTokens)
```

Un-marks tokens previously marked to be modified.

Parameters:

currentOrNewTokens - List of tokens to be un-marked (current or modified).

unremoveAllBuiltinTokens

```
public static final void unremoveAllBuiltinTokens()
```

Un-marks all tokens previously marked to be removed.

unremoveBuiltinTokens

```
public static final void unremoveBuiltinTokens(java.lang.String[] tokens)
```

Un-marks tokens previously marked to be removed.

Parameters:

tokens - List of tokens to un-mark.

wait

```
public static void wait(int n)
```

Waits given number of milliseconds

Parameters:

n - Number of milliseconds

Package

org.mariuszgromada.math.mxparser.mathcollec

Class Summary

[AstronomicalConstants](#)

AstronomicalConstants - class representing the most important astronomical constants.

[BinaryRelations](#)

BinaryRelations - class for dealing with binary relations on integers or doubles.

[BooleanAlgebra](#)

BooleanAlgebra - class for boolean operators.

[Calculus](#)

Calculus - numerical integration, differentiation, etc...

[Coefficients](#)

Coefficients - various coefficients supporting numerical computation.

[Evaluate](#)

Evaluate - currently only polynomial evaluation based on provided coefficients.

[MathConstants](#)

MathConstants - class representing the most important math constants.

[MathFunctions](#)

MathFunctions - the most popular math functions.

[NumberTheory](#)

NumberTheory - summation / products etc...

[PhysicalConstants](#)

PhysicalConstants - class representing the most important physical constants.

[PrimesCache](#)

Class for generating prime numbers cache using Eratosthenes Sieve.

[ProbabilityDistributions](#)

ProbabilityDistributions - random number generators, PDF - Probability Distribution Functions, CDF - Cumulative Distribution Functions, QNT - Quantile Functions (Inverse Cumulative Distribution Functions).

[SpecialFunctions](#)

SpecialFunctions - special (non-elementary functions).

[Statistics](#)

Statistics - i.e.: mean, variance, standard deviation, etc.

Units

Units - class representing the most important units (length, area, volume, mass).

org.mariuszgromada.math.mxparser.mathcollection

Class AstronomicalConstants

java.lang.Object

|--org.mariuszgromada.math.mxparser.mathcollection.AstronomicalConstants

< [Fields](#) > < [Constructors](#) >

```
public final class AstronomicalConstants
    extends java.lang.Object
```

AstronomicalConstants - class representing the most important astronomical constants.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

[MathSpace.pl](#)

[MathParser.org - mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.0.0

Fields

ASTRONOMICAL_UNIT

```
public static final double ASTRONOMICAL_UNIT
    Astronomical unit
```

EARTH_MASS

```
public static final double EARTH_MASS  
    Earth mass
```

EARTH_RADIUS_EQUATORIAL

```
public static final double EARTH_RADIUS_EQUATORIAL  
    Earth equatorial radius
```

EARTH_RADIUS_MEAN

```
public static final double EARTH_RADIUS_MEAN  
    Earth mean radius
```

EARTH_RADIUS_POLAR

```
public static final double EARTH_RADIUS_POLAR  
    Earth polar radius
```

EARTH_SEMI_MAJOR_AXIS

```
public static final double EARTH_SEMI_MAJOR_AXIS  
    Earth semi-major axis
```

JUPITER_MASS

```
public static final double JUPITER_MASS  
    Jupiter mass
```

JUPITER_RADIUS_MEAN

```
public static final double JUPITER_RADIUS_MEAN  
    Jupiter radius
```

JUPITER_SEMI_MAJOR_AXIS

```
public static final double JUPITER_SEMI_MAJOR_AXIS  
    Jupiter semi-major axis
```

KILOPARSEC

```
public static final double KILOPARSEC  
    Kiloparsec
```

LIGHT_YEAR

```
public static final double LIGHT_YEAR  
    Light year
```

MARS_MASS

```
public static final double MARS_MASS  
    Mars mass
```

MARS_RADIUS_MEAN

```
public static final double MARS_RADIUS_MEAN  
    Mars radius
```

MARS_SEMI_MAJOR_AXIS

```
public static final double MARS_SEMI_MAJOR_AXIS  
    Mars semi-major axis
```

MERCURY_MASS

```
public static final double MERCURY_MASS  
    Mercury mass
```

MERCURY_RADIUS_MEAN

```
public static final double MERCURY_RADIUS_MEAN  
    Mercury radius
```

MERCURY_SEMI_MAJOR_AXIS

```
public static final double MERCURY_SEMI_MAJOR_AXIS  
    Mercury semi-major axis
```

MONN_SEMI_MAJOR_AXIS

```
public static final double MONN_SEMI_MAJOR_AXIS  
    Moon semi-major axis
```

MOON_MASS

```
public static final double MOON_MASS  
    Moon mass
```

MOON_RADIUS_MEAN

```
public static final double MOON_RADIUS_MEAN  
    Moon mean radius
```

NEPTUNE_MASS

```
public static final double NEPTUNE_MASS  
    Neptune mass
```

NEPTUNE_RADIUS_MEAN

```
public static final double NEPTUNE_RADIUS_MEAN  
    Neptune radius
```

NEPTUNE_SEMI_MAJOR_AXIS

```
public static final double NEPTUNE_SEMI_MAJOR_AXIS  
    Neptune semi-major axis
```

PARSEC

```
public static final double PARSEC  
    Parsec
```

SATURN_MASS

```
public static final double SATURN_MASS  
    Saturn mass
```

SATURN_RADIUS_MEAN

```
public static final double SATURN_RADIUS_MEAN
    Saturn radius
```

SATURN_SEMI_MAJOR_AXIS

```
public static final double SATURN_SEMI_MAJOR_AXIS
    Saturn semi-major axis
```

SOLAR_MASS

```
public static final double SOLAR_MASS
    Solar mass
```

SOLAR_RADIUS

```
public static final double SOLAR_RADIUS
    Solar radius
```

URANUS_MASS

```
public static final double URANUS_MASS
    Uranus mass
```

URANUS_RADIUS_MEAN

```
public static final double URANUS_RADIUS_MEAN
    Uranus radius
```

URANUS_SEMI_MAJOR_AXIS

```
public static final double URANUS_SEMI_MAJOR_AXIS
    Uranus semi-major axis
```

VENUS_MASS

```
public static final double VENUS_MASS
    Venus mass
```

VENUS_RADIUS_MEAN

```
public static final double VENUS_RADIUS_MEAN
    Venus radius
```

VENUS_SEMI_MAJOR_AXIS

```
public static final double VENUS_SEMI_MAJOR_AXIS
    Venus semi-major axis
```

Constructors

AstronomicalConstants

```
public AstronomicalConstants()
```

org.mariuszgromada.math.mxparser.mathcollection

Class BinaryRelations

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.mathcollection.BinaryRelations
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public final class BinaryRelations
    extends java.lang.Object
```

BinaryRelations - class for dealing with binary relations on integers or doubles.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](https://bitbucket.org/mariuszgromada/mxparser)

[mXparser on CodePlex](https://codeplex.com/mxparser)

[Janet Sudoku - project web page](http://JanetSudoku.com)

[Janet Sudoku on GitHub](https://github.com/mariuszgromada/JanetSudoku)

[Janet Sudoku on CodePlex](https://codeplex.com/janetsudoku)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Fields

DEFAULT_COMPARISON_EPSILON

```
public static final double DEFAULT_COMPARISON_EPSILON  
    Default epsilon for comparison
```

epsilon

```
static double epsilon  
    Epsilon for comparison
```

epsilonComparison

```
static boolean epsilonComparison  
    CComparison mode indicator
```

Constructors

BinaryRelations

```
public BinaryRelations()
```

Methods

checkIfEpsilonMode

```
public static final boolean checkIfEpsilonMode()
```

Checks if epsilon comparison mode is active;

Returns:

True if epsilon mode is active, otherwise returns false.

checkIfExactMode

```
public static final boolean checkIfExactMode()
```

Checks if exact comparison mode is active;

Returns:

True if exact mode is active, otherwise returns false.

eq

```
public static final double eq(double a,  
                               double b)
```

Equality relation.

Parameters:

a - the a number (a = b)

b - the b number (a = b)

Returns:

if a = Double.NaN or b = Double.NaN return Double.NaN, else if a = b return 1, otherwise return 0.

geq

```
public static final double geq(double a,  
                                double b)
```

Greater or equal relation.

Parameters:

a - the a number (a >= b)

b - the b number (a >= b)

Returns:

if a = Double.NaN or b = Double.NaN return Double.NaN, else if a >= b return 1, otherwise return 0.

getEpsilon

```
public static final double getEpsilon()
```

Returns current epsilon value.

Returns:

Returns current epsilon value.

gt

```
public static final double gt(double a,  
                               double b)
```

Greater than relation.

Parameters:

a - the a number ($a > b$)

b - the b number ($a > b$)

Returns:

if $a = \text{Double.NaN}$ or $b = \text{Double.NaN}$ return Double.NaN , else if $a > b$ return 1, otherwise return 0.

leq

```
public static final double leq(double a,  
                                double b)
```

Lower or equal relation.

Parameters:

a - the a number ($a \leq b$)

b - the b number ($a \leq b$)

Returns:

if $a = \text{Double.NaN}$ or $b = \text{Double.NaN}$ return Double.NaN , else if $a \leq b$ return 1, otherwise return 0.

lt

```
public static final double lt(double a,  
                               double b)
```

Lower than relation.

Parameters:

a - the a number ($a < b$)

b - the b number ($a < b$)

Returns:

if $a = \text{Double.NaN}$ or $b = \text{Double.NaN}$ return Double.NaN , else if $a < b$ return 1, otherwise return 0.

neq

```
public static final double neq(double a,  
                                double b)
```

Inequality relation.

Parameters:

a - the a number (a <> b)

b - the b number (a <> b)

Returns:

if a = Double.NaN or b = Double.NaN return Double.NaN, else if a <> b return 1, otherwise return 0.

setDefaultEpsilon

```
public static final void setDefaultEpsilon()
```

Sets default epsilon value.

setEpsilon

```
public static final void setEpsilon(double epsilon)
```

Sets epsilon value.

Parameters:

epsilon - Epsilon value (grater than 0).

setEpsilonComparison

```
public static final void setEpsilonComparison()
```

Sets comparison mode to EPSILON.

setExactComparison

```
public static final void setExactComparison()
```

Sets comparison mode to EXACT.

org.mariuszgromada.math.mxparser.mathcollection

Class BooleanAlgebra

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.mathcollection.BooleanAlgebra
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public final class BooleanAlgebra
extends java.lang.Object
```

BooleanAlgebra - class for boolean operators.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](https://bitbucket.org/mariuszgromada/mxparser/)

[mXparser on CodePlex](https://codeplex.com/mxparser/)

[Janet Sudoku - project web page](http://JanetSudoku.com)

[Janet Sudoku on GitHub](https://github.com/mariuszgromada/JanetSudoku)

[Janet Sudoku on CodePlex](https://codeplex.com/janetsudoku/)

[Janet Sudoku on SourceForge](https://sourceforge.net/projects/janetsudoku/)

[Janet Sudoku on BitBucket](https://bitbucket.org/mariuszgromada/janetsudoku/)

Version:

4.1.0

Fields

AND_TRUTH_TABLE

```
public static final double[][] AND_TRUTH_TABLE
    AND truth table
```

CIMP_TRUTH_TABLE

```
public static final double[][] CIMP_TRUTH_TABLE
    CIMP truth table
```

CNIMP_TRUTH_TABLE

```
public static final double[][] CNIMP_TRUTH_TABLE  
    CNIMP truth table
```

EQV_TRUTH_TABLE

```
public static final double[][] EQV_TRUTH_TABLE  
    EQV truth table
```

F

```
public static final double F  
    False as double
```

FALSE

```
public static final int FALSE  
    False as integer
```

IMP_TRUTH_TABLE

```
public static final double[][] IMP_TRUTH_TABLE  
    IMP truth table
```

N

```
public static final double N  
    Null as double
```

NAND_TRUTH_TABLE

```
public static final double[][] NAND_TRUTH_TABLE  
    NAND truth table
```

NIMP_TRUTH_TABLE

```
public static final double[][] NIMP_TRUTH_TABLE  
    NIMP truth table
```

NOR_TRUTH_TABLE

```
public static final double[][] NOR_TRUTH_TABLE  
    NOR truth table
```

NOT_TRUTH_TABLE

```
public static final double[] NOT_TRUTH_TABLE  
    NOT truth table
```

NULL

```
public static final int NULL  
    Null as integer
```

OR_TRUTH_TABLE

```
public static final double[][] OR_TRUTH_TABLE  
    OR truth table
```

T

```
public static final double T  
    True as double
```

TRUE

```
public static final int TRUE  
    True as integer
```

XNOR_TRUTH_TABLE

```
public static final double[][] XNOR_TRUTH_TABLE  
    XNOR truth table
```

XOR_TRUTH_TABLE

```
public static final double[][] XOR_TRUTH_TABLE  
    XOR truth table
```

Constructors

BooleanAlgebra

```
public BooleanAlgebra()
```

Methods

and

```
public static final double and(double a,  
                                double b)
```

Boolean AND

Parameters:

a - the a number (a AND b)

b - the b number (a AND b)

Returns:

Truth table element AND[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

andVariadic

```
public static final double andVariadic(double[] values)
```

Boolean AND variadic

Parameters:

values - List of values

Returns:

Returns BooleanAlgebra.TRUE if all values on the list are BooleanAlgebra.TURE, otherwise returns BooleanAlgebra.FALSE

cimp

```
public static final double cimp(double a,  
                                double b)
```

Boolean CIMP

Parameters:

a - the a number (a CIMP b)

b - the b number (a CIMP b)

Returns:

Truth table element CIMP[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

cnimp

```
public static final double cnimp(double a,  
                                double b)
```

Boolean CNIMP

Parameters:

a - the a number (a CNIMP b)

b - the b number (a CNIMP b)

Returns:

Truth table element CNIMP[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

double2IntBoolean

```
public static final int double2IntBoolean(double a)
```

Double to integer boolean translation

Parameters:

a - the double number

Returns:

If a = Double.NaN return NULL, else if a <> 0 return TRUE, else return FALSE.

eqv

```
public static final double eqv(double a,  
                                double b)
```

Boolean EQV

Parameters:

a - the a number (a EQV b)

b - the b number (a EQV b)

Returns:

Truth table element EQV[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

imp

```
public static final double imp(double a,  
                                double b)
```

Boolean IMP

Parameters:

a - the a number (a IMP b)

b - the b number (a IMP b)

Returns:

Truth table element IMP[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

nand

```
public static final double nand(double a,  
                                double b)
```

Boolean NAND

Parameters:

a - the a number (a NAND b)

b - the b number (a NAND b)

Returns:

Truth table element NAND[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

nimp

```
public static final double nimp(double a,  
                                double b)
```

Boolean NIMP

Parameters:

a - the a number (a NIMP b)

b - the b number (a NIMP b)

Returns:

Truth table element NIMP[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

nor

```
public static final double nor(double a,  
                                double b)
```

Boolean NOR

Parameters:

a - the a number (a NOR b)

b - the b number (a NOR b)

Returns:

Truth table element NOR[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

not

```
public static final double not(double a)
```

Boolean NOT

Parameters:

a - the a number (NOT a)

Returns:

Truth table element NOT[A] where A = double2IntBoolean(a)

or

```
public static final double or(double a,  
                                double b)
```

Boolean OR

Parameters:

a - the a number (a OR b)

b - the b number (a OR b)

Returns:

Truth table element OR[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

orVariadic

```
public static final double orVariadic(double[] values)
```

Boolean OR variadic

Parameters:

values - List of values

Returns:

Returns BooleanAlgebra.TRUE if at least one value on the list is BooleanAlgebra.TURE, otherwise returns BooleanAlgebra.FALSE

xnor

```
public static final double xnor(double a,  
                                double b)
```

Boolean XNOR

Parameters:

a - the a number (a XNOR b)

b - the b number (a XNOR b)

Returns:

Truth table element XNOR[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

XOR

```
public static final double xor(double a,  
                                double b)
```

Boolean XOR

Parameters:

a - the a number (a XOR b)

b - the b number (a XOR b)

Returns:

Truth table element XOR[A][B] where A = double2IntBoolean(a), B = double2IntBoolean(b)

xorVariadic

```
public static final double xorVariadic(double[] values)
```

Boolean XOR variadic

Parameters:

values - List of values

Returns:

Returns BooleanAlgebra.TRUE if exactly one value on the list is BooleanAlgebra.TURE, otherwise returns BooleanAlgebra.FALSE

org.mariuszgromada.math.mxparser.mathcollection

Class Calculus

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.mathcollection.Calculus
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public final class Calculus  
extends java.lang.Object
```

Calculus - numerical integration, differentiation, etc...

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.0.0

Fields

GENERAL_DERIVATIVE

```
public static final int GENERAL_DERIVATIVE
```

LEFT_DERIVATIVE

```
public static final int LEFT_DERIVATIVE
```

Derivative type specification

RIGHT_DERIVATIVE

```
public static final int RIGHT_DERIVATIVE
```

Constructors

Calculus

```
public Calculus()
```

Methods

backwardDifference

```
public static final double backwardDifference(Expression f,  
                                             double h,  
                                             Argument x)
```

Backward difference(h) operator (at the current value of the argument x)

Parameters:

f - the expression
h - the difference
x - the argument name

Returns:

Backward difference(h) value calculated at at the current value of the argument x.

backwardDifference

```
public static final double backwardDifference(Expression f,  
                                             double h,  
                                             Argument x,  
                                             double x0)
```

Backward difference(h) operator (at x = x0)

Parameters:

f - the expression
h - the difference
x - the argument name
x0 - x = x0

Returns:

Backward difference(h) value calculated at x0.

backwardDifference

```
public static final double backwardDifference(Expression f,  
                                             Argument x)
```

Backward difference(1) operator (at current value of argument x)

Parameters:

f - the expression
x - the argument name

Returns:

Backward difference(1) value calculated at the current value of argument x.

backwardDifference

```
public static final double backwardDifference(Expression f,  
                                             Argument x,  
                                             double x0)
```

Backward difference(1) operator (at $x = x_0$).

Parameters:

f - the expression
x - the argument name
 x_0 - $x = x_0$

Returns:

Backward difference value calculated at x_0 .

derivative

```
public static final double derivative(Expression f,  
                                     Argument x,  
                                     double x0,  
                                     int derType,  
                                     double eps,  
                                     int maxSteps)
```

Numerical derivative at $x = x_0$

Parameters:

f - the expression
x - the argument
 x_0 - at point $x = x_0$
derType - derivative type (LEFT_DERIVATIVE, RIGHT_DERIVATIVE,
GENERAL_DERIVATIVE)
eps - the epsilon (error)
maxSteps - the maximum number of steps

Returns:

Derivative value as double.

derivativeNth

```
public static final double derivativeNth(Expression f,  
                                         double n,  
                                         Argument x,  
                                         double x0,  
                                         int derType,  
                                         double eps,  
                                         int maxSteps)
```

Numerical n-th derivative at $x = x_0$ (you should avoid calculation of derivatives with order higher than 2).

Parameters:

- f - the expression
- n - the derivative order
- x - the argument
- x_0 - at point $x = x_0$
- derType - derivative type (LEFT_DERIVATIVE, RIGHT_DERIVATIVE, GENERAL_DERIVATIVE)
- eps - the epsilon (error)
- maxSteps - the maximum number of steps

Returns:

Derivative value as double.

forwardDifference

```
public static final double forwardDifference(Expression f,  
                                             double h,  
                                             Argument x)
```

Forward difference(h) operator (at the current value of the argument x)

Parameters:

- f - the expression
- h - the difference
- x - the argument name

Returns:

Forward difference(h) value calculated at at the current value of the argument x.

forwardDifference

```
public static final double forwardDifference(Expression f,  
                                             double h,  
                                             Argument x,  
                                             double x0)
```

Forward difference(h) operator (at $x = x_0$)

Parameters:

f - the expression
h - the difference
x - the argument name
 $x_0 - x = x_0$

Returns:

Forward difference(h) value calculated at x_0 .

forwardDifference

```
public static final double forwardDifference(Expression f,  
                                             Argument x)
```

Forward difference(1) operator (at current value of argument x)

Parameters:

f - the expression
x - the argument name

Returns:

Forward difference(1) value calculated at the current value of argument x.

forwardDifference

```
public static final double forwardDifference(Expression f,  
                                             Argument x,  
                                             double x0)
```

Forward difference(1) operator (at $x = x_0$)

Parameters:

f - the expression
x - the argument name
 $x_0 - x = x_0$

Returns:

Forward difference(1) value calculated at x_0 .

integralTrapezoid

```
public static final double integralTrapezoid(Expression f,  
                                             Argument x,  
                                             double a,  
                                             double b,  
                                             double eps,  
                                             int maxSteps)
```

Trapezoid numerical integration

Parameters:

- f - the expression
- x - the argument
- a - from a ...
- b - ... to b
- eps - the epsilon (error)
- maxSteps - the maximum number of steps

Returns:

Integral value as double.

solveBrent

```
public static final double solveBrent(Expression f,  
                                       Argument x,  
                                       double a,  
                                       double b,  
                                       double eps,  
                                       double maxSteps)
```

Brent solver (Brent root finder)

Parameters:

- f - Function given in the Expression form
- x - Argument
- a - Left limit
- b - Right limit
- eps - Epsilon value (accuracy)
- maxSteps - Maximum number of iterations

Returns:

Function root - if found, otherwise Double.NaN.

org.mariuszgromada.math.mxparser.mathcollection

Class Coefficients

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.mathcollection.Coefficients
```

< [Fields](#) > < [Constructors](#) >

final class **Coefficients**
extends java.lang.Object

Coefficients - various coefficients supporting numerical computation.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

[MathSpace.pl](#)

[MathParser.org - mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Fields

EI

static final double[] **EI**
Supporting function while Exponential integral function $Ei(x)$ calculation

erfImpAd

static final double[] **erfImpAd**
Polynomial coefficients for denominator of erfImp calculation for erf(x) in the interval [1e-10, 0.5].

erfImpAn

static final double[] **erfImpAn**
Polynomial coefficients for a numerator of erfImp calculation for erf(x) in the interval [1e-10, 0.5].

erfImpBd

```
static final double[] erfImpBd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [0.5, 0.75].

erfImpBn

```
static final double[] erfImpBn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [0.5, 0.75].

erfImpCd

```
static final double[] erfImpCd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [0.75, 1.25].

erfImpCn

```
static final double[] erfImpCn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [0.75, 1.25].

erfImpDd

```
static final double[] erfImpDd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [1.25, 2.25].

erfImpDn

```
static final double[] erfImpDn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [1.25, 2.25].

erfImpEd

```
static final double[] erfImpEd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [2.25, 3.5].

erfImpEn

```
static final double[] erfImpEn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [2.25, 3.5].

erfImpFd

```
static final double[] erfImpFd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [3.5, 5.25].

erfImpFn

```
static final double[] erfImpFn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [3.5, 5.25].

erfImpGd

```
static final double[] erfImpGd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [5.25, 8].

erfImpGn

```
static final double[] erfImpGn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [5.25, 8].

erfImpHd

```
static final double[] erfImpHd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [8, 11.5].

erfImpHn

```
static final double[] erfImpHn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [8, 11.5].

erfImpId

```
static final double[] erfImpId
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [11.5, 17].

erfImpIn

```
static final double[] erfImpIn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [11.5, 17].

erfImpJd

```
static final double[] erfImpJd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [17, 24].

erfImpJn

```
static final double[] erfImpJn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [17, 24].

erfImpKd

```
static final double[] erfImpKd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [24, 38].

erfImpKn

```
static final double[] erfImpKn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [24, 38].

erfImpLd

```
static final double[] erfImpLd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [38, 60].

erfImpLn

```
static final double[] erfImpLn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [38, 60].

erfImpMd

```
static final double[] erfImpMd
```

Polynomial coefficients for a denominator in erfImp calculation for erfc(x) in the interval [60, 85].

erfImpMn

```
static final double[] erfImpMn
```

Polynomial coefficients for a numerator in erfImp calculation for erfc(x) in the interval [60, 85].

erfImpNd

```
static final double[] erfImpNd
```

Polynomial coefficients for a denominator in erfImp calculation for $\operatorname{erfc}(x)$ in the interval $[85, 110]$.

erfImpNn

```
static final double[] erfImpNn
```

Polynomial coefficients for a numerator in erfImp calculation for $\operatorname{erfc}(x)$ in the interval $[85, 110]$.

ervInvImpAd

```
static final double[] ervInvImpAd
```

Polynomial coefficients for a denominator of erfInvImp calculation for $\operatorname{erf}^{-1}(z)$ in the interval $[0, 0.5]$.

ervInvImpAn

```
static final double[] ervInvImpAn
```

Polynomial coefficients for a numerator of erfInvImp calculation for $\operatorname{erf}^{-1}(z)$ in the interval $[0, 0.5]$.

ervInvImpBd

```
static final double[] ervInvImpBd
```

Polynomial coefficients for a denominator of erfInvImp calculation for $\operatorname{erf}^{-1}(z)$ in the interval $[0.5, 0.75]$.

ervInvImpBn

```
static final double[] ervInvImpBn
```

Polynomial coefficients for a numerator of erfInvImp calculation for $\operatorname{erf}^{-1}(z)$ in the interval $[0.5, 0.75]$.

ervInvImpCd

```
static final double[] ervInvImpCd
```

Polynomial coefficients for a denominator of erfInvImp calculation for $\operatorname{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x less than 3.

ervInvImpCn

```
static final double[] ervInvImpCn
```

Polynomial coefficients for a numerator of erflnImp calculation for $\text{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x less than 3.

ervInvImpDd

```
static final double[] ervInvImpDd
```

Polynomial coefficients for a denominator of erflnImp calculation for $\text{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x between 3 and 6.

ervInvImpDn

```
static final double[] ervInvImpDn
```

Polynomial coefficients for a numerator of erflnImp calculation for $\text{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x between 3 and 6.

ervInvImpEd

```
static final double[] ervInvImpEd
```

Polynomial coefficients for a denominator of erflnImp calculation for $\text{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x between 6 and 18.

ervInvImpEn

```
static final double[] ervInvImpEn
```

Polynomial coefficients for a numerator of erflnImp calculation for $\text{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x between 6 and 18.

ervInvImpFd

```
static final double[] ervInvImpFd
```

Polynomial coefficients for a denominator of erflnImp calculation for $\text{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x between 18 and 44.

ervInvImpFn

```
static final double[] ervInvImpFn
```

Polynomial coefficients for a numerator of erflnImp calculation for $\text{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x between 18 and 44.

ervInvImpGd

```
static final double[] ervInvImpGd
```


Polynomial coefficients for a denominator of erflnvlmp calculation for $\text{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x greater than 44.

ervlnvlmpGn

```
static final double[] ervlnvlmpGn
```

Polynomial coefficients for a numerator of erflnvlmp calculation for $\text{erf}^{-1}(z)$ in the interval $[0.75, 1]$ with x greater than 44.

Constructors

Coefficients

```
Coefficients()
```

org.mariuszgromada.math.mxparser.mathcollection

Class Evaluate

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.mathcollection.Evaluate
```

< [Constructors](#) > < [Methods](#) >

```
public final class Evaluate  
extends java.lang.Object
```

Evaluate - currently only polynomial evaluation based on provided coefficients.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

[MathSpace.pl](#)

[MathParser.org - mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Constructors

Evaluate

```
public Evaluate()
```

Methods

polynomial

```
public static final double polynomial(double x,  
                                         double[] coefficients)
```

Polynomial evaluation based on provided coefficients.

Parameters:

x - Point at which polynomial will be evaluated
coefficients - Polynomial coefficients

Returns:

Polynomial value

org.mariuszgromada.math.mxparser.mathcollection

Class MathConstants

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.mathcollection.MathConstants
```

< [Fields](#) > < [Constructors](#) >

```
public final class MathConstants  
extends java.lang.Object
```

MathConstants - class representing the most important math constants.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

ALLADI_GRINSTEAD

```
public static final double ALLADI_GRINSTEAD  
    Alladi-Grinstead constant
```

APERY

```
public static final double APERY  
    Apery's constant
```

BACKHOUSE

```
public static final double BACKHOUSE  
    Backhouse's constant
```

BERNSTEIN

```
public static final double BERNSTEIN  
    Bernstein's constant
```

BRAUN_PRIME_QUADR

```
public static final double BRAUN_PRIME_QUADR  
    Brun's constant for prime quadruplets
```

BRAUN_TWIN_PRIME

```
public static final double BRAUN_TWIN_PRIME  
    Brun's constant for twin primes
```

BRUIJN_NEWMAN

```
public static final double BRUIJN_NEWMAN  
    de Bruijn-Newman constant
```

CAHEN

```
public static final double CAHEN  
    Cahen's constant
```

CATALAN

```
public static final double CATALAN  
    Catalan's constant
```

E

```
public static final double E  
    Napier's constant, or Euler's number, base of Natural logarithm
```

EMBREE_TREFETHEN

```
public static final double EMBREE_TREFETHEN  
    Embree-Trefethen constant
```

ERDOS_BORWEIN

```
public static final double ERDOS_BORWEIN  
    Erdos-Borwein constant
```

EULER_MASCHERONI

```
public static final double EULER_MASCHERONI  
    Euler-Mascheroni constant
```

FEIGENBAUM_ALFA

```
public static final double FEIGENBAUM_ALFA
    Feigenbaum constant
```

FEIGENBAUM_DELTA

```
public static final double FEIGENBAUM_DELTA
    Feigenbaum constant
```

FRANSEN_ROBINSON

```
public static final double FRANSEN_ROBINSON
    Frans n-Robinson constant
```

GAUSS_KUZMIN_WIRSING

```
public static final double GAUSS_KUZMIN_WIRSING
    Gauss-Kuzmin-Wirsing constant
```

GOLDEN_RATIO

```
public static final double GOLDEN_RATIO
    Golden ratio
```

GOLOMB_DICKMAN

```
public static final double GOLOMB_DICKMAN
    Golomb-Dickman constant
```

GOMPERTZ

```
public static final double GOMPERTZ
    Gompertz Constant OEIS A073003
```

HAFNER_SARNAK_MCCURLEY

```
public static final double HAFNER_SARNAK_MCCURLEY
    Hafner-Sarnak-McCurley constant
```

KHINCHIN

```
public static final double KHINCHIN
    Khinchin's constant
```

LANDAU

```
public static final double LANDAU
    Landau's constant
```

LANDAU_RAMANUJAN

```
public static final double LANDAU_RAMANUJAN
    Landau-Ramanujan constant
```

LAPLACE_LIMIT

```
public static final double LAPLACE_LIMIT
    Laplace limit
```

LEGENDRE

```
public static final double LEGENDRE
    Legendre's constant
```

LENGYEL

```
public static final double LENGYEL
    Lengyel's constant
```

LEVY

```
public static final double LEVY
    Levy's constant
```

LI2

```
public static final double LI2
    A069284 - Logarithmic integral function li(2)
```

LIEB_QUARE_ICE

```
public static final double LIEB_QUARE_ICE
    Porter's constant
```

MEISSEL_MERTEENS

```
public static final double MEISSEL_MERTEENS
    Meissel-Mertens constant
```

MILLS

```
public static final double MILLS
    Mills' constant
```

MRB

```
public static final double MRB
    MRB constant
```

NIVEN

```
public static final double NIVEN
    Niven's constant
```

NOT_A_NUMBER

```
public static final double NOT_A_NUMBER
    Not-a-Number
```

OMEGA

```
public static final double OMEGA
    Omega constant
```

PARABOLIC

```
public static final double PARABOLIC
    Parabolic constant
```

PI

```
public static final double PI  
    Pi, Archimedes' constant or Ludolph's number
```

PLASTIC

```
public static final double PLASTIC  
    Plastic constant
```

PORTER

```
public static final double PORTER  
    Porter's constant
```

RAMANUJAN_SOLDNER

```
public static final double RAMANUJAN_SOLDNER  
    Ramanujan-Soldner constant
```

SIERPINSKI

```
public static final double SIERPINSKI  
    Sierpiński's constant
```

SQRT2

```
public static final double SQRT2  
    Square root of 2
```

SQRT2Pi

```
public static final double SQRT2Pi  
    Square root of 2*pi
```

TWIN_PRIME

```
public static final double TWIN_PRIME  
    Feigenbaum constant
```

VISWANATH

```
public static final double VISWANATH  
    Viswanath's constant
```

Constructors

MathConstants

```
public MathConstants()
```

```
org.mariuszgromada.math.mxparser.mathcollection
```

Class MathFunctions

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.mathcollection.MathFunctions
```

< [Constructors](#) > < [Methods](#) >

```
public final class MathFunctions  
    extends java.lang.Object
```

MathFunctions - the most popular math functions. Many of function implemented by this class could be found in java Math package (in fact functions from MathFunctions typically calls original functions from the Math package). The reason why it was "re-implemented" is: if you decide to implement your own function you do not need to change anything in the parser, jut modify function implementation in this class.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](https://bitbucket.org/mariuszgromada/mxparser/)

[mXparser on CodePlex](https://codeplex.com/mxparser/)

[Janet Sudoku - project web page](http://JanetSudoku.org)

[Janet Sudoku on GitHub](https://github.com/mariuszgromada/JanetSudoku)

[Janet Sudoku on CodePlex](https://sourceforge.net/projects/janetsudoku/)

[Janet Sudoku on SourceForge](https://sourceforge.net/projects/janetsudoku/)

[Janet Sudoku on BitBucket](https://bitbucket.org/mariuszgromada/janetsudoku/)

Version:

4.1.0

Constructors

MathFunctions

```
public MathFunctions()
```

Methods

Sirling1Number

```
public static final double Sirling1Number(double n,  
                                           double k)
```

Stirling numbers of the first kind

Parameters:

n - the n function parameter

k - the k function parameter

Returns:

if n, k <> Double.NaN returns Sirling1Number((int)Math.round(n), (int)Math.round(k)),
otherwise returns Double.NaN.

Sirling1Number

```
public static final double Sirling1Number(int n,  
                                           int k)
```

Stirling numbers of the first kind

Parameters:

n - the n function parameter

k - the k function parameter

Returns:

Stirling numbers of the first kind

Sirling2Number

```
public static final double Sirling2Number(double n,  
                                           double k)
```

Striling numbers of the second kind

Parameters:

n - the n function parameter

k - the k function parameter

Returns:

if n, k <> Double.NaN returns Sirling2Number((int)Math.round(n), (int)Math.round(k)),
otherwise returns Double.NaN.

Sirling2Number

```
public static final double Sirling2Number(int n,  
                                           int k)
```

Striling numbers of the second kind

Parameters:

n - the n function parameter

k - the k function parameter

Returns:

Striling numbers of the second kind

abs

```
public static final double abs(double a)
```

Absolute value.

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.abs(a), otherwise returns Double.NaN.

acos

```
public static final double acos(double a)
```

Arcus cosine - inverse trigonometric cosine function

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ returns $\text{Math.acos}(a)$, otherwise returns Double.NaN .

actan

```
public static final double actan(double a)
```

Arcus cotangent - inverse trigonometric cotangent function

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $a \neq 0$ returns $\text{Math.atan}(1/a)$, otherwise returns Double.NaN .

arcosh

```
public static final double arcosh(double a)
```

Arcus hyperbolic cosine - inverse hyperbolic cosine function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ returns $\text{Math.log}(a + \text{Math.sqrt}(a^2 - 1))$, otherwise returns Double.NaN .

arcoth

```
public static final double arcoth(double a)
```

Arcus hyperbolic tangent - inverse hyperbolic tangent function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $a - 1 \neq 0$ returns $0.5 * \text{Math.log}((a+1)/(a-1))$; , otherwise returns Double.NaN .

arcsch

```
public static final double arcsch(double a)
```

Arcus hyperbolic cosecant - inverse hyperbolic cosecant function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $a \neq 0$ returns $\text{Math.log}((1 + \text{Math.sqrt}(1 - a * a)) / a)$;; otherwise returns Double.NaN .

arsech

```
public static final double arsech(double a)
```

Arcus hyperbolic secant - inverse hyperbolic secant function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $a \neq 0$ returns $\text{Math.log}((1 + \text{Math.sqrt}(1 - a * a)) / a)$;; otherwise returns Double.NaN .

arsinh

```
public static final double arsinh(double a)
```

Arcus hyperbolic sine - inverse hyperbolic sine function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ returns $\text{Math.log}(a + \text{Math.sqrt}(a * a + 1))$, otherwise returns Double.NaN .

artanh

```
public static final double artanh(double a)
```

Arcus hyperbolic tangent - inverse hyperbolic tangent function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $1 - a \neq 0$ returns $0.5 * \text{Math.log}((1 + a) / (1 - a))$, otherwise returns Double.NaN .

asin

```
public static final double asin(double a)
```

Arcus sine - inverse trigonometric sine function

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.asin(a), otherwise returns Double.NaN.

atan

```
public static final double atan(double a)
```

Arcus tangent - inverse trigonometric tangent function

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.atan(a), otherwise returns Double.NaN.

bellNumber

```
public static final double bellNumber(double n)
```

Bell number

Parameters:

n - the n

Returns:

if n <> Double.NaN return bellNumber((int)Math.round(n)), otherwise return Double.NaN.

bellNumber

```
public static final double bellNumber(int n)
```

Bell Numbers

Parameters:

n - the n

Returns:

if n >= 0 returns Bell numbers, otherwise returns Double.NaN.

bernoulliNumber

```
public static final double bernoulliNumber(double m,  
                                             double n)
```

Bernoulli numbers

Parameters:

m - the m function parameter

n - the n function parameter

Returns:

if n, m <> Double.NaN returns bernoulliNumber((int)Math.round(m), (int)Math.round(n)),
otherwise returns Double.NaN.

bernoulliNumber

```
public static final double bernoulliNumber(int m,  
                                             int n)
```

Bernoulli numbers

Parameters:

m - the m function parameter

n - the n function parameter

Returns:

if n, m >= 0 returns Bernoulli number, otherwise returns Double.NaN.

binomCoeff

```
public static final double binomCoeff(double n,  
                                       double k)
```

Generalized binomial coefficient

Parameters:

n - the n function parameter

k - the k function parameter

Returns:

if n, k <> Double.NaN returns binomCoeff(n, (int)Math.round(k)), otherwise returns
Double.NaN.

binomCoeff

```
public static final double binomCoeff(double n,  
                                       int k)
```

Generalized binomial coefficient

Parameters:

n - the n function parameter
k - k the k function parameter

Returns:

Generalized binomial coefficient, if n = Double.NaN or k < 0 returns Double.NaN.

catalanNumber

```
public static final double catalanNumber(double n)
```

Catalan numbers

Parameters:

n - the n function parameter

Returns:

if n <> Double.NaN returns catalanNumber((int)Math.round(n)), otherwise returns Double.NaN.

catalanNumber

```
public static final double catalanNumber(int n)
```

Catalan numbers

Parameters:

n - the n function parameter

Returns:

Catalan numbers

ceil

```
public static final double ceil(double a)
```

Ceiling function.

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.ceil(a), otherwise returns Double.NaN.

chi

```
public static final double chi(double x,  
                                double a,  
                                double b)
```

Characteristic function x in (a,b)

Parameters:

x - the x value
 a - the left (lower) limit
 b - the right (upper) limit

Returns:

if $x, a, b \neq \text{Double.NaN}$ returns characteristic function value on the (a,b) range.

chi_L

```
public static final double chi_L(double x,  
                                   double a,  
                                   double b)
```

Characteristic function x in $[a,b)$

Parameters:

x - the x value
 a - the left (lower) limit
 b - the right (upper) limit

Returns:

if $x, a, b \neq \text{Double.NaN}$ returns characteristic function value on the $[a,b)$ range.

chi_LR

```
public static final double chi_LR(double x,  
                                    double a,  
                                    double b)
```

Characteristic function x in $[a,b]$

Parameters:

x - the x value
 a - the left (lower) limit
 b - the right (upper) limit

Returns:

if $x, a, b \neq \text{Double.NaN}$ returns characteristic function value on the $[a,b]$ range.

chi_R

```
public static final double chi_R(double x,  
                                   double a,  
                                   double b)
```

Characteristic function x in (a,b]

Parameters:

x - the x value
a - the left (lower) limit
b - the right (upper) limit

Returns:

if x, a, b <> Double.NaN returns characteristic function value on the (a,b] range.

coalesce

```
public static final double coalesce(double[] values)
```

Returns the first non-NaN value

Parameters:

values - List of values

Returns:

Returns the first non-NaN value, if list is null then returns Double.NaN, if list contains no elements then returns Double.NaN.

continuedFraction

```
public static final double continuedFraction(double[] sequence)
```

Continued fraction

Parameters:

sequence - the numbers

Returns:

if each number form the sequence <> Double.NaN and there is no division by 0 while computing returns continued fraction value, otherwise returns Double.NaN.

continuedPolynomial

```
public static final double continuedPolynomial(double[] x)
```

Continued polynomial

Parameters:

x - the x values

Returns:

if each number for x is different the Double.NaN returns continued polynomial, otherwise returns Double.NaN.

cos

```
public static final double cos(double a)
```

Cosine trigonometric function

Parameters:

a - the a function parameter

Returns:

if a \neq Double.NaN returns Math.cos(a), otherwise returns Double.NaN.

cosec

```
public static final double cosec(double a)
```

Cosecant trigonometric function

Parameters:

a - the a function parameter

Returns:

if a \neq Double.NaN and $\sin(a) \neq 0$ returns $1 / \sin(a)$, otherwise returns Double.NaN.

cosh

```
public static final double cosh(double a)
```

Hyperbolic cosine function.

Parameters:

a - the a function parameter

Returns:

if a \neq Double.NaN returns Math.cosh(a), otherwise returns Double.NaN.

coth

```
public static final double coth(double a)
```

Hyperbolic cotangent function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $\tanh(a) \neq 0$ returns $1 / \text{Math.tanh}(a)$, otherwise returns Double.NaN .

csch

```
public static final double csch(double a)
```

Hyperbolic cosecant function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $\sinh(a) \neq 0$ returns $1 / \text{Math.sinh}(a)$, otherwise returns Double.NaN .

ctan

```
public static final double ctan(double a)
```

Cotangent trigonometric function

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $\tan(a) \neq 0$ returns $1 / \text{Math.tan}(a)$, otherwise returns Double.NaN .

decimalDigitsBefore

```
public static final int decimalDigitsBefore(double value)
```

For very small number returns number of zeros before first significant digit.

Parameters:

value - Double value, small one.

Returns:

Number of digits, number of places.

deg

```
public static final double deg(double a)
```

Radius to degrees translation.

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.toDegrees(a), otherwise returns Double.NaN.

div

```
public static final double div(double a,  
                                double b)
```

Division a/b

Parameters:

a - the a function parameter

b - the b function parameter

Returns:

if a,b <> Double.NaN and b <> 0 returns a/b, otherwise return Double.NaN.

eulerNumber

```
public static final double eulerNumber(double n,  
                                          double k)
```

Euler numbers

Parameters:

n - the n function param

k - the k function param

Returns:

if n, k <> Double.NaN returns eulerNumber((int)Math.round(n), (int)Math.round(k)),
otherwise return Double.NaN.

eulerNumber

```
public static final double eulerNumber(int n,  
                                         int k)
```

Euler numbers

Parameters:

n - the n function param

k - the k function param

Returns:

if $n \geq 0$ returns Euler number, otherwise return Double.NaN.

eulerPolynomial

```
public static final double eulerPolynomial(double m,  
                                             double x)
```

Euler polynomial

Parameters:

m - the m parameter

x - the x parameter

Returns:

if $x, m \neq \text{Double.NaN}$ returns `eulerPolynomial((int)Math.round(m), (int)Math.round(x))`,
otherwise returns Double.NaN.

eulerPolynomial

```
public static final double eulerPolynomial(int m,  
                                             double x)
```

Euler polynomial

Parameters:

m - the m parameter

x - the x parameter

Returns:

if $x \neq \text{Double.NaN}$ and $m \geq 0$ returns polynomial value, otherwise returns Double.NaN.

exp

```
public static final double exp(double a)
```

Exponential function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ returns $\text{Math.exp}(a)$, otherwise returns Double.NaN .

factorial

```
public static final double factorial(double n)
```

Factorial

Parameters:

n - the n function parameter

Returns:

if $n \neq \text{Double.NaN}$ return $\text{factorial}((\text{int})\text{Math.round}(n))$, otherwise returns Double.NaN .

factorial

```
public static final double factorial(int n)
```

Factorial

Parameters:

n - the n function parameter

Returns:

Factorial if $n \geq 0$, otherwise returns Double.NaN .

fibonacciNumber

```
public static final double fibonacciNumber(double n)
```

Fibonacci numbers

Parameters:

n - the n function parameter

Returns:

if $n \neq \text{Double.NaN}$ returns $\text{fibonacciNumber}((\text{int})\text{Math.round}(n))$, otherwise returns Double.NaN .

fibonacciNumber

```
public static final double fibonacciNumber(int n)
```

Fibonacci numbers

Parameters:

n - the n function parameter

Returns:

if $n \geq 0$ returns fibonacci numbers, otherwise returns Double.NaN.

floor

```
public static final double floor(double a)
```

Floor function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ returns Math.floor(a), otherwise returns Double.NaN.

harmonicNumber

```
public static final double harmonicNumber(double n)
```

Harmonic number

Parameters:

n - the n function parameter

Returns:

if $n \neq \text{Double.NaN}$ returns harmonicNumber((int)Math.round(n)), otherwise returns Double.NaN

harmonicNumber

```
public static final double harmonicNumber(double x,  
                                           double n)
```

Harmonic number $1/1 + 1/2^x + \dots + 1/n^x$

Parameters:

x - the x function parameter

n - the n function parameter

Returns:

if $x, n \neq \text{Double.NaN}$ returns `harmonicNumber(x, (int)Math.round(n))`, otherwise returns `Double.NaN`.

harmonicNumber

```
public static final double harmonicNumber(double x,  
                                           int n)
```

Harmonic number $1/1 + 1/2^x + \dots + 1/n^x$

Parameters:

x - the x function parameter

n - the n function parameter

Returns:

if $x \neq \text{Double.NaN}$ and $x \geq 0$ Harmonic number, otherwise returns `Double.NaN`.

harmonicNumber

```
public static final double harmonicNumber(int n)
```

Harmonic number

Parameters:

n - the n function parameter

Returns:

if $n > 0$ returns harmonic number, otherwise returns 0 (empty summation operator)

kroneckerDelta

```
public static final double kroneckerDelta(double i,  
                                           double j)
```

Kronecker delta

Parameters:

i - the i function parameter

j - the j function parameter

Returns:

if i,j <> Double.NaN returns Kronecker delta, otherwise returns Double.NaN.

kroneckerDelta

```
public static final double kroneckerDelta(int i,  
                                           int j)
```

Kronecker delta

Parameters:

i - the i function parameter

j - the j function parameter

Returns:

Kronecker delta

ln

```
public static final double ln(double a)
```

Natural logarithm

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.log(1/a), otherwise returns Double.NaN.

log

```
public static final double log(double a,  
                                double b)
```

General logarithm.

Parameters:

- a - the a function parameter (base)
- b - the b function parameter (number)

Returns:

if a,b <> Double.NaN and log(b) <> 0 returns Math.log(a) / Math.log(b), otherwise returns Double.NaN.

log10

```
public static final double log10(double a)
```

Common logarithm

Parameters:

- a - the a function parameter

Returns:

if a <> Double.NaN returns Math.log10(a), otherwise returns Double.NaN.

log2

```
public static final double log2(double a)
```

Binary logarithm

Parameters:

- a - the a function parameter

Returns:

if a <> Double.NaN returns Math.log(a)/Math.log(2.0), otherwise returns Double.NaN.

lucasNumber

```
public static final double lucasNumber(double n)
```

Lucas numebers

Parameters:

n - the n function parameter

Returns:

if $n \neq \text{Double.NaN}$ returns `lucasNumber((int)Math.round(n))`, otherwise returns `Double.NaN`.

lucasNumber

```
public static final double lucasNumber(int n)
```

Lucas numebers

Parameters:

n - the n function parameter

Returns:

if $n \geq 0$ returns Lucas numbers, otherwise returns `Double.NaN`.

mod

```
public static final double mod(double a,  
                                double b)
```

Modulo operator $a \% b$

Parameters:

a - the a function parameter

b - the b function parameter

Returns:

if $a, b \neq \text{Double.NaN}$ returns $a \% b$.

power

```
public static final double power(double a,  
                                double b)
```

Power function a^b

Parameters:

a - the a function parameter

b - the b function parameter

Returns:

if $a, b \neq \text{Double.NaN}$ returns $\text{Math.pow}(a, b)$, otherwise returns Double.NaN .

rad

```
public static final double rad(double a)
```

Degrees to radius translation.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ returns $\text{Math.toRadians}(a)$, otherwise returns Double.NaN .

round

```
public static final double round(double value,  
                                int places)
```

Double rounding

Parameters:

value - double value to be rounded

places - decimal places

Returns:

Rounded value

sa

```
public static final double sa(double a)
```

Normalized sinc function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $a \neq 0$ returns $\text{Math.sin}(\text{PI} * a) / (\text{PI} * a)$;; otherwise returns Double.NaN .

sec

```
public static final double sec(double a)
```

Secant trigonometric function

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $\cos(a) \neq 0$ returns $1 / \text{Math.cos}(a)$, otherwise returns Double.NaN .

sech

```
public static final double sech(double a)
```

Hyperbolic secant function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ and $\cosh(a) \neq 0$ returns $1 / \text{Math.cosh}(a)$, otherwise returns Double.NaN .

sgn

```
public static final double sgn(double a)
```

Signum function.

Parameters:

a - the a function parameter

Returns:

if $a \neq \text{Double.NaN}$ returns $\text{Math.signum}(a)$, otherwise returns Double.NaN .

sin

```
public static final double sin(double a)
```

Sine trigonometric function

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN return Math.sin(a), otherwise return Double.NaN.

sinc

```
public static final double sinc(double a)
```

Sinc function.

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN and a <> 0 returns Math.sin(a) / (a), otherwise returns Double.NaN.

sinh

```
public static final double sinh(double a)
```

Hyperbolic sine function.

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.sinh(a), otherwise returns Double.NaN.

sqrt

```
public static final double sqrt(double a)
```

Square root.

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.sqrt(a), otherwise returns Double.NaN.

tan

```
public static final double tan(double a)
```

Tangent trigonometric function

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.tan(a), otherwise returns Double.NaN.

tanh

```
public static final double tanh(double a)
```

Hyperbolic tangent function.

Parameters:

a - the a function parameter

Returns:

if a <> Double.NaN returns Math.tanh(a), otherwise returns Double.NaN.

ulp

```
public static final double ulp(double value)
```

Unit in the last place(ULP) for double

Parameters:

value - Double number

Returns:

ULP for a given double.

ulpDecimalDigitsBefore

```
public static final int ulpDecimalDigitsBefore(double value)
```

Unit in The Last Place - number of decimal digits before

Parameters:

value - Double number

Returns:

Positive number of digits N for ulp = $1e^{-\{N+1\}}$, if ulp is > 1 then -1 is returned. Returned proper value is always between -1 and +99. If value is NaN then -2 is returned.

worpitzkyNumber

```
public static final double worpitzkyNumber(double n,  
                                             double k)
```

Worpitzky numbers

Parameters:

n - the n function parameter
k - the k function parameter

Returns:

if $n, k \geq 0$ returns `worpitzkyNumber((int)Math.round(n), (int)Math.round(k))`,
otherwise return `Double.NaN`.

worpitzkyNumber

```
public static final double worpitzkyNumber(int n,  
                                             int k)
```

Worpitzky numbers

Parameters:

n - the n function parameter
k - the k function parameter

Returns:

if $n, k \geq 0$ and $k \leq n$ return Worpitzky number, otherwise return `Double.NaN`.

org.mariuszgromada.math.mxparser.mathcollection

Class NumberTheory

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.mathcollection.NumberTheory
```

< [Constructors](#) > < [Methods](#) >

```
public final class NumberTheory  
extends java.lang.Object
```

NumberTheory - summation / products etc...

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - [mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Constructors

NumberTheory

```
public NumberTheory()
```

Methods

gcd

```
public static final double gcd(double a,  
                               double b)
```

Greatest common divisor (GCD)

Parameters:

a - the a function parameter

b - the b function parameter

Returns:

if a, b <> Double.NaN returns gcd((int)Math.round(a),(int)Math.round(b)), otherwise returns Double.NaN.

gcd

```
public static final double gcd(double[] numbers)
```

Greatest common divisor (GCD)

Parameters:

numbers - the numbers

Returns:

if each number from numbers \neq Double.NaN returns GCD(a₁,...,a_n) a₁,...,a_n in numbers, otherwise returns Double.NaN.

gcd

```
public static final double gcd(int a,  
                                int b)
```

Greatest common divisor (GCD)

Parameters:

a - the a function parameter

b - the b function parameter

Returns:

GCD(a,b)

gcd

```
public static final double gcd(int[] numbers)
```

Greatest common divisor (GCD)

Parameters:

numbers - the numbers

Returns:

GCD(a₁,...,a_n) a₁,...,a_n in numbers

lcm

```
public static final double lcm(double a,  
                                double b)
```

Latest common multiply (LCM)

Parameters:

a - the a function parameter

b - the b function parameter

Returns:

if a, b <> Double.NaN returns lcm((int)Math.round(a), (int)Math.round(b)), otherwise returns Double.NaN.

lcm

```
public static final double lcm(double[] numbers)
```

Latest common multiply (LCM)

Parameters:

numbers - the numbers

Returns:

if each number form numbers <> Double.NaN returns LCM(a_1,...,a_n) a_1,...,a_n in numbers, otherwise returns Double.NaN.

lcm

```
public static final double lcm(int a,  
                                int b)
```

Latest common multiply (LCM)

Parameters:

a - the a function parameter

b - the b function parameter

Returns:

LCM(a,b)

lcm

```
public static final double lcm(int[] numbers)
```

Latest common multiply (LCM)

Parameters:

numbers - the numbers

Returns:

LCM(a₁,...,a_n) a₁,...,a_n in numbers

max

```
public static final double max(double a,  
                                double b)
```

Maximum function.

Parameters:

a - the a function parameter

b - the b function parameter

Returns:

if a,b <> Double.NaN returns Math.max(a, b), otherwise returns Double.NaN.

max

```
public static final double max(double[] numbers)
```

Maximum function.

Parameters:

numbers - the a function parameter

Returns:

if each number form numbers <> Double.NaN returns the highest number, otherwise returns Double.NaN.

max

```
public static final double max(Expression f,  
                                Argument index,  
                                double from,  
                                double to,  
                                double delta)
```

Maximum value - iterative operator.

Parameters:

f - the expression
index - the name of index argument
from - FROM index = from
to - TO index = to
delta - BY delta

Returns:

product operation (for empty product operations returns 1).

min

```
public static final double min(double a,  
                                double b)
```

Minimum function.

Parameters:

a - the a function parameter
b - the b function parameter

Returns:

if a,b <> Double.NaN returns Math.min(a, b), otherwise returns Double.NaN.

min

```
public static final double min(double[] numbers)
```

Minimum function.

Parameters:

numbers - the a function parameter

Returns:

if each number form numbers <> Double.NaN returns the smallest number, otherwise returns Double.NaN.

min

```
public static final double min(Expression f,  
                               Argument index,  
                               double from,  
                               double to,  
                               double delta)
```

Minimum value - iterative operator.

Parameters:

f - the expression
index - the name of index argument
from - FROM index = form
to - TO index = to
delta - BY delta

Returns:

product operation (for empty product operations returns 1).

piProduct

```
public static final double piProduct(Expression f,  
                                     Argument index,  
                                     double from,  
                                     double to,  
                                     double delta)
```

Product operator

Parameters:

f - the expression
index - the name of index argument
from - FROM index = form
to - TO index = to
delta - BY delta

Returns:

product operation (for empty product operations returns 1).

primeCount

```
public static final double primeCount(double n)
```

Prime counting function

Parameters:

n - number

Returns:

Number of primes below or equal x

primeCount

```
public static final long primeCount(long n)
```

Prime counting function

Parameters:

n - number

Returns:

Number of primes below or equal x

primeTest

```
public static final double primeTest(double n)
```

Prime test

Parameters:

n - The number to be tested.

Returns:

true if number is prime, otherwise false

primeTest

```
public static final boolean primeTest(long n)
```

Prime test

Parameters:

n - The number to be tested.

Returns:

true if number is prime, otherwise false

prod

```
public static final double prod(double[] numbers)
```

Numbers multiplication.

Parameters:

numbers - the numbers

Returns:

if each number from numbers \neq Double.NaN returns $\text{prod}(a_1, \dots, a_n)$ a_1, \dots, a_n in numbers, otherwise returns Double.NaN.

sigmaSummation

```
public static final double sigmaSummation(Expression f,  
                                           Argument index,  
                                           double from,  
                                           double to,  
                                           double delta)
```

Summation operator (SIGMA FROM i = a, to b, f(i) by delta

Parameters:

f - the expression
index - the name of index argument
from - FROM index = from
to - TO index = to
delta - BY delta

Returns:

summation operation (for empty summation operations returns 0).

sum

```
public static final double sum(double[] numbers)
```

Adding numbers.

Parameters:

numbers - the numbers

Returns:

if each number from numbers <> Double.NaN returns sum(a₁,...,a_n) a₁,...,a_n in numbers, otherwise returns Double.NaN.

org.mariuszgromada.math.mxparser.mathcollection

Class PhysicalConstants

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.mathcollection.PhysicalConstants
```

< [Fields](#) > < [Constructors](#) >

```
public final class PhysicalConstants  
extends java.lang.Object
```

PhysicalConstants - class representing the most important physical constants.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

[MathSpace.pl](#)

[MathParser.org - mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.0.0

Fields

GRAVITATIONAL_CONSTANT

```
public static final double GRAVITATIONAL_CONSTANT  
    Gravitational constant
```

GRAVIT_ACC_EARTH

```
public static final double GRAVIT_ACC_EARTH  
    Gravitational acceleration - Earth (normal)
```

LIGHT_SPEED

```
public static final double LIGHT_SPEED  
    Light speed
```

PLANCK_CONSTANT

```
public static final double PLANCK_CONSTANT  
    Planck constant
```

PLANCK_CONSTANT_REDUCED

```
public static final double PLANCK_CONSTANT_REDUCED  
    Reduced Planck constant
```

PLANCK_LENGTH

```
public static final double PLANCK_LENGTH  
    Planck length
```

PLANCK_MASS

```
public static final double PLANCK_MASS  
    Planck mass
```

PLANCK_TIME

```
public static final double PLANCK_TIME  
    Planck time
```

Constructors

PhysicalConstants

```
public PhysicalConstants()
```

org.mariuszgromada.math.mxparser.mathcollection

Class PrimesCache

```
java.lang.Object  
|  
|--org.mariuszgromada.math.mxparser.mathcollection.PrimesCache
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class PrimesCache  
    extends java.lang.Object
```

Class for generating prime numbers cache using Eratosthenes Sieve.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Fields

CACHE_EMPTY

```
public static final boolean CACHE_EMPTY  
    Empty cache status
```

CACHING_FINISHED

```
public static final boolean CACHING_FINISHED  
    Cache ready to use
```

DEFAULT_MAX_NUM_IN_CACHE

```
public static final int DEFAULT_MAX_NUM_IN_CACHE  
    Default range of integer to store in cache
```

IS_NOT_PRIME

```
public static final int IS_NOT_PRIME  
    Indicator if given number is not a prime
```

IS_PRIME

```
public static final int IS_PRIME  
    Indicator if given number is a prime
```

NOT_IN_CACHE

```
public static final int NOT_IN_CACHE  
    Indicator that the value is not stored in cache
```

cacheStatus

```
boolean cacheStatus  
    Caching process status
```

computingTime

```
double computingTime  
    Time in seconds showing how long did it take to finalize prime numbers caching.
```

isPrime

```
boolean[] isPrime
```

maxNumInCache

```
int maxNumInCache  
    Primes between 0 ... and ... maximumNumberInCache will be cached
```

numberOfPrimes

```
int numberOfPrimes  
    Number of cached prime numbers
```

Constructors

PrimesCache

```
public PrimesCache()  
    Default constructor - setting prime cache for a default range if integers
```

PrimesCache

```
public PrimesCache(int maxNumInCache)  
    Constructor - setting prime cache for a given range if integers  
Parameters:  
    maxNumInCache - Range of integers to be stored in prime cache
```

Methods

getCacheStatus

```
public boolean getCacheStatus()
```

Returns cache status

Returns:

PrimesCache.CACHE_EMPTY or PrimesCache.CACHING_FINISHED;

getComputingTime

```
public double getComputingTime()
```

Returns computing time of Eratosthenes Sieve

Returns:

Computing time in seconds

getMaxNumInCache

```
public int getMaxNumInCache()
```

Returns cache range.

Returns:

Maximum integera number in cache/

getNumberOfPrimes

```
public int getNumberOfPrimes()
```

Returns number of found primes.

Returns:

Number of found primes.

getPrimes

```
boolean[] getPrimes()
```

Gets underlying primes cache boolean table

Returns:

Underlying primes cache boolean table

primeTest

```
public int primeTest(int n)
```

Check whether given number is prime

Parameters:

n - Given integer number.

Returns:

PrimesCache.IS_PRIME or PrimesCache.IS_NOT_PRIME or
PrimesCache.NOT_IN_CACHE

org.mariuszgromada.math.mxparser.mathcollection

Class ProbabilityDistributions

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.mathcollection.ProbabilityDistributions
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public final class ProbabilityDistributions
extends java.lang.Object
```

ProbabilityDistributions - random number generators, PDF - Probability Distribution Functions, CDF - Cumulative Distribution Functions, QNT - Quantile Functions (Inverse Cumulative Distribution Functions).

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Fields

randomGenerator

```
public static java.util.Random randomGenerator  
    Random number generator
```

Constructors

ProbabilityDistributions

```
public ProbabilityDistributions()
```

Methods

cdfNormal

```
public static final double cdfNormal(double x,  
                                     double mean,  
                                     double stddev)
```

CDF - Cumulative Distribution Function - Normal distribution $N(\text{mean}, \text{stddev})$.

Parameters:

x - Point to evaluate pdf function.
mean - Mean value.
stddev - Standard deviation.

Returns:

Double.NaN if mean or stddev is null or stddev is lower than 0 - otherwise function value.

cdfUniformContinuous

```
public static final double cdfUniformContinuous(double x,  
                                                  double a,  
                                                  double b)
```

CDF - Cumulative Distribution Function - Uniform Continuous distribution over interval $[a, b)$.

Parameters:

x - Point to evaluate cdf function.
a - Interval limit - left / lower.
b - Interval limit - right / upper.

Returns:

Double.NaN if a or b is null, or b is lower than a - otherwise function value.

pdfNormal

```
public static final double pdfNormal(double x,  
                                     double mean,  
                                     double stddev)
```

PDF - Probability Distribution Function - Normal distribution $N(\text{mean}, \text{stddev})$.

Parameters:

x - Point to evaluate pdf function.
mean - Mean value.
stddev - Standard deviation.

Returns:

Double.NaN if mean or stddev is null or stddev is lower than 0 - otherwise function value.

pdfUniformContinuous

```
public static final double pdfUniformContinuous(double x,  
                                                 double a,  
                                                 double b)
```

PDF - Probability Distribution Function - Uniform Continuous distribution over interval $[a, b)$.

Parameters:

x - Point to evaluate pdf function.
a - Interval limit - left / lower.
b - Interval limit - right / upper.

Returns:

Double.NaN if a or b is null, or b is lower than a - otherwise function value.

qntNormal

```
public static final double qntNormal(double q,  
                                     double mean,  
                                     double stddev)
```

QNT - Quantile Function - Normal distribution $N(\text{mean}, \text{stddev})$. (Inverse of Cumulative Distribution Function).

Parameters:

q - Quantile.
mean - Mean value.
stddev - Standard deviation.

Returns:

Double.NaN if mean or stddev is null or stddev is lower than 0 or q is lower than 0 or q is greater than 1 - otherwise function value.

qntUniformContinuous

```
public static final double qntUniformContinuous(double q,  
                                                  double a,  
                                                  double b)
```

QNT - Quantile Function - Uniform Continuous distribution over interval [a, b). (Inverse of Cumulative Distribution Function).

Parameters:

q - Quantile.
a - Interval limit - left / lower.
b - Interval limit - right / upper.

Returns:

Double.NaN if a or b is null, or b is lower than a or q is lower than 0 or q is greater than 1 - otherwise function value.

randomUniformContinuous

```
public static final double randomUniformContinuous()
```

Random number from Uniform Continuous distribution over interval [0, 1).

Returns:

Random number.

rndIndex

```
public static final int rndIndex(int n)
```

Random index from 0 to n-1,

Parameters:

n - Bound.

Returns:

if n < 0 returns -1, otherwise random index.

rndIndex

```
public static final int rndIndex(int n,  
                                java.util.Random rnd)
```

Random index from 0 to n-1,

Parameters:

n - Bound.

rnd - Random number generator.

Returns:

if n < 0 returns -1, otherwise random index.

rndInteger

```
public static final int rndInteger()
```

Random integer.

Returns:

Double.NaN if a or b is null, or b is lower than a - otherwise returns random number.

rndInteger

```
public static final double rndInteger(int a,  
                                       int b)
```

Random number from Uniform Discrete distribution. over set interval (a, a+1, ..., b-1, b).

Parameters:

a - Interval limit - left / lower.

b - Interval limit - right / upper.

Returns:

Double.NaN if a or b is null, or b is lower than a - otherwise returns random number.

rndInteger

```
public static final double rndInteger(int a,  
                                       int b,  
                                       java.util.Random rnd)
```

Random number from Uniform Discrete distribution. over set interval (a, a+1, ..., b-1, b).

Parameters:

a - Interval limit - left / lower.
b - Interval limit - right / upper.
rnd - Random number generator.

Returns:

Double.NaN if a or b is null, or b is lower than a - otherwise returns random number.

rndInteger

```
public static final int rndInteger(java.util.Random rnd)
```

Random integer.

Parameters:

rnd - Random number generator.

Returns:

Returns random number.

rndNormal

```
public static final double rndNormal(double mean,  
                                       double stddev)
```

Random number from normal distribution N(mean, stddev).

Parameters:

mean - Mean value.
stddev - Standard deviation.

Returns:

Double.NaN if mean or stddev is null or stddev is lower than 0 - otherwise random number.

rndNormal

```
public static final double rndNormal(double mean,  
                                     double stddev,  
                                     java.util.Random rnd)
```

Random number from normal distribution $N(\text{mean}, \text{stddev})$.

Parameters:

mean - Mean value.
stddev - Standard deviation.
rnd - Random number generator.

Returns:

Double.NaN if mean or stddev or rnd is null or stddev is lower than 0 - otherwise random number.

rndUniformContinuous

```
public static final double rndUniformContinuous(double a,  
                                                  double b)
```

Random number from dUniform Continuous distribution over interval [a, b).

Parameters:

a - Interval limit - left / lower.
b - Interval limit - right / upper.

Returns:

Double.NaN if a or b is null, or b is lower than a - otherwise returns random number.

rndUniformContinuous

```
public static final double rndUniformContinuous(double a,  
                                                  double b,  
                                                  java.util.Random rnd)
```

Random number from Uniform Continuous distribution over interval [a, b).

Parameters:

a - Interval limit - left / lower.
b - Interval limit - right / upper.
rnd - Random number generator.

Returns:

Double.NaN if a or b is null, or b is lower than a - otherwise returns random number.

rndUniformContinuous

```
public static final double rndUniformContinuous(java.util.Random rnd)
```

Random number from Uniform Continuous distribution over interval [0, 1).

Parameters:

rnd - Random number generator.

Returns:

Random number.

org.mariuszgromada.math.mxparser.mathcollection

Class SpecialFunctions

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.mathcollection.SpecialFunctions
```

< [Constructors](#) > < [Methods](#) >

```
public final class SpecialFunctions
extends java.lang.Object
```

SpecialFunctions - special (non-elementary functions).

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Constructors

SpecialFunctions

```
public SpecialFunctions()
```

Methods

erf

```
public static final double erf(double x)
```

Calculates the error function

Parameters:

x - Point at which function will be evaluated.

Returns:

Error function erf(x)

erflnv

```
public static final double erfInv(double x)
```

Calculates the inverse error function evaluated at x.

Parameters:

x - Point at which function will be evaluated.

Returns:

Inverse error function erfInv(x)

erfc

```
public static final double erfc(double x)
```

Calculates the complementary error function.

Parameters:

x - Point at which function will be evaluated.

Returns:

Complementary error function erfc(x)

erfcInv

```
public static final double erfcInv(double z)
```

Calculates the complementary inverse error function evaluated at x.

Parameters:

z - Point at which function will be evaluated.

Returns:

Inverse of complementary inverse error function erfcInv(x)

exponentialIntegralEi

```
public static double exponentialIntegralEi(double x)
```

Exponential integral function Ei(x)

Parameters:

x - Point at which function will be evaluated.

Returns:

Exponential integral function Ei(x)

logarithmicIntegralLi

```
public static final double logarithmicIntegralLi(double x)
```

Logarithmic integral function li(x)

Parameters:

x - Point at which function will be evaluated.

Returns:

Logarithmic integral function li(x)

offsetLogarithmicIntegralLi

```
public static final double offsetLogarithmicIntegralLi(double x)
```

Offset logarithmic integral function Li(x)

Parameters:

x - Point at which function will be evaluated.

Returns:

Offset logarithmic integral function Li(x)

org.mariuszgromada.math.mxparser.mathcollection

Class Statistics

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.mathcollection.Statistics
```

< [Constructors](#) > < [Methods](#) >

```
public final class Statistics
extends java.lang.Object
```

Statistics - i.e.: mean, variance, standard deviation, etc.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Constructors

Statistics

```
public Statistics()
```

Methods

avg

```
public static final double avg(double[] numbers)
```

Sample average.

Parameters:

numbers - the numbers

Returns:

if each number from numbers \neq Double.NaN returns avg(a₁,...,a_n) a₁,...,a_n in numbers, otherwise returns Double.NaN.

avg

```
public static final double avg(Expression f,  
                                Argument index,  
                                double from,  
                                double to,  
                                double delta)
```

Average from sample function values - iterative operator.

Parameters:

f - the expression

index - the name of index argument

from - FROM index = from

to - TO index = to

delta - BY delta

Returns:

product operation (for empty product operations returns 1).

std

```
public static final double std(double[] numbers)
```

Sample standard deviation (biased-corrected).

Parameters:

numbers - the numbers

Returns:

if each number from numbers \neq Double.NaN returns Std(a₁,...,a_n) a₁,...,a_n in numbers, otherwise returns Double.NaN.

std

```
public static final double std(Expression f,  
                             Argument index,  
                             double from,  
                             double to,  
                             double delta)
```

Bias-corrected standard deviation from sample function values - iterative operator.

Parameters:

f - the expression
index - the name of index argument
from - FROM index = from
to - TO index = to
delta - BY delta

Returns:

product operation (for empty product operations returns 1).

var

```
public static final double var(double[] numbers)
```

Sample variance (biased-corrected).

Parameters:

numbers - the numbers

Returns:

if each number from numbers \neq Double.NaN returns Var(a₁,...,a_n) a₁,...,a_n in numbers, otherwise returns Double.NaN.

var

```
public static final double var(Expression f,  
                             Argument index,  
                             double from,  
                             double to,  
                             double delta)
```

Bias-corrected variance from sample function values - iterative operator.

Parameters:

f - the expression
index - the name of index argument
from - FROM index = from
to - TO index = to
delta - BY delta

Returns:

product operation (for empty product operations returns 1).

org.mariuszgromada.math.mxparser.mathcollection

Class Units

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.mathcollection.Units
```

< [Fields](#) > < [Constructors](#) >

```
public final class Units
extends java.lang.Object
```

Units - class representing the most important units (length, area, volume, mass).

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.0.0

Fields

ACRE

```
public static final double ACRE
    Acre
```

ARE

```
public static final double ARE
    Are
```

ATTO

public static final double **ATTO**
Atto

BIT

public static final double **BIT**
Bit

BYTE

public static final double **BYTE**
Byte

CENTI

public static final double **CENTI**
Centi

CENTIMETRE

public static final double **CENTIMETRE**
Centimeter

CENTIMETRE2

public static final double **CENTIMETRE2**
Square centimetre

CENTIMETRE3

public static final double **CENTIMETRE3**
Qubic centimetre

DAY

public static final double **DAY**
DAY

DECA

```
public static final double DECA
    Deca
```

DECAGRAM

```
public static final double DECAGRAM
    Decagram
```

DECI

```
public static final double DECI
    Deci
```

DEGREE_ARC

```
public static final double DEGREE_ARC
    Degree (angle)
```

ELECTRONO_VOLT

```
public static final double ELECTRONO_VOLT
    Electrono-Volt
```

EXA

```
public static final double EXA
    Exa
```

EXABIT

```
public static final double EXABIT
    Exabit
```

EXABYTE

```
public static final double EXABYTE
    Exabyte
```

FEET

public static final double **FEET**
Feet

FEMTO

public static final double **FEMTO**
Femto

GALLON

public static final double **GALLON**
Gallon

GIGA

public static final double **GIGA**
Giga

GIGABIT

public static final double **GIGABIT**
Gigabit

GIGABYTE

public static final double **GIGABYTE**
Gigabyte

GIGA_ELECTRONO_VOLT

public static final double **GIGA_ELECTRONO_VOLT**
Giga Electrono-Volt

GRAM

public static final double **GRAM**
Gram

HECTARE

```
public static final double HECTARE
    Hectare
```

HECTO

```
public static final double HECTO
    Hecto
```

HOURL

```
public static final double HOURL
    HOURL
```

INCH

```
public static final double INCH
    Inch
```

JOULE

```
public static final double JOULE
    Jule
```

JULIAN_YEAR

```
public static final double JULIAN_YEAR
    JULIAN YEAR
```

KILO

```
public static final double KILO
    Kilo
```

KILOBIT

```
public static final double KILOBIT
    Kilobit
```

KILOBYTE

```
public static final double KILOBYTE  
    Kilobyte
```

KILOGRAM

```
public static final double KILOGRAM  
    Kilogram
```

KILOMETRE

```
public static final double KILOMETRE  
    Kilometer
```

KILOMETRE2

```
public static final double KILOMETRE2  
    Square kilometre
```

KILOMETRE3

```
public static final double KILOMETRE3  
    Qubic kilometre
```

KILOMETRE_PER_HOUR

```
public static final double KILOMETRE_PER_HOUR  
    Kilometre per hour
```

KILOMETRE_PER_HOUR2

```
public static final double KILOMETRE_PER_HOUR2  
    Kilometre per squared hour
```

KILO_ELECTRONO_VOLT

```
public static final double KILO_ELECTRONO_VOLT  
    Kilo Electrono-Volt
```

KNOT

public static final double **KNOT**
Knot

LITRE

public static final double **LITRE**
Litre

MEGA

public static final double **MEGA**
Mega

MEGABIT

public static final double **MEGABIT**
Megabit

MEGABYTE

public static final double **MEGABYTE**
Megabyte

MEGA_ELECTRONO_VOLT

public static final double **MEGA_ELECTRONO_VOLT**
Mega Electrono-Volt

METRE

public static final double **METRE**
Meter

METRE2

public static final double **METRE2**
Square metre

METRE3

```
public static final double METRE3
    Qubic metre
```

METRE_PER_SECOND

```
public static final double METRE_PER_SECOND
    Metre per second
```

METRE_PER_SECOND2

```
public static final double METRE_PER_SECOND2
    Metre per squared second
```

MICRO

```
public static final double MICRO
    Micro
```

MILE

```
public static final double MILE
    Mile
```

MILE_PER_HOUR

```
public static final double MILE_PER_HOUR
    Mile per hour
```

MILE_PER_HOUR2

```
public static final double MILE_PER_HOUR2
    Mile per squared hour
```

MILLI

```
public static final double MILLI
    Milli
```

MILLIGRAM

```
public static final double MILLIGRAM  
    Milligram
```

MILLILITRE

```
public static final double MILLILITRE  
    Millilitre
```

MILLIMETRE

```
public static final double MILLIMETRE  
    Millimetre
```

MILLIMETRE2

```
public static final double MILLIMETRE2  
    Square millimetre
```

MILLIMETRE3

```
public static final double MILLIMETRE3  
    Qubic millimetre
```

MILLISECOND

```
public static final double MILLISECOND  
    Millisecond
```

MINUTE

```
public static final double MINUTE  
    MINUTE
```

MINUTE_ARC

```
public static final double MINUTE_ARC  
    Minute (angle)
```

NANO

public static final double **NANO**
Nano

NAUTICAL_MILE

public static final double **NAUTICAL_MILE**
Nautical mile

OUNCE

public static final double **OUNCE**
Ounce

PERC

public static final double **PERC**
Percentage

PETA

public static final double **PETA**
Peta

PETABIT

public static final double **PETABIT**
Petabit

PETABYTE

public static final double **PETABYTE**
Petabyte

PICO

public static final double **PICO**
Pico

PINT

```
public static final double PINT  
    Pint
```

POUND

```
public static final double POUND  
    Pound
```

PROMIL

```
public static final double PROMIL  
    Promil, Per mille
```

RADIAN_ARC

```
public static final double RADIAN_ARC  
    Radian (angle)
```

SECOND

```
public static final double SECOND  
    Second
```

SECOND_ARC

```
public static final double SECOND_ARC  
    Second (angle)
```

TERA

```
public static final double TERA  
    Tera
```

TERABIT

```
public static final double TERABIT  
    Terabit
```

TERABYTE

public static final double **TERABYTE**
Terabyte

TERA_ELECTRONO_VOLT

public static final double **TERA_ELECTRONO_VOLT**
Tera Electrono-Volt

TONNE

public static final double **TONNE**
Tonne

WEEK

public static final double **WEEK**
WEEK

YARD

public static final double **YARD**
Yard

YOCTO

public static final double **YOCTO**
Yocto

YOTTA

public static final double **YOTTA**
Yotta

YOTTABIT

public static final double **YOTTABIT**
Yottabit

YOTTABYTE

```
public static final double YOTTABYTE  
    Yottabyte
```

ZEPTO

```
public static final double ZEPTO  
    Zepto
```

ZETTA

```
public static final double ZETTA  
    Zetta
```

ZETTABIT

```
public static final double ZETTABIT  
    Zettabit
```

ZETTABYTE

```
public static final double ZETTABYTE  
    Zettabyte
```

Constructors

Units

```
public Units()
```


Package org.mariuszgromada.math.mxparser.parsertoke

Class Summary

[BinaryRelation](#)

Binary Relations - mXparser tokens definition.

[BitwiseOperator](#)

Bitwise Operators - mXparser tokens definition.

[BooleanOperator](#)

Boolean Operators - mXparser tokens definition.

[CalculusOperator](#)

Calculus Operators - mXparser tokens definition.

[ConstantValue](#)

Constant Values - mXparser tokens definition.

[Function1Arg](#)

Unary functions (1 argument) - mXparser tokens definition.

[Function2Arg](#)

Binary functions (2 arguments) - mXparser tokens definition.

[Function3Arg](#)

Functions with 3 arguments - mXparser tokens definition.

[FunctionVariadic](#)

Variadic functions (n parameters)- mXparser tokens definition.

[KeyWord](#)

Class representing key words known to the parser

[Operator](#)

Operators - mXparser tokens definition.

[ParserSymbol](#)

Parser symbols - mXparser tokens definition.

[RandomVariable](#)

Random variables - mXparser tokens definition.

[Token](#)

Token recognized by mXparser after string tokenization process.

[Unit](#)

Units - mXparser tokens definition.

org.mariuszgromada.math.mxparser.parsertokens

Class BinaryRelation

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.parsertokens.BinaryRelation
```

< [Fields](#) > < [Constructors](#) >

```
public final class BinaryRelation
extends java.lang.Object
```

Binary Relations - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

EQ1_STR

```
public static final java.lang.String EQ1_STR
```

EQ1_SYN

```
public static final java.lang.String EQ1_SYN
```

EQ_DESC

```
public static final java.lang.String EQ_DESC
```

EQ_ID

```
public static final int EQ_ID
```

EQ_SINCE

```
public static final java.lang.String EQ_SINCE
```

EQ_STR

```
public static final java.lang.String EQ_STR
```

EQ_SYN

```
public static final java.lang.String EQ_SYN
```

GEQ_DESC

```
public static final java.lang.String GEQ_DESC
```

GEQ_ID

```
public static final int GEQ_ID
```

GEQ_SINCE

```
public static final java.lang.String GEQ_SINCE
```

GEQ_STR

```
public static final java.lang.String GEQ_STR
```

GEQ_SYN

```
public static final java.lang.String GEQ_SYN
```

GT_DESC

```
public static final java.lang.String GT_DESC
```

GT_ID

```
public static final int GT_ID
```

GT_SINCE

```
public static final java.lang.String GT_SINCE
```

GT_STR

```
public static final java.lang.String GT_STR
```

GT_SYN

```
public static final java.lang.String GT_SYN
```

LEQ_DESC

```
public static final java.lang.String LEQ_DESC
```

LEQ_ID

```
public static final int LEQ_ID
```

LEQ_SINCE

```
public static final java.lang.String LEQ_SINCE
```

LEQ_STR

```
public static final java.lang.String LEQ_STR
```

LEQ_SYN

```
public static final java.lang.String LEQ_SYN
```

LT_DESC

```
public static final java.lang.String LT_DESC
```

LT_ID

```
public static final int LT_ID
```

LT_SINCE

```
public static final java.lang.String LT_SINCE
```

LT_STR

```
public static final java.lang.String LT_STR
```

LT_SYN

```
public static final java.lang.String LT_SYN
```

NEQ1_STR

```
public static final java.lang.String NEQ1_STR
```

NEQ1_SYN

```
public static final java.lang.String NEQ1_SYN
```

NEQ2_STR

```
public static final java.lang.String NEQ2_STR
```

NEQ2_SYN

```
public static final java.lang.String NEQ2_SYN
```

NEQ_DESC

```
public static final java.lang.String NEQ_DESC
```

NEQ_ID

```
public static final int NEQ_ID
```

NEQ_SINCE

```
public static final java.lang.String NEQ_SINCE
```

NEQ_STR

```
public static final java.lang.String NEQ_STR
```

NEQ_SYN

```
public static final java.lang.String NEQ_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

Constructors

BinaryRelation

```
public BinaryRelation()
```

org.mariuszgromada.math.mxparser.parsertokens

Class BitwiseOperator

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.parsertokens.BitwiseOperator
```

< [Fields](#) > < [Constructors](#) >

```
public final class BitwiseOperator
extends java.lang.Object
```

Bitwise Operators - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

AND_DESC

```
public static final java.lang.String AND_DESC
```

AND_ID

```
public static final int AND_ID
```

AND_SINCE

```
public static final java.lang.String AND_SINCE
```

AND_STR

```
public static final java.lang.String AND_STR
```

AND_SYN

```
public static final java.lang.String AND_SYN
```

COMPL_DESC

```
public static final java.lang.String COMPL_DESC
```

COMPL_ID

```
public static final int COMPL_ID
```

COMPL_SINCE

```
public static final java.lang.String COMPL_SINCE
```

COMPL_STR

```
public static final java.lang.String COMPL_STR
```

COMPL_SYN

```
public static final java.lang.String COMPL_SYN
```

LEFT_SHIFT_DESC

```
public static final java.lang.String LEFT_SHIFT_DESC
```

LEFT_SHIFT_ID


```
public static final int LEFT_SHIFT_ID
```

LEFT_SHIFT_SINCE

```
public static final java.lang.String LEFT_SHIFT_SINCE
```

LEFT_SHIFT_STR

```
public static final java.lang.String LEFT_SHIFT_STR
```

LEFT_SHIFT_SYN

```
public static final java.lang.String LEFT_SHIFT_SYN
```

OR_DESC

```
public static final java.lang.String OR_DESC
```

OR_ID

```
public static final int OR_ID
```

OR_SINCE

```
public static final java.lang.String OR_SINCE
```

OR_STR

```
public static final java.lang.String OR_STR
```

OR_SYN

```
public static final java.lang.String OR_SYN
```

RIGHT_SHIFT_DESC

```
public static final java.lang.String RIGHT_SHIFT_DESC
```

RIGHT_SHIFT_ID

```
public static final int RIGHT_SHIFT_ID
```

RIGHT_SHIFT_SINCE

```
public static final java.lang.String RIGHT_SHIFT_SINCE
```

RIGHT_SHIFT_STR

```
public static final java.lang.String RIGHT_SHIFT_STR
```

RIGHT_SHIFT_SYN

```
public static final java.lang.String RIGHT_SHIFT_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

XOR_DESC

```
public static final java.lang.String XOR_DESC
```

XOR_ID

```
public static final int XOR_ID
```

XOR_SINCE

```
public static final java.lang.String XOR_SINCE
```

XOR_STR

```
public static final java.lang.String XOR_STR
```

XOR_SYN

```
public static final java.lang.String XOR_SYN
```

Constructors

BitwiseOperator

```
public BitwiseOperator()
```

org.mariuszgromada.math.mxparser.parsertokens

Class BooleanOperator

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.parsertokens.BooleanOperator
```

< [Fields](#) > < [Constructors](#) >

```
public final class BooleanOperator
extends java.lang.Object
```

Boolean Operators - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

AND1_STR

```
public static final java.lang.String AND1_STR
```

AND1_SYN

```
public static final java.lang.String AND1_SYN
```

AND2_STR

```
public static final java.lang.String AND2_STR
```

AND2_SYN

```
public static final java.lang.String AND2_SYN
```

AND_DESC

```
public static final java.lang.String AND_DESC
```

AND_ID

```
public static final int AND_ID
```

AND_SINCE

```
public static final java.lang.String AND_SINCE
```

AND_STR

```
public static final java.lang.String AND_STR
```

AND_SYN

```
public static final java.lang.String AND_SYN
```

CIMP_DESC

```
public static final java.lang.String CIMP_DESC
```

CIMP_ID

```
public static final int CIMP_ID
```

CIMP_SINCE

```
public static final java.lang.String CIMP_SINCE
```

CIMP_STR

```
public static final java.lang.String CIMP_STR
```

CIMP_SYN

```
public static final java.lang.String CIMP_SYN
```

CNIMP_DESC

```
public static final java.lang.String CNIMP_DESC
```

CNIMP_ID

```
public static final int CNIMP_ID
```

CNIMP_SINCE

```
public static final java.lang.String CNIMP_SINCE
```

CNIMP_STR

```
public static final java.lang.String CNIMP_STR
```

CNIMP_SYN

```
public static final java.lang.String CNIMP_SYN
```

EQV_DESC

```
public static final java.lang.String EQV_DESC
```

EQV_ID

```
public static final int EQV_ID
```

EQV_SINCE

```
public static final java.lang.String EQV_SINCE
```

EQV_STR

```
public static final java.lang.String EQV_STR
```

EQV_SYN

```
public static final java.lang.String EQV_SYN
```

IMP_DESC

```
public static final java.lang.String IMP_DESC
```

IMP_ID

```
public static final int IMP_ID
```

IMP_SINCE

```
public static final java.lang.String IMP_SINCE
```

IMP_STR

```
public static final java.lang.String IMP_STR
```

IMP_SYN

```
public static final java.lang.String IMP_SYN
```

NAND1_STR

```
public static final java.lang.String NAND1_STR
```

NAND1_SYN

```
public static final java.lang.String NAND1_SYN
```

NAND2_STR

```
public static final java.lang.String NAND2_STR
```

NAND2_SYN

```
public static final java.lang.String NAND2_SYN
```

NAND_DESC

```
public static final java.lang.String NAND_DESC
```

NAND_ID

```
public static final int NAND_ID
```

NAND_SINCE

```
public static final java.lang.String NAND_SINCE
```

NAND_STR

```
public static final java.lang.String NAND_STR
```

NAND_SYN

```
public static final java.lang.String NAND_SYN
```

NEG_DESC

```
public static final java.lang.String NEG_DESC
```

NEG_ID

```
public static final int NEG_ID
```

NEG_SINCE

```
public static final java.lang.String NEG_SINCE
```

NEG_STR

```
public static final java.lang.String NEG_STR
```

NEG_SYN

```
public static final java.lang.String NEG_SYN
```

NIMP_DESC

```
public static final java.lang.String NIMP_DESC
```

NIMP_ID

```
public static final int NIMP_ID
```

NIMP_SINCE

```
public static final java.lang.String NIMP_SINCE
```

NIMP_STR


```
public static final java.lang.String NIMP_STR
```

NIMP_SYN

```
public static final java.lang.String NIMP_SYN
```

NOR1_STR

```
public static final java.lang.String NOR1_STR
```

NOR1_SYN

```
public static final java.lang.String NOR1_SYN
```

NOR2_STR

```
public static final java.lang.String NOR2_STR
```

NOR2_SYN

```
public static final java.lang.String NOR2_SYN
```

NOR_DESC

```
public static final java.lang.String NOR_DESC
```

NOR_ID

```
public static final int NOR_ID
```

NOR_SINCE

```
public static final java.lang.String NOR_SINCE
```

NOR_STR

```
public static final java.lang.String NOR_STR
```

NOR_SYN

```
public static final java.lang.String NOR_SYN
```

OR1_STR

```
public static final java.lang.String OR1_STR
```

OR1_SYN

```
public static final java.lang.String OR1_SYN
```

OR2_STR

```
public static final java.lang.String OR2_STR
```

OR2_SYN

```
public static final java.lang.String OR2_SYN
```

OR_DESC

```
public static final java.lang.String OR_DESC
```

OR_ID

```
public static final int OR_ID
```

OR_SINCE

```
public static final java.lang.String OR_SINCE
```

OR_STR

```
public static final java.lang.String OR_STR
```

OR_SYN

```
public static final java.lang.String OR_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

XOR_DESC

```
public static final java.lang.String XOR_DESC
```

XOR_ID

```
public static final int XOR_ID
```

XOR_SINCE

```
public static final java.lang.String XOR_SINCE
```

XOR_STR

```
public static final java.lang.String XOR_STR
```

XOR_SYN

```
public static final java.lang.String XOR_SYN
```

Constructors

BooleanOperator

```
public BooleanOperator()
```

org.mariuszgromada.math.mxparser.parsertokens

Class CalculusOperator

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.parsertokens.CalculusOperator
```

< [Fields](#) > < [Constructors](#) >

```
public final class CalculusOperator
extends java.lang.Object
```

Calculus Operators - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

AVG_DESC

```
public static final java.lang.String AVG_DESC
```

AVG_ID

```
public static final int AVG_ID
```

AVG_SINCE

```
public static final java.lang.String AVG_SINCE
```

AVG_STR

```
public static final java.lang.String AVG_STR
```

AVG_SYN

```
public static final java.lang.String AVG_SYN
```

BACKW_DIFF_DESC

```
public static final java.lang.String BACKW_DIFF_DESC
```

BACKW_DIFF_ID

```
public static final int BACKW_DIFF_ID
```

BACKW_DIFF_SINCE

```
public static final java.lang.String BACKW_DIFF_SINCE
```

BACKW_DIFF_STR

```
public static final java.lang.String BACKW_DIFF_STR
```

BACKW_DIFF_SYN

```
public static final java.lang.String BACKW_DIFF_SYN
```

DERN_DESC

```
public static final java.lang.String DERN_DESC
```

DERN_ID

```
public static final int DERN_ID
```

DERN_SINCE

```
public static final java.lang.String DERN_SINCE
```

DERN_STR

```
public static final java.lang.String DERN_STR
```

DERN_SYN

```
public static final java.lang.String DERN_SYN
```

DER_DESC

```
public static final java.lang.String DER_DESC
```

DER_ID

```
public static final int DER_ID
```

DER_LEFT_DESC

```
public static final java.lang.String DER_LEFT_DESC
```

DER_LEFT_ID

```
public static final int DER_LEFT_ID
```

DER_LEFT_SINCE

```
public static final java.lang.String DER_LEFT_SINCE
```

DER_LEFT_STR

```
public static final java.lang.String DER_LEFT_STR
```

DER_LEFT_SYN

```
public static final java.lang.String DER_LEFT_SYN
```

DER_RIGHT_DESC

```
public static final java.lang.String DER_RIGHT_DESC
```

DER_RIGHT_ID

```
public static final int DER_RIGHT_ID
```

DER_RIGHT_SINCE

```
public static final java.lang.String DER_RIGHT_SINCE
```

DER_RIGHT_STR

```
public static final java.lang.String DER_RIGHT_STR
```

DER_RIGHT_SYN

```
public static final java.lang.String DER_RIGHT_SYN
```

DER_SINCE

```
public static final java.lang.String DER_SINCE
```

DER_STR

```
public static final java.lang.String DER_STR
```

DER_SYN

```
public static final java.lang.String DER_SYN
```

FORW_DIFF_DESC

```
public static final java.lang.String FORW_DIFF_DESC
```

FORW_DIFF_ID

```
public static final int FORW_DIFF_ID
```

FORW_DIFF_SINCE

```
public static final java.lang.String FORW_DIFF_SINCE
```

FORW_DIFF_STR

```
public static final java.lang.String FORW_DIFF_STR
```

FORW_DIFF_SYN

```
public static final java.lang.String FORW_DIFF_SYN
```

INT_DESC

```
public static final java.lang.String INT_DESC
```

INT_ID

```
public static final int INT_ID
```

INT_SINCE

```
public static final java.lang.String INT_SINCE
```

INT_STR

```
public static final java.lang.String INT_STR
```

INT_SYN

```
public static final java.lang.String INT_SYN
```

MAX_DESC

```
public static final java.lang.String MAX_DESC
```

MAX_ID

```
public static final int MAX_ID
```

MAX_SINCE

```
public static final java.lang.String MAX_SINCE
```

MAX_STR

```
public static final java.lang.String MAX_STR
```

MAX_SYN

```
public static final java.lang.String MAX_SYN
```

MIN_DESC

```
public static final java.lang.String MIN_DESC
```

MIN_ID

```
public static final int MIN_ID
```

MIN_SINCE

```
public static final java.lang.String MIN_SINCE
```

MIN_STR

```
public static final java.lang.String MIN_STR
```

MIN_SYN

```
public static final java.lang.String MIN_SYN
```

PROD_DESC

```
public static final java.lang.String PROD_DESC
```

PROD_ID

```
public static final int PROD_ID
```

PROD_SINCE

```
public static final java.lang.String PROD_SINCE
```

PROD_STR

```
public static final java.lang.String PROD_STR
```

PROD_SYN

```
public static final java.lang.String PROD_SYN
```

SOLVE_DESC

```
public static final java.lang.String SOLVE_DESC
```

SOLVE_ID

```
public static final int SOLVE_ID
```

SOLVE_SINCE

```
public static final java.lang.String SOLVE_SINCE
```

SOLVE_STR

```
public static final java.lang.String SOLVE_STR
```

SOLVE_SYN

```
public static final java.lang.String SOLVE_SYN
```

STD_DESC

```
public static final java.lang.String STD_DESC
```

STD_ID

```
public static final int STD_ID
```

STD_SINCE

```
public static final java.lang.String STD_SINCE
```

STD_STR

```
public static final java.lang.String STD_STR
```

STD_SYN

```
public static final java.lang.String STD_SYN
```

SUM_DESC

```
public static final java.lang.String SUM_DESC
```

SUM_ID

```
public static final int SUM_ID
```

SUM_SINCE

```
public static final java.lang.String SUM_SINCE
```

SUM_STR

```
public static final java.lang.String SUM_STR
```

SUM_SYN

```
public static final java.lang.String SUM_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

VAR_DESC

```
public static final java.lang.String VAR_DESC
```

VAR_ID

```
public static final int VAR_ID
```

VAR_SINCE

```
public static final java.lang.String VAR_SINCE
```

VAR_STR

```
public static final java.lang.String VAR_STR
```

VAR_SYN

```
public static final java.lang.String VAR_SYN
```

Constructors

CalculusOperator

```
public CalculusOperator()
```

org.mariuszgromada.math.mxparser.parsertokens

Class ConstantValue

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.parsertokens.ConstantValue
```

< [Fields](#) > < [Constructors](#) >

```
public final class ConstantValue
extends java.lang.Object
```

Constant Values - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

ALLADI_GRINSTEAD_DESC

```
public static final java.lang.String ALLADI_GRINSTEAD_DESC
```

ALLADI_GRINSTEAD_ID

```
public static final int ALLADI_GRINSTEAD_ID
```

ALLADI_GRINSTEAD_SINCE

```
public static final java.lang.String ALLADI_GRINSTEAD_SINCE
```

ALLADI_GRINSTEAD_STR

```
public static final java.lang.String ALLADI_GRINSTEAD_STR
```

ALLADI_GRINSTEAD_SYN

```
public static final java.lang.String ALLADI_GRINSTEAD_SYN
```

APERY_DESC

```
public static final java.lang.String APERY_DESC
```

APERY_ID

```
public static final int APERY_ID
```

APERY_SINCE

```
public static final java.lang.String APERY_SINCE
```

APERY_STR

```
public static final java.lang.String APERY_STR
```

APERY_SYN

```
public static final java.lang.String APERY_SYN
```

ASTRONOMICAL_UNIT_DESC

```
public static final java.lang.String ASTRONOMICAL_UNIT_DESC
```

ASTRONOMICAL_UNIT_ID

```
public static final int ASTRONOMICAL_UNIT_ID
```

ASTRONOMICAL_UNIT_SINCE

```
public static final java.lang.String ASTRONOMICAL_UNIT_SINCE
```

ASTRONOMICAL_UNIT_STR

```
public static final java.lang.String ASTRONOMICAL_UNIT_STR
```

ASTRONOMICAL_UNIT_SYN

```
public static final java.lang.String ASTRONOMICAL_UNIT_SYN
```

BACKHOUSE_DESC

```
public static final java.lang.String BACKHOUSE_DESC
```

BACKHOUSE_ID

```
public static final int BACKHOUSE_ID
```

BACKHOUSE_SINCE

```
public static final java.lang.String BACKHOUSE_SINCE
```

BACKHOUSE_STR

```
public static final java.lang.String BACKHOUSE_STR
```

BACKHOUSE_SYN

```
public static final java.lang.String BACKHOUSE_SYN
```

BERNSTEIN_DESC

```
public static final java.lang.String BERNSTEIN_DESC
```

BERNSTEIN_ID

```
public static final int BERNSTEIN_ID
```

BERNSTEIN_SINCE

```
public static final java.lang.String BERNSTEIN_SINCE
```

BERNSTEIN_STR

```
public static final java.lang.String BERNSTEIN_STR
```

BERNSTEIN_SYN

```
public static final java.lang.String BERNSTEIN_SYN
```

BRAUN_PRIME_QUADR_DESC

```
public static final java.lang.String BRAUN_PRIME_QUADR_DESC
```

BRAUN_PRIME_QUADR_ID

```
public static final int BRAUN_PRIME_QUADR_ID
```

BRAUN_PRIME_QUADR_SINCE

```
public static final java.lang.String BRAUN_PRIME_QUADR_SINCE
```

BRAUN_PRIME_QUADR_STR

```
public static final java.lang.String BRAUN_PRIME_QUADR_STR
```

BRAUN_PRIME_QUADR_SYN


```
public static final java.lang.String BRAUN_PRIME_QUADR_SYN
```

BRAUN_TWIN_PRIME_DESC

```
public static final java.lang.String BRAUN_TWIN_PRIME_DESC
```

BRAUN_TWIN_PRIME_ID

```
public static final int BRAUN_TWIN_PRIME_ID
```

BRAUN_TWIN_PRIME_SINCE

```
public static final java.lang.String BRAUN_TWIN_PRIME_SINCE
```

BRAUN_TWIN_PRIME_STR

```
public static final java.lang.String BRAUN_TWIN_PRIME_STR
```

BRAUN_TWIN_PRIME_SYN

```
public static final java.lang.String BRAUN_TWIN_PRIME_SYN
```

BRUIJN_NEWMAN_DESC

```
public static final java.lang.String BRUIJN_NEWMAN_DESC
```

BRUIJN_NEWMAN_ID

```
public static final int BRUIJN_NEWMAN_ID
```

BRUIJN_NEWMAN_SINCE

```
public static final java.lang.String BRUIJN_NEWMAN_SINCE
```

BRUIJN_NEWMAN_STR

```
public static final java.lang.String BRUIJN_NEWMAN_STR
```

BRUIJN_NEWMAN_SYN

```
public static final java.lang.String BRUIJN_NEWMAN_SYN
```

CAHEN_DESC

```
public static final java.lang.String CAHEN_DESC
```

CAHEN_ID

```
public static final int CAHEN_ID
```

CAHEN_SINCE

```
public static final java.lang.String CAHEN_SINCE
```

CAHEN_STR

```
public static final java.lang.String CAHEN_STR
```

CAHEN_SYN

```
public static final java.lang.String CAHEN_SYN
```

CATALAN_DESC

```
public static final java.lang.String CATALAN_DESC
```

CATALAN_ID

```
public static final int CATALAN_ID
```

CATALAN_SINCE

```
public static final java.lang.String CATALAN_SINCE
```

CATALAN_STR

```
public static final java.lang.String CATALAN_STR
```

CATALAN_SYN

```
public static final java.lang.String CATALAN_SYN
```

EARTH_MASS_DESC

```
public static final java.lang.String EARTH_MASS_DESC
```

EARTH_MASS_ID

```
public static final int EARTH_MASS_ID
```

EARTH_MASS_SINCE

```
public static final java.lang.String EARTH_MASS_SINCE
```

EARTH_MASS_STR

```
public static final java.lang.String EARTH_MASS_STR
```

EARTH_MASS_SYN

```
public static final java.lang.String EARTH_MASS_SYN
```

EARTH_RADIUS_EQUATORIAL_DESC

```
public static final java.lang.String EARTH_RADIUS_EQUATORIAL_DESC
```

EARTH_RADIUS_EQUATORIAL_ID

```
public static final int EARTH_RADIUS_EQUATORIAL_ID
```

EARTH_RADIUS_EQUATORIAL_SINCE

```
public static final java.lang.String EARTH_RADIUS_EQUATORIAL_SINCE
```

EARTH_RADIUS_EQUATORIAL_STR

```
public static final java.lang.String EARTH_RADIUS_EQUATORIAL_STR
```

EARTH_RADIUS_EQUATORIAL_SYN

```
public static final java.lang.String EARTH_RADIUS_EQUATORIAL_SYN
```

EARTH_RADIUS_MEAN_DESC

```
public static final java.lang.String EARTH_RADIUS_MEAN_DESC
```

EARTH_RADIUS_MEAN_ID

```
public static final int EARTH_RADIUS_MEAN_ID
```

EARTH_RADIUS_MEAN_SINCE

```
public static final java.lang.String EARTH_RADIUS_MEAN_SINCE
```

EARTH_RADIUS_MEAN_STR

```
public static final java.lang.String EARTH_RADIUS_MEAN_STR
```

EARTH_RADIUS_MEAN_SYN

```
public static final java.lang.String EARTH_RADIUS_MEAN_SYN
```

EARTH_RADIUS_POLAR_DESC

```
public static final java.lang.String EARTH_RADIUS_POLAR_DESC
```

EARTH_RADIUS_POLAR_ID

```
public static final int EARTH_RADIUS_POLAR_ID
```

EARTH_RADIUS_POLAR_SINCE

```
public static final java.lang.String EARTH_RADIUS_POLAR_SINCE
```

EARTH_RADIUS_POLAR_STR

```
public static final java.lang.String EARTH_RADIUS_POLAR_STR
```

EARTH_RADIUS_POLAR_SYN

```
public static final java.lang.String EARTH_RADIUS_POLAR_SYN
```

EARTH_SEMI_MAJOR_AXIS_DESC

```
public static final java.lang.String EARTH_SEMI_MAJOR_AXIS_DESC
```

EARTH_SEMI_MAJOR_AXIS_ID

```
public static final int EARTH_SEMI_MAJOR_AXIS_ID
```

EARTH_SEMI_MAJOR_AXIS_SINCE

```
public static final java.lang.String EARTH_SEMI_MAJOR_AXIS_SINCE
```

EARTH_SEMI_MAJOR_AXIS_STR

```
public static final java.lang.String EARTH_SEMI_MAJOR_AXIS_STR
```

EARTH_SEMI_MAJOR_AXIS_SYN

```
public static final java.lang.String EARTH_SEMI_MAJOR_AXIS_SYN
```

EMBREE_TREFETHEN_DESC

```
public static final java.lang.String EMBREE_TREFETHEN_DESC
```

EMBREE_TREFETHEN_ID

```
public static final int EMBREE_TREFETHEN_ID
```

EMBREE_TREFETHEN_SINCE

```
public static final java.lang.String EMBREE_TREFETHEN_SINCE
```

EMBREE_TREFETHEN_STR

```
public static final java.lang.String EMBREE_TREFETHEN_STR
```

EMBREE_TREFETHEN_SYN

```
public static final java.lang.String EMBREE_TREFETHEN_SYN
```

ERDOS_BORWEIN_DESC

```
public static final java.lang.String ERDOS_BORWEIN_DESC
```

ERDOS_BORWEIN_ID

```
public static final int ERDOS_BORWEIN_ID
```

ERDOS_BORWEIN_SINCE

```
public static final java.lang.String ERDOS_BORWEIN_SINCE
```

ERDOS_BORWEIN_STR

```
public static final java.lang.String ERDOS_BORWEIN_STR
```

ERDOS_BORWEIN_SYN

```
public static final java.lang.String ERDOS_BORWEIN_SYN
```

EULER_DESC

```
public static final java.lang.String EULER_DESC
```

EULER_ID

```
public static final int EULER_ID
```

EULER_MASCHERONI_DESC

```
public static final java.lang.String EULER_MASCHERONI_DESC
```

EULER_MASCHERONI_ID

```
public static final int EULER_MASCHERONI_ID
```

EULER_MASCHERONI_SINCE

```
public static final java.lang.String EULER_MASCHERONI_SINCE
```

EULER_MASCHERONI_STR

```
public static final java.lang.String EULER_MASCHERONI_STR
```

EULER_MASCHERONI_SYN

```
public static final java.lang.String EULER_MASCHERONI_SYN
```

EULER_SINCE

```
public static final java.lang.String EULER_SINCE
```

EULER_STR

```
public static final java.lang.String EULER_STR
```

EULER_SYN

```
public static final java.lang.String EULER_SYN
```

FALSE_DESC

```
public static final java.lang.String FALSE_DESC
```

FALSE_ID

```
public static final int FALSE_ID
```

FALSE_SINCE

```
public static final java.lang.String FALSE_SINCE
```

FALSE_STR

```
public static final java.lang.String FALSE_STR
```

FALSE_SYN

```
public static final java.lang.String FALSE_SYN
```

FEIGENBAUM_ALFA_DESC

```
public static final java.lang.String FEIGENBAUM_ALFA_DESC
```

FEIGENBAUM_ALFA_ID

```
public static final int FEIGENBAUM_ALFA_ID
```

FEIGENBAUM_ALFA_SINCE

```
public static final java.lang.String FEIGENBAUM_ALFA_SINCE
```

FEIGENBAUM_ALFA_STR

```
public static final java.lang.String FEIGENBAUM_ALFA_STR
```

FEIGENBAUM_ALFA_SYN

```
public static final java.lang.String FEIGENBAUM_ALFA_SYN
```

FEIGENBAUM_DELTA_DESC


```
public static final java.lang.String FEIGENBAUM_DELTA_DESC
```

FEIGENBAUM_DELTA_ID

```
public static final int FEIGENBAUM_DELTA_ID
```

FEIGENBAUM_DELTA_SINCE

```
public static final java.lang.String FEIGENBAUM_DELTA_SINCE
```

FEIGENBAUM_DELTA_STR

```
public static final java.lang.String FEIGENBAUM_DELTA_STR
```

FEIGENBAUM_DELTA_SYN

```
public static final java.lang.String FEIGENBAUM_DELTA_SYN
```

FRANSEN_ROBINSON_DESC

```
public static final java.lang.String FRANSEN_ROBINSON_DESC
```

FRANSEN_ROBINSON_ID

```
public static final int FRANSEN_ROBINSON_ID
```

FRANSEN_ROBINSON_SINCE

```
public static final java.lang.String FRANSEN_ROBINSON_SINCE
```

FRANSEN_ROBINSON_STR

```
public static final java.lang.String FRANSEN_ROBINSON_STR
```

FRANSEN_ROBINSON_SYN

```
public static final java.lang.String FRANSEN_ROBINSON_SYN
```

GAUSS_KUZMIN_WIRSING_DESC

```
public static final java.lang.String GAUSS_KUZMIN_WIRSING_DESC
```

GAUSS_KUZMIN_WIRSING_ID

```
public static final int GAUSS_KUZMIN_WIRSING_ID
```

GAUSS_KUZMIN_WIRSING_SINCE

```
public static final java.lang.String GAUSS_KUZMIN_WIRSING_SINCE
```

GAUSS_KUZMIN_WIRSING_STR

```
public static final java.lang.String GAUSS_KUZMIN_WIRSING_STR
```

GAUSS_KUZMIN_WIRSING_SYN

```
public static final java.lang.String GAUSS_KUZMIN_WIRSING_SYN
```

GOLDEN_RATIO_DESC

```
public static final java.lang.String GOLDEN_RATIO_DESC
```

GOLDEN_RATIO_ID

```
public static final int GOLDEN_RATIO_ID
```

GOLDEN_RATIO_SINCE

```
public static final java.lang.String GOLDEN_RATIO_SINCE
```

GOLDEN_RATIO_STR

```
public static final java.lang.String GOLDEN_RATIO_STR
```

GOLDEN_RATIO_SYN

```
public static final java.lang.String GOLDEN_RATIO_SYN
```

GOLOMB_DICKMAN_DESC

```
public static final java.lang.String GOLOMB_DICKMAN_DESC
```

GOLOMB_DICKMAN_ID

```
public static final int GOLOMB_DICKMAN_ID
```

GOLOMB_DICKMAN_SINCE

```
public static final java.lang.String GOLOMB_DICKMAN_SINCE
```

GOLOMB_DICKMAN_STR

```
public static final java.lang.String GOLOMB_DICKMAN_STR
```

GOLOMB_DICKMAN_SYN

```
public static final java.lang.String GOLOMB_DICKMAN_SYN
```

GOMPERTZ_DESC

```
public static final java.lang.String GOMPERTZ_DESC
```

GOMPERTZ_ID

```
public static final int GOMPERTZ_ID
```

GOMPERTZ_SINCE

```
public static final java.lang.String GOMPERTZ_SINCE
```

GOMPERTZ_STR

```
public static final java.lang.String GOMPERTZ_STR
```

GOMPERTZ_SYN

```
public static final java.lang.String GOMPERTZ_SYN
```

GRAVITATIONAL_CONSTANT_DESC

```
public static final java.lang.String GRAVITATIONAL_CONSTANT_DESC
```

GRAVITATIONAL_CONSTANT_ID

```
public static final int GRAVITATIONAL_CONSTANT_ID
```

GRAVITATIONAL_CONSTANT_SINCE

```
public static final java.lang.String GRAVITATIONAL_CONSTANT_SINCE
```

GRAVITATIONAL_CONSTANT_STR

```
public static final java.lang.String GRAVITATIONAL_CONSTANT_STR
```

GRAVITATIONAL_CONSTANT_SYN

```
public static final java.lang.String GRAVITATIONAL_CONSTANT_SYN
```

GRAVIT_ACC_EARTH_DESC

```
public static final java.lang.String GRAVIT_ACC_EARTH_DESC
```

GRAVIT_ACC_EARTH_ID

```
public static final int GRAVIT_ACC_EARTH_ID
```

GRAVIT_ACC_EARTH_SINCE

```
public static final java.lang.String GRAVIT_ACC_EARTH_SINCE
```

GRAVIT_ACC_EARTH_STR

```
public static final java.lang.String GRAVIT_ACC_EARTH_STR
```

GRAVIT_ACC_EARTH_SYN

```
public static final java.lang.String GRAVIT_ACC_EARTH_SYN
```

HAFNER_SARNAK_MCCURLEY_DESC

```
public static final java.lang.String HAFNER_SARNAK_MCCURLEY_DESC
```

HAFNER_SARNAK_MCCURLEY_ID

```
public static final int HAFNER_SARNAK_MCCURLEY_ID
```

HAFNER_SARNAK_MCCURLEY_SINCE

```
public static final java.lang.String HAFNER_SARNAK_MCCURLEY_SINCE
```

HAFNER_SARNAK_MCCURLEY_STR

```
public static final java.lang.String HAFNER_SARNAK_MCCURLEY_STR
```

HAFNER_SARNAK_MCCURLEY_SYN

```
public static final java.lang.String HAFNER_SARNAK_MCCURLEY_SYN
```

JUPITER_MASS_DESC

```
public static final java.lang.String JUPITER_MASS_DESC
```

JUPITER_MASS_ID

```
public static final int JUPITER_MASS_ID
```

JUPITER_MASS_SINCE

```
public static final java.lang.String JUPITER_MASS_SINCE
```

JUPITER_MASS_STR

```
public static final java.lang.String JUPITER_MASS_STR
```

JUPITER_MASS_SYN

```
public static final java.lang.String JUPITER_MASS_SYN
```

JUPITER_RADIUS_MEAN_DESC

```
public static final java.lang.String JUPITER_RADIUS_MEAN_DESC
```

JUPITER_RADIUS_MEAN_ID

```
public static final int JUPITER_RADIUS_MEAN_ID
```

JUPITER_RADIUS_MEAN_SINCE

```
public static final java.lang.String JUPITER_RADIUS_MEAN_SINCE
```

JUPITER_RADIUS_MEAN_STR

```
public static final java.lang.String JUPITER_RADIUS_MEAN_STR
```

JUPITER_RADIUS_MEAN_SYN

```
public static final java.lang.String JUPITER_RADIUS_MEAN_SYN
```

JUPITER_SEMI_MAJOR_AXIS_DESC

```
public static final java.lang.String JUPITER_SEMI_MAJOR_AXIS_DESC
```

JUPITER_SEMI_MAJOR_AXIS_ID

```
public static final int JUPITER_SEMI_MAJOR_AXIS_ID
```

JUPITER_SEMI_MAJOR_AXIS_SINCE

```
public static final java.lang.String JUPITER_SEMI_MAJOR_AXIS_SINCE
```

JUPITER_SEMI_MAJOR_AXIS_STR

```
public static final java.lang.String JUPITER_SEMI_MAJOR_AXIS_STR
```

JUPITER_SEMI_MAJOR_AXIS_SYN

```
public static final java.lang.String JUPITER_SEMI_MAJOR_AXIS_SYN
```

KHINCHIN_DESC

```
public static final java.lang.String KHINCHIN_DESC
```

KHINCHIN_ID

```
public static final int KHINCHIN_ID
```

KHINCHIN_SINCE

```
public static final java.lang.String KHINCHIN_SINCE
```

KHINCHIN_STR

```
public static final java.lang.String KHINCHIN_STR
```

KHINCHIN_SYN

```
public static final java.lang.String KHINCHIN_SYN
```

KILOPARSEC_DESC

```
public static final java.lang.String KILOPARSEC_DESC
```

KILOPARSEC_ID

```
public static final int KILOPARSEC_ID
```

KILOPARSEC_SINCE

```
public static final java.lang.String KILOPARSEC_SINCE
```

KILOPARSEC_STR

```
public static final java.lang.String KILOPARSEC_STR
```

KILOPARSEC_SYN

```
public static final java.lang.String KILOPARSEC_SYN
```

LANDAU_DESC

```
public static final java.lang.String LANDAU_DESC
```

LANDAU_ID

```
public static final int LANDAU_ID
```

LANDAU_RAMANUJAN_DESC

```
public static final java.lang.String LANDAU_RAMANUJAN_DESC
```

LANDAU_RAMANUJAN_ID

```
public static final int LANDAU_RAMANUJAN_ID
```

LANDAU_RAMANUJAN_SINCE

```
public static final java.lang.String LANDAU_RAMANUJAN_SINCE
```

LANDAU_RAMANUJAN_STR

```
public static final java.lang.String LANDAU_RAMANUJAN_STR
```

LANDAU_RAMANUJAN_SYN


```
public static final java.lang.String LANDAU_RAMANUJAN_SYN
```

LANDAU_SINCE

```
public static final java.lang.String LANDAU_SINCE
```

LANDAU_STR

```
public static final java.lang.String LANDAU_STR
```

LANDAU_SYN

```
public static final java.lang.String LANDAU_SYN
```

LAPLACE_LIMIT_DESC

```
public static final java.lang.String LAPLACE_LIMIT_DESC
```

LAPLACE_LIMIT_ID

```
public static final int LAPLACE_LIMIT_ID
```

LAPLACE_LIMIT_SINCE

```
public static final java.lang.String LAPLACE_LIMIT_SINCE
```

LAPLACE_LIMIT_STR

```
public static final java.lang.String LAPLACE_LIMIT_STR
```

LAPLACE_LIMIT_SYN

```
public static final java.lang.String LAPLACE_LIMIT_SYN
```

LEGENDRE_DESC

```
public static final java.lang.String LEGENDRE_DESC
```

LEGENDRE_ID

```
public static final int LEGENDRE_ID
```

LEGENDRE_SINCE

```
public static final java.lang.String LEGENDRE_SINCE
```

LEGENDRE_STR

```
public static final java.lang.String LEGENDRE_STR
```

LEGENDRE_SYN

```
public static final java.lang.String LEGENDRE_SYN
```

LENGYEL_DESC

```
public static final java.lang.String LENGYEL_DESC
```

LENGYEL_ID

```
public static final int LENGYEL_ID
```

LENGYEL_SINCE

```
public static final java.lang.String LENGYEL_SINCE
```

LENGYEL_STR

```
public static final java.lang.String LENGYEL_STR
```

LENGYEL_SYN

```
public static final java.lang.String LENGYEL_SYN
```

LEVY_DESC

```
public static final java.lang.String LEVY_DESC
```

LEVY_ID

```
public static final int LEVY_ID
```

LEVY_SINCE

```
public static final java.lang.String LEVY_SINCE
```

LEVY_STR

```
public static final java.lang.String LEVY_STR
```

LEVY_SYN

```
public static final java.lang.String LEVY_SYN
```

LI2_DESC

```
public static final java.lang.String LI2_DESC
```

LI2_ID

```
public static final int LI2_ID
```

LI2_SINCE

```
public static final java.lang.String LI2_SINCE
```

LI2_STR

```
public static final java.lang.String LI2_STR
```

LI2_SYN

```
public static final java.lang.String LI2_SYN
```

LIEB_QUARE_ICE_DESC

```
public static final java.lang.String LIEB_QUARE_ICE_DESC
```

LIEB_QUARE_ICE_ID

```
public static final int LIEB_QUARE_ICE_ID
```

LIEB_QUARE_ICE_SINCE

```
public static final java.lang.String LIEB_QUARE_ICE_SINCE
```

LIEB_QUARE_ICE_STR

```
public static final java.lang.String LIEB_QUARE_ICE_STR
```

LIEB_QUARE_ICE_SYN

```
public static final java.lang.String LIEB_QUARE_ICE_SYN
```

LIGHT_SPEED_DESC

```
public static final java.lang.String LIGHT_SPEED_DESC
```

LIGHT_SPEED_ID

```
public static final int LIGHT_SPEED_ID
```

LIGHT_SPEED_SINCE

```
public static final java.lang.String LIGHT_SPEED_SINCE
```

LIGHT_SPEED_STR

```
public static final java.lang.String LIGHT_SPEED_STR
```

LIGHT_SPEED_SYN

```
public static final java.lang.String LIGHT_SPEED_SYN
```

LIGHT_YEAR_DESC

```
public static final java.lang.String LIGHT_YEAR_DESC
```

LIGHT_YEAR_ID

```
public static final int LIGHT_YEAR_ID
```

LIGHT_YEAR_SINCE

```
public static final java.lang.String LIGHT_YEAR_SINCE
```

LIGHT_YEAR_STR

```
public static final java.lang.String LIGHT_YEAR_STR
```

LIGHT_YEAR_SYN

```
public static final java.lang.String LIGHT_YEAR_SYN
```

MARS_MASS_DESC

```
public static final java.lang.String MARS_MASS_DESC
```

MARS_MASS_ID

```
public static final int MARS_MASS_ID
```

MARS_MASS_SINCE

```
public static final java.lang.String MARS_MASS_SINCE
```

MARS_MASS_STR

```
public static final java.lang.String MARS_MASS_STR
```

MARS_MASS_SYN

```
public static final java.lang.String MARS_MASS_SYN
```

MARS_RADIUS_MEAN_DESC

```
public static final java.lang.String MARS_RADIUS_MEAN_DESC
```

MARS_RADIUS_MEAN_ID

```
public static final int MARS_RADIUS_MEAN_ID
```

MARS_RADIUS_MEAN_SINCE

```
public static final java.lang.String MARS_RADIUS_MEAN_SINCE
```

MARS_RADIUS_MEAN_STR

```
public static final java.lang.String MARS_RADIUS_MEAN_STR
```

MARS_RADIUS_MEAN_SYN

```
public static final java.lang.String MARS_RADIUS_MEAN_SYN
```

MARS_SEMI_MAJOR_AXIS_DESC

```
public static final java.lang.String MARS_SEMI_MAJOR_AXIS_DESC
```

MARS_SEMI_MAJOR_AXIS_ID

```
public static final int MARS_SEMI_MAJOR_AXIS_ID
```

MARS_SEMI_MAJOR_AXIS_SINCE

```
public static final java.lang.String MARS_SEMI_MAJOR_AXIS_SINCE
```

MARS_SEMI_MAJOR_AXIS_STR

```
public static final java.lang.String MARS_SEMI_MAJOR_AXIS_STR
```

MARS_SEMI_MAJOR_AXIS_SYN

```
public static final java.lang.String MARS_SEMI_MAJOR_AXIS_SYN
```

MEISSEL_MERTEENS_DESC

```
public static final java.lang.String MEISSEL_MERTEENS_DESC
```

MEISSEL_MERTEENS_ID

```
public static final int MEISSEL_MERTEENS_ID
```

MEISSEL_MERTEENS_SINCE

```
public static final java.lang.String MEISSEL_MERTEENS_SINCE
```

MEISSEL_MERTEENS_STR

```
public static final java.lang.String MEISSEL_MERTEENS_STR
```

MEISSEL_MERTEENS_SYN

```
public static final java.lang.String MEISSEL_MERTEENS_SYN
```

MERCURY_MASS_DESC

```
public static final java.lang.String MERCURY_MASS_DESC
```

MERCURY_MASS_ID

```
public static final int MERCURY_MASS_ID
```

MERCURY_MASS_SINCE

```
public static final java.lang.String MERCURY_MASS_SINCE
```

MERCURY_MASS_STR

```
public static final java.lang.String MERCURY_MASS_STR
```

MERCURY_MASS_SYN

```
public static final java.lang.String MERCURY_MASS_SYN
```

MERCURY_RADIUS_MEAN_DESC

```
public static final java.lang.String MERCURY_RADIUS_MEAN_DESC
```

MERCURY_RADIUS_MEAN_ID

```
public static final int MERCURY_RADIUS_MEAN_ID
```

MERCURY_RADIUS_MEAN_SINCE

```
public static final java.lang.String MERCURY_RADIUS_MEAN_SINCE
```

MERCURY_RADIUS_MEAN_STR

```
public static final java.lang.String MERCURY_RADIUS_MEAN_STR
```

MERCURY_RADIUS_MEAN_SYN

```
public static final java.lang.String MERCURY_RADIUS_MEAN_SYN
```

MERCURY_SEMI_MAJOR_AXIS_DESC

```
public static final java.lang.String MERCURY_SEMI_MAJOR_AXIS_DESC
```

MERCURY_SEMI_MAJOR_AXIS_ID

```
public static final int MERCURY_SEMI_MAJOR_AXIS_ID
```

MERCURY_SEMI_MAJOR_AXIS_SINCE


```
public static final java.lang.String MERCURY_SEMI_MAJOR_AXIS_SINCE
```

MERCURY_SEMI_MAJOR_AXIS_STR

```
public static final java.lang.String MERCURY_SEMI_MAJOR_AXIS_STR
```

MERCURY_SEMI_MAJOR_AXIS_SYN

```
public static final java.lang.String MERCURY_SEMI_MAJOR_AXIS_SYN
```

MILLS_DESC

```
public static final java.lang.String MILLS_DESC
```

MILLS_ID

```
public static final int MILLS_ID
```

MILLS_SINCE

```
public static final java.lang.String MILLS_SINCE
```

MILLS_STR

```
public static final java.lang.String MILLS_STR
```

MILLS_SYN

```
public static final java.lang.String MILLS_SYN
```

MONN_SEMI_MAJOR_AXIS_DESC

```
public static final java.lang.String MONN_SEMI_MAJOR_AXIS_DESC
```

MONN_SEMI_MAJOR_AXIS_ID

```
public static final int MONN_SEMI_MAJOR_AXIS_ID
```

MONN_SEMI_MAJOR_AXIS_SINCE

```
public static final java.lang.String MONN_SEMI_MAJOR_AXIS_SINCE
```

MONN_SEMI_MAJOR_AXIS_STR

```
public static final java.lang.String MONN_SEMI_MAJOR_AXIS_STR
```

MONN_SEMI_MAJOR_AXIS_SYN

```
public static final java.lang.String MONN_SEMI_MAJOR_AXIS_SYN
```

MOON_MASS_DESC

```
public static final java.lang.String MOON_MASS_DESC
```

MOON_MASS_ID

```
public static final int MOON_MASS_ID
```

MOON_MASS_SINCE

```
public static final java.lang.String MOON_MASS_SINCE
```

MOON_MASS_STR

```
public static final java.lang.String MOON_MASS_STR
```

MOON_MASS_SYN

```
public static final java.lang.String MOON_MASS_SYN
```

MOON_RADIUS_MEAN_DESC

```
public static final java.lang.String MOON_RADIUS_MEAN_DESC
```

MOON_RADIUS_MEAN_ID

```
public static final int MOON_RADIUS_MEAN_ID
```

MOON_RADIUS_MEAN_SINCE

```
public static final java.lang.String MOON_RADIUS_MEAN_SINCE
```

MOON_RADIUS_MEAN_STR

```
public static final java.lang.String MOON_RADIUS_MEAN_STR
```

MOON_RADIUS_MEAN_SYN

```
public static final java.lang.String MOON_RADIUS_MEAN_SYN
```

MRB_DESC

```
public static final java.lang.String MRB_DESC
```

MRB_ID

```
public static final int MRB_ID
```

MRB_SINCE

```
public static final java.lang.String MRB_SINCE
```

MRB_STR

```
public static final java.lang.String MRB_STR
```

MRB_SYN

```
public static final java.lang.String MRB_SYN
```

NAN_DESC

```
public static final java.lang.String NAN_DESC
```

NAN_ID

```
public static final int NAN_ID
```

NAN_SINCE

```
public static final java.lang.String NAN_SINCE
```

NAN_STR

```
public static final java.lang.String NAN_STR
```

NAN_SYN

```
public static final java.lang.String NAN_SYN
```

NEPTUNE_MASS_DESC

```
public static final java.lang.String NEPTUNE_MASS_DESC
```

NEPTUNE_MASS_ID

```
public static final int NEPTUNE_MASS_ID
```

NEPTUNE_MASS_SINCE

```
public static final java.lang.String NEPTUNE_MASS_SINCE
```

NEPTUNE_MASS_STR

```
public static final java.lang.String NEPTUNE_MASS_STR
```

NEPTUNE_MASS_SYN

```
public static final java.lang.String NEPTUNE_MASS_SYN
```

NEPTUNE_RADIUS_MEAN_DESC

```
public static final java.lang.String NEPTUNE_RADIUS_MEAN_DESC
```

NEPTUNE_RADIUS_MEAN_ID

```
public static final int NEPTUNE_RADIUS_MEAN_ID
```

NEPTUNE_RADIUS_MEAN_SINCE

```
public static final java.lang.String NEPTUNE_RADIUS_MEAN_SINCE
```

NEPTUNE_RADIUS_MEAN_STR

```
public static final java.lang.String NEPTUNE_RADIUS_MEAN_STR
```

NEPTUNE_RADIUS_MEAN_SYN

```
public static final java.lang.String NEPTUNE_RADIUS_MEAN_SYN
```

NEPTUNE_SEMI_MAJOR_AXIS_DESC

```
public static final java.lang.String NEPTUNE_SEMI_MAJOR_AXIS_DESC
```

NEPTUNE_SEMI_MAJOR_AXIS_ID

```
public static final int NEPTUNE_SEMI_MAJOR_AXIS_ID
```

NEPTUNE_SEMI_MAJOR_AXIS_SINCE

```
public static final java.lang.String NEPTUNE_SEMI_MAJOR_AXIS_SINCE
```

NEPTUNE_SEMI_MAJOR_AXIS_STR

```
public static final java.lang.String NEPTUNE_SEMI_MAJOR_AXIS_STR
```

NEPTUNE_SEMI_MAJOR_AXIS_SYN

```
public static final java.lang.String NEPTUNE_SEMI_MAJOR_AXIS_SYN
```

NIVEN_DESC

```
public static final java.lang.String NIVEN_DESC
```

NIVEN_ID

```
public static final int NIVEN_ID
```

NIVEN_SINCE

```
public static final java.lang.String NIVEN_SINCE
```

NIVEN_STR

```
public static final java.lang.String NIVEN_STR
```

NIVEN_SYN

```
public static final java.lang.String NIVEN_SYN
```

NaN

```
public static final int NaN
```

OMEGA_DESC

```
public static final java.lang.String OMEGA_DESC
```

OMEGA_ID

```
public static final int OMEGA_ID
```

OMEGA_SINCE

```
public static final java.lang.String OMEGA_SINCE
```

OMEGA_STR

```
public static final java.lang.String OMEGA_STR
```

OMEGA_SYN

```
public static final java.lang.String OMEGA_SYN
```

PARABOLIC_DESC

```
public static final java.lang.String PARABOLIC_DESC
```

PARABOLIC_ID

```
public static final int PARABOLIC_ID
```

PARABOLIC_SINCE

```
public static final java.lang.String PARABOLIC_SINCE
```

PARABOLIC_STR

```
public static final java.lang.String PARABOLIC_STR
```

PARABOLIC_SYN

```
public static final java.lang.String PARABOLIC_SYN
```

PARSEC_DESC

```
public static final java.lang.String PARSEC_DESC
```

PARSEC_ID

```
public static final int PARSEC_ID
```

PARSEC_SINCE

```
public static final java.lang.String PARSEC_SINCE
```

PARSEC_STR

```
public static final java.lang.String PARSEC_STR
```

PARSEC_SYN

```
public static final java.lang.String PARSEC_SYN
```

PI_DESC

```
public static final java.lang.String PI_DESC
```

PI_ID

```
public static final int PI_ID
```

PI_SINCE

```
public static final java.lang.String PI_SINCE
```

PI_STR

```
public static final java.lang.String PI_STR
```

PI_SYN

```
public static final java.lang.String PI_SYN
```

PLANCK_CONSTANT_DESC

```
public static final java.lang.String PLANCK_CONSTANT_DESC
```

PLANCK_CONSTANT_ID

```
public static final int PLANCK_CONSTANT_ID
```

PLANCK_CONSTANT_REDUCED_DESC


```
public static final java.lang.String PLANCK_CONSTANT_REDUCED_DESC
```

PLANCK_CONSTANT_REDUCED_ID

```
public static final int PLANCK_CONSTANT_REDUCED_ID
```

PLANCK_CONSTANT_REDUCED_SINCE

```
public static final java.lang.String PLANCK_CONSTANT_REDUCED_SINCE
```

PLANCK_CONSTANT_REDUCED_STR

```
public static final java.lang.String PLANCK_CONSTANT_REDUCED_STR
```

PLANCK_CONSTANT_REDUCED_SYN

```
public static final java.lang.String PLANCK_CONSTANT_REDUCED_SYN
```

PLANCK_CONSTANT_SINCE

```
public static final java.lang.String PLANCK_CONSTANT_SINCE
```

PLANCK_CONSTANT_STR

```
public static final java.lang.String PLANCK_CONSTANT_STR
```

PLANCK_CONSTANT_SYN

```
public static final java.lang.String PLANCK_CONSTANT_SYN
```

PLANCK_LENGTH_DESC

```
public static final java.lang.String PLANCK_LENGTH_DESC
```

PLANCK_LENGTH_ID

```
public static final int PLANCK_LENGTH_ID
```

PLANCK_LENGTH_SINCE

```
public static final java.lang.String PLANCK_LENGTH_SINCE
```

PLANCK_LENGTH_STR

```
public static final java.lang.String PLANCK_LENGTH_STR
```

PLANCK_LENGTH_SYN

```
public static final java.lang.String PLANCK_LENGTH_SYN
```

PLANCK_MASS_DESC

```
public static final java.lang.String PLANCK_MASS_DESC
```

PLANCK_MASS_ID

```
public static final int PLANCK_MASS_ID
```

PLANCK_MASS_SINCE

```
public static final java.lang.String PLANCK_MASS_SINCE
```

PLANCK_MASS_STR

```
public static final java.lang.String PLANCK_MASS_STR
```

PLANCK_MASS_SYN

```
public static final java.lang.String PLANCK_MASS_SYN
```

PLANCK_TIME_DESC

```
public static final java.lang.String PLANCK_TIME_DESC
```

PLANCK_TIME_ID

```
public static final int PLANCK_TIME_ID
```

PLANCK_TIME_SINCE

```
public static final java.lang.String PLANCK_TIME_SINCE
```

PLANCK_TIME_STR

```
public static final java.lang.String PLANCK_TIME_STR
```

PLANCK_TIME_SYN

```
public static final java.lang.String PLANCK_TIME_SYN
```

PLASTIC_DESC

```
public static final java.lang.String PLASTIC_DESC
```

PLASTIC_ID

```
public static final int PLASTIC_ID
```

PLASTIC_SINCE

```
public static final java.lang.String PLASTIC_SINCE
```

PLASTIC_STR

```
public static final java.lang.String PLASTIC_STR
```

PLASTIC_SYN

```
public static final java.lang.String PLASTIC_SYN
```

PORTER_DESC

```
public static final java.lang.String PORTER_DESC
```

PORTER_ID

```
public static final int PORTER_ID
```

PORTER_SINCE

```
public static final java.lang.String PORTER_SINCE
```

PORTER_STR

```
public static final java.lang.String PORTER_STR
```

PORTER_SYN

```
public static final java.lang.String PORTER_SYN
```

RAMANUJAN_SOLDNER_DESC

```
public static final java.lang.String RAMANUJAN_SOLDNER_DESC
```

RAMANUJAN_SOLDNER_ID

```
public static final int RAMANUJAN_SOLDNER_ID
```

RAMANUJAN_SOLDNER_SINCE

```
public static final java.lang.String RAMANUJAN_SOLDNER_SINCE
```

RAMANUJAN_SOLDNER_STR

```
public static final java.lang.String RAMANUJAN_SOLDNER_STR
```

RAMANUJAN_SOLDNER_SYN

```
public static final java.lang.String RAMANUJAN_SOLDNER_SYN
```

SATURN_MASS_DESC

```
public static final java.lang.String SATURN_MASS_DESC
```

SATURN_MASS_ID

```
public static final int SATURN_MASS_ID
```

SATURN_MASS_SINCE

```
public static final java.lang.String SATURN_MASS_SINCE
```

SATURN_MASS_STR

```
public static final java.lang.String SATURN_MASS_STR
```

SATURN_MASS_SYN

```
public static final java.lang.String SATURN_MASS_SYN
```

SATURN_RADIUS_MEAN_DESC

```
public static final java.lang.String SATURN_RADIUS_MEAN_DESC
```

SATURN_RADIUS_MEAN_ID

```
public static final int SATURN_RADIUS_MEAN_ID
```

SATURN_RADIUS_MEAN_SINCE

```
public static final java.lang.String SATURN_RADIUS_MEAN_SINCE
```

SATURN_RADIUS_MEAN_STR

```
public static final java.lang.String SATURN_RADIUS_MEAN_STR
```

SATURN_RADIUS_MEAN_SYN

```
public static final java.lang.String SATURN_RADIUS_MEAN_SYN
```

SATURN_SEMI_MAJOR_AXIS_DESC

```
public static final java.lang.String SATURN_SEMI_MAJOR_AXIS_DESC
```

SATURN_SEMI_MAJOR_AXIS_ID

```
public static final int SATURN_SEMI_MAJOR_AXIS_ID
```

SATURN_SEMI_MAJOR_AXIS_SINCE

```
public static final java.lang.String SATURN_SEMI_MAJOR_AXIS_SINCE
```

SATURN_SEMI_MAJOR_AXIS_STR

```
public static final java.lang.String SATURN_SEMI_MAJOR_AXIS_STR
```

SATURN_SEMI_MAJOR_AXIS_SYN

```
public static final java.lang.String SATURN_SEMI_MAJOR_AXIS_SYN
```

SIERPINSKI_DESC

```
public static final java.lang.String SIERPINSKI_DESC
```

SIERPINSKI_ID

```
public static final int SIERPINSKI_ID
```

SIERPINSKI_SINCE

```
public static final java.lang.String SIERPINSKI_SINCE
```

SIERPINSKI_STR

```
public static final java.lang.String SIERPINSKI_STR
```

SIERPINSKI_SYN

```
public static final java.lang.String SIERPINSKI_SYN
```

SOLAR_MASS_DESC

```
public static final java.lang.String SOLAR_MASS_DESC
```

SOLAR_MASS_ID

```
public static final int SOLAR_MASS_ID
```

SOLAR_MASS_SINCE

```
public static final java.lang.String SOLAR_MASS_SINCE
```

SOLAR_MASS_STR

```
public static final java.lang.String SOLAR_MASS_STR
```

SOLAR_MASS_SYN

```
public static final java.lang.String SOLAR_MASS_SYN
```

SOLAR_RADIUS_DESC

```
public static final java.lang.String SOLAR_RADIUS_DESC
```

SOLAR_RADIUS_ID

```
public static final int SOLAR_RADIUS_ID
```

SOLAR_RADIUS_SINCE

```
public static final java.lang.String SOLAR_RADIUS_SINCE
```

SOLAR_RADIUS_STR

```
public static final java.lang.String SOLAR_RADIUS_STR
```

SOLAR_RADIUS_SYN

```
public static final java.lang.String SOLAR_RADIUS_SYN
```

TRUE_DESC

```
public static final java.lang.String TRUE_DESC
```

TRUE_ID

```
public static final int TRUE_ID
```

TRUE_SINCE

```
public static final java.lang.String TRUE_SINCE
```

TRUE_STR

```
public static final java.lang.String TRUE_STR
```

TRUE_SYN

```
public static final java.lang.String TRUE_SYN
```

TWIN_PRIME_DESC

```
public static final java.lang.String TWIN_PRIME_DESC
```

TWIN_PRIME_ID

```
public static final int TWIN_PRIME_ID
```

TWIN_PRIME_SINCE

```
public static final java.lang.String TWIN_PRIME_SINCE
```

TWIN_PRIME_STR


```
public static final java.lang.String TWIN_PRIME_STR
```

TWIN_PRIME_SYN

```
public static final java.lang.String TWIN_PRIME_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

URANUS_MASS_DESC

```
public static final java.lang.String URANUS_MASS_DESC
```

URANUS_MASS_ID

```
public static final int URANUS_MASS_ID
```

URANUS_MASS_SINCE

```
public static final java.lang.String URANUS_MASS_SINCE
```

URANUS_MASS_STR

```
public static final java.lang.String URANUS_MASS_STR
```

URANUS_MASS_SYN

```
public static final java.lang.String URANUS_MASS_SYN
```

URANUS_RADIUS_MEAN_DESC

```
public static final java.lang.String URANUS_RADIUS_MEAN_DESC
```

URANUS_RADIUS_MEAN_ID

```
public static final int URANUS_RADIUS_MEAN_ID
```

URANUS_RADIUS_MEAN_SINCE

```
public static final java.lang.String URANUS_RADIUS_MEAN_SINCE
```

URANUS_RADIUS_MEAN_STR

```
public static final java.lang.String URANUS_RADIUS_MEAN_STR
```

URANUS_RADIUS_MEAN_SYN

```
public static final java.lang.String URANUS_RADIUS_MEAN_SYN
```

URANUS_SEMI_MAJOR_AXIS_DESC

```
public static final java.lang.String URANUS_SEMI_MAJOR_AXIS_DESC
```

URANUS_SEMI_MAJOR_AXIS_ID

```
public static final int URANUS_SEMI_MAJOR_AXIS_ID
```

URANUS_SEMI_MAJOR_AXIS_SINCE

```
public static final java.lang.String URANUS_SEMI_MAJOR_AXIS_SINCE
```

URANUS_SEMI_MAJOR_AXIS_STR

```
public static final java.lang.String URANUS_SEMI_MAJOR_AXIS_STR
```

URANUS_SEMI_MAJOR_AXIS_SYN

```
public static final java.lang.String URANUS_SEMI_MAJOR_AXIS_SYN
```

VENUS_MASS_DESC

```
public static final java.lang.String VENUS_MASS_DESC
```

VENUS_MASS_ID

```
public static final int VENUS_MASS_ID
```

VENUS_MASS_SINCE

```
public static final java.lang.String VENUS_MASS_SINCE
```

VENUS_MASS_STR

```
public static final java.lang.String VENUS_MASS_STR
```

VENUS_MASS_SYN

```
public static final java.lang.String VENUS_MASS_SYN
```

VENUS_RADIUS_MEAN_DESC

```
public static final java.lang.String VENUS_RADIUS_MEAN_DESC
```

VENUS_RADIUS_MEAN_ID

```
public static final int VENUS_RADIUS_MEAN_ID
```

VENUS_RADIUS_MEAN_SINCE

```
public static final java.lang.String VENUS_RADIUS_MEAN_SINCE
```

VENUS_RADIUS_MEAN_STR

```
public static final java.lang.String VENUS_RADIUS_MEAN_STR
```

VENUS_RADIUS_MEAN_SYN

```
public static final java.lang.String VENUS_RADIUS_MEAN_SYN
```

VENUS_SEMI_MAJOR_AXIS_DESC

```
public static final java.lang.String VENUS_SEMI_MAJOR_AXIS_DESC
```

VENUS_SEMI_MAJOR_AXIS_ID

```
public static final int VENUS_SEMI_MAJOR_AXIS_ID
```

VENUS_SEMI_MAJOR_AXIS_SINCE

```
public static final java.lang.String VENUS_SEMI_MAJOR_AXIS_SINCE
```

VENUS_SEMI_MAJOR_AXIS_STR

```
public static final java.lang.String VENUS_SEMI_MAJOR_AXIS_STR
```

VENUS_SEMI_MAJOR_AXIS_SYN

```
public static final java.lang.String VENUS_SEMI_MAJOR_AXIS_SYN
```

VISWANATH_DESC

```
public static final java.lang.String VISWANATH_DESC
```

VISWANATH_ID

```
public static final int VISWANATH_ID
```

VISWANATH_SINCE

```
public static final java.lang.String VISWANATH_SINCE
```

VISWANATH_STR

```
public static final java.lang.String VISWANATH_STR
```

VISWANATH_SYN

```
public static final java.lang.String VISWANATH_SYN
```

Constructors

ConstantValue

```
public ConstantValue()
```

```
org.mariuszgromada.math.mxparser.parsertokens
```

Class Function1Arg

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.parsertokens.Function1Arg
```

< [Fields](#) > < [Constructors](#) >

```
public final class Function1Arg
extends java.lang.Object
```

Unary functions (1 argument) - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

[MathSpace.pl](#)

[MathParser.org - mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

ABS_DESC

```
public static final java.lang.String ABS_DESC
```

ABS_ID

```
public static final int ABS_ID
```

ABS_SINCE

```
public static final java.lang.String ABS_SINCE
```

ABS_STR

```
public static final java.lang.String ABS_STR
```

ABS_SYN

```
public static final java.lang.String ABS_SYN
```

ACOSECH_STR

```
public static final java.lang.String ACOSECH_STR
```

ACOSECH_SYN

```
public static final java.lang.String ACOSECH_SYN
```

ACOSH_STR

```
public static final java.lang.String ACOSH_STR
```

ACOSH_SYN

```
public static final java.lang.String ACOSH_SYN
```

ACOS_DESC

```
public static final java.lang.String ACOS_DESC
```

ACOS_ID

```
public static final int ACOS_ID
```

ACOS_SINCE

```
public static final java.lang.String ACOS_SINCE
```

ACOS_STR

```
public static final java.lang.String ACOS_STR
```

ACOS_SYN

```
public static final java.lang.String ACOS_SYN
```

ACOTH_STR

```
public static final java.lang.String ACOTH_STR
```

ACOTH_SYN

```
public static final java.lang.String ACOTH_SYN
```

ACOT_STR

```
public static final java.lang.String ACOT_STR
```

ACOT_SYN

```
public static final java.lang.String ACOT_SYN
```

ACSCH_STR

```
public static final java.lang.String ACSCH_STR
```

ACSCH_SYN

```
public static final java.lang.String ACSCH_SYN
```

ACTANH_STR

```
public static final java.lang.String ACTANH_STR
```

ACTANH_SYN

```
public static final java.lang.String ACTANH_SYN
```

ACTAN_DESC

```
public static final java.lang.String ACTAN_DESC
```

ACTAN_ID

```
public static final int ACTAN_ID
```

ACTAN_SINCE

```
public static final java.lang.String ACTAN_SINCE
```

ACTAN_STR

```
public static final java.lang.String ACTAN_STR
```

ACTAN_SYN

```
public static final java.lang.String ACTAN_SYN
```

ACTGH_STR

```
public static final java.lang.String ACTGH_STR
```

ACTGH_SYN


```
public static final java.lang.String ACTGH_SYN
```

ACTG_STR

```
public static final java.lang.String ACTG_STR
```

ACTG_SYN

```
public static final java.lang.String ACTG_SYN
```

ARCCOSECH_STR

```
public static final java.lang.String ARCCOSECH_STR
```

ARCCOSECH_SYN

```
public static final java.lang.String ARCCOSECH_SYN
```

ARCCOSH_STR

```
public static final java.lang.String ARCCOSH_STR
```

ARCCOSH_SYN

```
public static final java.lang.String ARCCOSH_SYN
```

ARCCOS_STR

```
public static final java.lang.String ARCCOS_STR
```

ARCCOS_SYN

```
public static final java.lang.String ARCCOS_SYN
```

ARCCOTH_STR

```
public static final java.lang.String ARCCOTH_STR
```

ARCCOTH_SYN

```
public static final java.lang.String ARCCOTH_SYN
```

ARCCOT_STR

```
public static final java.lang.String ARCCOT_STR
```

ARCCOT_SYN

```
public static final java.lang.String ARCCOT_SYN
```

ARCCSCH_STR

```
public static final java.lang.String ARCCSCH_STR
```

ARCCSCH_SYN

```
public static final java.lang.String ARCCSCH_SYN
```

ARCCTANH_STR

```
public static final java.lang.String ARCCTANH_STR
```

ARCCTANH_SYN

```
public static final java.lang.String ARCCTANH_SYN
```

ARCCTAN_STR

```
public static final java.lang.String ARCCTAN_STR
```

ARCCTAN_SYN

```
public static final java.lang.String ARCCTAN_SYN
```

ARCCTGH_STR

```
public static final java.lang.String ARCCTGH_STR
```

ARCCTGH_SYN

```
public static final java.lang.String ARCCTGH_SYN
```

ARCCTG_STR

```
public static final java.lang.String ARCCTG_STR
```

ARCCTG_SYN

```
public static final java.lang.String ARCCTG_SYN
```

ARCOSECH_STR

```
public static final java.lang.String ARCOSECH_STR
```

ARCOSECH_SYN

```
public static final java.lang.String ARCOSECH_SYN
```

ARCOSH_DESC

```
public static final java.lang.String ARCOSH_DESC
```

ARCOSH_ID

```
public static final int ARCOSH_ID
```

ARCOSH_SINCE

```
public static final java.lang.String ARCOSH_SINCE
```

ARCOSH_STR

```
public static final java.lang.String ARCOSH_STR
```

ARCOSH_SYN

```
public static final java.lang.String ARCOSH_SYN
```

ARCOS_STR

```
public static final java.lang.String ARCOS_STR
```

ARCOS_SYN

```
public static final java.lang.String ARCOS_SYN
```

ARCOTH_DESC

```
public static final java.lang.String ARCOTH_DESC
```

ARCOTH_ID

```
public static final int ARCOTH_ID
```

ARCOTH_SINCE

```
public static final java.lang.String ARCOTH_SINCE
```

ARCOTH_STR

```
public static final java.lang.String ARCOTH_STR
```

ARCOTH_SYN

```
public static final java.lang.String ARCOTH_SYN
```

ARCSCH_DESC

```
public static final java.lang.String ARCSCH_DESC
```

ARCSCH_ID

```
public static final int ARCSCH_ID
```

ARCSCH_SINCE

```
public static final java.lang.String ARCSCH_SINCE
```

ARCSCH_STR

```
public static final java.lang.String ARCSCH_STR
```

ARCSCH_SYN

```
public static final java.lang.String ARCSCH_SYN
```

ARCSECH_STR

```
public static final java.lang.String ARCSECH_STR
```

ARCSECH_SYN

```
public static final java.lang.String ARCSECH_SYN
```

ARCSINH_STR

```
public static final java.lang.String ARCSINH_STR
```

ARCSINH_SYN

```
public static final java.lang.String ARCSINH_SYN
```

ARCSIN_STR

```
public static final java.lang.String ARCSIN_STR
```

ARCSIN_SYN

```
public static final java.lang.String ARCSIN_SYN
```

ARCTANH_STR

```
public static final java.lang.String ARCTANH_STR
```

ARCTANH_SYN

```
public static final java.lang.String ARCTANH_SYN
```

ARCTAN_STR

```
public static final java.lang.String ARCTAN_STR
```

ARCTAN_SYN

```
public static final java.lang.String ARCTAN_SYN
```

ARCTGH_STR

```
public static final java.lang.String ARCTGH_STR
```

ARCTGH_SYN

```
public static final java.lang.String ARCTGH_SYN
```

ARCTG_STR

```
public static final java.lang.String ARCTG_STR
```

ARCTG_SYN

```
public static final java.lang.String ARCTG_SYN
```

ARSECH_DESC

```
public static final java.lang.String ARSECH_DESC
```

ARSECH_ID

```
public static final int ARSECH_ID
```

ARSECH_SINCE

```
public static final java.lang.String ARSECH_SINCE
```

ARSECH_STR

```
public static final java.lang.String ARSECH_STR
```

ARSECH_SYN

```
public static final java.lang.String ARSECH_SYN
```

ARSINH_DESC

```
public static final java.lang.String ARSINH_DESC
```

ARSINH_ID

```
public static final int ARSINH_ID
```

ARSINH_SINCE

```
public static final java.lang.String ARSINH_SINCE
```

ARSINH_STR

```
public static final java.lang.String ARSINH_STR
```

ARSINH_SYN

```
public static final java.lang.String ARSINH_SYN
```

ARSIN_STR

```
public static final java.lang.String ARSIN_STR
```

ARSIN_SYN

```
public static final java.lang.String ARSIN_SYN
```

ARTANH_DESC

```
public static final java.lang.String ARTANH_DESC
```

ARTANH_ID

```
public static final int ARTANH_ID
```

ARTANH_SINCE

```
public static final java.lang.String ARTANH_SINCE
```

ASECH_STR

```
public static final java.lang.String ASECH_STR
```

ASECH_SYN

```
public static final java.lang.String ASECH_SYN
```

ASINH_STR

```
public static final java.lang.String ASINH_STR
```

ASINH_SYN

```
public static final java.lang.String ASINH_SYN
```

ASIN_DESC

```
public static final java.lang.String ASIN_DESC
```

ASIN_ID


```
public static final int ASIN_ID
```

ASIN_SINCE

```
public static final java.lang.String ASIN_SINCE
```

ASIN_STR

```
public static final java.lang.String ASIN_STR
```

ASIN_SYN

```
public static final java.lang.String ASIN_SYN
```

ATANH_STR

```
public static final java.lang.String ATANH_STR
```

ATANH_SYN

```
public static final java.lang.String ATANH_SYN
```

ATAN_DESC

```
public static final java.lang.String ATAN_DESC
```

ATAN_ID

```
public static final int ATAN_ID
```

ATAN_SINCE

```
public static final java.lang.String ATAN_SINCE
```

ATAN_STR

```
public static final java.lang.String ATAN_STR
```

ATAN_SYN

```
public static final java.lang.String ATAN_SYN
```

ATGH_STR

```
public static final java.lang.String ATGH_STR
```

ATGH_SYN

```
public static final java.lang.String ATGH_SYN
```

ATG_STR

```
public static final java.lang.String ATG_STR
```

ATG_SYN

```
public static final java.lang.String ATG_SYN
```

BELL_NUMBER_DESC

```
public static final java.lang.String BELL_NUMBER_DESC
```

BELL_NUMBER_ID

```
public static final int BELL_NUMBER_ID
```

BELL_NUMBER_SINCE

```
public static final java.lang.String BELL_NUMBER_SINCE
```

BELL_NUMBER_STR

```
public static final java.lang.String BELL_NUMBER_STR
```

BELL_NUMBER_SYN

```
public static final java.lang.String BELL_NUMBER_SYN
```

CEIL_DESC

```
public static final java.lang.String CEIL_DESC
```

CEIL_ID

```
public static final int CEIL_ID
```

CEIL_SINCE

```
public static final java.lang.String CEIL_SINCE
```

CEIL_STR

```
public static final java.lang.String CEIL_STR
```

CEIL_SYN

```
public static final java.lang.String CEIL_SYN
```

COSECH_STR

```
public static final java.lang.String COSECH_STR
```

COSECH_SYN

```
public static final java.lang.String COSECH_SYN
```

COSEC_DESC

```
public static final java.lang.String COSEC_DESC
```

COSEC_ID

```
public static final int COSEC_ID
```

COSEC_SINCE

```
public static final java.lang.String COSEC_SINCE
```

COSEC_STR

```
public static final java.lang.String COSEC_STR
```

COSEC_SYN

```
public static final java.lang.String COSEC_SYN
```

COSH_DESC

```
public static final java.lang.String COSH_DESC
```

COSH_ID

```
public static final int COSH_ID
```

COSH_SINCE

```
public static final java.lang.String COSH_SINCE
```

COSH_STR

```
public static final java.lang.String COSH_STR
```

COSH_SYN

```
public static final java.lang.String COSH_SYN
```

COS_DESC

```
public static final java.lang.String COS_DESC
```

COS_ID

```
public static final int COS_ID
```

COS_SINCE

```
public static final java.lang.String COS_SINCE
```

COS_STR

```
public static final java.lang.String COS_STR
```

COS_SYN

```
public static final java.lang.String COS_SYN
```

COTH_DESC

```
public static final java.lang.String COTH_DESC
```

COTH_ID

```
public static final int COTH_ID
```

COTH_SINCE

```
public static final java.lang.String COTH_SINCE
```

COTH_STR

```
public static final java.lang.String COTH_STR
```

COTH_SYN

```
public static final java.lang.String COTH_SYN
```

COT_STR

```
public static final java.lang.String COT_STR
```

COT_SYN

```
public static final java.lang.String COT_SYN
```

CSCH_DESC

```
public static final java.lang.String CSCH_DESC
```

CSCH_ID

```
public static final int CSCH_ID
```

CSCH_SINCE

```
public static final java.lang.String CSCH_SINCE
```

CSCH_STR

```
public static final java.lang.String CSCH_STR
```

CSCH_SYN

```
public static final java.lang.String CSCH_SYN
```

CSC_STR

```
public static final java.lang.String CSC_STR
```

CSC_SYN

```
public static final java.lang.String CSC_SYN
```

CTANH_STR

```
public static final java.lang.String CTANH_STR
```

CTANH_SYN

```
public static final java.lang.String CTANH_SYN
```

CTAN_DESC

```
public static final java.lang.String CTAN_DESC
```

CTAN_ID

```
public static final int CTAN_ID
```

CTAN_SINCE

```
public static final java.lang.String CTAN_SINCE
```

CTAN_STR

```
public static final java.lang.String CTAN_STR
```

CTAN_SYN

```
public static final java.lang.String CTAN_SYN
```

CTGH_STR

```
public static final java.lang.String CTGH_STR
```

CTGH_SYN

```
public static final java.lang.String CTGH_SYN
```

CTG_STR

```
public static final java.lang.String CTG_STR
```

CTG_SYN

```
public static final java.lang.String CTG_SYN
```

DEG_DESC

```
public static final java.lang.String DEG_DESC
```

DEG_ID

```
public static final int DEG_ID
```

DEG_SINCE

```
public static final java.lang.String DEG_SINCE
```

DEG_STR

```
public static final java.lang.String DEG_STR
```

DEG_SYN

```
public static final java.lang.String DEG_SYN
```

EXP_DESC

```
public static final java.lang.String EXP_DESC
```

EXP_ID

```
public static final int EXP_ID
```

EXP_INT_DESC

```
public static final java.lang.String EXP_INT_DESC
```

EXP_INT_ID

```
public static final int EXP_INT_ID
```

EXP_INT_SINCE


```
public static final java.lang.String EXP_INT_SINCE
```

EXP_INT_STR

```
public static final java.lang.String EXP_INT_STR
```

EXP_INT_SYN

```
public static final java.lang.String EXP_INT_SYN
```

EXP_SINCE

```
public static final java.lang.String EXP_SINCE
```

EXP_STR

```
public static final java.lang.String EXP_STR
```

EXP_SYN

```
public static final java.lang.String EXP_SYN
```

FIBONACCI_NUMBER_DESC

```
public static final java.lang.String FIBONACCI_NUMBER_DESC
```

FIBONACCI_NUMBER_ID

```
public static final int FIBONACCI_NUMBER_ID
```

FIBONACCI_NUMBER_SINCE

```
public static final java.lang.String FIBONACCI_NUMBER_SINCE
```

FIBONACCI_NUMBER_STR

```
public static final java.lang.String FIBONACCI_NUMBER_STR
```

FIBONACCI_NUMBER_SYN

```
public static final java.lang.String FIBONACCI_NUMBER_SYN
```

FLOOR_DESC

```
public static final java.lang.String FLOOR_DESC
```

FLOOR_ID

```
public static final int FLOOR_ID
```

FLOOR_SINCE

```
public static final java.lang.String FLOOR_SINCE
```

FLOOR_STR

```
public static final java.lang.String FLOOR_STR
```

FLOOR_SYN

```
public static final java.lang.String FLOOR_SYN
```

GAUSS_ERFC_DESC

```
public static final java.lang.String GAUSS_ERFC_DESC
```

GAUSS_ERFC_ID

```
public static final int GAUSS_ERFC_ID
```

GAUSS_ERFC_INV_DESC

```
public static final java.lang.String GAUSS_ERFC_INV_DESC
```

GAUSS_ERFC_INV_ID

```
public static final int GAUSS_ERFC_INV_ID
```

GAUSS_ERFC_INV_SINCE

```
public static final java.lang.String GAUSS_ERFC_INV_SINCE
```

GAUSS_ERFC_INV_STR

```
public static final java.lang.String GAUSS_ERFC_INV_STR
```

GAUSS_ERFC_INV_SYN

```
public static final java.lang.String GAUSS_ERFC_INV_SYN
```

GAUSS_ERFC_SINCE

```
public static final java.lang.String GAUSS_ERFC_SINCE
```

GAUSS_ERFC_STR

```
public static final java.lang.String GAUSS_ERFC_STR
```

GAUSS_ERFC_SYN

```
public static final java.lang.String GAUSS_ERFC_SYN
```

GAUSS_ERF_DESC

```
public static final java.lang.String GAUSS_ERF_DESC
```

GAUSS_ERF_ID

```
public static final int GAUSS_ERF_ID
```

GAUSS_ERF_INV_DESC

```
public static final java.lang.String GAUSS_ERF_INV_DESC
```

GAUSS_ERF_INV_ID

```
public static final int GAUSS_ERF_INV_ID
```

GAUSS_ERF_INV_SINCE

```
public static final java.lang.String GAUSS_ERF_INV_SINCE
```

GAUSS_ERF_INV_STR

```
public static final java.lang.String GAUSS_ERF_INV_STR
```

GAUSS_ERF_INV_SYN

```
public static final java.lang.String GAUSS_ERF_INV_SYN
```

GAUSS_ERF_SINCE

```
public static final java.lang.String GAUSS_ERF_SINCE
```

GAUSS_ERF_STR

```
public static final java.lang.String GAUSS_ERF_STR
```

GAUSS_ERF_SYN

```
public static final java.lang.String GAUSS_ERF_SYN
```

HARMONIC_NUMBER_DESC

```
public static final java.lang.String HARMONIC_NUMBER_DESC
```

HARMONIC_NUMBER_ID

```
public static final int HARMONIC_NUMBER_ID
```

HARMONIC_NUMBER_SINCE

```
public static final java.lang.String HARMONIC_NUMBER_SINCE
```

HARMONIC_NUMBER_STR

```
public static final java.lang.String HARMONIC_NUMBER_STR
```

HARMONIC_NUMBER_SYN

```
public static final java.lang.String HARMONIC_NUMBER_SYN
```

ISNAN_DESC

```
public static final java.lang.String ISNAN_DESC
```

ISNAN_ID

```
public static final int ISNAN_ID
```

ISNAN_SINCE

```
public static final java.lang.String ISNAN_SINCE
```

ISNAN_STR

```
public static final java.lang.String ISNAN_STR
```

ISNAN_SYN

```
public static final java.lang.String ISNAN_SYN
```

IS_PRIME_DESC

```
public static final java.lang.String IS_PRIME_DESC
```

IS_PRIME_ID

```
public static final int IS_PRIME_ID
```

IS_PRIME_SINCE

```
public static final java.lang.String IS_PRIME_SINCE
```

IS_PRIME_STR

```
public static final java.lang.String IS_PRIME_STR
```

IS_PRIME_SYN

```
public static final java.lang.String IS_PRIME_SYN
```

LN_DESC

```
public static final java.lang.String LN_DESC
```

LN_ID

```
public static final int LN_ID
```

LN_SINCE

```
public static final java.lang.String LN_SINCE
```

LN_STR

```
public static final java.lang.String LN_STR
```

LN_SYN

```
public static final java.lang.String LN_SYN
```

LOG10_DESC

```
public static final java.lang.String LOG10_DESC
```

LOG10_ID

```
public static final int LOG10_ID
```

LOG10_SINCE

```
public static final java.lang.String LOG10_SINCE
```

LOG10_STR

```
public static final java.lang.String LOG10_STR
```

LOG10_SYN

```
public static final java.lang.String LOG10_SYN
```

LOG2_DESC

```
public static final java.lang.String LOG2_DESC
```

LOG2_ID

```
public static final int LOG2_ID
```

LOG2_SINCE

```
public static final java.lang.String LOG2_SINCE
```

LOG2_STR

```
public static final java.lang.String LOG2_STR
```

LOG2_SYN

```
public static final java.lang.String LOG2_SYN
```

LOG_INT_DESC

```
public static final java.lang.String LOG_INT_DESC
```

LOG_INT_ID

```
public static final int LOG_INT_ID
```

LOG_INT_SINCE

```
public static final java.lang.String LOG_INT_SINCE
```

LOG_INT_STR

```
public static final java.lang.String LOG_INT_STR
```

LOG_INT_SYN

```
public static final java.lang.String LOG_INT_SYN
```

LUCAS_NUMBER_DESC

```
public static final java.lang.String LUCAS_NUMBER_DESC
```

LUCAS_NUMBER_ID

```
public static final int LUCAS_NUMBER_ID
```

LUCAS_NUMBER_SINCE

```
public static final java.lang.String LUCAS_NUMBER_SINCE
```

LUCAS_NUMBER_STR

```
public static final java.lang.String LUCAS_NUMBER_STR
```

LUCAS_NUMBER_SYN

```
public static final java.lang.String LUCAS_NUMBER_SYN
```

NOT_DESC


```
public static final java.lang.String NOT_DESC
```

NOT_ID

```
public static final int NOT_ID
```

NOT_SINCE

```
public static final java.lang.String NOT_SINCE
```

NOT_STR

```
public static final java.lang.String NOT_STR
```

NOT_SYN

```
public static final java.lang.String NOT_SYN
```

OFF_LOG_INT_DESC

```
public static final java.lang.String OFF_LOG_INT_DESC
```

OFF_LOG_INT_ID

```
public static final int OFF_LOG_INT_ID
```

OFF_LOG_INT_SINCE

```
public static final java.lang.String OFF_LOG_INT_SINCE
```

OFF_LOG_INT_STR

```
public static final java.lang.String OFF_LOG_INT_STR
```

OFF_LOG_INT_SYN

```
public static final java.lang.String OFF_LOG_INT_SYN
```

PRIME_COUNT_DESC

```
public static final java.lang.String PRIME_COUNT_DESC
```

PRIME_COUNT_ID

```
public static final int PRIME_COUNT_ID
```

PRIME_COUNT_SINCE

```
public static final java.lang.String PRIME_COUNT_SINCE
```

PRIME_COUNT_STR

```
public static final java.lang.String PRIME_COUNT_STR
```

PRIME_COUNT_SYN

```
public static final java.lang.String PRIME_COUNT_SYN
```

RAD_DESC

```
public static final java.lang.String RAD_DESC
```

RAD_ID

```
public static final int RAD_ID
```

RAD_SINCE

```
public static final java.lang.String RAD_SINCE
```

RAD_STR

```
public static final java.lang.String RAD_STR
```

RAD_SYN

```
public static final java.lang.String RAD_SYN
```

SA1_STR

```
public static final java.lang.String SA1_STR
```

SA1_SYN

```
public static final java.lang.String SA1_SYN
```

SA_DESC

```
public static final java.lang.String SA_DESC
```

SA_ID

```
public static final int SA_ID
```

SA_SINCE

```
public static final java.lang.String SA_SINCE
```

SA_STR

```
public static final java.lang.String SA_STR
```

SA_SYN

```
public static final java.lang.String SA_SYN
```

SECH_DESC

```
public static final java.lang.String SECH_DESC
```

SECH_ID

```
public static final int SECH_ID
```

SECH_SINCE

```
public static final java.lang.String SECH_SINCE
```

SECH_STR

```
public static final java.lang.String SECH_STR
```

SECH_SYN

```
public static final java.lang.String SECH_SYN
```

SEC_DESC

```
public static final java.lang.String SEC_DESC
```

SEC_ID

```
public static final int SEC_ID
```

SEC_SINCE

```
public static final java.lang.String SEC_SINCE
```

SEC_STR

```
public static final java.lang.String SEC_STR
```

SEC_SYN

```
public static final java.lang.String SEC_SYN
```

SGN_DESC

```
public static final java.lang.String SGN_DESC
```

SGN_ID

```
public static final int SGN_ID
```

SGN_SINCE

```
public static final java.lang.String SGN_SINCE
```

SGN_STR

```
public static final java.lang.String SGN_STR
```

SGN_SYN

```
public static final java.lang.String SGN_SYN
```

SINC_DESC

```
public static final java.lang.String SINC_DESC
```

SINC_ID

```
public static final int SINC_ID
```

SINC_SINCE

```
public static final java.lang.String SINC_SINCE
```

SINC_STR

```
public static final java.lang.String SINC_STR
```

SINC_SYN

```
public static final java.lang.String SINC_SYN
```

SINH_DESC

```
public static final java.lang.String SINH_DESC
```

SINH_ID

```
public static final int SINH_ID
```

SINH_SINCE

```
public static final java.lang.String SINH_SINCE
```

SINH_STR

```
public static final java.lang.String SINH_STR
```

SINH_SYN

```
public static final java.lang.String SINH_SYN
```

SIN_DESC

```
public static final java.lang.String SIN_DESC
```

SIN_ID

```
public static final int SIN_ID
```

SIN_SINCE

```
public static final java.lang.String SIN_SINCE
```

SIN_STR

```
public static final java.lang.String SIN_STR
```

SIN_SYN

```
public static final java.lang.String SIN_SYN
```

SQRT_DESC

```
public static final java.lang.String SQRT_DESC
```

SQRT_ID

```
public static final int SQRT_ID
```

SQRT_SINCE

```
public static final java.lang.String SQRT_SINCE
```

SQRT_STR

```
public static final java.lang.String SQRT_STR
```

SQRT_SYN

```
public static final java.lang.String SQRT_SYN
```

TANH_DESC

```
public static final java.lang.String TANH_DESC
```

TANH_ID

```
public static final int TANH_ID
```

TANH_SINCE

```
public static final java.lang.String TANH_SINCE
```

TANH_STR

```
public static final java.lang.String TANH_STR
```

TANH_SYN

```
public static final java.lang.String TANH_SYN
```

TAN_DESC

```
public static final java.lang.String TAN_DESC
```

TAN_ID

```
public static final int TAN_ID
```

TAN_SINCE

```
public static final java.lang.String TAN_SINCE
```

TAN_STR

```
public static final java.lang.String TAN_STR
```

TAN_SYN

```
public static final java.lang.String TAN_SYN
```

TGH_STR

```
public static final java.lang.String TGH_STR
```

TGH_SYN

```
public static final java.lang.String TGH_SYN
```

TG_STR

```
public static final java.lang.String TG_STR
```

TG_SYN

```
public static final java.lang.String TG_SYN
```

TYPE_DESC


```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

ULP_DESC

```
public static final java.lang.String ULP_DESC
```

ULP_ID

```
public static final int ULP_ID
```

ULP_SINCE

```
public static final java.lang.String ULP_SINCE
```

ULP_STR

```
public static final java.lang.String ULP_STR
```

ULP_SYN

```
public static final java.lang.String ULP_SYN
```

Constructors

Function1Arg

```
public Function1Arg()
```

org.mariuszgromada.math.mxparser.parsertokens

Class Function2Arg

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.parsertokens.Function2Arg
```

< [Fields](#) > < [Constructors](#) >

```
public final class Function2Arg
extends java.lang.Object
```

Binary functions (2 arguments) - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

BERNOULLI_NUMBER_DESC

```
public static final java.lang.String BERNOULLI_NUMBER_DESC
```

BERNOULLI_NUMBER_ID

```
public static final int BERNOULLI_NUMBER_ID
```

BERNOULLI_NUMBER_SINCE

```
public static final java.lang.String BERNOULLI_NUMBER_SINCE
```

BERNOULLI_NUMBER_STR

```
public static final java.lang.String BERNOULLI_NUMBER_STR
```

BERNOULLI_NUMBER_SYN

```
public static final java.lang.String BERNOULLI_NUMBER_SYN
```

BINOM_COEFF_DESC

```
public static final java.lang.String BINOM_COEFF_DESC
```

BINOM_COEFF_ID

```
public static final int BINOM_COEFF_ID
```

BINOM_COEFF_SINCE

```
public static final java.lang.String BINOM_COEFF_SINCE
```

BINOM_COEFF_STR

```
public static final java.lang.String BINOM_COEFF_STR
```

BINOM_COEFF_SYN

```
public static final java.lang.String BINOM_COEFF_SYN
```

EULER_NUMBER_DESC

```
public static final java.lang.String EULER_NUMBER_DESC
```

EULER_NUMBER_ID

```
public static final int EULER_NUMBER_ID
```

EULER_NUMBER_SINCE

```
public static final java.lang.String EULER_NUMBER_SINCE
```

EULER_NUMBER_STR

```
public static final java.lang.String EULER_NUMBER_STR
```

EULER_NUMBER_SYN

```
public static final java.lang.String EULER_NUMBER_SYN
```

EULER_POLYNOMIAL_DESC

```
public static final java.lang.String EULER_POLYNOMIAL_DESC
```

EULER_POLYNOMIAL_ID

```
public static final int EULER_POLYNOMIAL_ID
```

EULER_POLYNOMIAL_SINCE

```
public static final java.lang.String EULER_POLYNOMIAL_SINCE
```

EULER_POLYNOMIAL_STR

```
public static final java.lang.String EULER_POLYNOMIAL_STR
```

EULER_POLYNOMIAL_SYN

```
public static final java.lang.String EULER_POLYNOMIAL_SYN
```

HARMONIC_NUMBER_DESC

```
public static final java.lang.String HARMONIC_NUMBER_DESC
```

HARMONIC_NUMBER_ID

```
public static final int HARMONIC_NUMBER_ID
```

HARMONIC_NUMBER_SINCE

```
public static final java.lang.String HARMONIC_NUMBER_SINCE
```

HARMONIC_NUMBER_STR

```
public static final java.lang.String HARMONIC_NUMBER_STR
```

HARMONIC_NUMBER_SYN

```
public static final java.lang.String HARMONIC_NUMBER_SYN
```

KRONECKER_DELTA_DESC

```
public static final java.lang.String KRONECKER_DELTA_DESC
```

KRONECKER_DELTA_ID

```
public static final int KRONECKER_DELTA_ID
```

KRONECKER_DELTA_SINCE

```
public static final java.lang.String KRONECKER_DELTA_SINCE
```

KRONECKER_DELTA_STR

```
public static final java.lang.String KRONECKER_DELTA_STR
```

KRONECKER_DELTA_SYN

```
public static final java.lang.String KRONECKER_DELTA_SYN
```

LOG_DESC

```
public static final java.lang.String LOG_DESC
```

LOG_ID

```
public static final int LOG_ID
```

LOG_SINCE

```
public static final java.lang.String LOG_SINCE
```

LOG_STR

```
public static final java.lang.String LOG_STR
```

LOG_SYN

```
public static final java.lang.String LOG_SYN
```

MOD_DESC

```
public static final java.lang.String MOD_DESC
```

MOD_ID

```
public static final int MOD_ID
```

MOD_SINCE

```
public static final java.lang.String MOD_SINCE
```

MOD_STR

```
public static final java.lang.String MOD_STR
```

MOD_SYN

```
public static final java.lang.String MOD_SYN
```

RND_NORMAL_DESC

```
public static final java.lang.String RND_NORMAL_DESC
```

RND_NORMAL_ID

```
public static final int RND_NORMAL_ID
```

RND_NORMAL_SINCE

```
public static final java.lang.String RND_NORMAL_SINCE
```

RND_NORMAL_STR

```
public static final java.lang.String RND_NORMAL_STR
```

RND_NORMAL_SYN

```
public static final java.lang.String RND_NORMAL_SYN
```

RND_UNIFORM_CONT_DESC

```
public static final java.lang.String RND_UNIFORM_CONT_DESC
```

RND_UNIFORM_CONT_ID

```
public static final int RND_UNIFORM_CONT_ID
```

RND_UNIFORM_CONT_SINCE

```
public static final java.lang.String RND_UNIFORM_CONT_SINCE
```

RND_UNIFORM_CONT_STR

```
public static final java.lang.String RND_UNIFORM_CONT_STR
```

RND_UNIFORM_CONT_SYN

```
public static final java.lang.String RND_UNIFORM_CONT_SYN
```

RND_UNIFORM_DISCR_DESC

```
public static final java.lang.String RND_UNIFORM_DISCR_DESC
```

RND_UNIFORM_DISCR_ID

```
public static final int RND_UNIFORM_DISCR_ID
```

RND_UNIFORM_DISCR_SINCE

```
public static final java.lang.String RND_UNIFORM_DISCR_SINCE
```

RND_UNIFORM_DISCR_STR

```
public static final java.lang.String RND_UNIFORM_DISCR_STR
```

RND_UNIFORM_DISCR_SYN

```
public static final java.lang.String RND_UNIFORM_DISCR_SYN
```

ROUND_DESC

```
public static final java.lang.String ROUND_DESC
```

ROUND_ID

```
public static final int ROUND_ID
```

ROUND_SINCE

```
public static final java.lang.String ROUND_SINCE
```

ROUND_STR

```
public static final java.lang.String ROUND_STR
```

ROUND_SYN

```
public static final java.lang.String ROUND_SYN
```

STIRLING1_NUMBER_DESC

```
public static final java.lang.String STIRLING1_NUMBER_DESC
```

STIRLING1_NUMBER_ID

```
public static final int STIRLING1_NUMBER_ID
```

STIRLING1_NUMBER_SINCE

```
public static final java.lang.String STIRLING1_NUMBER_SINCE
```

STIRLING1_NUMBER_STR

```
public static final java.lang.String STIRLING1_NUMBER_STR
```

STIRLING1_NUMBER_SYN

```
public static final java.lang.String STIRLING1_NUMBER_SYN
```

STIRLING2_NUMBER_DESC

```
public static final java.lang.String STIRLING2_NUMBER_DESC
```

STIRLING2_NUMBER_ID

```
public static final int STIRLING2_NUMBER_ID
```

STIRLING2_NUMBER_SINCE

```
public static final java.lang.String STIRLING2_NUMBER_SINCE
```

STIRLING2_NUMBER_STR

```
public static final java.lang.String STIRLING2_NUMBER_STR
```

STIRLING2_NUMBER_SYN

```
public static final java.lang.String STIRLING2_NUMBER_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

WORPITZKY_NUMBER_DESC

```
public static final java.lang.String WORPITZKY_NUMBER_DESC
```

WORPITZKY_NUMBER_ID

```
public static final int WORPITZKY_NUMBER_ID
```

WORPITZKY_NUMBER_SINCE

```
public static final java.lang.String WORPITZKY_NUMBER_SINCE
```

WORPITZKY_NUMBER_STR

```
public static final java.lang.String WORPITZKY_NUMBER_STR
```

WORPITZKY_NUMBER_SYN

```
public static final java.lang.String WORPITZKY_NUMBER_SYN
```

Constructors

Function2Arg

```
public Function2Arg()
```

org.mariuszgromada.math.mxparser.parsertokens

Class Function3Arg

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.parsertokens.Function3Arg
```

< [Fields](#) > < [Constructors](#) >

```
public final class Function3Arg
extends java.lang.Object
```

Functions with 3 arguments - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

CDF_NORMAL_DESC

```
public static final java.lang.String CDF_NORMAL_DESC
```

CDF_NORMAL_ID

```
public static final int CDF_NORMAL_ID
```

CDF_NORMAL_SINCE

```
public static final java.lang.String CDF_NORMAL_SINCE
```

CDF_NORMAL_STR

```
public static final java.lang.String CDF_NORMAL_STR
```

CDF_NORMAL_SYN

```
public static final java.lang.String CDF_NORMAL_SYN
```

CDF_UNIFORM_CONT_DESC

```
public static final java.lang.String CDF_UNIFORM_CONT_DESC
```

CDF_UNIFORM_CONT_ID

```
public static final int CDF_UNIFORM_CONT_ID
```

CDF_UNIFORM_CONT_SINCE

```
public static final java.lang.String CDF_UNIFORM_CONT_SINCE
```

CDF_UNIFORM_CONT_STR

```
public static final java.lang.String CDF_UNIFORM_CONT_STR
```

CDF_UNIFORM_CONT_SYN

```
public static final java.lang.String CDF_UNIFORM_CONT_SYN
```

CHI_DESC

```
public static final java.lang.String CHI_DESC
```

CHI_ID

```
public static final int CHI_ID
```

CHI_LR_DESC

```
public static final java.lang.String CHI_LR_DESC
```

CHI_LR_ID

```
public static final int CHI_LR_ID
```

CHI_LR_SINCE

```
public static final java.lang.String CHI_LR_SINCE
```

CHI_LR_STR

```
public static final java.lang.String CHI_LR_STR
```

CHI_LR_SYN

```
public static final java.lang.String CHI_LR_SYN
```

CHI_L_DESC

```
public static final java.lang.String CHI_L_DESC
```

CHI_L_ID

```
public static final int CHI_L_ID
```

CHI_L_SINCE

```
public static final java.lang.String CHI_L_SINCE
```

CHI_L_STR

```
public static final java.lang.String CHI_L_STR
```

CHI_L_SYN

```
public static final java.lang.String CHI_L_SYN
```

CHI_R_DESC

```
public static final java.lang.String CHI_R_DESC
```

CHI_R_ID

```
public static final int CHI_R_ID
```

CHI_R_SINCE

```
public static final java.lang.String CHI_R_SINCE
```

CHI_R_STR

```
public static final java.lang.String CHI_R_STR
```

CHI_R_SYN

```
public static final java.lang.String CHI_R_SYN
```

CHI_SINCE

```
public static final java.lang.String CHI_SINCE
```

CHI_STR

```
public static final java.lang.String CHI_STR
```

CHI_SYN

```
public static final java.lang.String CHI_SYN
```

IF_CONDITION_ID

```
public static final int IF_CONDITION_ID
```

IF_DESC

```
public static final java.lang.String IF_DESC
```

IF_ID

```
public static final int IF_ID
```

IF_SINCE

```
public static final java.lang.String IF_SINCE
```

IF_STR

```
public static final java.lang.String IF_STR
```

IF_SYN

```
public static final java.lang.String IF_SYN
```

PDF_NORMAL_DESC

```
public static final java.lang.String PDF_NORMAL_DESC
```

PDF_NORMAL_ID

```
public static final int PDF_NORMAL_ID
```

PDF_NORMAL_SINCE

```
public static final java.lang.String PDF_NORMAL_SINCE
```

PDF_NORMAL_STR

```
public static final java.lang.String PDF_NORMAL_STR
```

PDF_NORMAL_SYN

```
public static final java.lang.String PDF_NORMAL_SYN
```

PDF_UNIFORM_CONT_DESC

```
public static final java.lang.String PDF_UNIFORM_CONT_DESC
```

PDF_UNIFORM_CONT_ID

```
public static final int PDF_UNIFORM_CONT_ID
```

PDF_UNIFORM_CONT_SINCE

```
public static final java.lang.String PDF_UNIFORM_CONT_SINCE
```

PDF_UNIFORM_CONT_STR

```
public static final java.lang.String PDF_UNIFORM_CONT_STR
```

PDF_UNIFORM_CONT_SYN

```
public static final java.lang.String PDF_UNIFORM_CONT_SYN
```

QNT_NORMAL_DESC

```
public static final java.lang.String QNT_NORMAL_DESC
```

QNT_NORMAL_ID

```
public static final int QNT_NORMAL_ID
```

QNT_NORMAL_SINCE


```
public static final java.lang.String QNT_NORMAL_SINCE
```

QNT_NORMAL_STR

```
public static final java.lang.String QNT_NORMAL_STR
```

QNT_NORMAL_SYN

```
public static final java.lang.String QNT_NORMAL_SYN
```

QNT_UNIFORM_CONT_DESC

```
public static final java.lang.String QNT_UNIFORM_CONT_DESC
```

QNT_UNIFORM_CONT_ID

```
public static final int QNT_UNIFORM_CONT_ID
```

QNT_UNIFORM_CONT_SINCE

```
public static final java.lang.String QNT_UNIFORM_CONT_SINCE
```

QNT_UNIFORM_CONT_STR

```
public static final java.lang.String QNT_UNIFORM_CONT_STR
```

QNT_UNIFORM_CONT_SYN

```
public static final java.lang.String QNT_UNIFORM_CONT_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

Constructors

Function3Arg

```
public Function3Arg()
```

org.mariuszgromada.math.mxparser.parsertokens

Class FunctionVariadic

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.parsertokens.FunctionVariadic
```

< [Fields](#) > < [Constructors](#) >

```
public final class FunctionVariadic  
extends java.lang.Object
```

Variadic functions (n parameters)- mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

[MathSpace.pl](#)

[MathParser.org - mXparser project page](#)

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

AND_DESC

```
public static final java.lang.String AND_DESC
```

AND_ID

```
public static final int AND_ID
```

AND_SINCE

```
public static final java.lang.String AND_SINCE
```

AND_STR

```
public static final java.lang.String AND_STR
```

AND_SYN

```
public static final java.lang.String AND_SYN
```

ARGMAX_DESC

```
public static final java.lang.String ARGMAX_DESC
```

ARGMAX_ID

```
public static final int ARGMAX_ID
```

ARGMAX_SINCE

```
public static final java.lang.String ARGMAX_SINCE
```

ARGMAX_STR

```
public static final java.lang.String ARGMAX_STR
```

ARGMAX_SYN

```
public static final java.lang.String ARGMAX_SYN
```

ARGMIN_DESC

```
public static final java.lang.String ARGMIN_DESC
```

ARGMIN_ID

```
public static final int ARGMIN_ID
```

ARGMIN_SINCE

```
public static final java.lang.String ARGMIN_SINCE
```

ARGMIN_STR

```
public static final java.lang.String ARGMIN_STR
```

ARGMIN_SYN

```
public static final java.lang.String ARGMIN_SYN
```

AVG_DESC

```
public static final java.lang.String AVG_DESC
```

AVG_ID

```
public static final int AVG_ID
```

AVG_SINCE

```
public static final java.lang.String AVG_SINCE
```

AVG_STR

```
public static final java.lang.String AVG_STR
```

AVG_SYN

```
public static final java.lang.String AVG_SYN
```

COALESCE_DESC

```
public static final java.lang.String COALESCE_DESC
```

COALESCE_ID

```
public static final int COALESCE_ID
```

COALESCE_SINCE

```
public static final java.lang.String COALESCE_SINCE
```

COALESCE_STR

```
public static final java.lang.String COALESCE_STR
```

COALESCE_SYN

```
public static final java.lang.String COALESCE_SYN
```

CONT_FRAC_DESC

```
public static final java.lang.String CONT_FRAC_DESC
```

CONT_FRAC_ID

```
public static final int CONT_FRAC_ID
```

CONT_FRAC_SINCE

```
public static final java.lang.String CONT_FRAC_SINCE
```

CONT_FRAC_STR

```
public static final java.lang.String CONT_FRAC_STR
```

CONT_FRAC_SYN

```
public static final java.lang.String CONT_FRAC_SYN
```

CONT_POL_DESC

```
public static final java.lang.String CONT_POL_DESC
```

CONT_POL_ID

```
public static final int CONT_POL_ID
```

CONT_POL_SINCE

```
public static final java.lang.String CONT_POL_SINCE
```

CONT_POL_STR

```
public static final java.lang.String CONT_POL_STR
```

CONT_POL_SYN

```
public static final java.lang.String CONT_POL_SYN
```

GCD_DESC

```
public static final java.lang.String GCD_DESC
```

GCD_ID

```
public static final int GCD_ID
```

GCD_SINCE

```
public static final java.lang.String GCD_SINCE
```

GCD_STR

```
public static final java.lang.String GCD_STR
```

GCD_SYN

```
public static final java.lang.String GCD_SYN
```

IFF_DESC

```
public static final java.lang.String IFF_DESC
```

IFF_ID

```
public static final int IFF_ID
```

IFF_SINCE

```
public static final java.lang.String IFF_SINCE
```

IFF_STR

```
public static final java.lang.String IFF_STR
```

IFF_SYN

```
public static final java.lang.String IFF_SYN
```

LCM_DESC

```
public static final java.lang.String LCM_DESC
```

LCM_ID

```
public static final int LCM_ID
```

LCM_SINCE

```
public static final java.lang.String LCM_SINCE
```

LCM_STR

```
public static final java.lang.String LCM_STR
```

LCM_SYN

```
public static final java.lang.String LCM_SYN
```

MAX_DESC

```
public static final java.lang.String MAX_DESC
```

MAX_ID

```
public static final int MAX_ID
```

MAX_SINCE

```
public static final java.lang.String MAX_SINCE
```

MAX_STR

```
public static final java.lang.String MAX_STR
```

MAX_SYN

```
public static final java.lang.String MAX_SYN
```

MEDIAN_DESC

```
public static final java.lang.String MEDIAN_DESC
```

MEDIAN_ID

```
public static final int MEDIAN_ID
```

MEDIAN_SINCE


```
public static final java.lang.String MEDIAN_SINCE
```

MEDIAN_STR

```
public static final java.lang.String MEDIAN_STR
```

MEDIAN_SYN

```
public static final java.lang.String MEDIAN_SYN
```

MIN_DESC

```
public static final java.lang.String MIN_DESC
```

MIN_ID

```
public static final int MIN_ID
```

MIN_SINCE

```
public static final java.lang.String MIN_SINCE
```

MIN_STR

```
public static final java.lang.String MIN_STR
```

MIN_SYN

```
public static final java.lang.String MIN_SYN
```

OR_DESC

```
public static final java.lang.String OR_DESC
```

OR_ID

```
public static final int OR_ID
```

OR_SINCE

```
public static final java.lang.String OR_SINCE
```

OR_STR

```
public static final java.lang.String OR_STR
```

OR_SYN

```
public static final java.lang.String OR_SYN
```

PROD_DESC

```
public static final java.lang.String PROD_DESC
```

PROD_ID

```
public static final int PROD_ID
```

PROD_SINCE

```
public static final java.lang.String PROD_SINCE
```

PROD_STR

```
public static final java.lang.String PROD_STR
```

PROD_SYN

```
public static final java.lang.String PROD_SYN
```

RND_LIST_DESC

```
public static final java.lang.String RND_LIST_DESC
```

RND_LIST_ID

```
public static final int RND_LIST_ID
```

RND_LIST_SINCE

```
public static final java.lang.String RND_LIST_SINCE
```

RND_LIST_STR

```
public static final java.lang.String RND_LIST_STR
```

RND_LIST_SYN

```
public static final java.lang.String RND_LIST_SYN
```

STD_DESC

```
public static final java.lang.String STD_DESC
```

STD_ID

```
public static final int STD_ID
```

STD_SINCE

```
public static final java.lang.String STD_SINCE
```

STD_STR

```
public static final java.lang.String STD_STR
```

STD_SYN

```
public static final java.lang.String STD_SYN
```

SUM_DESC

```
public static final java.lang.String SUM_DESC
```

SUM_ID

```
public static final int SUM_ID
```

SUM_SINCE

```
public static final java.lang.String SUM_SINCE
```

SUM_STR

```
public static final java.lang.String SUM_STR
```

SUM_SYN

```
public static final java.lang.String SUM_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

VAR_DESC

```
public static final java.lang.String VAR_DESC
```

VAR_ID

```
public static final int VAR_ID
```

VAR_SINCE

```
public static final java.lang.String VAR_SINCE
```

VAR_STR

```
public static final java.lang.String VAR_STR
```

VAR_SYN

```
public static final java.lang.String VAR_SYN
```

XOR_DESC

```
public static final java.lang.String XOR_DESC
```

XOR_ID

```
public static final int XOR_ID
```

XOR_SINCE

```
public static final java.lang.String XOR_SINCE
```

XOR_STR

```
public static final java.lang.String XOR_STR
```

XOR_SYN

```
public static final java.lang.String XOR_SYN
```

Constructors

FunctionVariadic

```
public FunctionVariadic()
```

org.mariuszgromada.math.mxparser.parsertokens

Class Keyword

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.parsertokens.Keyword
```

< [Fields](#) > < [Constructors](#) >

```
public class Keyword
extends java.lang.Object
```

Class representing key words known to the parser

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

description

```
public java.lang.String description
```

since

```
public java.lang.String since
```

syntax

```
public java.lang.String syntax
```

wordId

```
public int wordId
```

wordString

```
public java.lang.String wordString
```

wordTypeId

```
public int wordTypeId
```

Constructors

Keyword

```
public Keyword()
```

Keyword

```
public Keyword(java.lang.String wordString,  
               java.lang.String description,  
               int wordId,  
               java.lang.String syntax,  
               java.lang.String since,  
               int wordTypeId)
```

Constructor - creates key words form wordString wordId wordTypeId syntax since

Parameters:

- wordString - the word string (refers to below interfaces)
- wordId - the word identifier (refers to below interfaces)
- wordTypeId - the word type (refers to below interfaces)
- description - the word description
- syntax - the word syntax
- since - the word version since

org.mariuszgromada.math.mxparser.parsertokens

Class Operator

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.parsertokens.Operator
```

< [Fields](#) > < [Constructors](#) >

```
public final class Operator
extends java.lang.Object
```

Operators - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

DIVIDE_DESC

```
public static final java.lang.String DIVIDE_DESC
```

DIVIDE_ID

```
public static final int DIVIDE_ID
```

DIVIDE_SINCE

```
public static final java.lang.String DIVIDE_SINCE
```

DIVIDE_STR

```
public static final java.lang.String DIVIDE_STR
```

DIVIDE_SYN

```
public static final java.lang.String DIVIDE_SYN
```

FACT_DESC

```
public static final java.lang.String FACT_DESC
```

FACT_ID

```
public static final int FACT_ID
```

FACT_SINCE

```
public static final java.lang.String FACT_SINCE
```

FACT_STR

```
public static final java.lang.String FACT_STR
```

FACT_SYN

```
public static final java.lang.String FACT_SYN
```

MINUS_DESC

```
public static final java.lang.String MINUS_DESC
```

MINUS_ID

```
public static final int MINUS_ID
```

MINUS_SINCE

```
public static final java.lang.String MINUS_SINCE
```

MINUS_STR

```
public static final java.lang.String MINUS_STR
```

MINUS_SYN

```
public static final java.lang.String MINUS_SYN
```

MOD_DESC

```
public static final java.lang.String MOD_DESC
```

MOD_ID

```
public static final int MOD_ID
```

MOD_SINCE

```
public static final java.lang.String MOD_SINCE
```

MOD_STR

```
public static final java.lang.String MOD_STR
```

MOD_SYN

```
public static final java.lang.String MOD_SYN
```

MULTIPLY_DESC

```
public static final java.lang.String MULTIPLY_DESC
```

MULTIPLY_ID

```
public static final int MULTIPLY_ID
```

MULTIPLY_SINCE

```
public static final java.lang.String MULTIPLY_SINCE
```

MULTIPLY_STR

```
public static final java.lang.String MULTIPLY_STR
```

MULTIPLY_SYN

```
public static final java.lang.String MULTIPLY_SYN
```

PERC_DESC

```
public static final java.lang.String PERC_DESC
```

PERC_ID

```
public static final int PERC_ID
```

PERC_SINCE

```
public static final java.lang.String PERC_SINCE
```

PERC_STR

```
public static final java.lang.String PERC_STR
```

PERC_SYN

```
public static final java.lang.String PERC_SYN
```

PLUS_DESC

```
public static final java.lang.String PLUS_DESC
```

PLUS_ID

```
public static final int PLUS_ID
```

PLUS_SINCE

```
public static final java.lang.String PLUS_SINCE
```

PLUS_STR

```
public static final java.lang.String PLUS_STR
```

PLUS_SYN

```
public static final java.lang.String PLUS_SYN
```

POWER_DESC

```
public static final java.lang.String POWER_DESC
```

POWER_ID

```
public static final int POWER_ID
```

POWER_SINCE

```
public static final java.lang.String POWER_SINCE
```

POWER_STR

```
public static final java.lang.String POWER_STR
```

POWER_SYN

```
public static final java.lang.String POWER_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

Constructors

Operator

```
public Operator()
```

org.mariuszgromada.math.mxparser.parsertokens

Class ParserSymbol

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.parsertokens.ParserSymbol
```

< [Fields](#) > < [Constructors](#) >

```
public final class ParserSymbol
extends java.lang.Object
```

Parser symbols - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](https://bitbucket.org/mariuszgromada/mxparser/)

[mXparser on CodePlex](https://codeplex.com/mxparser/)

[Janet Sudoku - project web page](http://janet.sudoku.org/)

[Janet Sudoku on GitHub](https://github.com/mariuszgromada/janet-sudoku)

[Janet Sudoku on CodePlex](https://codeplex.com/janet-sudoku/)

[Janet Sudoku on SourceForge](https://sourceforge.net/projects/janet-sudoku/)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

COMMA_DESC

```
public static final java.lang.String COMMA_DESC
```

COMMA_ID

```
public static final int COMMA_ID
```

COMMA_SINCE

```
public static final java.lang.String COMMA_SINCE
```

COMMA_STR

```
public static final java.lang.String COMMA_STR
```

COMMA_SYN

```
public static final java.lang.String COMMA_SYN
```

DIGIT

```
public static final java.lang.String DIGIT
```

INTEGER

```
public static final java.lang.String INTEGER
```

LEFT_PARENTHESSES_DESC

```
public static final java.lang.String LEFT_PARENTHESSES_DESC
```

LEFT_PARENTHESSES_ID

```
public static final int LEFT_PARENTHESSES_ID
```

LEFT_PARENTHESSES_SINCE

```
public static final java.lang.String LEFT_PARENTHESSES_SINCE
```

LEFT_PARENTHESSES_STR

```
public static final java.lang.String LEFT_PARENTHESSES_STR
```

LEFT_PARENTHESSES_SYN

```
public static final java.lang.String LEFT_PARENTHESSES_SYN
```

NUMBER

```
public static final java.lang.String NUMBER
```

NUMBER_CONST

```
public static final java.lang.String NUMBER_CONST
```

NUMBER_DESC

```
public static final java.lang.String NUMBER_DESC
```

NUMBER_ID

```
public static final int NUMBER_ID
```

NUMBER_REG_DESC

```
public static final java.lang.String NUMBER_REG_DESC
```

NUMBER_REG_EXP

```
public static final java.lang.String NUMBER_REG_EXP
```

NUMBER_SINCE

```
public static final java.lang.String NUMBER_SINCE
```

NUMBER_STR

```
public static final java.lang.String NUMBER_STR
```

NUMBER_SYN

```
public static final java.lang.String NUMBER_SYN
```

NUMBER_TYPE_ID

```
public static final int NUMBER_TYPE_ID
```

REAL

```
public static final java.lang.String REAL
```

RIGHT_PARENTHESSES_DESC

```
public static final java.lang.String RIGHT_PARENTHESSES_DESC
```

RIGHT_PARENTHESSES_ID

```
public static final int RIGHT_PARENTHESSES_ID
```

RIGHT_PARENTHESSES_SINCE

```
public static final java.lang.String RIGHT_PARENTHESSES_SINCE
```

RIGHT_PARENTHESSES_STR

```
public static final java.lang.String RIGHT_PARENTHESSES_STR
```

RIGHT_PARENTHESSES_SYN

```
public static final java.lang.String RIGHT_PARENTHESSES_SYN
```

SEMI_DESC

```
public static final java.lang.String SEMI_DESC
```

SEMI_SINCE

```
public static final java.lang.String SEMI_SINCE
```

SEMI_STR

```
public static final java.lang.String SEMI_STR
```

SEMI_SYN

```
public static final java.lang.String SEMI_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

constArgDefStrRegExp

```
public static final java.lang.String constArgDefStrRegExp
```

function1ArgDefStrRegExp

```
public static final java.lang.String function1ArgDefStrRegExp
```

functionDefStrRegExp

```
public static final java.lang.String functionDefStrRegExp
```

nameOnlyTokenRegExp

```
public static final java.lang.String nameOnlyTokenRegExp
```

nameTokenRegExp

```
public static final java.lang.String nameTokenRegExp
```

paramsTokenRegeExp

```
public static final java.lang.String paramsTokenRegeExp
```

Constructors

ParserSymbol

```
public ParserSymbol()
```

```
org.mariuszgromada.math.mxparser.parsertokens
```

Class RandomVariable

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.parsertokens.RandomVariable
```

< [Fields](#) > < [Constructors](#) >

```
public final class RandomVariable  
extends java.lang.Object
```

Random variables - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Fields

INT1_DESC

```
public static final java.lang.String INT1_DESC
```

INT1_ID

```
public static final int INT1_ID
```

INT1_SINCE

```
public static final java.lang.String INT1_SINCE
```

INT1_STR

```
public static final java.lang.String INT1_STR
```

INT1_SYN

```
public static final java.lang.String INT1_SYN
```

INT2_DESC

```
public static final java.lang.String INT2_DESC
```

INT2_ID

```
public static final int INT2_ID
```

INT2_SINCE

```
public static final java.lang.String INT2_SINCE
```

INT2_STR

```
public static final java.lang.String INT2_STR
```

INT2_SYN

```
public static final java.lang.String INT2_SYN
```

INT3_DESC

```
public static final java.lang.String INT3_DESC
```

INT3_ID

```
public static final int INT3_ID
```

INT3_SINCE

```
public static final java.lang.String INT3_SINCE
```

INT3_STR

```
public static final java.lang.String INT3_STR
```

INT3_SYN

```
public static final java.lang.String INT3_SYN
```

INT4_DESC

```
public static final java.lang.String INT4_DESC
```

INT4_ID

```
public static final int INT4_ID
```

INT4_SINCE

```
public static final java.lang.String INT4_SINCE
```

INT4_STR

```
public static final java.lang.String INT4_STR
```

INT4_SYN

```
public static final java.lang.String INT4_SYN
```

INT5_DESC

```
public static final java.lang.String INT5_DESC
```

INT5_ID

```
public static final int INT5_ID
```

INT5_SINCE

```
public static final java.lang.String INT5_SINCE
```

INT5_STR

```
public static final java.lang.String INT5_STR
```

INT5_SYN

```
public static final java.lang.String INT5_SYN
```

INT6_DESC

```
public static final java.lang.String INT6_DESC
```

INT6_ID

```
public static final int INT6_ID
```

INT6_SINCE

```
public static final java.lang.String INT6_SINCE
```

INT6_STR

```
public static final java.lang.String INT6_STR
```

INT6_SYN

```
public static final java.lang.String INT6_SYN
```

INT7_DESC

```
public static final java.lang.String INT7_DESC
```

INT7_ID

```
public static final int INT7_ID
```

INT7_SINCE

```
public static final java.lang.String INT7_SINCE
```

INT7_STR

```
public static final java.lang.String INT7_STR
```

INT7_SYN

```
public static final java.lang.String INT7_SYN
```

INT8_DESC

```
public static final java.lang.String INT8_DESC
```

INT8_ID

```
public static final int INT8_ID
```

INT8_SINCE

```
public static final java.lang.String INT8_SINCE
```

INT8_STR

```
public static final java.lang.String INT8_STR
```

INT8_SYN

```
public static final java.lang.String INT8_SYN
```

INT9_DESC

```
public static final java.lang.String INT9_DESC
```

INT9_ID

```
public static final int INT9_ID
```

INT9_SINCE

```
public static final java.lang.String INT9_SINCE
```

INT9_STR

```
public static final java.lang.String INT9_STR
```

INT9_SYN

```
public static final java.lang.String INT9_SYN
```

INT_DESC

```
public static final java.lang.String INT_DESC
```

INT_ID

```
public static final int INT_ID
```

INT_SINCE

```
public static final java.lang.String INT_SINCE
```

INT_STR

```
public static final java.lang.String INT_STR
```

INT_SYN

```
public static final java.lang.String INT_SYN
```

NAT0_1_DESC

```
public static final java.lang.String NAT0_1_DESC
```

NAT0_1_ID

```
public static final int NAT0_1_ID
```

NAT0_1_SINCE

```
public static final java.lang.String NAT0_1_SINCE
```

NAT0_1_STR

```
public static final java.lang.String NAT0_1_STR
```

NAT0_1_SYN

```
public static final java.lang.String NAT0_1_SYN
```

NAT0_2_DESC

```
public static final java.lang.String NAT0_2_DESC
```

NAT0_2_ID

```
public static final int NAT0_2_ID
```

NAT0_2_SINCE

```
public static final java.lang.String NAT0_2_SINCE
```

NAT0_2_STR

```
public static final java.lang.String NAT0_2_STR
```

NAT0_2_SYN

```
public static final java.lang.String NAT0_2_SYN
```

NAT0_3_DESC

```
public static final java.lang.String NAT0_3_DESC
```

NAT0_3_ID

```
public static final int NAT0_3_ID
```

NAT0_3_SINCE

```
public static final java.lang.String NAT0_3_SINCE
```

NAT0_3_STR

```
public static final java.lang.String NAT0_3_STR
```

NAT0_3_SYN

```
public static final java.lang.String NAT0_3_SYN
```

NAT0_4_DESC

```
public static final java.lang.String NAT0_4_DESC
```

NAT0_4_ID

```
public static final int NAT0_4_ID
```

NAT0_4_SINCE

```
public static final java.lang.String NAT0_4_SINCE
```

NAT0_4_STR

```
public static final java.lang.String NAT0_4_STR
```

NAT0_4_SYN

```
public static final java.lang.String NAT0_4_SYN
```

NAT0_5_DESC

```
public static final java.lang.String NAT0_5_DESC
```

NAT0_5_ID

```
public static final int NAT0_5_ID
```

NAT0_5_SINCE

```
public static final java.lang.String NAT0_5_SINCE
```

NAT0_5_STR

```
public static final java.lang.String NAT0_5_STR
```

NAT0_5_SYN

```
public static final java.lang.String NAT0_5_SYN
```

NAT0_6_DESC

```
public static final java.lang.String NAT0_6_DESC
```

NAT0_6_ID

```
public static final int NAT0_6_ID
```

NAT0_6_SINCE

```
public static final java.lang.String NAT0_6_SINCE
```

NAT0_6_STR

```
public static final java.lang.String NAT0_6_STR
```

NAT0_6_SYN

```
public static final java.lang.String NAT0_6_SYN
```

NAT0_7_DESC

```
public static final java.lang.String NAT0_7_DESC
```

NAT0_7_ID

```
public static final int NAT0_7_ID
```

NAT0_7_SINCE

```
public static final java.lang.String NAT0_7_SINCE
```

NAT0_7_STR

```
public static final java.lang.String NAT0_7_STR
```

NAT0_7_SYN

```
public static final java.lang.String NAT0_7_SYN
```

NAT0_8_DESC

```
public static final java.lang.String NAT0_8_DESC
```

NAT0_8_ID

```
public static final int NAT0_8_ID
```

NAT0_8_SINCE

```
public static final java.lang.String NAT0_8_SINCE
```

NAT0_8_STR

```
public static final java.lang.String NAT0_8_STR
```

NAT0_8_SYN

```
public static final java.lang.String NAT0_8_SYN
```

NAT0_9_DESC

```
public static final java.lang.String NAT0_9_DESC
```

NAT0_9_ID

```
public static final int NAT0_9_ID
```

NAT0_9_SINCE

```
public static final java.lang.String NAT0_9_SINCE
```

NAT0_9_STR

```
public static final java.lang.String NAT0_9_STR
```

NAT0_9_SYN

```
public static final java.lang.String NAT0_9_SYN
```

NAT0_DESC

```
public static final java.lang.String NAT0_DESC
```

NAT0_ID

```
public static final int NAT0_ID
```

NAT0_SINCE

```
public static final java.lang.String NAT0_SINCE
```

NAT0_STR

```
public static final java.lang.String NAT0_STR
```

NAT0_SYN

```
public static final java.lang.String NAT0_SYN
```

NAT1_1_DESC

```
public static final java.lang.String NAT1_1_DESC
```

NAT1_1_ID

```
public static final int NAT1_1_ID
```

NAT1_1_SINCE

```
public static final java.lang.String NAT1_1_SINCE
```

NAT1_1_STR

```
public static final java.lang.String NAT1_1_STR
```

NAT1_1_SYN

```
public static final java.lang.String NAT1_1_SYN
```

NAT1_2_DESC

```
public static final java.lang.String NAT1_2_DESC
```

NAT1_2_ID

```
public static final int NAT1_2_ID
```

NAT1_2_SINCE

```
public static final java.lang.String NAT1_2_SINCE
```

NAT1_2_STR

```
public static final java.lang.String NAT1_2_STR
```

NAT1_2_SYN

```
public static final java.lang.String NAT1_2_SYN
```

NAT1_3_DESC

```
public static final java.lang.String NAT1_3_DESC
```

NAT1_3_ID

```
public static final int NAT1_3_ID
```

NAT1_3_SINCE

```
public static final java.lang.String NAT1_3_SINCE
```

NAT1_3_STR

```
public static final java.lang.String NAT1_3_STR
```

NAT1_3_SYN

```
public static final java.lang.String NAT1_3_SYN
```

NAT1_4_DESC

```
public static final java.lang.String NAT1_4_DESC
```

NAT1_4_ID

```
public static final int NAT1_4_ID
```

NAT1_4_SINCE

```
public static final java.lang.String NAT1_4_SINCE
```

NAT1_4_STR

```
public static final java.lang.String NAT1_4_STR
```

NAT1_4_SYN

```
public static final java.lang.String NAT1_4_SYN
```

NAT1_5_DESC

```
public static final java.lang.String NAT1_5_DESC
```

NAT1_5_ID

```
public static final int NAT1_5_ID
```

NAT1_5_SINCE

```
public static final java.lang.String NAT1_5_SINCE
```

NAT1_5_STR

```
public static final java.lang.String NAT1_5_STR
```

NAT1_5_SYN

```
public static final java.lang.String NAT1_5_SYN
```

NAT1_6_DESC

```
public static final java.lang.String NAT1_6_DESC
```

NAT1_6_ID

```
public static final int NAT1_6_ID
```

NAT1_6_SINCE

```
public static final java.lang.String NAT1_6_SINCE
```

NAT1_6_STR

```
public static final java.lang.String NAT1_6_STR
```

NAT1_6_SYN

```
public static final java.lang.String NAT1_6_SYN
```

NAT1_7_DESC

```
public static final java.lang.String NAT1_7_DESC
```

NAT1_7_ID

```
public static final int NAT1_7_ID
```

NAT1_7_SINCE

```
public static final java.lang.String NAT1_7_SINCE
```

NAT1_7_STR

```
public static final java.lang.String NAT1_7_STR
```

NAT1_7_SYN

```
public static final java.lang.String NAT1_7_SYN
```

NAT1_8_DESC

```
public static final java.lang.String NAT1_8_DESC
```

NAT1_8_ID

```
public static final int NAT1_8_ID
```

NAT1_8_SINCE

```
public static final java.lang.String NAT1_8_SINCE
```

NAT1_8_STR

```
public static final java.lang.String NAT1_8_STR
```

NAT1_8_SYN

```
public static final java.lang.String NAT1_8_SYN
```

NAT1_9_DESC

```
public static final java.lang.String NAT1_9_DESC
```

NAT1_9_ID

```
public static final int NAT1_9_ID
```

NAT1_9_SINCE

```
public static final java.lang.String NAT1_9_SINCE
```

NAT1_9_STR

```
public static final java.lang.String NAT1_9_STR
```

NAT1_9_SYN

```
public static final java.lang.String NAT1_9_SYN
```

NAT1_DESC

```
public static final java.lang.String NAT1_DESC
```

NAT1_ID

```
public static final int NAT1_ID
```

NAT1_SINCE

```
public static final java.lang.String NAT1_SINCE
```

NAT1_STR

```
public static final java.lang.String NAT1_STR
```

NAT1_SYN

```
public static final java.lang.String NAT1_SYN
```

NOR_DESC

```
public static final java.lang.String NOR_DESC
```

NOR_ID

```
public static final int NOR_ID
```

NOR_SINCE

```
public static final java.lang.String NOR_SINCE
```

NOR_STR

```
public static final java.lang.String NOR_STR
```

NOR_SYN

```
public static final java.lang.String NOR_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

UNIFORM_DESC

```
public static final java.lang.String UNIFORM_DESC
```

UNIFORM_ID

```
public static final int UNIFORM_ID
```

UNIFORM_SINCE

```
public static final java.lang.String UNIFORM_SINCE
```

UNIFORM_STR

```
public static final java.lang.String UNIFORM_STR
```

UNIFORM_SYN

```
public static final java.lang.String UNIFORM_SYN
```

Constructors

RandomVariable

```
public RandomVariable()
```

```
org.mariuszgromada.math.mxparser.parsertokens
```

Class Token

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.parsertokens.Token
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class Token
extends java.lang.Object
```

Token recognized by mXparser after string tokenization process.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.0.0

Fields

NOT_MATCHED

```
public static final int NOT_MATCHED  
    Indicator that token was not matched
```

keyWord

```
public java.lang.String keyWord  
    Key word string (if matched)
```

looksLike

```
public java.lang.String looksLike  
    If token was not matched then looksLike functionality is trying asses the kind of token
```

tokenId

```
public int tokenId  
    Token identifier
```

tokenLevel

```
public int tokenLevel  
    Token level
```

tokenStr

```
public java.lang.String tokenStr
```

String token

tokenTypeId

```
public int tokenTypeId  
    Token type
```

tokenValue

```
public double tokenValue  
    Token value if number
```

Constructors

Token

```
public Token()  
    Default constructor
```

Methods

clone

```
public Token clone()  
    Token cloning.  
    Overrides:  
        clone in class java.lang.Object
```

org.mariuszgromada.math.mxparser.parsertokens

Class Unit

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.parsertokens.Unit
```

< [Fields](#) > < [Constructors](#) >

```
public final class Unit  
    extends java.lang.Object
```

Units - mXparser tokens definition.

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](https://bitbucket.org/mariuszgromada/mxparser/)

[mXparser on CodePlex](https://codeplex.com/mxparser/)

[Janet Sudoku](http://janet.sudoku.pl) - project web page

[Janet Sudoku on GitHub](https://github.com/mariuszgromada/janet-sudoku)

[Janet Sudoku on CodePlex](https://codeplex.com/janet-sudoku/)

[Janet Sudoku on SourceForge](https://sourceforge.net/projects/janet-sudoku/)

[Janet Sudoku on BitBucket](https://bitbucket.org/mariuszgromada/janet-sudoku/)

Version:

4.1.0

Fields

ACRE_DESC

```
public static final java.lang.String ACRE_DESC
```

ACRE_ID

```
public static final int ACRE_ID
```

ACRE_SINCE

```
public static final java.lang.String ACRE_SINCE
```

ACRE_STR

```
public static final java.lang.String ACRE_STR
```

ACRE_SYN

```
public static final java.lang.String ACRE_SYN
```

ARE_DESC

```
public static final java.lang.String ARE_DESC
```

ARE_ID

```
public static final int ARE_ID
```

ARE_SINCE

```
public static final java.lang.String ARE_SINCE
```

ARE_STR

```
public static final java.lang.String ARE_STR
```

ARE_SYN

```
public static final java.lang.String ARE_SYN
```

ATTO_DESC

```
public static final java.lang.String ATTO_DESC
```

ATTO_ID

```
public static final int ATTO_ID
```

ATTO_SINCE

```
public static final java.lang.String ATTO_SINCE
```

ATTO_STR

```
public static final java.lang.String ATTO_STR
```

ATTO_SYN


```
public static final java.lang.String ATTO_SYN
```

BIT_DESC

```
public static final java.lang.String BIT_DESC
```

BIT_ID

```
public static final int BIT_ID
```

BIT_SINCE

```
public static final java.lang.String BIT_SINCE
```

BIT_STR

```
public static final java.lang.String BIT_STR
```

BIT_SYN

```
public static final java.lang.String BIT_SYN
```

BYTE_DESC

```
public static final java.lang.String BYTE_DESC
```

BYTE_ID

```
public static final int BYTE_ID
```

BYTE_SINCE

```
public static final java.lang.String BYTE_SINCE
```

BYTE_STR

```
public static final java.lang.String BYTE_STR
```

BYTE_SYN

```
public static final java.lang.String BYTE_SYN
```

CENTIMETRE2_DESC

```
public static final java.lang.String CENTIMETRE2_DESC
```

CENTIMETRE2_ID

```
public static final int CENTIMETRE2_ID
```

CENTIMETRE2_SINCE

```
public static final java.lang.String CENTIMETRE2_SINCE
```

CENTIMETRE2_STR

```
public static final java.lang.String CENTIMETRE2_STR
```

CENTIMETRE2_SYN

```
public static final java.lang.String CENTIMETRE2_SYN
```

CENTIMETRE3_DESC

```
public static final java.lang.String CENTIMETRE3_DESC
```

CENTIMETRE3_ID

```
public static final int CENTIMETRE3_ID
```

CENTIMETRE3_SINCE

```
public static final java.lang.String CENTIMETRE3_SINCE
```

CENTIMETRE3_STR

```
public static final java.lang.String CENTIMETRE3_STR
```

CENTIMETRE3_SYN

```
public static final java.lang.String CENTIMETRE3_SYN
```

CENTIMETRE_DESC

```
public static final java.lang.String CENTIMETRE_DESC
```

CENTIMETRE_ID

```
public static final int CENTIMETRE_ID
```

CENTIMETRE_SINCE

```
public static final java.lang.String CENTIMETRE_SINCE
```

CENTIMETRE_STR

```
public static final java.lang.String CENTIMETRE_STR
```

CENTIMETRE_SYN

```
public static final java.lang.String CENTIMETRE_SYN
```

CENTI_DESC

```
public static final java.lang.String CENTI_DESC
```

CENTI_ID

```
public static final int CENTI_ID
```

CENTI_SINCE

```
public static final java.lang.String CENTI_SINCE
```

CENTI_STR

```
public static final java.lang.String CENTI_STR
```

CENTI_SYN

```
public static final java.lang.String CENTI_SYN
```

DAY_DESC

```
public static final java.lang.String DAY_DESC
```

DAY_ID

```
public static final int DAY_ID
```

DAY_SINCE

```
public static final java.lang.String DAY_SINCE
```

DAY_STR

```
public static final java.lang.String DAY_STR
```

DAY_SYN

```
public static final java.lang.String DAY_SYN
```

DECAGRAM_DESC

```
public static final java.lang.String DECAGRAM_DESC
```

DECAGRAM_ID

```
public static final int DECAGRAM_ID
```

DECAGRAM_SINCE

```
public static final java.lang.String DECAGRAM_SINCE
```

DECAGRAM_STR

```
public static final java.lang.String DECAGRAM_STR
```

DECAGRAM_SYN

```
public static final java.lang.String DECAGRAM_SYN
```

DECA_DESC

```
public static final java.lang.String DECA_DESC
```

DECA_ID

```
public static final int DECA_ID
```

DECA_SINCE

```
public static final java.lang.String DECA_SINCE
```

DECA_STR

```
public static final java.lang.String DECA_STR
```

DECA_SYN

```
public static final java.lang.String DECA_SYN
```

DECA_TEN_SINCE

```
public static final java.lang.String DECA_TEN_SINCE
```

DECA_TEN_STR

```
public static final java.lang.String DECA_TEN_STR
```

DECA_TEN_SYN

```
public static final java.lang.String DECA_TEN_SYN
```

DECI_DESC

```
public static final java.lang.String DECI_DESC
```

DECI_ID

```
public static final int DECI_ID
```

DECI_SINCE

```
public static final java.lang.String DECI_SINCE
```

DECI_STR

```
public static final java.lang.String DECI_STR
```

DECI_SYN

```
public static final java.lang.String DECI_SYN
```

DEGREE_ARC_DESC

```
public static final java.lang.String DEGREE_ARC_DESC
```

DEGREE_ARC_ID

```
public static final int DEGREE_ARC_ID
```

DEGREE_ARC_SINCE

```
public static final java.lang.String DEGREE_ARC_SINCE
```

DEGREE_ARC_STR

```
public static final java.lang.String DEGREE_ARC_STR
```

DEGREE_ARC_SYN

```
public static final java.lang.String DEGREE_ARC_SYN
```

ELECTRONO_VOLT_DESC

```
public static final java.lang.String ELECTRONO_VOLT_DESC
```

ELECTRONO_VOLT_ID

```
public static final int ELECTRONO_VOLT_ID
```

ELECTRONO_VOLT_SINCE

```
public static final java.lang.String ELECTRONO_VOLT_SINCE
```

ELECTRONO_VOLT_STR

```
public static final java.lang.String ELECTRONO_VOLT_STR
```

ELECTRONO_VOLT_SYN

```
public static final java.lang.String ELECTRONO_VOLT_SYN
```

EXABIT_DESC

```
public static final java.lang.String EXABIT_DESC
```

EXABIT_ID

```
public static final int EXABIT_ID
```

EXABIT_SINCE

```
public static final java.lang.String EXABIT_SINCE
```

EXABIT_STR

```
public static final java.lang.String EXABIT_STR
```

EXABIT_SYN

```
public static final java.lang.String EXABIT_SYN
```

EXABYTE_DESC

```
public static final java.lang.String EXABYTE_DESC
```

EXABYTE_ID

```
public static final int EXABYTE_ID
```

EXABYTE_SINCE

```
public static final java.lang.String EXABYTE_SINCE
```

EXABYTE_STR

```
public static final java.lang.String EXABYTE_STR
```

EXABYTE_SYN

```
public static final java.lang.String EXABYTE_SYN
```

EXA_DESC

```
public static final java.lang.String EXA_DESC
```

EXA_ID

```
public static final int EXA_ID
```

EXA_QUINT_SINCE


```
public static final java.lang.String EXA_QUINT_SINCE
```

EXA_QUINT_STR

```
public static final java.lang.String EXA_QUINT_STR
```

EXA_QUINT_SYN

```
public static final java.lang.String EXA_QUINT_SYN
```

EXA_SINCE

```
public static final java.lang.String EXA_SINCE
```

EXA_STR

```
public static final java.lang.String EXA_STR
```

EXA_SYN

```
public static final java.lang.String EXA_SYN
```

FEET_DESC

```
public static final java.lang.String FEET_DESC
```

FEET_ID

```
public static final int FEET_ID
```

FEET_SINCE

```
public static final java.lang.String FEET_SINCE
```

FEET_STR

```
public static final java.lang.String FEET_STR
```

FEET_SYN

```
public static final java.lang.String FEET_SYN
```

FEMTO_DESC

```
public static final java.lang.String FEMTO_DESC
```

FEMTO_ID

```
public static final int FEMTO_ID
```

FEMTO_SINCE

```
public static final java.lang.String FEMTO_SINCE
```

FEMTO_STR

```
public static final java.lang.String FEMTO_STR
```

FEMTO_SYN

```
public static final java.lang.String FEMTO_SYN
```

GALLON_DESC

```
public static final java.lang.String GALLON_DESC
```

GALLON_ID

```
public static final int GALLON_ID
```

GALLON_SINCE

```
public static final java.lang.String GALLON_SINCE
```

GALLON_STR

```
public static final java.lang.String GALLON_STR
```

GALLON_SYN

```
public static final java.lang.String GALLON_SYN
```

GIGABIT_DESC

```
public static final java.lang.String GIGABIT_DESC
```

GIGABIT_ID

```
public static final int GIGABIT_ID
```

GIGABIT_SINCE

```
public static final java.lang.String GIGABIT_SINCE
```

GIGABIT_STR

```
public static final java.lang.String GIGABIT_STR
```

GIGABIT_SYN

```
public static final java.lang.String GIGABIT_SYN
```

GIGABYTE_DESC

```
public static final java.lang.String GIGABYTE_DESC
```

GIGABYTE_ID

```
public static final int GIGABYTE_ID
```

GIGABYTE_SINCE

```
public static final java.lang.String GIGABYTE_SINCE
```

GIGABYTE_STR

```
public static final java.lang.String GIGABYTE_STR
```

GIGABYTE_SYN

```
public static final java.lang.String GIGABYTE_SYN
```

GIGA_BIL_SINCE

```
public static final java.lang.String GIGA_BIL_SINCE
```

GIGA_BIL_STR

```
public static final java.lang.String GIGA_BIL_STR
```

GIGA_BIL_SYN

```
public static final java.lang.String GIGA_BIL_SYN
```

GIGA_DESC

```
public static final java.lang.String GIGA_DESC
```

GIGA_ELECTRONO_VOLT_DESC

```
public static final java.lang.String GIGA_ELECTRONO_VOLT_DESC
```

GIGA_ELECTRONO_VOLT_ID

```
public static final int GIGA_ELECTRONO_VOLT_ID
```

GIGA_ELECTRONO_VOLT_SINCE

```
public static final java.lang.String GIGA_ELECTRONO_VOLT_SINCE
```

GIGA_ELECTRONO_VOLT_STR

```
public static final java.lang.String GIGA_ELECTRONO_VOLT_STR
```

GIGA_ELECTRONO_VOLT_SYN

```
public static final java.lang.String GIGA_ELECTRONO_VOLT_SYN
```

GIGA_ID

```
public static final int GIGA_ID
```

GIGA_SINCE

```
public static final java.lang.String GIGA_SINCE
```

GIGA_STR

```
public static final java.lang.String GIGA_STR
```

GIGA_SYN

```
public static final java.lang.String GIGA_SYN
```

GRAM_DESC

```
public static final java.lang.String GRAM_DESC
```

GRAM_ID

```
public static final int GRAM_ID
```

GRAM_SINCE

```
public static final java.lang.String GRAM_SINCE
```

GRAM_STR

```
public static final java.lang.String GRAM_STR
```

GRAM_SYN

```
public static final java.lang.String GRAM_SYN
```

HECTARE_DESC

```
public static final java.lang.String HECTARE_DESC
```

HECTARE_ID

```
public static final int HECTARE_ID
```

HECTARE_SINCE

```
public static final java.lang.String HECTARE_SINCE
```

HECTARE_STR

```
public static final java.lang.String HECTARE_STR
```

HECTARE_SYN

```
public static final java.lang.String HECTARE_SYN
```

HECTO_DESC

```
public static final java.lang.String HECTO_DESC
```

HECTO_HUND_SINCE

```
public static final java.lang.String HECTO_HUND_SINCE
```

HECTO_HUND_STR

```
public static final java.lang.String HECTO_HUND_STR
```

HECTO_HUND_SYN

```
public static final java.lang.String HECTO_HUND_SYN
```

HECTO_ID

```
public static final int HECTO_ID
```

HECTO_SINCE

```
public static final java.lang.String HECTO_SINCE
```

HECTO_STR

```
public static final java.lang.String HECTO_STR
```

HECTO_SYN

```
public static final java.lang.String HECTO_SYN
```

HOURL_DESC

```
public static final java.lang.String HOURL_DESC
```

HOURL_ID

```
public static final int HOURL_ID
```

HOURL_SINCE

```
public static final java.lang.String HOURL_SINCE
```

HOURL_STR

```
public static final java.lang.String HOURL_STR
```

HOURL_SYN

```
public static final java.lang.String HOURL_SYN
```

INCH_DESC

```
public static final java.lang.String INCH_DESC
```

INCH_ID

```
public static final int INCH_ID
```

INCH_SINCE

```
public static final java.lang.String INCH_SINCE
```

INCH_STR

```
public static final java.lang.String INCH_STR
```

INCH_SYN

```
public static final java.lang.String INCH_SYN
```

JOULE_DESC

```
public static final java.lang.String JOULE_DESC
```

JOULE_ID

```
public static final int JOULE_ID
```

JOULE_SINCE

```
public static final java.lang.String JOULE_SINCE
```

JOULE_STR

```
public static final java.lang.String JOULE_STR
```

JOULE_SYN


```
public static final java.lang.String JOULE_SYN
```

JULIAN_YEAR_DESC

```
public static final java.lang.String JULIAN_YEAR_DESC
```

JULIAN_YEAR_ID

```
public static final int JULIAN_YEAR_ID
```

JULIAN_YEAR_SINCE

```
public static final java.lang.String JULIAN_YEAR_SINCE
```

JULIAN_YEAR_STR

```
public static final java.lang.String JULIAN_YEAR_STR
```

JULIAN_YEAR_SYN

```
public static final java.lang.String JULIAN_YEAR_SYN
```

KILOBIT_DESC

```
public static final java.lang.String KILOBIT_DESC
```

KILOBIT_ID

```
public static final int KILOBIT_ID
```

KILOBIT_SINCE

```
public static final java.lang.String KILOBIT_SINCE
```

KILOBIT_STR

```
public static final java.lang.String KILOBIT_STR
```

KILOBIT_SYN

```
public static final java.lang.String KILOBIT_SYN
```

KILOBYTE_DESC

```
public static final java.lang.String KILOBYTE_DESC
```

KILOBYTE_ID

```
public static final int KILOBYTE_ID
```

KILOBYTE_SINCE

```
public static final java.lang.String KILOBYTE_SINCE
```

KILOBYTE_STR

```
public static final java.lang.String KILOBYTE_STR
```

KILOBYTE_SYN

```
public static final java.lang.String KILOBYTE_SYN
```

KILOGRAM_DESC

```
public static final java.lang.String KILOGRAM_DESC
```

KILOGRAM_ID

```
public static final int KILOGRAM_ID
```

KILOGRAM_SINCE

```
public static final java.lang.String KILOGRAM_SINCE
```

KILOGRAM_STR

```
public static final java.lang.String KILOGRAM_STR
```

KILOGRAM_SYN

```
public static final java.lang.String KILOGRAM_SYN
```

KILOMETRE2_DESC

```
public static final java.lang.String KILOMETRE2_DESC
```

KILOMETRE2_ID

```
public static final int KILOMETRE2_ID
```

KILOMETRE2_SINCE

```
public static final java.lang.String KILOMETRE2_SINCE
```

KILOMETRE2_STR

```
public static final java.lang.String KILOMETRE2_STR
```

KILOMETRE2_SYN

```
public static final java.lang.String KILOMETRE2_SYN
```

KILOMETRE3_DESC

```
public static final java.lang.String KILOMETRE3_DESC
```

KILOMETRE3_ID

```
public static final int KILOMETRE3_ID
```

KILOMETRE3_SINCE

```
public static final java.lang.String KILOMETRE3_SINCE
```

KILOMETRE3_STR

```
public static final java.lang.String KILOMETRE3_STR
```

KILOMETRE3_SYN

```
public static final java.lang.String KILOMETRE3_SYN
```

KILOMETRE_DESC

```
public static final java.lang.String KILOMETRE_DESC
```

KILOMETRE_ID

```
public static final int KILOMETRE_ID
```

KILOMETRE_PER_HOUR2_DESC

```
public static final java.lang.String KILOMETRE_PER_HOUR2_DESC
```

KILOMETRE_PER_HOUR2_ID

```
public static final int KILOMETRE_PER_HOUR2_ID
```

KILOMETRE_PER_HOUR2_SINCE

```
public static final java.lang.String KILOMETRE_PER_HOUR2_SINCE
```

KILOMETRE_PER_HOUR2_STR

```
public static final java.lang.String KILOMETRE_PER_HOUR2_STR
```

KILOMETRE_PER_HOUR2_SYN

```
public static final java.lang.String KILOMETRE_PER_HOUR2_SYN
```

KILOMETRE_PER_HOUR_DESC

```
public static final java.lang.String KILOMETRE_PER_HOUR_DESC
```

KILOMETRE_PER_HOUR_ID

```
public static final int KILOMETRE_PER_HOUR_ID
```

KILOMETRE_PER_HOUR_SINCE

```
public static final java.lang.String KILOMETRE_PER_HOUR_SINCE
```

KILOMETRE_PER_HOUR_STR

```
public static final java.lang.String KILOMETRE_PER_HOUR_STR
```

KILOMETRE_PER_HOUR_SYN

```
public static final java.lang.String KILOMETRE_PER_HOUR_SYN
```

KILOMETRE_SINCE

```
public static final java.lang.String KILOMETRE_SINCE
```

KILOMETRE_STR

```
public static final java.lang.String KILOMETRE_STR
```

KILOMETRE_SYN

```
public static final java.lang.String KILOMETRE_SYN
```

KILO_DESC

```
public static final java.lang.String KILO_DESC
```

KILO_ELECTRONO_VOLT_DESC

```
public static final java.lang.String KILO_ELECTRONO_VOLT_DESC
```

KILO_ELECTRONO_VOLT_ID

```
public static final int KILO_ELECTRONO_VOLT_ID
```

KILO_ELECTRONO_VOLT_SINCE

```
public static final java.lang.String KILO_ELECTRONO_VOLT_SINCE
```

KILO_ELECTRONO_VOLT_STR

```
public static final java.lang.String KILO_ELECTRONO_VOLT_STR
```

KILO_ELECTRONO_VOLT_SYN

```
public static final java.lang.String KILO_ELECTRONO_VOLT_SYN
```

KILO_ID

```
public static final int KILO_ID
```

KILO_SINCE

```
public static final java.lang.String KILO_SINCE
```

KILO_STR

```
public static final java.lang.String KILO_STR
```

KILO_SYN

```
public static final java.lang.String KILO_SYN
```

KILO_TH_SINCE

```
public static final java.lang.String KILO_TH_SINCE
```

KILO_TH_STR

```
public static final java.lang.String KILO_TH_STR
```

KILO_TH_SYN

```
public static final java.lang.String KILO_TH_SYN
```

KNOT_DESC

```
public static final java.lang.String KNOT_DESC
```

KNOT_ID

```
public static final int KNOT_ID
```

KNOT_SINCE

```
public static final java.lang.String KNOT_SINCE
```

KNOT_STR

```
public static final java.lang.String KNOT_STR
```

KNOT_SYN

```
public static final java.lang.String KNOT_SYN
```

LITRE_DESC

```
public static final java.lang.String LITRE_DESC
```

LITRE_ID

```
public static final int LITRE_ID
```

LITRE_SINCE

```
public static final java.lang.String LITRE_SINCE
```

LITRE_STR

```
public static final java.lang.String LITRE_STR
```

LITRE_SYN

```
public static final java.lang.String LITRE_SYN
```

MEGABIT_DESC

```
public static final java.lang.String MEGABIT_DESC
```

MEGABIT_ID

```
public static final int MEGABIT_ID
```

MEGABIT_SINCE

```
public static final java.lang.String MEGABIT_SINCE
```

MEGABIT_STR

```
public static final java.lang.String MEGABIT_STR
```

MEGABIT_SYN

```
public static final java.lang.String MEGABIT_SYN
```

MEGABYTE_DESC

```
public static final java.lang.String MEGABYTE_DESC
```

MEGABYTE_ID

```
public static final int MEGABYTE_ID
```

MEGABYTE_SINCE


```
public static final java.lang.String MEGABYTE_SINCE
```

MEGABYTE_STR

```
public static final java.lang.String MEGABYTE_STR
```

MEGABYTE_SYN

```
public static final java.lang.String MEGABYTE_SYN
```

MEGA_DESC

```
public static final java.lang.String MEGA_DESC
```

MEGA_ELECTRONO_VOLT_DESC

```
public static final java.lang.String MEGA_ELECTRONO_VOLT_DESC
```

MEGA_ELECTRONO_VOLT_ID

```
public static final int MEGA_ELECTRONO_VOLT_ID
```

MEGA_ELECTRONO_VOLT_SINCE

```
public static final java.lang.String MEGA_ELECTRONO_VOLT_SINCE
```

MEGA_ELECTRONO_VOLT_STR

```
public static final java.lang.String MEGA_ELECTRONO_VOLT_STR
```

MEGA_ELECTRONO_VOLT_SYN

```
public static final java.lang.String MEGA_ELECTRONO_VOLT_SYN
```

MEGA_ID

```
public static final int MEGA_ID
```

MEGA_MIL_SINCE

```
public static final java.lang.String MEGA_MIL_SINCE
```

MEGA_MIL_STR

```
public static final java.lang.String MEGA_MIL_STR
```

MEGA_MIL_SYN

```
public static final java.lang.String MEGA_MIL_SYN
```

MEGA_SINCE

```
public static final java.lang.String MEGA_SINCE
```

MEGA_STR

```
public static final java.lang.String MEGA_STR
```

MEGA_SYN

```
public static final java.lang.String MEGA_SYN
```

METRE2_DESC

```
public static final java.lang.String METRE2_DESC
```

METRE2_ID

```
public static final int METRE2_ID
```

METRE2_SINCE

```
public static final java.lang.String METRE2_SINCE
```

METRE2_STR

```
public static final java.lang.String METRE2_STR
```

METRE2_SYN

```
public static final java.lang.String METRE2_SYN
```

METRE3_DESC

```
public static final java.lang.String METRE3_DESC
```

METRE3_ID

```
public static final int METRE3_ID
```

METRE3_SINCE

```
public static final java.lang.String METRE3_SINCE
```

METRE3_STR

```
public static final java.lang.String METRE3_STR
```

METRE3_SYN

```
public static final java.lang.String METRE3_SYN
```

METRE_DESC

```
public static final java.lang.String METRE_DESC
```

METRE_ID

```
public static final int METRE_ID
```

METRE_PER_SECOND2_DESC

```
public static final java.lang.String METRE_PER_SECOND2_DESC
```

METRE_PER_SECOND2_ID

```
public static final int METRE_PER_SECOND2_ID
```

METRE_PER_SECOND2_SINCE

```
public static final java.lang.String METRE_PER_SECOND2_SINCE
```

METRE_PER_SECOND2_STR

```
public static final java.lang.String METRE_PER_SECOND2_STR
```

METRE_PER_SECOND2_SYN

```
public static final java.lang.String METRE_PER_SECOND2_SYN
```

METRE_PER_SECOND_DESC

```
public static final java.lang.String METRE_PER_SECOND_DESC
```

METRE_PER_SECOND_ID

```
public static final int METRE_PER_SECOND_ID
```

METRE_PER_SECOND_SINCE

```
public static final java.lang.String METRE_PER_SECOND_SINCE
```

METRE_PER_SECOND_STR

```
public static final java.lang.String METRE_PER_SECOND_STR
```

METRE_PER_SECOND_SYN

```
public static final java.lang.String METRE_PER_SECOND_SYN
```

METRE_SINCE

```
public static final java.lang.String METRE_SINCE
```

METRE_STR

```
public static final java.lang.String METRE_STR
```

METRE_SYN

```
public static final java.lang.String METRE_SYN
```

MICRO_DESC

```
public static final java.lang.String MICRO_DESC
```

MICRO_ID

```
public static final int MICRO_ID
```

MICRO_SINCE

```
public static final java.lang.String MICRO_SINCE
```

MICRO_STR

```
public static final java.lang.String MICRO_STR
```

MICRO_SYN

```
public static final java.lang.String MICRO_SYN
```

MILE_DESC

```
public static final java.lang.String MILE_DESC
```

MILE_ID

```
public static final int MILE_ID
```

MILE_PER_HOUR2_DESC

```
public static final java.lang.String MILE_PER_HOUR2_DESC
```

MILE_PER_HOUR2_ID

```
public static final int MILE_PER_HOUR2_ID
```

MILE_PER_HOUR2_SINCE

```
public static final java.lang.String MILE_PER_HOUR2_SINCE
```

MILE_PER_HOUR2_STR

```
public static final java.lang.String MILE_PER_HOUR2_STR
```

MILE_PER_HOUR2_SYN

```
public static final java.lang.String MILE_PER_HOUR2_SYN
```

MILE_PER_HOUR_DESC

```
public static final java.lang.String MILE_PER_HOUR_DESC
```

MILE_PER_HOUR_ID

```
public static final int MILE_PER_HOUR_ID
```

MILE_PER_HOUR_SINCE

```
public static final java.lang.String MILE_PER_HOUR_SINCE
```

MILE_PER_HOUR_STR

```
public static final java.lang.String MILE_PER_HOUR_STR
```

MILE_PER_HOUR_SYN

```
public static final java.lang.String MILE_PER_HOUR_SYN
```

MILE_SINCE

```
public static final java.lang.String MILE_SINCE
```

MILE_STR

```
public static final java.lang.String MILE_STR
```

MILE_SYN

```
public static final java.lang.String MILE_SYN
```

MILLIGRAM_DESC

```
public static final java.lang.String MILLIGRAM_DESC
```

MILLIGRAM_ID

```
public static final int MILLIGRAM_ID
```

MILLIGRAM_SINCE

```
public static final java.lang.String MILLIGRAM_SINCE
```

MILLIGRAM_STR

```
public static final java.lang.String MILLIGRAM_STR
```

MILLIGRAM_SYN

```
public static final java.lang.String MILLIGRAM_SYN
```

MILLILITRE_DESC

```
public static final java.lang.String MILLILITRE_DESC
```

MILLILITRE_ID

```
public static final int MILLILITRE_ID
```

MILLILITRE_SINCE

```
public static final java.lang.String MILLILITRE_SINCE
```

MILLILITRE_STR

```
public static final java.lang.String MILLILITRE_STR
```

MILLILITRE_SYN

```
public static final java.lang.String MILLILITRE_SYN
```

MILLIMETRE2_DESC

```
public static final java.lang.String MILLIMETRE2_DESC
```

MILLIMETRE2_ID

```
public static final int MILLIMETRE2_ID
```

MILLIMETRE2_SINCE

```
public static final java.lang.String MILLIMETRE2_SINCE
```

MILLIMETRE2_STR

```
public static final java.lang.String MILLIMETRE2_STR
```

MILLIMETRE2_SYN

```
public static final java.lang.String MILLIMETRE2_SYN
```

MILLIMETRE3_DESC


```
public static final java.lang.String MILLIMETRE3_DESC
```

MILLIMETRE3_ID

```
public static final int MILLIMETRE3_ID
```

MILLIMETRE3_SINCE

```
public static final java.lang.String MILLIMETRE3_SINCE
```

MILLIMETRE3_STR

```
public static final java.lang.String MILLIMETRE3_STR
```

MILLIMETRE3_SYN

```
public static final java.lang.String MILLIMETRE3_SYN
```

MILLIMETRE_DESC

```
public static final java.lang.String MILLIMETRE_DESC
```

MILLIMETRE_ID

```
public static final int MILLIMETRE_ID
```

MILLIMETRE_SINCE

```
public static final java.lang.String MILLIMETRE_SINCE
```

MILLIMETRE_STR

```
public static final java.lang.String MILLIMETRE_STR
```

MILLIMETRE_SYN

```
public static final java.lang.String MILLIMETRE_SYN
```

MILLISECOND_DESC

```
public static final java.lang.String MILLISECOND_DESC
```

MILLISECOND_ID

```
public static final int MILLISECOND_ID
```

MILLISECOND_SINCE

```
public static final java.lang.String MILLISECOND_SINCE
```

MILLISECOND_STR

```
public static final java.lang.String MILLISECOND_STR
```

MILLISECOND_SYN

```
public static final java.lang.String MILLISECOND_SYN
```

MILLI_DESC

```
public static final java.lang.String MILLI_DESC
```

MILLI_ID

```
public static final int MILLI_ID
```

MILLI_SINCE

```
public static final java.lang.String MILLI_SINCE
```

MILLI_STR

```
public static final java.lang.String MILLI_STR
```

MILLI_SYN

```
public static final java.lang.String MILLI_SYN
```

MINUTE_ARC_DESC

```
public static final java.lang.String MINUTE_ARC_DESC
```

MINUTE_ARC_ID

```
public static final int MINUTE_ARC_ID
```

MINUTE_ARC_SINCE

```
public static final java.lang.String MINUTE_ARC_SINCE
```

MINUTE_ARC_STR

```
public static final java.lang.String MINUTE_ARC_STR
```

MINUTE_ARC_SYN

```
public static final java.lang.String MINUTE_ARC_SYN
```

MINUTE_DESC

```
public static final java.lang.String MINUTE_DESC
```

MINUTE_ID

```
public static final int MINUTE_ID
```

MINUTE_SINCE

```
public static final java.lang.String MINUTE_SINCE
```

MINUTE_STR

```
public static final java.lang.String MINUTE_STR
```

MINUTE_SYN

```
public static final java.lang.String MINUTE_SYN
```

NANO_DESC

```
public static final java.lang.String NANO_DESC
```

NANO_ID

```
public static final int NANO_ID
```

NANO_SINCE

```
public static final java.lang.String NANO_SINCE
```

NANO_STR

```
public static final java.lang.String NANO_STR
```

NANO_SYN

```
public static final java.lang.String NANO_SYN
```

NAUTICAL_MILE_DESC

```
public static final java.lang.String NAUTICAL_MILE_DESC
```

NAUTICAL_MILE_ID

```
public static final int NAUTICAL_MILE_ID
```

NAUTICAL_MILE_SINCE

```
public static final java.lang.String NAUTICAL_MILE_SINCE
```

NAUTICAL_MILE_STR

```
public static final java.lang.String NAUTICAL_MILE_STR
```

NAUTICAL_MILE_SYN

```
public static final java.lang.String NAUTICAL_MILE_SYN
```

OUNCE_DESC

```
public static final java.lang.String OUNCE_DESC
```

OUNCE_ID

```
public static final int OUNCE_ID
```

OUNCE_SINCE

```
public static final java.lang.String OUNCE_SINCE
```

OUNCE_STR

```
public static final java.lang.String OUNCE_STR
```

OUNCE_SYN

```
public static final java.lang.String OUNCE_SYN
```

PERC_DESC

```
public static final java.lang.String PERC_DESC
```

PERC_ID

```
public static final int PERC_ID
```

PERC_SINCE

```
public static final java.lang.String PERC_SINCE
```

PERC_STR

```
public static final java.lang.String PERC_STR
```

PERC_SYN

```
public static final java.lang.String PERC_SYN
```

PETABIT_DESC

```
public static final java.lang.String PETABIT_DESC
```

PETABIT_ID

```
public static final int PETABIT_ID
```

PETABIT_SINCE

```
public static final java.lang.String PETABIT_SINCE
```

PETABIT_STR

```
public static final java.lang.String PETABIT_STR
```

PETABIT_SYN

```
public static final java.lang.String PETABIT_SYN
```

PETABYTE_DESC

```
public static final java.lang.String PETABYTE_DESC
```

PETABYTE_ID

```
public static final int PETABYTE_ID
```

PETABYTE_SINCE

```
public static final java.lang.String PETABYTE_SINCE
```

PETABYTE_STR

```
public static final java.lang.String PETABYTE_STR
```

PETABYTE_SYN

```
public static final java.lang.String PETABYTE_SYN
```

PETA_DESC

```
public static final java.lang.String PETA_DESC
```

PETA_ID

```
public static final int PETA_ID
```

PETA_QUAD_SINCE

```
public static final java.lang.String PETA_QUAD_SINCE
```

PETA_QUAD_STR

```
public static final java.lang.String PETA_QUAD_STR
```

PETA_QUAD_SYN

```
public static final java.lang.String PETA_QUAD_SYN
```

PETA_SINCE

```
public static final java.lang.String PETA_SINCE
```

PETA_STR

```
public static final java.lang.String PETA_STR
```

PETA_SYN

```
public static final java.lang.String PETA_SYN
```

PICO_DESC

```
public static final java.lang.String PICO_DESC
```

PICO_ID

```
public static final int PICO_ID
```

PICO_SINCE

```
public static final java.lang.String PICO_SINCE
```

PICO_STR

```
public static final java.lang.String PICO_STR
```

PICO_SYN

```
public static final java.lang.String PICO_SYN
```

PINT_DESC

```
public static final java.lang.String PINT_DESC
```

PINT_ID

```
public static final int PINT_ID
```

PINT_SINCE

```
public static final java.lang.String PINT_SINCE
```

PINT_STR


```
public static final java.lang.String PINT_STR
```

PINT_SYN

```
public static final java.lang.String PINT_SYN
```

POUND_DESC

```
public static final java.lang.String POUND_DESC
```

POUND_ID

```
public static final int POUND_ID
```

POUND_SINCE

```
public static final java.lang.String POUND_SINCE
```

POUND_STR

```
public static final java.lang.String POUND_STR
```

POUND_SYN

```
public static final java.lang.String POUND_SYN
```

PROMIL_DESC

```
public static final java.lang.String PROMIL_DESC
```

PROMIL_ID

```
public static final int PROMIL_ID
```

PROMIL_SINCE

```
public static final java.lang.String PROMIL_SINCE
```

PROMIL_STR

```
public static final java.lang.String PROMIL_STR
```

PROMIL_SYN

```
public static final java.lang.String PROMIL_SYN
```

RADIAN_ARC_DESC

```
public static final java.lang.String RADIAN_ARC_DESC
```

RADIAN_ARC_ID

```
public static final int RADIAN_ARC_ID
```

RADIAN_ARC_SINCE

```
public static final java.lang.String RADIAN_ARC_SINCE
```

RADIAN_ARC_STR

```
public static final java.lang.String RADIAN_ARC_STR
```

RADIAN_ARC_SYN

```
public static final java.lang.String RADIAN_ARC_SYN
```

SECOND_ARC_DESC

```
public static final java.lang.String SECOND_ARC_DESC
```

SECOND_ARC_ID

```
public static final int SECOND_ARC_ID
```

SECOND_ARC_SINCE

```
public static final java.lang.String SECOND_ARC_SINCE
```

SECOND_ARC_STR

```
public static final java.lang.String SECOND_ARC_STR
```

SECOND_ARC_SYN

```
public static final java.lang.String SECOND_ARC_SYN
```

SECOND_DESC

```
public static final java.lang.String SECOND_DESC
```

SECOND_ID

```
public static final int SECOND_ID
```

SECOND_SINCE

```
public static final java.lang.String SECOND_SINCE
```

SECOND_STR

```
public static final java.lang.String SECOND_STR
```

SECOND_SYN

```
public static final java.lang.String SECOND_SYN
```

TERABIT_DESC

```
public static final java.lang.String TERABIT_DESC
```

TERABIT_ID

```
public static final int TERABIT_ID
```

TERABIT_SINCE

```
public static final java.lang.String TERABIT_SINCE
```

TERABIT_STR

```
public static final java.lang.String TERABIT_STR
```

TERABIT_SYN

```
public static final java.lang.String TERABIT_SYN
```

TERABYTE_DESC

```
public static final java.lang.String TERABYTE_DESC
```

TERABYTE_ID

```
public static final int TERABYTE_ID
```

TERABYTE_SINCE

```
public static final java.lang.String TERABYTE_SINCE
```

TERABYTE_STR

```
public static final java.lang.String TERABYTE_STR
```

TERABYTE_SYN

```
public static final java.lang.String TERABYTE_SYN
```

TERA_DESC

```
public static final java.lang.String TERA_DESC
```

TERA_ELECTRONO_VOLT_DESC

```
public static final java.lang.String TERA_ELECTRONO_VOLT_DESC
```

TERA_ELECTRONO_VOLT_ID

```
public static final int TERA_ELECTRONO_VOLT_ID
```

TERA_ELECTRONO_VOLT_SINCE

```
public static final java.lang.String TERA_ELECTRONO_VOLT_SINCE
```

TERA_ELECTRONO_VOLT_STR

```
public static final java.lang.String TERA_ELECTRONO_VOLT_STR
```

TERA_ELECTRONO_VOLT_SYN

```
public static final java.lang.String TERA_ELECTRONO_VOLT_SYN
```

TERA_ID

```
public static final int TERA_ID
```

TERA_SINCE

```
public static final java.lang.String TERA_SINCE
```

TERA_STR

```
public static final java.lang.String TERA_STR
```

TERA_SYN

```
public static final java.lang.String TERA_SYN
```

TERA_TRIL_SINCE

```
public static final java.lang.String TERA_TRIL_SINCE
```

TERA_TRIL_STR

```
public static final java.lang.String TERA_TRIL_STR
```

TERA_TRIL_SYN

```
public static final java.lang.String TERA_TRIL_SYN
```

TONNE_DESC

```
public static final java.lang.String TONNE_DESC
```

TONNE_ID

```
public static final int TONNE_ID
```

TONNE_SINCE

```
public static final java.lang.String TONNE_SINCE
```

TONNE_STR

```
public static final java.lang.String TONNE_STR
```

TONNE_SYN

```
public static final java.lang.String TONNE_SYN
```

TYPE_DESC

```
public static final java.lang.String TYPE_DESC
```

TYPE_ID

```
public static final int TYPE_ID
```

WEEK_DESC

```
public static final java.lang.String WEEK_DESC
```

WEEK_ID

```
public static final int WEEK_ID
```

WEEK_SINCE

```
public static final java.lang.String WEEK_SINCE
```

WEEK_STR

```
public static final java.lang.String WEEK_STR
```

WEEK_SYN

```
public static final java.lang.String WEEK_SYN
```

YARD_DESC

```
public static final java.lang.String YARD_DESC
```

YARD_ID

```
public static final int YARD_ID
```

YARD_SINCE

```
public static final java.lang.String YARD_SINCE
```

YARD_STR

```
public static final java.lang.String YARD_STR
```

YARD_SYN

```
public static final java.lang.String YARD_SYN
```

YOCTO_DESC

```
public static final java.lang.String YOCTO_DESC
```

YOCTO_ID

```
public static final int YOCTO_ID
```

YOCTO_SINCE

```
public static final java.lang.String YOCTO_SINCE
```

YOCTO_STR

```
public static final java.lang.String YOCTO_STR
```

YOCTO_SYN

```
public static final java.lang.String YOCTO_SYN
```

YOTTABIT_DESC

```
public static final java.lang.String YOTTABIT_DESC
```

YOTTABIT_ID

```
public static final int YOTTABIT_ID
```

YOTTABIT_SINCE

```
public static final java.lang.String YOTTABIT_SINCE
```

YOTTABIT_STR

```
public static final java.lang.String YOTTABIT_STR
```

YOTTABIT_SYN


```
public static final java.lang.String YOTTABIT_SYN
```

YOTTABYTE_DESC

```
public static final java.lang.String YOTTABYTE_DESC
```

YOTTABYTE_ID

```
public static final int YOTTABYTE_ID
```

YOTTABYTE_SINCE

```
public static final java.lang.String YOTTABYTE_SINCE
```

YOTTABYTE_STR

```
public static final java.lang.String YOTTABYTE_STR
```

YOTTABYTE_SYN

```
public static final java.lang.String YOTTABYTE_SYN
```

YOTTA_DESC

```
public static final java.lang.String YOTTA_DESC
```

YOTTA_ID

```
public static final int YOTTA_ID
```

YOTTA_SEPT_SINCE

```
public static final java.lang.String YOTTA_SEPT_SINCE
```

YOTTA_SEPT_STR

```
public static final java.lang.String YOTTA_SEPT_STR
```

YOTTA_SEPT_SYN

```
public static final java.lang.String YOTTA_SEPT_SYN
```

YOTTA_SINCE

```
public static final java.lang.String YOTTA_SINCE
```

YOTTA_STR

```
public static final java.lang.String YOTTA_STR
```

YOTTA_SYN

```
public static final java.lang.String YOTTA_SYN
```

ZEPTO_DESC

```
public static final java.lang.String ZEPTO_DESC
```

ZEPTO_ID

```
public static final int ZEPTO_ID
```

ZEPTO_SINCE

```
public static final java.lang.String ZEPTO_SINCE
```

ZEPTO_STR

```
public static final java.lang.String ZEPTO_STR
```

ZEPTO_SYN

```
public static final java.lang.String ZEPTO_SYN
```

ZETTABIT_DESC

```
public static final java.lang.String ZETTABIT_DESC
```

ZETTABIT_ID

```
public static final int ZETTABIT_ID
```

ZETTABIT_SINCE

```
public static final java.lang.String ZETTABIT_SINCE
```

ZETTABIT_STR

```
public static final java.lang.String ZETTABIT_STR
```

ZETTABIT_SYN

```
public static final java.lang.String ZETTABIT_SYN
```

ZETTABYTE_DESC

```
public static final java.lang.String ZETTABYTE_DESC
```

ZETTABYTE_ID

```
public static final int ZETTABYTE_ID
```

ZETTABYTE_SINCE

```
public static final java.lang.String ZETTABYTE_SINCE
```

ZETTABYTE_STR

```
public static final java.lang.String ZETTABYTE_STR
```

ZETTABYTE_SYN

```
public static final java.lang.String ZETTABYTE_SYN
```

ZETTA_DESC

```
public static final java.lang.String ZETTA_DESC
```

ZETTA_ID

```
public static final int ZETTA_ID
```

ZETTA_SEXT_SINCE

```
public static final java.lang.String ZETTA_SEXT_SINCE
```

ZETTA_SEXT_STR

```
public static final java.lang.String ZETTA_SEXT_STR
```

ZETTA_SEXT_SYN

```
public static final java.lang.String ZETTA_SEXT_SYN
```

ZETTA_SINCE

```
public static final java.lang.String ZETTA_SINCE
```

ZETTA_STR

```
public static final java.lang.String ZETTA_STR
```

ZETTA_SYN

```
public static final java.lang.String ZETTA_SYN
```

Constructors

Unit

```
public Unit()
```

Package org.mariuszgromada.math.mxparser.regression

Class Summary

[FunExt](#)

Example of implementation FunctionExtension interface

[PerformanceTestResult](#)

Package level class to keep performance test result/

[PerformanceTests](#)

PerformanceTests - mXparser performance tests

[RegTestExpression](#)

RegTestExpression - regression tests for the expression calculation

[RegTestExpressionAPI](#)

RegTestExpressionAPI - regression tests for the expression API

[RegTestSyntax](#)

RegTestSyntax - regression tests for the expression syntax checking

[RunTest](#)

Use this class to run one of the following test

- Param: reg - Expression regression test
- Param: api - mXparser API test
- Param: syn - Syntax checking test
- Param: perf - Performance test

[Test011Thread](#)

Multithreading implementation of class for Performance test: Simple calculations - addition with argument.

[Test012Thread](#)

Multithreading implementation of class for Performance test: User defined function $f(x,y)=3x+4y$.

[Test013Thread](#)

Multithreading implementation of class for Performance test: Creating constants: Iteration:
Constant c = new Constant("c", 5)

[Test014Thread](#)

Multithreading implementation of class for Performance test: Creating constants: Iteration:
Constant c = new Constant("c=5")

[Test015Thread](#)

Multithreading implementation of class for Performance test: Creating arguments: Iteration:
Argument x = new Argument("x", 5)

[Test016Thread](#)

Multithreading implementation of class for Performance test: Creating arguments: Iteration:
Argument x = new Argument("x = 5")

[Test017Thread](#)

Multithreading implementation of class for Performance test: Creating functions: Iteration: Function
f = new Function("f", "x+y", "x", "y")

[Test018Thread](#)

Multithreading implementation of class for Performance test: Creating functions: Iteration: Function
f = new Function("f(x,y)=x+y")

[Test019Thread](#)

Multithreading implementation of class for Performance test: Creating expressions: Iteration:
Expression e = new Expression("sin(2+(3*4)^2)/10")

[Test020Thread](#)

Multithreading implementation of class for Performance test: Creating expressions: Expression e =
new Expression("") Iteration: e.setExpressionString(sin(2+(3*4)^2)/10); e.checkSyntax();

[TestSimpleCalcThread](#)

Multithreading implementation of class for Performance test: Simple calculations

[TestThread](#)

Multithreading abstract implementation of test

org.mariuszgromada.math.mxparser.regressiontesting

Class FunExt

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.regressiontesting.FunExt
```

All Implemented Interfaces:

[FunctionExtension](#)

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
class FunExt
  extends java.lang.Object
  implements FunctionExtension
```

Example of implementation FunctionExtension interface

FunctionExtension

Fields

double **x**

y

double **y**

Constructors

FunExt

```
FunExt()
```

FunExt

```
FunExt(double x,  
        double y)
```

Methods

calculate

```
public double calculate()
```

clone

```
public FunExt clone()
```

Overrides:

clone in class java.lang.Object

getParameterName

```
public java.lang.String getParameterName(int parameterIndex)
```

getParametersNumber

```
public int getParametersNumber()
```

setParameterValue

```
public void setParameterValue(int parameterIndex,  
                             double parameterValue)
```

org.mariuszgromada.math.mxparser.regressiontesting

Class PerformanceTestResult

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.regressiontesting.PerformanceTestResult
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

class **PerformanceTestResult**
extends java.lang.Object

Package level class to keep performance test result/

Fields

Id

```
int Id
```

computingTimeSec

```
double computingTimeSec
```

description

```
java.lang.String description
```

endTime

long **endTime**

exprStr

java.lang.String **exprStr**

iterNum

int **iterNum**

iterPerSec

long **iterPerSec**

startTime

long **startTime**

threadsNum

int **threadsNum**

Constructors

PerformanceTestResult

PerformanceTestResult(int threadsNum)

Methods

testClose

void **testClose**()

testInit

```
void testInit()
```

org.mariuszgromada.math.mxparser.regressiontesting

Class PerformanceTests

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.regressiontesting.PerformanceTests
```

< [Constructors](#) > < [Methods](#) >

```
public class PerformanceTests
extends java.lang.Object
```

PerformanceTests - mXparser performance tests

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Expression

Constructors

PerformanceTests

```
public PerformanceTests()
```

Methods

createRunJoinThreads

```
static void createRunJoinThreads(PerformanceTestResult test,  
                                int classId)
```

Creates threads, executes them, then wait till each thread is finished

Parameters:

test - Test definition

classId - Class id specifying the implementation of test scenario

main

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

Performance test run with multithreading support.

Parameters:

args - If parameters are given then only the first one is verified, and is considered as number of threads.

start

```
public static int start()
```

Starts mXparser performance tests - number of threads given by the
mXparser.getThreadsNumber() List of performed tests:

- 00. Simple calculations - addition. Expression created once. Iteration: repeatedly recalculated same expression.
- 01. Simple calculations - multiplication. Expression created once. Iteration: repeatedly recalculated same expression.
- 02. Simple calculations - division. Expression created once. Iteration: repeatedly recalculated same expression.
- 03. Simple calculations - power. Expression created once. Iteration: repeatedly recalculated same expression.
- 04. Simple calculations - sinus. Expression created once. Iteration: repeatedly recalculated same expression.
- 05. Simple calculations - 2 additions. Expression created once. Iteration: repeatedly recalculated same expression.
- 06. Simple calculations - 3 additions. Expression created once. Iteration: repeatedly recalculated same expression.
- 07. Simple calculations - 3 additions + 1 parenthesis. Expression created once. Iteration: repeatedly recalculated same expression.
- 08. Simple calculations - 3 additions + 2 brackets. Expression created once. Iteration: repeatedly recalculated same expression.
- 09. Simple calculations - 3 additions + 2 brackets. Expression created once. Iteration: repeatedly recalculated same expression.
- 10. Combination of different operations. Expression created once. Iteration: repeatedly recalculated same expression.
- 11. Simple calculations - addition with argument. Expression created once, containing argument 'x'. Iteration: argument value is being modified (increased), then expression is recalculated
- 12. User defined function $f(x,y)=3x+4y$. Expression Function created once, containing argument 'x'. Iteration: argument value is being modified (increased), then expression is recalculated
- 13. Creating constants: Iteration: Constant $c = \text{new Constant}("c", 5)$
- 14. Creating constants: Iteration: Constant $c = \text{new Constant}("c=5")$
- 15. Creating arguments: Iteration: Argument $x = \text{new Argument}("x", 5)$
- 16. Creating arguments: Iteration: Argument $x = \text{new Argument}("x=5")$
- 17. Creating functions: Iteration: Function $f = \text{new Function}("f", "x+y", "x", "y")$
- 18. Creating functions: Iteration: Function $f = \text{new Function}("f(x,y)=x+y")$
- 19. Creating expressions: Iteration: Expression $e = \text{new Expression}("sin(2+(3*4)^2)/10")$
- 20. Creating expressions + checking syntax: Iteration: Expression $e = \text{new Expression}("sin(2+(3*4)^2)/10")$

Returns:

Number of tests that were not performed.

start

```
public static int start(int threadsNum)
```

Starts mXparser performance tests. List of performed tests:

- 00. Simple calculations - addition. Expression created once. Iteration: repeatedly recalculated same expression.
- 01. Simple calculations - multiplication. Expression created once. Iteration: repeatedly recalculated same expression.
- 02. Simple calculations - division. Expression created once. Iteration: repeatedly recalculated same expression.
- 03. Simple calculations - power. Expression created once. Iteration: repeatedly recalculated same expression.
- 04. Simple calculations - sinus. Expression created once. Iteration: repeatedly recalculated same expression.
- 05. Simple calculations - 2 additions. Expression created once. Iteration: repeatedly recalculated same expression.
- 06. Simple calculations - 3 additions. Expression created once. Iteration: repeatedly recalculated same expression.
- 07. Simple calculations - 3 additions + 1 parenthesis. Expression created once. Iteration: repeatedly recalculated same expression.
- 08. Simple calculations - 3 additions + 2 brackets. Expression created once. Iteration: repeatedly recalculated same expression.
- 09. Simple calculations - 3 additions + 2 brackets. Expression created once. Iteration: repeatedly recalculated same expression.
- 10. Combination of different operations. Expression created once. Iteration: repeatedly recalculated same expression.
- 11. Simple calculations - addition with argument. Expression created once, containing argument 'x'. Iteration: argument value is being modified (increased), then expression is recalculated
- 12. User defined function $f(x,y)=3x+4y$. Expression Function created once, containing argument 'x'. Iteration: argument value is being modified (increased), then expression is recalculated
- 13. Creating constants: Iteration: Constant $c = \text{new Constant}("c", 5)$
- 14. Creating constants: Iteration: Constant $c = \text{new Constant}("c=5")$
- 15. Creating arguments: Iteration: Argument $x = \text{new Argument}("x", 5)$
- 16. Creating arguments: Iteration: Argument $x = \text{new Argument}("x=5")$
- 17. Creating functions: Iteration: Function $f = \text{new Function}("f", "x+y", "x", "y")$
- 18. Creating functions: Iteration: Function $f = \text{new Function}("f(x,y)=x+y")$
- 19. Creating expressions: Iteration: Expression $e = \text{new Expression}("sin(2+(3*4)^2)/10")$
- 20. Creating expressions + checking syntax: Iteration: Expression $e = \text{new Expression}("sin(2+(3*4)^2)/10")$

Parameters:

threadsNum - Number of threads

Returns:

Number of tests that were not performed.

test000

```
static void test000(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - addition. Expression created once. Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test001

```
static void test001(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - multiplication. Expression created once. Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test002

```
static void test002(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - division. Expression created once. Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test003

```
static void test003(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - power. Expression created once. Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test004

```
static void test004(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - sinus. Expression created once. Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test005

```
static void test005(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - 2 additions. Expression created once. Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test006

```
static void test006(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - 3 additions. Expression created once. Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test007

```
static void test007(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - 3 additions + 1 parenthesis. Expression created once. Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test008

```
static void test008(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - 3 additions + 2 brackets. Expression created once.
Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test009

```
static void test009(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - 3 additions + 2 brackets. Expression created once.
Iteration: repeatedly recalculated same expression.

Parameters:

test -
testId -

test010

```
static void test010(PerformanceTestResult test,  
                  int testId)
```

Performance test: Combination of different operations. Expression created once. Iteration:
repeatedly recalculated same expression.

Parameters:

test -
testId -

test011

```
static void test011(PerformanceTestResult test,  
                  int testId)
```

Performance test: Simple calculations - addition with argument. Expression created once,
containing argument 'x'. Iteration: argument value is being modified (increased), then expression is
recalculated

Parameters:

test -
testId -

test012

```
static void test012(PerformanceTestResult test,  
                  int testId)
```

Performance test: User defined function $f(x,y)=3x+4y$. Expression &Function created once, containing argument 'x'. Iteration: argument value is being modified (increased), then expression is recalculated

Parameters:

test -
testId -

test013

```
static void test013(PerformanceTestResult test,  
                  int testId)
```

Performance test: Creating constants: Iteration: Constant c = new Constant("c", 5)

Parameters:

test -
testId -

test014

```
static void test014(PerformanceTestResult test,  
                  int testId)
```

Performance test: Creating constants: Iteration: Constant c = new Constant("c=5")

Parameters:

test -
testId -

test015

```
static void test015(PerformanceTestResult test,  
                  int testId)
```

Performance test: Creating arguments: Iteration: Argument x = new Argument("x", 5)

Parameters:

test -
testId -

test016

```
static void test016(PerformanceTestResult test,  
                  int testId)
```

Performance test: Creating arguments: Iteration: Argument x = new Argument("x", 5)

Parameters:

test -
testId -

test017

```
static void test017(PerformanceTestResult test,  
                  int testId)
```

Performance test: Creating functions: Iteration: Function f = new Function("f", "x+y", "x", "y")

Parameters:

test -
testId -

test018

```
static void test018(PerformanceTestResult test,  
                  int testId)
```

Performance test: Creating functions: Iteration: Function f = new Function("f(x,y)=x+y")

Parameters:

test -
testId -

test019

```
static void test019(PerformanceTestResult test,  
                  int testId)
```

Performance test: Creating expressions: Iteration: Expression e = new Expression("sin(2+(3*4)^2)/10")

Parameters:

test -
testId -

test020

```
static void test020(PerformanceTestResult test,  
                  int testId)
```

Performance test: Creating expressions + checking syntax: Iteration: Expression e = new Expression("sin(2+(3*4)^2)/10")

Parameters:

test -
testId -

org.mariuszgromada.math.mxparser.regressiontesting

Class RegTestExpression

```
java.lang.Object  
|  
|--org.mariuszgromada.math.mxparser.regressiontesting.RegTestExpression
```

< [Constructors](#) > < [Methods](#) >

```
public class RegTestExpression  
extends java.lang.Object
```

RegTestExpression - regression tests for the expression calculation

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Expression

Constructors

RegTestExpression

```
public RegTestExpression()
```

Methods

main

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

Runs main regression tests in the field of calculation.

Parameters:

args - no parameters are being considered

start

```
public static int start()
```

Runs main regression tests in the field of calculation.

Returns:

Number of tests with error result.

org.mariuszgromada.math.mxparser.regressiontesting

Class RegTestExpressionAPI

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.regressiontesting.RegTestExpressionAPI
```

< [Constructors](#) > < [Methods](#) >

```
public class RegTestExpressionAPI
extends java.lang.Object
```

RegTestExpressionAPI - regression tests for the expression API

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Expression

Constructors

RegTestExpressionAPI

```
public RegTestExpressionAPI()
```

Methods

main

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

Runs API regression tests.

Parameters:

args - Not used.

start

```
public static int start()
```

Runs API regression tests.

Returns:

Number of tests with error result.

org.mariuszgromada.math.mxparser.regressiontesting

Class RegTestSyntax

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.regressiontesting.RegTestSyntax
```

< [Constructors](#) > < [Methods](#) >

```
public class RegTestSyntax
extends java.lang.Object
```

RegTestSyntax - regression tests for the expression syntax checking

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](#)

[mXparser on SourceForge](#)

[mXparser on Bitbucket](#)

[mXparser on CodePlex](#)

[Janet Sudoku - project web page](#)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

4.1.0

Expression

Constructors

RegTestSyntax

```
public RegTestSyntax()
```

Methods

main

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

Runs syntax checking regression test.

Parameters:

args - no parameters are being considered

start

```
public static int start()
```

Runs syntax checking regression test.

Returns:

Number of errors.

org.mariuszgromada.math.mxparser.regressiontesting

Class RunTest

```
java.lang.Object
|
+--org.mariuszgromada.math.mxparser.regressiontesting.RunTest
```

< [Constructors](#) > < [Methods](#) >

```
public class RunTest
extends java.lang.Object
```

Use this class to run one of the following test

- Param: reg - Expression regression test
- Param: api - mXparser API test
- Param: syn - Syntax checking test
- Param: perf - Performance test

Author:

Mariusz Gromada

mariuszgromada.org@gmail.com

MathSpace.pl

MathParser.org - mXparser project page

[mXparser on GitHub](https://github.com/mariuszgromada/mXparser)

[mXparser on SourceForge](https://sourceforge.net/projects/mxparser/)

[mXparser on Bitbucket](https://bitbucket.org/mariuszgromada/mxparser/)

[mXparser on CodePlex](https://codeplex.com/mxparser/)

[Janet Sudoku - project web page](http://janet.sudoku.pl)

[Janet Sudoku on GitHub](#)

[Janet Sudoku on CodePlex](#)

[Janet Sudoku on SourceForge](#)

[Janet Sudoku on BitBucket](#)

Version:

3.0.0

Constructors

RunTest

```
public RunTest()
```

Methods

main

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

Use this class to run one of the following test

- Param: reg - Expression regression test
- Param: api - mXparser API test
- Param: syn - Syntax checking test
- Param: perf - Performance test

,

Parameters:

args - reg - Expression regression test, api - mXparser API test Param: syn - Syntax checking test, perf - Performance test

start

```
public static int start(java.lang.String[] args)
```

Use this class to run one of the following test

- Param: reg - Expression regression test
- Param: api - mXparser API test
- Param: syn - Syntax checking test
- Param: perf - Performance test

,

Parameters:

args - reg - Expression regression test, api - mXparser API test Param: syn - Syntax checking test, perf - Performance test

Returns:

Number of tests with error result.

org.mariuszgromada.math.mxparser.regressiontesting

Class Test011Thread

```
java.lang.Object
|
+--TestThread
|
+--org.mariuszgromada.math.mxparser.regressiontesting.Test011Thread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

class **Test011Thread**
extends [TestThread](#)

Multithreading implementation of class for Performance test: Simple calculations - addition with argument. Expression created once, containing argument 'x'. Iteration: argument value is being modified (increased), then expression is recalculated

Constructors

Test011Thread

Test011Thread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class Test012Thread

```
java.lang.Object
|
+--TestThread
    |
    +--org.mariuszgromada.math.mxparser.regressiontesting.Test012Thread
```

All Implemented Interfaces:

java.lang Runnable

< [Constructors](#) > < [Methods](#) >

class **Test012Thread**
 extends [TestThread](#)

Multithreading implementation of class for Performance test: User defined function $f(x,y)=3x+4y$. Expression &Function created once, containing argument 'x'. Iteration: argument value is being modified (increased), then expression is recalculated

Constructors

Test012Thread

Test012Thread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class Test013Thread

```
java.lang.Object
|
+--TestThread
|
+--org.mariuszgromada.math.mxparser.regressiontesting.Test013Thread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

class **Test013Thread**
extends [TestThread](#)

Multithreading implementation of class for Performance test: Creating constants: Iteration: Constant c = new Constant("c", 5)

Constructors

Test013Thread

Test013Thread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class Test014Thread

```
java.lang.Object
|
+--TestThread
|
+--org.mariuszgromada.math.mxparser.regressiontesting.Test014Thread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

```
class Test014Thread
extends TestThread
```

Multithreading implementation of class for Performance test: Creating constants: Iteration: Constant c = new Constant("c=5")

Constructors

Test014Thread

```
Test014Thread(PerformanceTestResult test)
```

Methods

testScenario

```
protected void testScenario()
```

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class Test015Thread

```
java.lang.Object
|
+--TestThread
|
+--org.mariuszgromada.math.mxparser.regressiontesting.Test015Thread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

class **Test015Thread**
extends [TestThread](#)

Multithreading implementation of class for Performance test: Creating arguments: Iteration: Argument x = new Argument("x", 5)

Constructors

Test015Thread

Test015Thread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class Test016Thread

```
java.lang.Object
|
+--TestThread
|
+--org.mariuszgromada.math.mxparser.regressiontesting.Test016Thread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

class **Test016Thread**
extends [TestThread](#)

Multithreading implementation of class for Performance test: Creating arguments: Iteration: Argument x = new Argument("x = 5")

Constructors

Test016Thread

Test016Thread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class Test017Thread

```
java.lang.Object
|
+--TestThread
|
+--org.mariuszgromada.math.mxparser.regressiontesting.Test017Thread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

class **Test017Thread**
 extends [TestThread](#)

Multithreading implementation of class for Performance test: Creating functions: Iteration: Function f = new Function("f", "x+y", "x", "y")

Constructors

Test017Thread

Test017Thread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class Test018Thread

```
java.lang.Object
|
+--TestThread
    |
    +--org.mariuszgromada.math.mxparser.regressiontesting.Test018Thread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

class **Test018Thread**
extends [TestThread](#)

Multithreading implementation of class for Performance test: Creating functions: Iteration: Function f = new Function("f(x,y)=x+y")

Constructors

Test018Thread

Test018Thread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class Test019Thread

```
java.lang.Object
|
+--TestThread
|
+--org.mariuszgromada.math.mxparser.regressiontesting.Test019Thread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

class **Test019Thread**
extends [TestThread](#)

Multithreading implementation of class for Performance test: Creating expressions: Iteration: Expression e
= new Expression("sin(2+(3*4)^2)/10")

Constructors

Test019Thread

Test019Thread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class Test020Thread

```
java.lang.Object
|
+--TestThread
    |
    +--org.mariuszgromada.math.mxparser.regressiontesting.Test020Thread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

class **Test020Thread**
 extends [TestThread](#)

Multithreading implementation of class for Performance test: Creating expressions: Expression e = new Expression("") Iteration: e.setExpressionString(sin(2+(3*4)^2)/10); e.checkSyntax();

Constructors

Test020Thread

Test020Thread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class TestSimpleCalcThread

```
java.lang.Object
|
+--TestThread
|
+--org.mariuszgromada.math.mxparser.regressiontesting.TestSimpleCalcThread
```

All Implemented Interfaces:

java.lang.Runnable

< [Constructors](#) > < [Methods](#) >

class **TestSimpleCalcThread**
extends [TestThread](#)

Multithreading implementation of class for Performance test: Simple calculations

Constructors

TestSimpleCalcThread

TestSimpleCalcThread([PerformanceTestResult](#) test)

Methods

testScenario

protected void **testScenario**()

Overrides:

[testScenario](#) in class [TestThread](#)

org.mariuszgromada.math.mxparser.regressiontesting

Class TestThread

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.regressiontesting.TestThread
```

All Implemented Interfaces:

java.lang.Runnable

Direct Known Subclasses:

[Test011Thread](#), [Test012Thread](#), [Test013Thread](#), [Test014Thread](#), [Test015Thread](#),
[Test016Thread](#), [Test017Thread](#), [Test018Thread](#), [Test019Thread](#), [Test020Thread](#),
[TestSimpleCalcThread](#)

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

abstract class **TestThread**

extends java.lang.Object

implements java.lang.Runnable

Multithreading abstract implementation of test

Fields

iterNum

protected int **iterNum**

Number of iterations for a single thread.

test

protected [PerformanceTestResult](#) **test**

Test parameters.

Constructors

TestThread

TestThread([PerformanceTestResult](#) test)

Default constructor - creates parameters for a single test thread

Parameters:

test - Test parameters

threadsNum - Number of threads

Methods

run

```
public void run()
```

Test scenario execution

testScenario

```
protected abstract void testScenario()
```

Test scenario implementation

Package

org.mariuszgromada.math.mxparser.syntaxchecker

Interface Summary

[SyntaxCheckerConstants](#)

Token literal values and constants.

Class Summary

[ParseException](#)

This exception is thrown when parse errors are encountered.

[SimpleCharStream](#)

An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (without unicode processing).

[SyntaxChecker](#)

[SyntaxCheckerTokenManager](#)

Token Manager.

[Token](#)

Describes the input token stream.

[TokenMgrError](#)

Token Manager Error.

org.mariuszgromada.math.mxparser.syntaxchecker

Class ParseException

```
java.lang.Object
|
+-- java.lang.Throwable
|   |
|   +-- java.lang.Exception
|       |
|       +-- org.mariuszgromada.math.mxparser.syntaxchecker.ParseException
```

All Implemented Interfaces:

java.io.Serializable

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class ParseException
extends java.lang.Exception
```

This exception is thrown when parse errors are encountered. You can explicitly create objects of this exception type by calling the method generateParseException in the generated parser. You can modify

this class to customize your error reporting mechanisms so long as you retain the public fields.

Fields

currentToken

```
public Token currentToken
```

This is the last token that has been consumed successfully. If this object has been created due to a parse error, the token following this token will (therefore) be the first error token.

eol

```
protected java.lang.String eol
```

The end of line string for this machine.

expectedTokenSequences

```
public int[][] expectedTokenSequences
```

Each entry in this array is an array of integers. Each array of integers represents a sequence of tokens (by their ordinal values) that is expected at this point of the parse.

tokenImage

```
public java.lang.String[] tokenImage
```

This is a reference to the "tokenImage" array of the generated parser within which the parse error occurred. This array is defined in the generated ...Constants interface.

Constructors

ParseException

```
public ParseException()
```

The following constructors are for use by you for whatever purpose you can think of. Constructing the exception in this manner makes the exception behave in the normal way - i.e., as documented in the class "Throwable". The fields "errorToken", "expectedTokenSequences", and "tokenImage" do not contain relevant information. The JavaCC generated code does not use these constructors.

ParseException

```
public ParseException(java.lang.String message)
```

Constructor with message.

ParseException

```
public ParseException(Token currentTokenVal,  
                    int[][] expectedTokenSequencesVal,  
                    java.lang.String[] tokenImageVal)
```

This constructor is used by the method "generateParseException" in the generated parser. Calling this constructor generates a new object of this type with the fields "currentToken", "expectedTokenSequences", and "tokenImage" set.

Methods

add_escapes

```
static java.lang.String add_escapes(java.lang.String str)
```

Used to convert raw characters to their escaped version when these raw version cannot be used as part of an ASCII string literal.

org.mariuszgromada.math.mxparser.syntaxchecker

Class SimpleCharStream

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxparser.syntaxchecker.SimpleCharStream
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class SimpleCharStream  
extends java.lang.Object
```

An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (without unicode processing).

Fields

available

```
int available
```

bufcolumn

```
protected int[] bufcolumn
```


buffer

```
protected char[] buffer
```

bufline

```
protected int[] bufline
```

bufpos

```
public int bufpos  
    Position in buffer.
```

bufsize

```
int bufsize
```

column

```
protected int column
```

inBuf

```
protected int inBuf
```

inputStream

```
protected java.io.Reader inputStream
```

line

```
protected int line
```

maxNextCharInd

```
protected int maxNextCharInd
```

prevCharIsLF

staticFlag

tabSize

tokenBegin

trackLineColumn

Constructors

SimpleCharStream

SimpleCharStream

Constructor.

SimpleCharStream

```
public SimpleCharStream(java.io.InputStream dstream,  
                        int startline,  
                        int startcolumn,  
                        int buffersize)
```

Constructor.

SimpleCharStream

```
public SimpleCharStream(java.io.InputStream dstream,  
                        java.lang.String encoding)
```

Constructor.

SimpleCharStream

```
public SimpleCharStream(java.io.InputStream dstream,  
                        java.lang.String encoding,  
                        int startline,  
                        int startcolumn)
```

Constructor.

SimpleCharStream

```
public SimpleCharStream(java.io.InputStream dstream,  
                        java.lang.String encoding,  
                        int startline,  
                        int startcolumn,  
                        int buffersize)
```

Constructor.

SimpleCharStream

```
public SimpleCharStream(java.io.Reader dstream)
```

Constructor.

SimpleCharStream

```
public SimpleCharStream(java.io.Reader dstream,  
                        int startline,  
                        int startcolumn)
```

Constructor.

SimpleCharStream

```
public SimpleCharStream(java.io.Reader dstream,  
                        int startline,  
                        int startcolumn,  
                        int buffersize)
```

Constructor.

Methods

BeginToken

```
public char BeginToken()
```

Start.

Done

```
public void Done()
```

Reset buffer when finished.

ExpandBuff

```
protected void ExpandBuff(boolean wrapAround)
```

FillBuff

```
protected void FillBuff()
```

GetImage

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

Get token literal value.

GetSuffix

```
public char[] GetSuffix(int len)
```

Get the suffix.

ReInit

```
public void ReInit(java.io.InputStream dstream)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.InputStream dstream,  
                  int startline,  
                  int startcolumn)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.InputStream dstream,  
                  int startline,  
                  int startcolumn,  
                  int buffersize)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.InputStream dstream,  
                  java.lang.String encoding)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.InputStream dstream,  
                  java.lang.String encoding,  
                  int startline,  
                  int startcolumn)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.InputStream dstream,  
                  java.lang.String encoding,  
                  int startline,  
                  int startcolumn,  
                  int buffersize)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.Reader dstream)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.Reader dstream,  
                  int startline,  
                  int startcolumn)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.Reader dstream,  
                  int startline,  
                  int startcolumn,  
                  int buffersize)
```

Reinitialise.

UpdateLineColumn

```
protected void UpdateLineColumn(char c)
```

adjustBeginLineColumn

```
public void adjustBeginLineColumn(int newLine,  
                                   int newCol)
```

Method to adjust line and column numbers for the start of a token.

backup

```
public void backup(int amount)
```

Backup a number of characters.

getBeginColumn

```
public int getBeginColumn()
```

Get token beginning column number.

getBeginLine

```
public int getBeginLine()
```

Get token beginning line number.

getColumn

```
public int getColumn()
```

getEndColumn

```
public int getEndColumn()
```

Get token end column number.

getEndLine

```
public int getEndLine()
```

Get token end line number.

getLine

```
public int getLine()
```

getTabSize

```
public int getTabSize()
```

getTrackLineColumn

```
boolean getTrackLineColumn()
```

readChar

```
public char readChar()
```

Read a character.

setTabSize

```
public void setTabSize(int i)
```

setTrackLineColumn

```
void setTrackLineColumn(boolean tlc)
```

org.mariuszgromada.math.mxparser.syntaxchecker

Class SyntaxChecker

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.syntaxchecker.SyntaxChecker
```

All Implemented Interfaces:

[SyntaxCheckerConstants](#)

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

public final class **SyntaxChecker**
extends java.lang.Object
implements [SyntaxCheckerConstants](#)

Fields

jj_input_stream

[SimpleCharStream](#) jj_input_stream

jj_nt

public [Token](#) jj_nt
Next token.

token

public [Token](#) token
Current token.

token_source

public [SyntaxCheckerTokenManager](#) token_source
Generated Token Manager.

Constructors

SyntaxChecker

```
public SyntaxChecker(java.io.InputStream stream)
```

Constructor with InputStream.

SyntaxChecker

```
public SyntaxChecker(java.io.InputStream stream,  
                     java.lang.String encoding)
```

Constructor with InputStream and supplied encoding

SyntaxChecker

```
public SyntaxChecker(java.io.Reader stream)
```

Constructor.

SyntaxChecker

```
public SyntaxChecker(SyntaxCheckerTokenManager tm)
```

Constructor with generated Token Manager.

Methods

ReInit

```
public void ReInit(java.io.InputStream stream)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.InputStream stream,  
                  java.lang.String encoding)
```

Reinitialise.

ReInit

```
public void ReInit(java.io.Reader stream)
    Reinitialise.
```

ReInit

```
public void ReInit(SyntaxCheckerTokenManager tm)
    Reinitialise.
```

argumentList

```
public final void argumentList()
```

binaryExpression

```
public final void binaryExpression()
```

checkSyntax

```
public final void checkSyntax()
```

disable_tracing

```
public final void disable_tracing()
    Disable tracing.
```

enable_tracing

```
public final void enable_tracing()
    Enable tracing.
```

expression

```
public final void expression()
```

generateParseException

```
public ParseException generateParseException()
```

Generate ParseException.

getNextToken

```
public final Token getNextToken()
```

Get the next Token.

getToken

```
public final Token getToken(int index)
```

Get the specific Token.

identifier

```
public final void identifier()
```

itemExpression

```
public final void itemExpression()
```

start

```
public final void start()
```

unaryLeftExpression

```
public final void unaryLeftExpression()
```

unaryRigthExpression

```
public final void unaryRigthExpression()
```

org.mariuszgromada.math.mxparser.syntaxchecker

Interface SyntaxCheckerConstants

< [Fields](#) >

```
public interface SyntaxCheckerConstants
```

Token literal values and constants. Generated by org.javacc.parser.OtherFilesGen#start()

Fields

AND

```
public static final int AND  
    RegularExpression Id.
```

BITNOT

```
public static final int BITNOT  
    RegularExpression Id.
```

BITWISE

```
public static final int BITWISE  
    RegularExpression Id.
```

CHAR

```
public static final int CHAR  
    RegularExpression Id.
```

CIMP

```
public static final int CIMP
    RegularExpression Id.
```

CNIMP

```
public static final int CNIMP
    RegularExpression Id.
```

COMMA

```
public static final int COMMA
    RegularExpression Id.
```

DEFAULT

```
public static final int DEFAULT
    Lexical state.
```

DIGIT

```
public static final int DIGIT
    RegularExpression Id.
```

DIV

```
public static final int DIV
    RegularExpression Id.
```

EOF

```
public static final int EOF
    End of File.
```

EQ

```
public static final int EQ
    RegularExpression Id.
```

EQV

```
public static final int EQV  
    RegularExpression Id.
```

FACTORIAL

```
public static final int FACTORIAL  
    RegularExpression Id.
```

FUNCTION

```
public static final int FUNCTION  
    RegularExpression Id.
```

GEQ

```
public static final int GEQ  
    RegularExpression Id.
```

GT

```
public static final int GT  
    RegularExpression Id.
```

IDENTIFIER

```
public static final int IDENTIFIER  
    RegularExpression Id.
```

IMP

```
public static final int IMP  
    RegularExpression Id.
```

INTEGER

```
public static final int INTEGER  
    RegularExpression Id.
```

INVALID_TOKEN

```
public static final int INVALID_TOKEN
    RegularExpression Id.
```

LEFT_PAR

```
public static final int LEFT_PAR
    RegularExpression Id.
```

LEQ

```
public static final int LEQ
    RegularExpression Id.
```

LETTER

```
public static final int LETTER
    RegularExpression Id.
```

LETTERS

```
public static final int LETTERS
    RegularExpression Id.
```

LT

```
public static final int LT
    RegularExpression Id.
```

MINUS

```
public static final int MINUS
    RegularExpression Id.
```

MODULO

```
public static final int MODULO
    RegularExpression Id.
```

MULTIPLY

```
public static final int MULTIPLY
    RegularExpression Id.
```

NAND

```
public static final int NAND
    RegularExpression Id.
```

NEQ

```
public static final int NEQ
    RegularExpression Id.
```

NIMP

```
public static final int NIMP
    RegularExpression Id.
```

NOR

```
public static final int NOR
    RegularExpression Id.
```

NOT

```
public static final int NOT
    RegularExpression Id.
```

NUMBER

```
public static final int NUMBER
    RegularExpression Id.
```

NUMBER_CONSTANT

```
public static final int NUMBER_CONSTANT
    RegularExpression Id.
```

OR

```
public static final int OR
    RegularExpression Id.
```

PERCENTAGE

```
public static final int PERCENTAGE
    RegularExpression Id.
```

PLUS

```
public static final int PLUS
    RegularExpression Id.
```

POWER

```
public static final int POWER
    RegularExpression Id.
```

REAL

```
public static final int REAL
    RegularExpression Id.
```

RIGHT_PAR

```
public static final int RIGHT_PAR
    RegularExpression Id.
```

SEMICOLON

```
public static final int SEMICOLON
    RegularExpression Id.
```

UNEXPECTED_CHAR

```
public static final int UNEXPECTED_CHAR
    RegularExpression Id.
```

UNIT

```
public static final int UNIT
    RegularExpression Id.
```

XOR

```
public static final int XOR
    RegularExpression Id.
```

tokenImage

```
public static final java.lang.String[] tokenImage
    Literal token values.
```

org.mariuszgromada.math.mxparser.syntaxchecker

Class SyntaxCheckerTokenManager

```
java.lang.Object
|
|--org.mariuszgromada.math.mxparser.syntaxchecker.SyntaxCheckerTokenManager
```

All Implemented Interfaces:

[SyntaxCheckerConstants](#)

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class SyntaxCheckerTokenManager
    extends java.lang.Object
    implements SyntaxCheckerConstants
```

Token Manager.

Fields

curChar

```
protected char curChar
```

curLexState

```
int curLexState
```

debugStream

```
public java.io.PrintStream debugStream  
    Debug output.
```

defaultLexState

```
int defaultLexState
```

input_stream

```
protected SimpleCharStream input_stream
```

jjmatchedKind

```
int jjmatchedKind
```

jjmatchedPos

```
int jjmatchedPos
```

jjnewStateCnt

```
int jjnewStateCnt
```

jjnextStates

```
static final int[] jjnextStates
```

jjround

```
int jjround
```

jjstrLiteralImages

```
public static final java.lang.String[] jjstrLiteralImages  
    Token literal values.
```

jjtoSkip

```
static final long[] jjtoSkip
```

jjtoToken

```
static final long[] jjtoToken
```

lexStateNames

```
public static final java.lang.String[] lexStateNames  
    Lexer state names.
```

Constructors

SyntaxCheckerTokenManager

```
public SyntaxCheckerTokenManager(SimpleCharStream stream)  
    Constructor.
```

SyntaxCheckerTokenManager

```
public SyntaxCheckerTokenManager(SimpleCharStream stream,  
                                int lexState)  
    Constructor.
```

Methods

ReInit

```
public void ReInit(SimpleCharStream stream)  
    Reinitialise parser.
```

ReInit

```
public void ReInit(SimpleCharStream stream,  
                  int lexState)  
    Reinitialise parser.
```

SwitchTo

```
public void SwitchTo(int lexState)
```

Switch to specified lex state.

getNextToken

```
public Token getNextToken()
```

Get the next Token.

jjFillToken

```
protected Token jjFillToken()
```

setDebugStream

```
public void setDebugStream(java.io.PrintStream ds)
```

Set debug output.

org.mariuszgromada.math.mxpaser.syntaxchecker

Class Token

```
java.lang.Object  
|  
+--org.mariuszgromada.math.mxpaser.syntaxchecker.Token
```

All Implemented Interfaces:

java.io.Serializable

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class Token  
extends java.lang.Object  
implements java.io.Serializable
```

Describes the input token stream.

Fields

beginColumn

`public int beginColumn`
The column number of the first character of this Token.

beginLine

`public int beginLine`
The line number of the first character of this Token.

endColumn

`public int endColumn`
The column number of the last character of this Token.

endLine

`public int endLine`
The line number of the last character of this Token.

image

`public java.lang.String image`
The string image of the token.

kind

`public int kind`
An integer that describes the kind of this token. This numbering system is determined by JavaCCParser, and a table of these numbers is stored in the file ...Constants.java.

next

`public Token next`
A reference to the next regular (non-special) token from the input stream. If this is the last token from the input stream, or if the token manager has not read tokens beyond this one, this field is set to null. This is true only if this token is also a regular token. Otherwise, see below for a description of the contents of this field.

specialToken

public [Token](#) **specialToken**

This field is used to access special tokens that occur prior to this token, but after the immediately preceding regular (non-special) token. If there are no such special tokens, this field is set to null. When there are more than one such special token, this field refers to the last of these special tokens, which in turn refers to the next previous special token through its `specialToken` field, and so on until the first special token (whose `specialToken` field is null). The next fields of special tokens refer to other special tokens that immediately follow it (without an intervening regular token). If there is no such token, this field is null.

Constructors

Token

public **Token**()

No-argument constructor

Token

public **Token**(int kind)

Constructs a new token for the specified Image.

Token

public **Token**(int kind,
java.lang.String image)

Constructs a new token for the specified Image and Kind.

Methods

getValue

public java.lang.Object **getValue**()

An optional attribute value of the Token. Tokens which are not used as syntactic sugar will often contain meaningful values that will be used later on by the compiler or interpreter. This attribute value is often different from the image. Any subclass of Token that actually wants to return a non-null value can override this method as appropriate.

newToken

public static [Token](#) **newToken**(int ofKind)

newToken

```
public static Token newToken(int ofKind,
                               java.lang.String image)
```

Returns a new Token object, by default. However, if you want, you can create and return subclass objects based on the value of ofKind. Simply add the cases to the switch for all those special cases. For example, if you have a subclass of Token called IDToken that you want to create if ofKind is ID, simply add something like : case MyParserConstants.ID : return new IDToken(ofKind, image); to the following switch statement. Then you can cast matchedToken variable to the appropriate type and use it in your lexical actions.

toString

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

Returns the image.

Overrides:

toString in class java.lang.Object

org.mariuszgromada.math.mxparser.syntaxchecker

Class TokenMgrError

```
java.lang.Object
|
+-- java.lang.Throwable
|   |
|   +-- java.lang.Error
|       |
|       +-- org.mariuszgromada.math.mxparser.syntaxchecker.TokenMgrError
```

All Implemented Interfaces:

java.io.Serializable

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class TokenMgrError
extends java.lang.Error
```

Token Manager Error.

Fields

INVALID_LEXICAL_STATE

```
static final int INVALID_LEXICAL_STATE
    Tried to change to an invalid lexical state.
```

LEXICAL_ERROR

```
static final int LEXICAL_ERROR
    Lexical error occurred.
```

LOOP_DETECTED

```
static final int LOOP_DETECTED
    Detected (and bailed out of) an infinite loop in the token manager.
```

STATIC_LEXER_ERROR

```
static final int STATIC_LEXER_ERROR
    An attempt was made to create a second instance of a static token manager.
```

errorCode

```
int errorCode
    Indicates the reason why the exception is thrown. It will have one of the above 4 values.
```

Constructors

TokenMgrError

```
public TokenMgrError()
    No arg constructor.
```

TokenMgrError

```
public TokenMgrError(boolean EOFSeen,
    int lexState,
    int errorLine,
    int errorColumn,
    java.lang.String errorAfter,
    char curChar,
    int reason)
```

Full Constructor.

TokenMgrError

```
public TokenMgrError(java.lang.String message,
                    int reason)
```

Constructor with message and reason.

Methods

LexicalError

```
protected static java.lang.String LexicalError(boolean EOFSeen,
                                              int lexState,
                                              int errorLine,
                                              int errorColumn,
                                              java.lang.String errorAfter,
                                              char curChar)
```

Returns a detailed message for the Error when it is thrown by the token manager to indicate a lexical error. Parameters : EOFSeen : indicates if EOF caused the lexical error curLexState : lexical state in which this error occurred errorLine : line number when the error occurred errorColumn : column number when the error occurred errorAfter : prefix that was seen before this error occurred curchar : the offending character Note: You can customize the lexical error message by modifying this method.

addEscapes

```
protected static final java.lang.String addEscapes(java.lang.String str)
```

Replaces unprintable characters by their escaped (or unicode escaped) equivalents in the given string

getMessage

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

You can also modify the body of this method to customize your error messages. For example, cases like LOOP_DETECTED and INVALID_LEXICAL_STATE are not of end-users concern, so you can return something like : "Internal Error : Please file a bug report " from this method for such cases in the release version of your parser.

Overrides:

getMessage in class java.lang.Throwable

3. Repositorio

- Para el control de versiones de nuestro proyecto mantenemos un Repositorio en Gitlab.

4. Conclusiones

Como resultado de la investigación sobre el Cálculo infinitesimal y de las bibliotecas externas "mXparser" "JFreeChart", en este trabajo se refleja el conocimiento obtenido acerca de calcular la integral definida, la derivada o un límite en un punto arbitrario de una función con una variable y el funcionamiento de dichas bibliotecas externas para la creación del programa, todo esto utilizando Programación Orientada a Objetos donde se crearon varias clases que trabajan en conjunto para el correcto funcionamiento tanto de la lógica como de la interfaz gráfica del programa.