

# DedSec Calculator

1.0

Generated by Doxygen 1.8.11

## Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>1</b>
2.1	Class Hierarchy . . . . .	1
<b>3</b>	<b>Class Index</b>	<b>1</b>
3.1	Class List . . . . .	1
<b>4</b>	<b>File Index</b>	<b>2</b>
4.1	File List . . . . .	2
<b>5</b>	<b>Namespace Documentation</b>	<b>2</b>
5.1	Package calculatorfx . . . . .	2
5.1.1	Detailed Description . . . . .	2
5.2	Package test . . . . .	3
<b>6</b>	<b>Class Documentation</b>	<b>3</b>
6.1	calculatorfx.Controller Class Reference . . . . .	3
6.1.1	Detailed Description . . . . .	4
6.1.2	Member Function Documentation . . . . .	4
6.1.3	Member Data Documentation . . . . .	5
6.2	calculatorfx.IMath Interface Reference . . . . .	9
6.2.1	Detailed Description . . . . .	9
6.2.2	Member Function Documentation . . . . .	9
6.3	calculatorfx.MainCalculatorFX Class Reference . . . . .	10
6.3.1	Detailed Description . . . . .	11
6.3.2	Member Function Documentation . . . . .	11
6.4	calculatorfx.Math Class Reference . . . . .	11
6.4.1	Detailed Description . . . . .	12
6.4.2	Member Function Documentation . . . . .	12
6.5	test.MathTest Class Reference . . . . .	16
6.5.1	Detailed Description . . . . .	17
6.5.2	Constructor & Destructor Documentation . . . . .	17
6.5.3	Member Function Documentation . . . . .	17
6.5.4	Member Data Documentation . . . . .	20
6.6	calculatorfx.StandartDeviation Class Reference . . . . .	20
6.6.1	Detailed Description . . . . .	21
6.6.2	Member Function Documentation . . . . .	21

7 File Documentation	21
7.1 src/calculatorfx/Controller.java File Reference	21
7.1.1 Detailed Description	22
7.2 src/calculatorfx/IMath.java File Reference	22
7.3 src/calculatorfx/MainCalculatorFX.java File Reference	22
7.3.1 Detailed Description	23
7.4 src/calculatorfx/Math.java File Reference	23
7.4.1 Detailed Description	23
7.5 src/StandartDeviation.java File Reference	23
7.5.1 Detailed Description	24
7.6 src/styles/Style.css File Reference	24
7.7 src/test/MathTest.java File Reference	24
7.7.1 Detailed Description	24

# 1 Namespace Index

## 1.1 Packages

Here are the packages with brief descriptions (if available):

<b>calculatorfx</b>	<b>2</b>
<b>test</b>	<b>3</b>

## 2 Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

<b>calculatorfx.IMath</b>	<b>9</b>
calculatorfx.Math	11
test.MathTest	16
calculatorfx.StandartDeviation Application	20
calculatorfx.MainCalculatorFX Initializable	10

<a href="#">calculatorfx.Controller</a>	<a href="#">3</a>
---	-------------------

## 3 Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">calculatorfx.Controller</a>		<a href="#">3</a>
The controller of the calculator		
<a href="#">calculatorfx.IMath</a>		<a href="#">9</a>
<a href="#">calculatorfx.MainCalculatorFX</a>		
Main class of the calculator launching the application		
<a href="#">calculatorfx.Math</a>		<a href="#">10</a>
Class containing the methods of the math library		
<a href="#">test.MathTest</a>		<a href="#">11</a>
Class containing the tests		
<a href="#">calculatorfx.StandartDeviation</a>		<a href="#">16</a>
Implementation of the SD, includes main		
		<a href="#">20</a>

## 4 File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">src/StandartDeviation.java</a>		<a href="#">23</a>
Program to calculate the SD		
<a href="#">src/calculatorfx/Controller.java</a>		<a href="#">21</a>
Implementation of a Controller for a java FXML application		
<a href="#">src/calculatorfx/IMath.java</a>		<a href="#">22</a>
<a href="#">src/calculatorfx/MainCalculatorFX.java</a>		<a href="#">22</a>
Implementation of a main class for a java FXML application		
<a href="#">src/calculatorfx/Math.java</a>		<a href="#">23</a>
Interface for a custom math library		
<a href="#">src/styles/Style.css</a>		<a href="#">24</a>
<a href="#">src/test/MathTest.java</a>		<a href="#">24</a>
Test the functionality of the custom math library		

## 5 Namespace Documentation

### 5.1 Package calculatorfx

#### Classes

- class [Controller](#)  
*The controller of the calculator.*
- interface [IMath](#)
- class [MainCalculatorFX](#)  
*Main class of the calculator launching the application.*
- class [Math](#)  
*Class containing the methods of the math library.*
- class [StandartDeviation](#)  
*Implementation of the SD, includes main.*

#### 5.1.1 Detailed Description

#### See also

[Math.java](#)

### 5.2 Package test

#### Classes

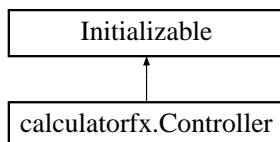
- class [MathTest](#)  
*Class containing the tests.*

## 6 Class Documentation

### 6.1 calculatorfx.Controller Class Reference

The controller of the calculator.

Inheritance diagram for calculatorfx.Controller:



### Public Member Functions

- int [getDigits](#) (double num)  
*Method getDigits.*
- void [displayResult](#) (double [result](#))  
*Method displayResult.*
- void [initialize](#) (URL url, ResourceBundle rb)

### Package Functions

- void [showHelp](#) (ActionEvent event)
- void [Calculation](#) (ActionEvent event) throws Exception

### Package Attributes

- double [result](#) = 0
- double [input1](#)
- double [input2](#)
- int [ps](#)
- int [point](#) = 0
- int [ispoint](#) = 0
- int [canneg](#) = 0
- int [emptydisplay](#) = 1
- String [oldvalue](#)
- [Math](#) [math](#) = new [Math](#)()
- DecimalFormat [td](#) = new DecimalFormat("0.#####")
- DecimalFormat [ed](#) = new DecimalFormat("0.#####E0")

### Private Attributes

- TextField [operation](#)
- TextField [display](#)
- Button [one](#)
- Button [two](#)
- Button [three](#)
- Button [four](#)
- Button [five](#)
- Button [six](#)
- Button [seven](#)
- Button [eight](#)
- Button [nine](#)
- Button [zero](#)
- Button [dot](#)
- Button [clear](#)
- Button [plus](#)
- Button [minus](#)
- Button [multiply](#)
- Button [divide](#)
- Button [fact](#)
- Button [power](#)
- Button [squareroot](#)
- Button [tobinary](#)
- Button [negate](#)
- Button [equal](#)

### 6.1.1 Detailed Description

The controller of the calculator.

Definition at line 30 of file Controller.java.

### 6.1.2 Member Function Documentation

#### 6.1.2.1 void calculatorfx.Controller.Calculation ( *ActionEvent event* ) throws Exception [package]

Definition at line 201 of file Controller.java.

#### 6.1.2.2 void calculatorfx.Controller.displayResult ( double *result* )

Method displayResult.

Method displayResult displays the result of mathematical operation onto display in decimal or scientific notation.

##### Parameters

<i>result</i>	is the number to be displayed in normal or scientific notation
---------------	--

< Number contains more than 16 digits and is displayed in scientific notation

< Number contains less or equal to 16 digits and is displayed in normal notation

Definition at line 173 of file Controller.java.

#### 6.1.2.3 int calculatorfx.Controller.getDigits ( double *num* )

Method getDigits.

Method getDigits counts the digits in number (decimal dot and minus included).

##### Parameters

<i>num</i>	method counts the digits in this number
------------	---

##### Returns

the number of digits (decimal dot and minus included)

< Number includes a minus

< Number has a decimal part thus includes a decimal dot

< Integer part is zero thus counting will be incorrect

< Some number needs to be added to integer part

< Number needs to be shifted left so the decimal part will be in integer part

< Number has max 8 decimal digits

< Number is shifted right and digits are counted

Definition at line 126 of file Controller.java.

#### 6.1.2.4 void calculatorfx.Controller.initialize ( URL url, ResourceBundle rb )

Definition at line 501 of file Controller.java.

#### 6.1.2.5 void calculatorfx.Controller.showHelp ((ActionEvent event) [package]

Definition at line 184 of file Controller.java.

### 6.1.3 Member Data Documentation

#### 6.1.3.1 int calculatorfx.Controller.canneg = 0 [package]

Flag which enables the negate operation (0 when enabled)

Definition at line 110 of file Controller.java.

#### 6.1.3.2 Button calculatorfx.Controller.clear [private]

Definition at line 72 of file Controller.java.

#### 6.1.3.3 TextField calculatorfx.Controller.display [private]

Definition at line 36 of file Controller.java.

#### 6.1.3.4 Button calculatorfx.Controller.divide [private]

Definition at line 84 of file Controller.java.

#### 6.1.3.5 Button calculatorfx.Controller.dot [private]

Definition at line 69 of file Controller.java.

#### 6.1.3.6 DecimalFormat calculatorfx.Controller.ed = new DecimalFormat("0.#####E0") [package]

Format for scientific notation of a number

Definition at line 117 of file Controller.java.

#### 6.1.3.7 Button calculatorfx.Controller.eight [private]

Definition at line 60 of file Controller.java.



**6.1.3.8** `int calculatorfx.Controller.emptydisplay = 1` [package]

Flag which indicates that a display is empty (1 when display is empty)

Definition at line 111 of file Controller.java.

**6.1.3.9** `Button calculatorfx.Controller.equal` [private]

Definition at line 102 of file Controller.java.

**6.1.3.10** `Button calculatorfx.Controller.fact` [private]

Definition at line 87 of file Controller.java.

**6.1.3.11** `Button calculatorfx.Controller.five` [private]

Definition at line 51 of file Controller.java.

**6.1.3.12** `Button calculatorfx.Controller.four` [private]

Definition at line 48 of file Controller.java.

**6.1.3.13** `double calculatorfx.Controller.input1` [package]

Definition at line 105 of file Controller.java.

**6.1.3.14** `double calculatorfx.Controller.input2` [package]

Definition at line 106 of file Controller.java.

**6.1.3.15** `int calculatorfx.Controller.ispoint = 0` [package]

Flag which indicates that a dot was typed or generated by operation (1 when true)

Definition at line 109 of file Controller.java.

**6.1.3.16** `Math calculatorfx.Controller.math = new Math()` [package]

Definition at line 115 of file Controller.java.

**6.1.3.17** `Button calculatorfx.Controller.minus` [private]

Definition at line 78 of file Controller.java.

**6.1.3.18** `Button calculatorfx.Controller.multiply` [private]

Definition at line 81 of file Controller.java.

**6.1.3.19 Button calculatorfx.Controller.negate** [private]

Definition at line 99 of file Controller.java.

**6.1.3.20 Button calculatorfx.Controller.nine** [private]

Definition at line 63 of file Controller.java.

**6.1.3.21 String calculatorfx.Controller.oldvalue** [package]

This String is used to store info about operation and input number. These are than displayed onto operation display.

Definition at line 113 of file Controller.java.

**6.1.3.22 Button calculatorfx.Controller.one** [private]

Definition at line 39 of file Controller.java.

**6.1.3.23 TextField calculatorfx.Controller.operation** [private]

Definition at line 33 of file Controller.java.

**6.1.3.24 Button calculatorfx.Controller.plus** [private]

Definition at line 75 of file Controller.java.

**6.1.3.25 int calculatorfx.Controller.point = 0** [package]

Flag which enables a dot to be typed (0 when enabled)

Definition at line 108 of file Controller.java.

**6.1.3.26 Button calculatorfx.Controller.power** [private]

Definition at line 90 of file Controller.java.

**6.1.3.27 int calculatorfx.Controller.ps** [package]

Number of operation is stored here (0 = clear, 1 = plus, 2 = minus, 3 = multiply, 4 = divide, 5 = fact, 6 = power, 7 = squareroot, 8 = tobinary)

Definition at line 107 of file Controller.java.

**6.1.3.28 double calculatorfx.Controller.result = 0** [package]

Definition at line 104 of file Controller.java.

**6.1.3.29 Button calculatorfx.Controller.seven** [private]

Definition at line 57 of file Controller.java.

**6.1.3.30** Button `calculatorfx.Controller.six` [private]

Definition at line 54 of file `Controller.java`.

**6.1.3.31** Button `calculatorfx.Controller.squareroot` [private]

Definition at line 93 of file `Controller.java`.

**6.1.3.32** DecimalFormat `calculatorfx.Controller.td = new DecimalFormat("0.#####")` [package]

Format for decimal notation of a number

Definition at line 116 of file `Controller.java`.

**6.1.3.33** Button `calculatorfx.Controller.three` [private]

Definition at line 45 of file `Controller.java`.

**6.1.3.34** Button `calculatorfx.Controller.tobinary` [private]

Definition at line 96 of file `Controller.java`.

**6.1.3.35** Button `calculatorfx.Controller.two` [private]

Definition at line 42 of file `Controller.java`.

**6.1.3.36** Button `calculatorfx.Controller.zero` [private]

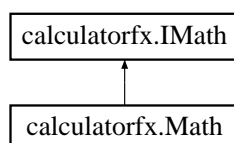
Definition at line 66 of file `Controller.java`.

The documentation for this class was generated from the following file:

- `src/calculatorfx/Controller.java`

**6.2** calculatorfx.IMath Interface Reference

Inheritance diagram for `calculatorfx.IMath`:



## Public Member Functions

- double [Sum](#) (double num1, double num2)
- double [Sub](#) (double minuend, double subtrahend)
- double [Div](#) (double dividend, double divisor) throws Exception
- double [Mult](#) (double num1, double num2)
- long [Fact](#) (long num) throws Exception
- double [Pow](#) (double basis, double exponent) throws Exception
- double [Root](#) (double n, double x) throws Exception
- long [toBinary](#) (long basis)
- double [Abs](#) (double nonAbs)

### 6.2.1 Detailed Description

Definition at line 12 of file IMath.java.

### 6.2.2 Member Function Documentation

#### 6.2.2.1 double calculatorfx.IMath.Abs ( double *nonAbs* )

Implemented in [calculatorfx.Math](#).

#### 6.2.2.2 double calculatorfx.IMath.Div ( double *divident*, double *divisor* ) throws Exception

Implemented in [calculatorfx.Math](#).

#### 6.2.2.3 long calculatorfx.IMath.Fact ( long *num* ) throws Exception

Implemented in [calculatorfx.Math](#).

#### 6.2.2.4 double calculatorfx.IMath.Mult ( double *num1*, double *num2* )

Implemented in [calculatorfx.Math](#).

#### 6.2.2.5 double calculatorfx.IMath.Pow ( double *basis*, double *exponent* ) throws Exception

Implemented in [calculatorfx.Math](#).

#### 6.2.2.6 double calculatorfx.IMath.Root ( double *n*, double *x* ) throws Exception

Implemented in [calculatorfx.Math](#).

#### 6.2.2.7 double calculatorfx.IMath.Sub ( double *minuend*, double *subtrahend* )

Implemented in [calculatorfx.Math](#).

#### 6.2.2.8 double calculatorfx.IMath.Sum ( double *num1*, double *num2* )

Implemented in [calculatorfx.Math](#).

6.2.2.9 long calculatorfx.IMath.toBinary ( long *basis* )

Implemented in [calculatorfx.Math](#).

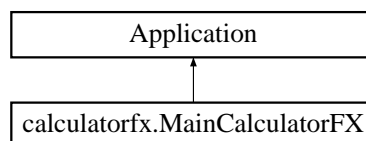
The documentation for this interface was generated from the following file:

- [src/calculatorfx/IMath.java](#)

## 6.3 calculatorfx.MainCalculatorFX Class Reference

Main class of the calculator launching the application.

Inheritance diagram for calculatorfx.MainCalculatorFX:



## Public Member Functions

- void [start](#) (Stage stage) throws Exception

## Static Public Member Functions

- static void [main](#) (String[] args)

## 6.3.1 Detailed Description

Main class of the calculator launching the application.

Definition at line 24 of file MainCalculatorFX.java.

## 6.3.2 Member Function Documentation

## 6.3.2.1 static void calculatorfx.MainCalculatorFX.main ( String[] args ) [static]

Definition at line 40 of file MainCalculatorFX.java.

## 6.3.2.2 void calculatorfx.MainCalculatorFX.start ( Stage stage ) throws Exception

< Added custom css styling

< Added title

< Added custom icon

Definition at line 27 of file MainCalculatorFX.java.

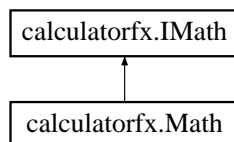
The documentation for this class was generated from the following file:

- [src/calculatorfx/MainCalculatorFX.java](#)

## 6.4 calculatorfx.Math Class Reference

Class containing the methods of the math library.

Inheritance diagram for calculatorfx.Math:



### Public Member Functions

- double [Sum](#) (double num1, double num2)  
*Method Sum.*
- double [Sub](#) (double minuend, double subtrahend)  
*Method Sub.*
- double [Div](#) (double dividend, double divisor) throws Exception  
*Method Div.*
- double [Mult](#) (double num1, double num2)  
*Method Mult.*
- long [Fact](#) (long num) throws Exception  
*Method Fact.*
- double [Pow](#) (double basis, double exponent) throws Exception  
*Method Pow.*
- double [Root](#) (double n, double x) throws Exception  
*Method Root.*
- long [toBinary](#) (long basis)  
*Method toBinary.*
- double [Abs](#) (double nonAbs)  
*Method Abs.*

### 6.4.1 Detailed Description

Class containing the methods of the math library.

See also

[MathTest.java](#) where the functionality is tested.

Definition at line 16 of file Math.java.

### 6.4.2 Member Function Documentation

#### 6.4.2.1 double calculatorfx.Math.Abs ( double nonAbs )

Method Abs.

Method Abs converts a number to its absolute value

## Parameters

<i>nonAbs</i>	is the number to be converted
---------------	-------------------------------

## Returns

absolute value of nonAbs

Implements [calculatorfx.IMath](#).

Definition at line 181 of file Math.java.

#### 6.4.2.2 double calculatorfx.Math.Div ( double *divident*, double *divisor* ) throws Exception

Method Div.

Method Div produces a division of two numbers

## Parameters

<i>divident</i>	is the number to be divided
<i>divisor</i>	is the number to be divided by

## Warning

exception thrown when divisor is 0

## Returns

division of divident and divisor

## Exceptions

<i>Exception</i>	(Division by zero.)
------------------	---------------------

Implements [calculatorfx.IMath](#).

Definition at line 57 of file Math.java.

#### 6.4.2.3 long calculatorfx.Math.Fact ( long *num* ) throws Exception

Method Fact.

Method Fact produces a factorial of a number

## Parameters

<i>num</i>	is the number to be factorised
------------	--------------------------------

**Returns**

factorial of num

**Exceptions**

<i>Exception</i>	(Factorial of negative number.)
------------------	---------------------------------

Implements [calculatorfx.IMath](#).

Definition at line 88 of file Math.java.

**6.4.2.4 double calculatorfx.Math.Mult ( double num1, double num2 )**

Method Mult.

Method Mult produces a multiplication of two numbers

**Parameters**

<i>num1</i>	is the first number to be multiplied
<i>num2</i>	is the second number to be multiplied

**Returns**

product of num1 and num2

Implements [calculatorfx.IMath](#).

Definition at line 74 of file Math.java.

**6.4.2.5 double calculatorfx.Math.Pow ( double basis, double exponent ) throws Exception**

Method Pow.

Method Pow produces an exponentiation of a number

**Parameters**

<i>basis</i>	is the basis
<i>exponent</i>	is the exponent

**Returns**

basis to the power of the exponent

**Exceptions**

<i>Exception</i>	(Negative exponent.)
------------------	----------------------



Implements [calculatorfx.IMath](#).

Definition at line 110 of file Math.java.

#### 6.4.2.6 double calculatorfx.Math.Root ( double *n*, double *x* ) throws Exception

Method Root.

Method Root produces a square root of a number

##### Parameters

<i>n</i>	is the grade of root
<i>x</i>	is the basis

##### Returns

square root of num

##### Exceptions

<i>Exception</i>	(Root not defined.)    (Even root from negative number.)
------------------	--

Implements [calculatorfx.IMath](#).

Definition at line 136 of file Math.java.

#### 6.4.2.7 double calculatorfx.Math.Sub ( double *minuend*, double *subtrahend* )

Method Sub.

Method Sub produces a difference of two numbers

##### Parameters

<i>minuend</i>	is the number to be subtracted from
<i>subtrahend</i>	is the number to be subtracted

##### Returns

difference between minuend and subtrahend

Implements [calculatorfx.IMath](#).

Definition at line 41 of file Math.java.

#### 6.4.2.8 double calculatorfx.Math.Sum ( double *num1*, double *num2* )

Method Sum.

Method Sum produces a sum of two numbers

**Parameters**

<i>num1</i>	is the first number to be added
<i>num2</i>	is the second number to be added

**Returns**

sum of num1 and num2

Implements [calculatorfx.IMath](#).

Definition at line 27 of file Math.java.

**6.4.2.9 long calculatorfx.Math.toBinary ( long *basis* )**

Method toBinary.

Method toBinary converts a number to binary

**Parameters**

<i>basis</i>	is the number to be converted
--------------	-------------------------------

**Returns**

basis written in binary

Implements [calculatorfx.IMath](#).

Definition at line 167 of file Math.java.

The documentation for this class was generated from the following file:

- [src/calculatorfx/Math.java](#)

**6.5 test.MathTest Class Reference**

Class containing the tests.

**Public Member Functions**

- [MathTest](#) ()
- void [setUp](#) ()
- void [tearDown](#) ()
- void [testSum](#) ()  
*Test of Sum method.*
- void [testSub](#) ()  
*Test of Sub method.*
- void [testDiv](#) () throws Exception

*Test of Div method.*

- void `testMult ()`

*Test of Mult method.*

- void `testFact ()` throws Exception

*Test of Fact method.*

- void `testPow ()` throws Exception

*Test of Pow method.*

- void `testRoot ()` throws Exception

*Test of Root method.*

- void `testToBinary ()`

*Test of ToBinary method.*

- void `testAbs ()`

*Test of Abs method.*

#### Static Public Member Functions

- static void `setUpClass ()`
- static void `tearDownClass ()`

#### Package Attributes

- Random `rand` = new Random()

#### 6.5.1 Detailed Description

Class containing the tests.

##### Postcondition

Custom math library is tested whether it works as desired or not.

Definition at line 31 of file MathTest.java.

#### 6.5.2 Constructor & Destructor Documentation

##### 6.5.2.1 test.MathTest.MathTest ( )

Definition at line 35 of file MathTest.java.

#### 6.5.3 Member Function Documentation

##### 6.5.3.1 void test.MathTest.setUp ( )

Definition at line 47 of file MathTest.java.

#### 6.5.3.2 static void test.MathTest.setUpClass ( ) [static]

Definition at line 39 of file MathTest.java.

#### 6.5.3.3 void test.MathTest.tearDown ( )

Definition at line 51 of file MathTest.java.

#### 6.5.3.4 static void test.MathTest.tearDownClass ( ) [static]

Definition at line 43 of file MathTest.java.

#### 6.5.3.5 void test.MathTest.testAbs ( )

Test of Abs method.

Testing Abs method by comparing its results to results of the standart method "abs" using random negative input numbers.

##### Postcondition

Method Abs produced the same results as method "abs" (or not)

Definition at line 316 of file MathTest.java.

#### 6.5.3.6 void test.MathTest.testDiv ( ) throws Exception

Test of Div method.

Testing Div method by comparing its results to results of the standart operation "/" using random input numbers.

##### Postcondition

Method div produced the same results as operation "/" (or not)

##### Exceptions

<code>java.lang.Exception</code>
----------------------------------

Definition at line 121 of file MathTest.java.

#### 6.5.3.7 void test.MathTest.testFact ( ) throws Exception

Test of Fact method.

Testing Fact method by comparing its results to expected results.

##### Postcondition

Method Fact produced the expected results (or not)

**Exceptions**

<code>java.lang.Exception</code>
----------------------------------

Definition at line 190 of file MathTest.java.

**6.5.3.8 void test.MathTest.testMult ( )**

Test of Mult method.

Testing Mult method by comparing its results to results of the standart operation "\*" using random input numbers.

**Postcondition**

Method Mult produced the same results as operation "\*" (or not)

Definition at line 159 of file MathTest.java.

**6.5.3.9 void test.MathTest.testPow ( ) throws Exception**

Test of Pow method.

Testing Pow method by comparing its results to results of the standart method "pow" using random input numbers.

**Postcondition**

Method Pow produced the same results as method "pow" (or not)

**Exceptions**

<code>java.lang.Exception</code>
----------------------------------

Definition at line 222 of file MathTest.java.

**6.5.3.10 void test.MathTest.testRoot ( ) throws Exception**

Test of Root method.

Testing Root method by comparing its results to results of the standart method "sqrt" using random input numbers.

**Postcondition**

Method Root produced the same results as method "sqrt" (or not)

**Exceptions**

<code>java.lang.Exception</code>
----------------------------------

Definition at line 250 of file MathTest.java.

#### 6.5.3.11 void test.MathTest.testSub ( )

Test of Sub method.

Testing Sub method by comparing its results to results of the standart operation "-" using random input numbers.

##### Postcondition

Method Sub produced the same results as operation "-" (or not)

Definition at line 90 of file MathTest.java.

#### 6.5.3.12 void test.MathTest.testSum ( )

Test of Sum method.

Testing Sum method by comparing its results to results of the standart operation "+" using random input numbers.

##### Postcondition

Method Sum produced the same results as operation "+" (or not)

Definition at line 61 of file MathTest.java.

#### 6.5.3.13 void test.MathTest.testToBinary ( )

Test of ToBinary method.

Testing ToBinary method by comparing its results to results of the standart methods "toString" and "parseInt" using random input numbers.

##### Postcondition

Method ToBinary produced the same results as methods "toString" and "parseInt" (or not)

Definition at line 288 of file MathTest.java.

### 6.5.4 Member Data Documentation

#### 6.5.4.1 Random test.MathTest.rand = new Random() [package]

Definition at line 33 of file MathTest.java.

The documentation for this class was generated from the following file:

- [src/test/MathTest.java](#)

## 6.6 calculatorfx.StandartDeviation Class Reference

Implementation of the SD, includes main.

### Static Public Member Functions

- static void [main](#) (String[] args) throws Exception

### 6.6.1 Detailed Description

Implementation of the SD, includes main.

#### Precondition

The data, out of which the SD is to be calculated, need to be available on the standart input after the program is executed.

#### Postcondition

SD is printed to the standart output.

Definition at line 20 of file StandartDeviation.java.

### 6.6.2 Member Function Documentation

#### 6.6.2.1 static void calculatorfx.StandartDeviation.main ( String[] args ) throws Exception [static]

< Import of the math library

< Read one line from the standart output

< Number currently read from the standart output

< Sum of all inputted numbers (a+b+...)

< Sum of all inputted numbers squared ( $a^2+b^2+\dots$ )

< The number of numbers (N)

< Mean of all inputted numbers

< Implementation of the equation of the SD

Definition at line 22 of file StandartDeviation.java.

The documentation for this class was generated from the following file:

- [src/StandartDeviation.java](#)

## 7 File Documentation

### 7.1 src/calculatorfx/Controller.java File Reference

Implementation of a Controller for a java FXML application.

#### Classes

- class [calculatorfx.Controller](#)  
*The controller of the calculator.*

#### Packages

- package [calculatorfx](#)

#### 7.1.1 Detailed Description

Implementation of a Controller for a java FXML application.

This file contains a java implementation of a Controller for a java FXML application. It is a part of a group project for the subject IVS of Brno University of Technology.

#### Author

Rene Bolf & Radoslav Grencik

### 7.2 src/calculatorfx/IMath.java File Reference

#### Classes

- interface [calculatorfx.IMath](#)

#### Packages

- package [calculatorfx](#)

### 7.3 src/calculatorfx/MainCalculatorFX.java File Reference

Implementation of a main class for a java FXML application.

#### Classes

- class [calculatorfx.MainCalculatorFX](#)  
*Main class of the calculator launching the application.*



#### Packages

- package [calculatorfx](#)

#### 7.3.1 Detailed Description

Implementation of a main class for a java FXML application.

This file contains a java implementation of a main class for a java FXML application. It is a part of a group project for the subject IVS of Brno University of Technology.

#### Author

Rene Bolf & Radoslav Grencik

### 7.4 src/calculatorfx/Math.java File Reference

Interface for a custom math library.

#### Classes

- class [calculatorfx.Math](#)  
*Class containing the methods of the math library.*

#### Packages

- package [calculatorfx](#)

#### 7.4.1 Detailed Description

Interface for a custom math library.

Implementation of a custom math library.

This file contains an interface of a custom math library for addition, subtraction, multiplication, division, calculating factorials, power and root functions, conversion to binary and conversion to absolute values. It is a part of a group project for the subject IVS of Brno University of Technology.

#### Author

Michal Vasicek

This file contains a java implementation of a custom math library for addition, subtraction, multiplication, division, calculating factorials, power and root functions, conversion to binary and conversion to absolute values. It is a part of a group project for the subject IVS of Brno University of Technology.

#### Author

Michal Vasicek

## 7.5 `src/StandartDeviation.java` File Reference

Program to calculate the SD.

### Classes

- class [calculatorfx.StandartDeviation](#)  
*Implementation of the SD, includes main.*

### Packages

- package [calculatorfx](#)

### 7.5.1 Detailed Description

Program to calculate the SD.

This file contains a java implementation of a program to calculate the standart deviation (SD) using custom math library. It is a part of a group project for the subject IVS of Brno University of Technology.

### Author

Ondrej Holub

### See also

[Math.java](#), [IMath.java](#)

## 7.6 `src/styles/Style.css` File Reference

## 7.7 `src/test/MathTest.java` File Reference

Test the functionality of the custom math library.

### Classes

- class [test.MathTest](#)  
*Class containing the tests.*

### Packages

- package [test](#)

### 7.7.1 Detailed Description

Test the functionality of the custom math library.

This file contains tests for the custom math library used for the calculator. Each method is tested by using it to calculate a result and then comparing the results with standart operations. It is a part of a group project for the subject IVS of Brno University of Technology.

### Author

Michal Vasicek

### See also

[Math.java](#), [IMath.java](#)