FACULTY OF INFORMATION TECHNOLOGY
BRNO UNIVERSITY OF TECHNOLOGY

PRACTICAL ASPECTS OF SOFTWARE DESIGN – CALCULATOR

SOFTWARE DOCUMENTATION

2019/2020

Generated by Doxygen 1.8.18

April 17, 2020 Matušák Tomáš (xmatus34)

Chapter 1

Namespace Index

1.1 Packages

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 Calculator Namespace Reference

Classes

- class App

  Interaction logic for App.xaml
- class MainWindow

  Interaction logic for MainWindow.xaml
- class Windowhelp

  Windowhelp

4.2 Calculator.Properties Namespace Reference

Classes

- class **Resources**

  Třída prostředků se silnými typy pro vyhledávání lokalizovaných řetězců atp.
- class **Settings**

4.3 ExpressionProcessor Namespace Reference

Classes

- class ExpressionProcessor
- class ShuntingYard

  Class implementing a shunting-yard algorithm
- class Solver
- class Splitter
- class Validator

4.4 MathLibrary Namespace Reference

Classes

- class **MathLib**

4.5 MathLibraryTests Namespace Reference

Classes

- class AdvancedFunctionsTests

  Tests of basic functions of mathematical library
- class BasicFunctionsTests

  Tests of basic functions of mathematical library

4.6 StandardDeviation Namespace Reference

Classes

- class Program

Chapter 5

Class Documentation

5.1 MathLibraryTests.AdvancedFunctionsTests Class Reference

Tests of basic functions of mathematical library

Public Member Functions

- void Fact (double num, double expected)

  Tests of factorial
- void Fact_ShouldThrowArgumentOutOfRangeException (double num)

  Tests of factorial, that should throw ArgumentOutOfRangeException when num is non Natural number except zero
- void Pow (double number, double exponent, double expected)

  Tests of exponentiation
- void Pow_ShouldThrowArgumentOutOfRangeException (double number, double exponent)

  Tests of exponentiation, that should throw ArgumentOutOfRangeException when exponent is non Natural number except zero
- void Root (double number, double degree, double expected)

  Tests of nth-Root
- void Root_ShouldThrowArgumentOutOfRangeException (double number, double degree)

  Tests of nth-Root, that should throw ArgumentOutOfRangeException, when degree is non Natural number or number is a negative number
- void Abs (double number, double expected)

  Tests of absolute value
- void Rnd ()

  Tests of random number generation

5.1.1 Detailed Description

Tests of basic functions of mathematical library

5.1.2 Member Function Documentation

5.1.2.1 Abs()

```
void MathLibraryTests.AdvancedFunctionsTests.Abs (
            double number,
            double expected )
```

Tests of absolute value

Parameters

number	Number
expected	Expected result of test

5.1.2.2 Fact()

```
void MathLibraryTests.AdvancedFunctionsTests.Fact (
            double num,
            double expected )
```

Tests of factorial

Parameters

num	Number
expected	Expected result of test

5.1.2.3 Fact_ShouldThrowArgumentOutOfRangeException()

```
void MathLibraryTests.AdvancedFunctionsTests.Fact_ShouldThrowArgumentOutOfRangeException (
            double num )
```

Tests of factorial, that should throw ArgumentOutOfRangeException when *num* is non Natural number except zero

Parameters

num	Number

5.1.2.4 Pow()

```
void MathLibraryTests.AdvancedFunctionsTests.Pow (
            double number,
```

```
            double exponent,
            double expected )
```

Tests of exponentiation

Parameters

number	Base
exponent	Exponent
expected	Expected result of test

5.1.2.5 Pow_ShouldThrowArgumentOutOfRangeException()

```
void MathLibraryTests.AdvancedFunctionsTests.Pow_ShouldThrowArgumentOutOfRangeException (
            double number,
            double exponent )
```

Tests of exponentiation, that should throw ArgumentOutOfRangeException when *exponent* is non Natural number except zero

Parameters

number	Base
exponent	Exponent

5.1.2.6 Rnd()

```
void MathLibraryTests.AdvancedFunctionsTests.Rnd ( )
```

Tests of random number generation

5.1.2.7 Root()

```
void MathLibraryTests.AdvancedFunctionsTests.Root (
            double number,
            double degree,
            double expected )
```

Tests of nth-Root

Parameters

number	Radicant
degree	Degree
expected	Expected result of test

5.1.2.8 Root_ShouldThrowArgumentOutOfRangeException()

```
void MathLibraryTests.AdvancedFunctionsTests.Root_ShouldThrowArgumentOutOfRangeException (
            double number,
            double degree )
```

Tests of nth-Root, that should throw ArgumentOutOfRangeException, when *degree* is non Natural number or *number* is a negative number

Parameters

number	Radicant
degree	Degree

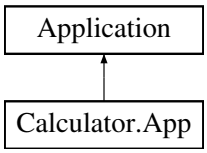
The documentation for this class was generated from the following file:

- MathLibraryTests/AdvancedFunctionsTests.cs

5.2 Calculator.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for Calculator.App:



Public Member Functions

- void InitializeComponent ()

  InitializeComponent

Static Public Member Functions

- static void Main ()

  Application Entry Point.

5.2.1 Detailed Description

Interaction logic for App.xaml

App

5.2.2 Member Function Documentation

5.2.2.1 InitializeComponent()

```
void Calculator.App.InitializeComponent ( )
```

InitializeComponent

5.2.2.2 Main()

```
static void Calculator.App.Main ( )  [static]
```

Application Entry Point.

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

- Calculator/App.xaml.cs
- Calculator/obj/Debug/netcoreapp3.1/App.g.i.cs

5.3 MathLibraryTests.BasicFunctionsTests Class Reference

Tests of basic functions of mathematical library

Public Member Functions

- void Add (double num1, double num2, double expected)
  Tests of addition
- void Sub (double num1, double num2, double expected)
  Tests of subtraction
- void Mul (double num1, double num2, double expected)
  Tests of multiplication
- void Div (double num1, double num2, double expected)
  Tests of division
- void Div_ShouldThrowDivideByZeroException (double num1, double num2)
  Test of division, that should throw DivideByZeroException

5.3.1 Detailed Description

Tests of basic functions of mathematical library

5.3.2 Member Function Documentation

5.3.2.1 Add()

```
void MathLibraryTests.BasicFunctionsTests.Add (
            double num1,
            double num2,
            double expected )
```

Tests of addition

Parameters

num1	Addend
num2	Addend
expected	Expected result of test

5.3.2.2 Div()

```
void MathLibraryTests.BasicFunctionsTests.Div (
            double num1,
            double num2,
            double expected )
```

Tests of division

Parameters

num1	Dividend
num2	Divisor
expected	Expected result of test

5.3.2.3 Div_ShouldThrowDivideByZeroException()

```
void MathLibraryTests.BasicFunctionsTests.Div_ShouldThrowDivideByZeroException (
            double num1,
            double num2 )
```

Test of division, that should throw DivideByZeroException

Parameters

num1	Dividend
num2	Divisor

5.3.2.4 Mul()

```
void MathLibraryTests.BasicFunctionsTests.Mul (
            double num1,
            double num2,
            double expected )
```

Tests of multiplication

Parameters

num1	Factor
num2	Factor
expected	Expected result of test

5.3.2.5 Sub()

```
void MathLibraryTests.BasicFunctionsTests.Sub (
            double num1,
            double num2,
            double expected )
```

Tests of subtraction

Parameters

num1	Minuend
num2	Subtrahend
expected	Expected result of test

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

- MathLibraryTests/BasicFunctionsTests.cs

5.4 ExpressionProcessor.ExpressionProcessor Class Reference

Static Public Member Functions

- static string **Process** (string expression)

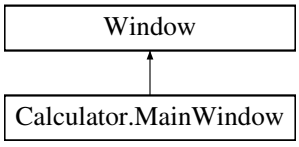
The documentation for this class was generated from the following file:

- ExpressionProcessor/ExpressionProcessor.cs

5.5 Calculator.MainWindow Class Reference

Interaction logic for MainWindow.xaml

Inheritance diagram for Calculator.MainWindow:



Public Member Functions

- void Get_Equation (object sender, RoutedEventArgs e)

  Actions to take when "=" button is clicked.
- void Abs (object sender, RoutedEventArgs e)

  Actions to take when "Abs" button is clicked.
- void InitializeComponent ()

  InitializeComponent

5.5.1 Detailed Description

Interaction logic for MainWindow.xaml

MainWindow

5.5.2 Member Function Documentation

5.5.2.1 Abs()

```
void Calculator.MainWindow.Abs (
            object sender,
            RoutedEventArgs e )
```

Actions to take when "Abs" button is clicked.

Parameters

sender	
e	

5.5.2.2 Get_Equation()

```
void Calculator.MainWindow.Get_Equation (
            object sender,
            RoutedEventArgs e )
```

Actions to take when "=" button is clicked.

Parameters

sender	
e	

5.5.2.3 InitializeComponent()

```
void Calculator.MainWindow.InitializeComponent ( )
```

InitializeComponent

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

- Calculator/MainWindow.xaml.cs
- Calculator/obj/Debug/netcoreapp3.1/MainWindow.g.i.cs

5.6 StandardDeviation.Program Class Reference

Static Public Member Functions

- static void **Main** (string[ ] args)

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

- StandardDeviation/Program.cs

5.7 ExpressionProcessor.ShuntingYard Class Reference

Class implementing a shunting-yard algorithm

Static Public Member Functions

- static List< string > ToPostfix (List< string > tokens)
  Convert mathematical expression to postfix notation

5.7.1 Detailed Description

Class implementing a shunting-yard algorithm

5.7.2 Member Function Documentation

5.7.2.1 ToPostfix()

```
static List<string> ExpressionProcessor.ShuntingYard.ToPostfix (
            List< string > tokens )  [static]
```

Convert mathematical expression to postfix notation

Parameters

tokens	List of tokens of mathematical expression

Returns

List of tokens in postfix notation

Exceptions

ArgumentException	Thrown when *tokens* contains unclosed expression

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

- ExpressionProcessor/ShuntingYard.cs

5.8 ExpressionProcessor.Solver Class Reference

Static Public Member Functions

- static string Solve (List< string > tokens)

  Calculates the given expression which is splitted into tokens.

5.8.1 Member Function Documentation

5.8.1.1 Solve()

```
static string ExpressionProcessor.Solver.Solve (
            List< string > tokens )  [static]
```

Calculates the given expression which is splitted into tokens.

Expression must be in postfix notation

Parameters

tokens	List of tokens

Returns

Calculation result

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

- ExpressionProcessor/Solver.cs

5.9 ExpressionProcessor.Splitter Class Reference

Static Public Member Functions

- static List< string > SplitToTokens (string exp)

  Splits a given string into tokens

5.9.1 Member Function Documentation

5.9.1.1 SplitToTokens()

```
static List<string> ExpressionProcessor.Splitter.SplitToTokens (
            string exp )  [static]
```

Splits a given string into tokens

Parameters

exp	The string to be splitted

Returns

List of tokens

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

- ExpressionProcessor/Splitter.cs

5.10 ExpressionProcessor.Validator Class Reference

Static Public Member Functions

- static bool IsValid (string exp, int num_of_param=1)

  Checks if a given string is a valid expression.

5.10.1 Member Function Documentation

5.10.1.1 IsValid()

```
static bool ExpressionProcessor.Validator.IsValid (
            string exp,
            int num_of_param = 1 )  [static]
```

Checks if a given string is a valid expression.

Parameters

exp	The string to be checked
num_of_param	Number of parameters (Only if the expression is inside a function)

Parameter `num_of_param` is optional

Returns

`True` if a string is a valid expression

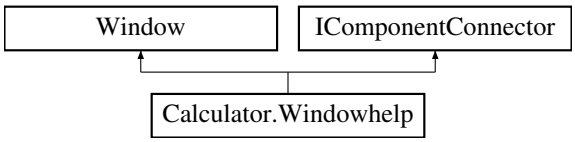
The documentation for this class was generated from the following file:

- ExpressionProcessor/Validator.cs

5.11 Calculator.Windowhelp Class Reference

Windowhelp

Inheritance diagram for Calculator.Windowhelp:



Public Member Functions

- void InitializeComponent ()

  InitializeComponent

5.11.1 Detailed Description

Windowhelp

Interaction logic for Windowhelp.xaml

5.11.2 Member Function Documentation

5.11.2.1 InitializeComponent()

```
void Calculator.Windowhelp.InitializeComponent ( )
```

InitializeComponent

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

- Calculator/obj/Debug/netcoreapp3.1/Windowhelp.g.i.cs
- Calculator/Windowhelp.xaml.cs

Index