

Floor11 Calculator

Generated by Doxygen 1.8.17

1 IVS Calculator	2
1.1 Introduction	2
2 Namespace Index	2
2.1 Packages	2
3 Hierarchical Index	2
3.1 Class Hierarchy	2
4 Class Index	2
4.1 Class List	2
5 File Index	3
5.1 File List	3
6 Namespace Documentation	3
6.1 calc_gui Namespace Reference	3
6.2 calc_main Namespace Reference	3
6.2.1 Variable Documentation	3
6.3 mathlib Namespace Reference	4
6.3.1 Function Documentation	4
7 Class Documentation	8
7.1 calc_main.calcLogic Class Reference	8
7.1.1 Detailed Description	11
7.1.2 Constructor & Destructor Documentation	11
7.1.3 Member Function Documentation	11
7.1.4 Member Data Documentation	25
7.2 object Class Reference	27
7.3 calc_gui.Ui_MainWindow Class Reference	27
7.3.1 Detailed Description	29
7.3.2 Member Function Documentation	29
7.3.3 Member Data Documentation	29
8 File Documentation	34
8.1 1000_num.txt File Reference	34
8.2 100_num.txt File Reference	34
8.3 10_num.txt File Reference	34
8.4 calc_gui.py File Reference	34
8.4.1 Detailed Description	34
8.5 calc_main.py File Reference	34
8.5.1 Detailed Description	35
8.6 mathlib.py File Reference	35
8.6.1 Detailed Description	36

1 IVS Calculator

1.1 Introduction

This is a documentation for Floor11 Calculator software created for IVS course, by xtalaj00 and xkovac57.

The software is a simple calculator capable of calculating basic arithmetic operations such as addition and subtraction, multiplication and division, exponents and roots or factorial. It's also capable to calculate with parentheses, has a simple memory and it can generate random numbers. It can also switch between dark and light modes to ease the strain on your eyes!

2 Namespace Index

2.1 Packages

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

calc_gui	3
calc_main	3
mathlib	4

3 Hierarchical Index

3.1 Class Hierarchy

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

object	27
calc_gui.Ui_MainWindow	27
calc_main.calcLogic	8
QMainWindow	
calc_main.calcLogic	8

4 Class Index

4.1 Class List

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

calc_main.calcLogic	
Class responsible for the logical aspect of the calculator	8
object	27
calc_gui.Ui_MainWindow	
Class responsible for the proper look of GUI of the calculator	27

5 File Index

5.1 File List

Here is a list of all files with brief descriptions:

calc_gui.py	
The interface of the calculator	34
calc_main.py	
The main logic of the calculator	34
mathlib.py	
Math library defining math operations (+,-,*,/,!,^,nth_root, RNG)	35

6 Namespace Documentation

6.1 calc_gui Namespace Reference

Classes

- class [Ui_MainWindow](#)
Class responsible for the proper look of GUI of the calculator.

6.2 calc_main Namespace Reference

Classes

- class [calcLogic](#)
Class responsible for the logical aspect of the calculator.

Variables

- [app](#) = QtWidgets.QApplication(sys.argv)
- [MainWindow](#) = QtWidgets.QMainWindow()
- [ui](#) = [calcLogic](#)()

6.2.1 Variable Documentation

6.2.1.1 app `calc_main.app = QtWidgets.QApplication(sys.argv)`

6.2.1.2 MainWindow `calc_main.MainWindow = QtWidgets.QMainWindow()`

6.2.1.3 ui `calc_main.ui = calcLogic()`

6.3 mathlib Namespace Reference

Functions

- def `add` (a, b)
Sum of 2 numbers.
- def `sub` (a, b)
Subtraction of 2 numbers.
- def `mul` (a, b)
Product of 2 numbers.
- def `div` (a, b)
Division of 2 numbers.
- def `fact` (n)
Factorial of a whole number.
- def `exp` (b, n)
*Exponentiation of a base *b* by an exponent *n*.*
- def `root` (n, x)
*Finding *n* th root of number *x*.*
- def `rng` (x)
Producing a random number in a given range.

6.3.1 Function Documentation

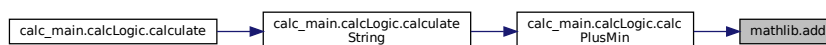
6.3.1.1 add() `def mathlib.add (`
 `a,`
 `b)`

Sum of 2 numbers.

Returns

`a + b`

Here is the caller graph for this function:



6.3.1.2 div() `def mathlib.div (`
`a,`
`b)`

Division of 2 numbers.

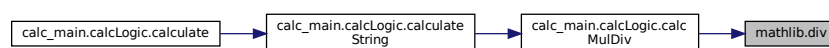
Exceptions

<i>ZeroDivisionError</i>	if b is equal to 0
--------------------------	--------------------

Returns

a / b

Here is the caller graph for this function:



6.3.1.3 exp() `def mathlib.exp (`
`b,`
`n)`

Exponentiation of a base b by an exponent n.

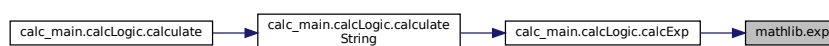
Exceptions

<i>ValueError</i>	if n is negative
<i>TypeError</i>	if n is a fraction

Returns

b^n

Here is the caller graph for this function:



6.3.1.4 fact() `def mathlib.fact (`
 `n)`

Factorial of a whole number.

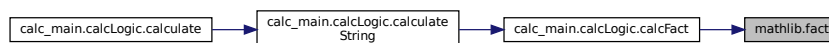
Exceptions

<i>ValueError</i>	if n is negative
<i>TypeError</i>	if n is a fraction

Returns

n!

Here is the caller graph for this function:



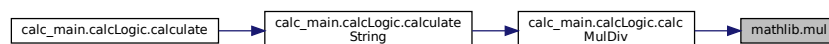
6.3.1.5 mul() `def mathlib.mul (`
 `a,`
 `b)`

Product of 2 numbers.

Returns

a * b

Here is the caller graph for this function:



6.3.1.6 rng() `def mathlib.rng (`
 `x)`

Producing a random number in a given range.

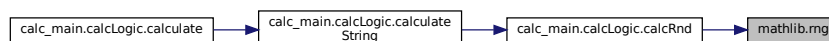
Exceptions

<i>ValueError</i>	if $x == 0$
<i>TypeError</i>	if x is a fraction

Returns

Random number from range $<0,x>$

Here is the caller graph for this function:



6.3.1.7 root() `def mathlib.root (`
`n,`
`x)`

Finding n th root of number x .

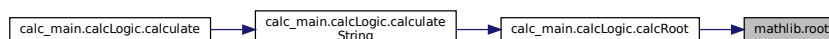
Exceptions

<i>ZeroDivisionError</i>	if n is zero
<i>ValueError</i>	if n is negative
<i>TypeError</i>	if n is a fraction
<i>ValueError</i>	if n is even and x is negative

Returns

$x^{(1/n)}$

Here is the caller graph for this function:



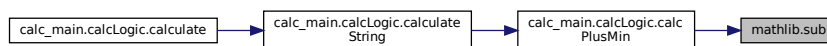

```
6.3.1.8 sub() def mathlib.sub (  
    a,  
    b )
```

Subtraction of 2 numbers.

Returns

a - b

Here is the caller graph for this function:

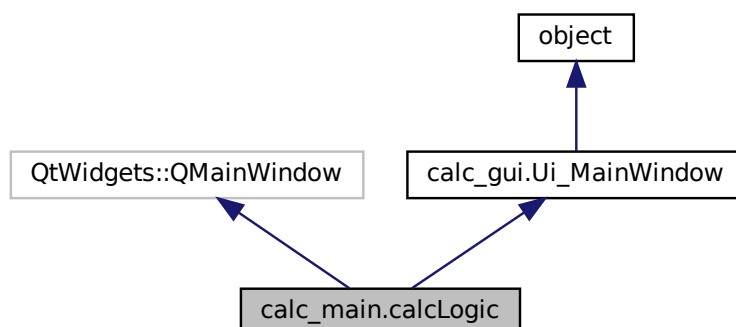


7 Class Documentation

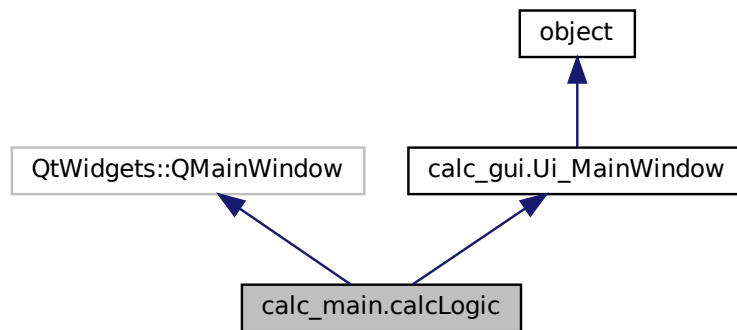
7.1 calc_main.calcLogic Class Reference

Class responsible for the logical aspect of the calculator.

Inheritance diagram for calc_main.calcLogic:



Collaboration diagram for calc_main.calcLogic:



Public Member Functions

- def `__init__` (self)
Default python initialization class used to initialize the variables to a starting state.
- def `aNum` (self, n)
Appends a number.
- def `aBinOp` (self, op)
Appends or replaces a binary operator.
- def `aUnOp` (self, op)
Appends a unary operator.
- def `aDecPoint` (self)
Appends a decimal point.
- def `backSpace` (self)
Deletes last input.
- def `aParenthesis` (self, symb)
Appends parentheses.
- def `clearEverything` (self)
Clears both main and secondary displays.
- def `changeSign` (self)
Changes the sign of an integer or float displayed.
- def `oPDF_g` (self)
Opens Guide PDF on a specific system (Windows/Linux)
- def `memSet` (self)
Stores text displayed on main display to memory.
- def `memLoad` (self)
Loads text from memory to main display (if not empty)
- def `memClear` (self)
Clears the memory.
- def `calculate` (self)
Calculates the result of equation.
- def `repairInput` (self)
Repairs any syntax errors still present in the input string.

- def `findAllPositions` (self, pattern, str)
Finds all positions of given patterns in string and returns them in a list.
- def `findParPairs` (self, text)
Finds corresponding pairs of parentheses.
- def `calculateString` (self, text)
Carries out a calculation on a string.
- def `calcRnd` (self, text)
Finds all rnd() and assigns random numbers for each one in a given text (if there are any)
- def `calcFact` (self, text)
Finds and calculates factorials in a given text (if there are any)
- def `calcRoot` (self, text)
Finds and calculates roots in a given text (if there are any)
- def `calcExp` (self, text)
Finds and calculates exponents in a given text (if there are any)
- def `calcMulDiv` (self, text)
Finds and calculates multiplications and divisions in a given text (if there are any)
- def `calcPlusMin` (self, text)
Finds and calculates additions and subtractions in a given text (if there are any)
- def `getLeftOperand` (self, text, pos)
Finds the left operand of an operation which is on a given position in a given text.
- def `getRightOperand` (self, text, pos)
Finds the right operand of an operation which is on a given position in a given text.
- def `errorHandler` (self, err_msg)
Handles error printing onto the main display and further setting of tags.
- def `repairOutput` (self)
Repairs the format of the output.
- def `sColor` (self, dark)
Function for UI color change (dark/white)

Public Attributes

- `md_text`
Text displayed on the main display.
- `sd_text`
Text displayed on the secondary (upper) display.
- `bin_ops`
List of basic binary operands ('+', '-', '/', '', '^')*
- `un_ops`
List of "unary" operands (root, '!') (although root is more of a binary operand it behaves more like factorial)
- `parentheses`
List of parenthesis characters ('(', ')')
- `open_par`
Number of open parentheses.
- `dec_p`
Bool that keeps track of whether a decimal point can be placed.
- `memory`
Variable stores a string that can later be loaded to the main display.
- `error`
Bool that keeps track of whether an error has occurred.
- `result`
Bool that informs us about whether a result from previous calculation is currently being shown.

7.1.1 Detailed Description

Class responsible for the logical aspect of the calculator.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 `__init__()` `def calc_main.calcLogic.__init__ (`
`self)`

Default python initialization class used to initialize the variables to a starting state.

7.1.3 Member Function Documentation

7.1.3.1 `aBinOp()` `def calc_main.calcLogic.aBinOp (`
`self,`
`op)`

Appends or replaces a binary operator.

Parameters

<i>op</i>	Operator that's being appended
-----------	--------------------------------

7.1.3.2 `aDecPoint()` `def calc_main.calcLogic.aDecPoint (`
`self)`

Appends a decimal point.

7.1.3.3 `aNum()` `def calc_main.calcLogic.aNum (`
`self,`
`n)`

Appends a number.

Parameters

<i>n</i>	Number that's being appended
----------	------------------------------

7.1.3.4 aParenthesis() `def calc_main.calcLogic.aParenthesis (`
 `self,`
 `symb)`

Spends parentheses.

Parameters

<i>symb</i>	Symbol that's being appended
-------------	------------------------------

7.1.3.5 aUnOp() `def calc_main.calcLogic.aUnOp (`
 `self,`
 `op)`

Appends a unary operator.

Parameters

<i>op</i>	Operator that's being appended
-----------	--------------------------------

7.1.3.6 backSpace() `def calc_main.calcLogic.backSpace (`
 `self)`

Deletes last input.

7.1.3.7 calcExp() `def calc_main.calcLogic.calcExp (`
 `self,`
 `text)`

Finds and calculates exponents in a given text (if there are any)

Parameters

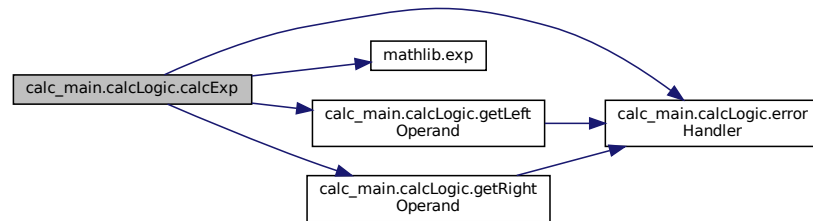
<i>text</i>	Text in which we calculate instances of '^'
-------------	---

Returns

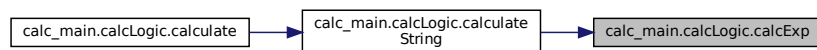
`text` with all exponents calculated

None if any of the called functions raise the error flag

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.8 calcFact() `def calc_main.calcLogic.calcFact (`
 `self,`
 `text)`

Finds and calculates factorials in a given text (if there are any)

Parameters

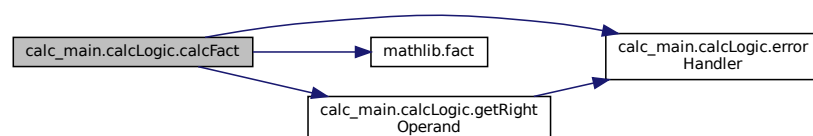
<i>text</i>	Text in which we calculate instances of factorial
-------------	---

Returns

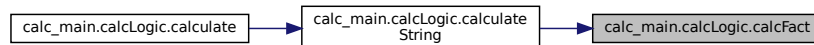
`text` with all factorials calculated

None if any of the called functions raise the error flag

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.9 calcMulDiv() `def calc_main.calcLogic.calcMulDiv (`
 `self,`
 `text)`

Finds and calculates multiplications and divisions in a given text (if there are any)

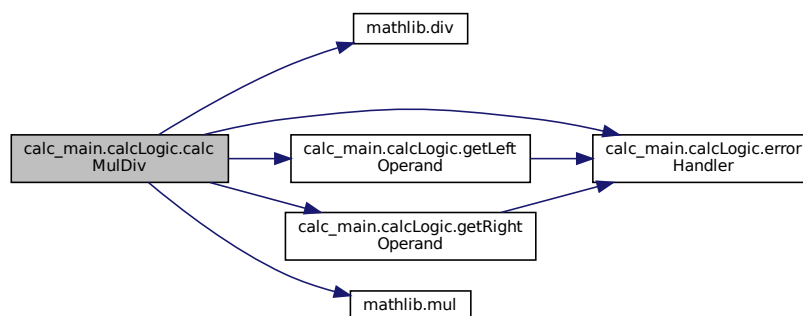
Parameters

<i>text</i>	Text in which we calculate instances of '*' and '/'
-------------	---

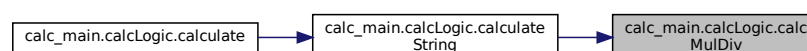
Returns

text with all multiplications and divisions calculated
 None if any of the called functions raise the error flag

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.10 calcPlusMin() `def calc_main.calcLogic.calcPlusMin (`
 `self,`
 `text)`

Finds and calculates additions and subtractions in a given text (if there are any)

Parameters

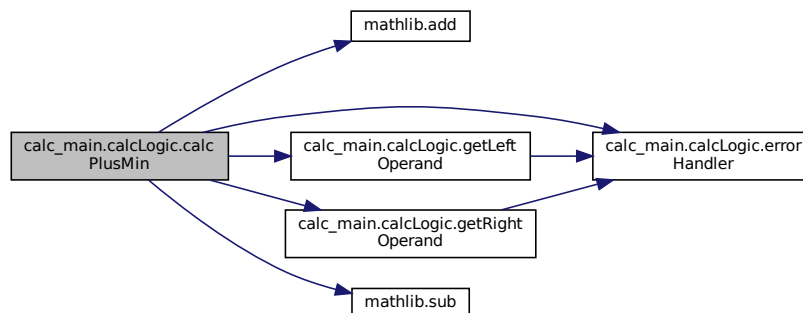
<i>text</i>	Text in which we calculate instances of '+' and '-'
-------------	---

Returns

`text` with all additions and subtractions calculated

None if any of the called functions raise the error flag

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.11 calcRnd() `def calc_main.calcLogic.calcRnd (`
 `self,`
 `text)`

Finds all `rnd()` and assigns random numbers for each one in a given text (if there are any)

Parameters

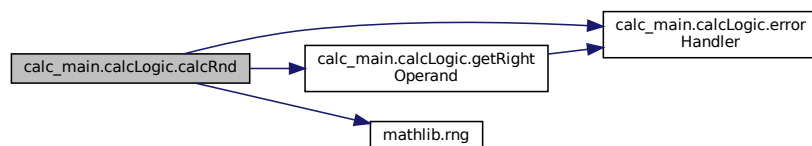
<i>text</i>	Text in which we generate random numbers if instances of <code>rnd()</code> are present
-------------	---

Returns

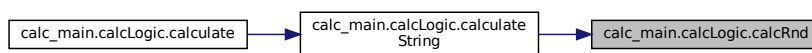
`text` with all random numbers assigned

None if any of the called functions raise the error flag

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.12 calcRoot() `def calc_main.calcLogic.calcRoot (`
 `self,`
 `text)`

Finds and calculates roots in a given text (if there are any)

Parameters

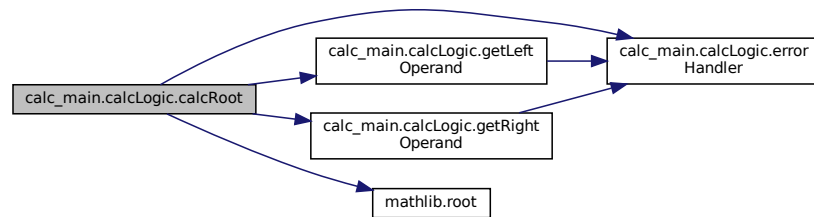
<code>text</code>	Text in which we calculate instances of root
-------------------	--

Returns

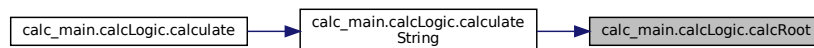
text with all roots calculated

None if any of the called functions raise the error flag

Here is the call graph for this function:



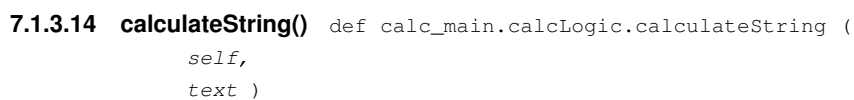
Here is the caller graph for this function:



7.1.3.13 calculate() `def calc_main.calcLogic.calculate (self)`

Calculates the result of equation.

Here is the call graph for this function:



Parameters

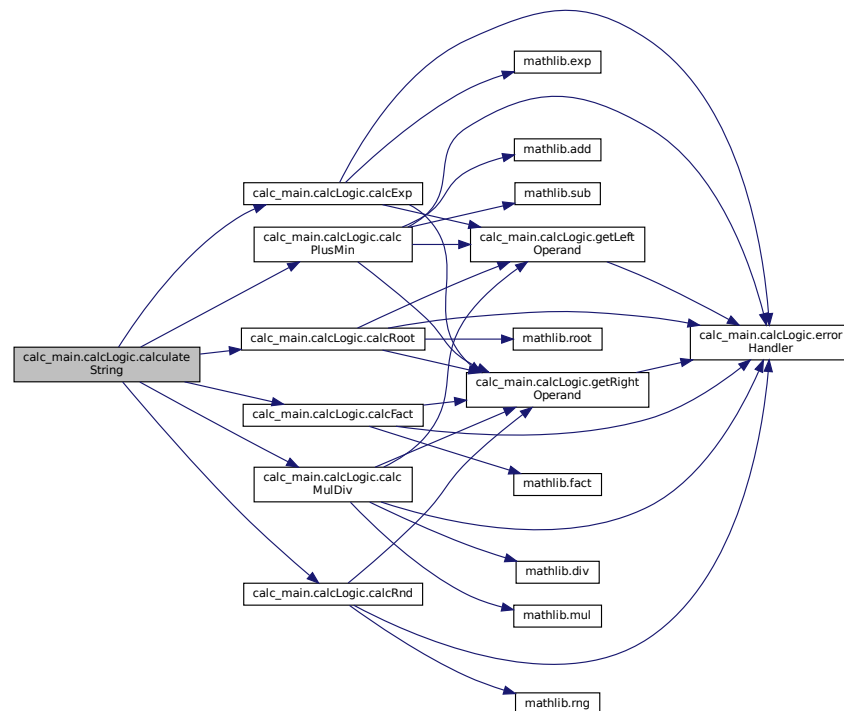
Generated by Doxygen

Returns

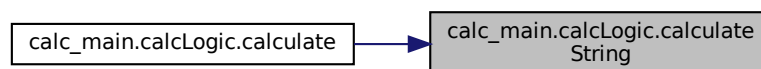
`text` after all the necessary calculations

None if any of the called functions raise the error flag

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.15 changeSign() `def calc_main.calcLogic.changeSign (self)`

Changes the sign of an integer or float displayed.

7.1.3.16 clearEverything() `def calc_main.calcLogic.clearEverything (`
 self)

Clears both main and secondary displays.

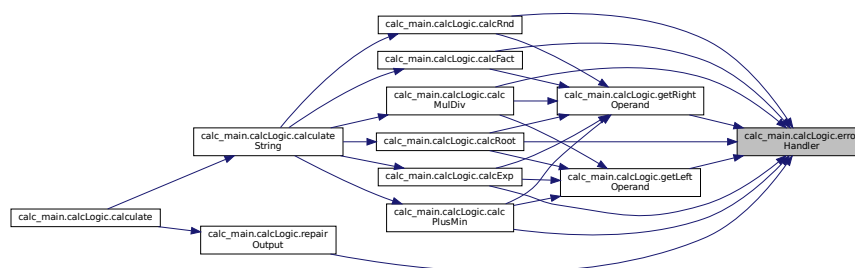
7.1.3.17 errorHandler() `def calc_main.calcLogic.errorHandler (`
 self,
 err_msg)

Handles error printing onto the main display and further setting of tags.

Parameters

<i>err_msg</i>	Error message that will be printed on the display
----------------	---

Here is the caller graph for this function:



7.1.3.18 findAllPositions() `def calc_main.calcLogic.findAllPositions (`
 self,
 pattern,
 str)

Finds all positions of given patterns in string and returns them in a list.

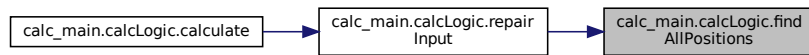
Parameters

<i>pattern</i>	Pattern the function is looking for
<i>str</i>	String that's being searched for the sought pattern

Returns

`positions` List with all the positions of sought pattern

Here is the caller graph for this function:



7.1.3.19 findParPairs() `def calc_main.calcLogic.findParPairs (`
 `self,`
 `text)`

Finds corresponding pairs of parentheses.

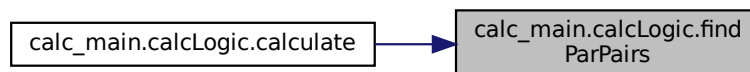
Parameters

<i>text</i>	Text that's being searched
-------------	----------------------------

Returns

`pairs` Dictionary with parenthesis pairs

Here is the caller graph for this function:



7.1.3.20 getLeftOperand() `def calc_main.calcLogic.getLeftOperand (`
 `self,`
 `text,`
 `pos)`

Finds the left operand of an operation which is on a given position in a given text.

Parameters

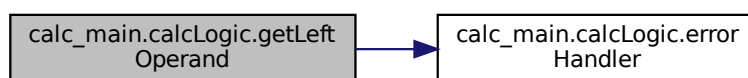
<i>text</i>	Text in which we are searching for the operand
<i>pos</i>	Position of the operand in the text

Returns

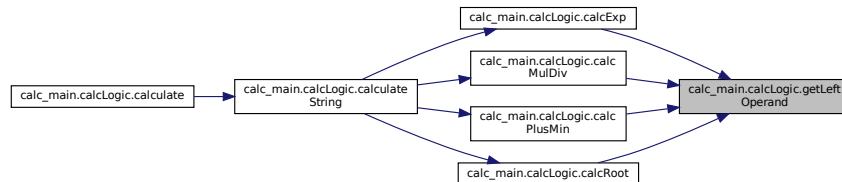
`left_op`

None if the operand is out of bounds

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.21 getRightOperand() `def calc_main.calcLogic.getRightOperand (`
 `self,`
 `text,`
 `pos)`

Finds the right operand of an operation which is on a given position in a given text.

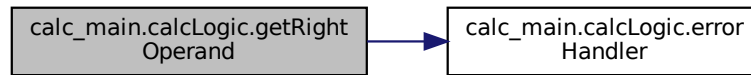
Parameters

<i>text</i>	Text in which we are searching for the operand
<i>pos</i>	Position of the operand in the text

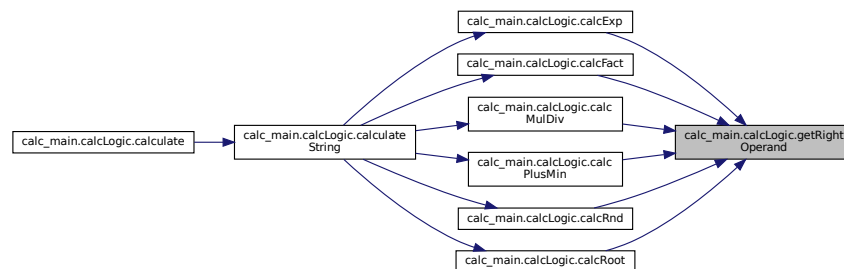
Returns`right_op`

None if the operand is out of bounds

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.22 memClear() `def calc_main.calcLogic.memClear (`
 `self)`

Clears the memory.

7.1.3.23 memLoad() `def calc_main.calcLogic.memLoad (`
 `self)`

Loads text from memory to main display (if not empty)

7.1.3.24 memSet() `def calc_main.calcLogic.memSet (`
 `self)`

Stores text displayed on main display to memory.

7.1.3.25 oPDF_g() `def calc_main.calcLogic.oPDF_g (`
`self)`

Opens Guide PDF on a specific system (Windows/Linux)

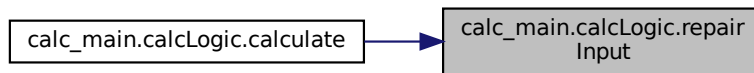
7.1.3.26 repairInput() `def calc_main.calcLogic.repairInput (`
`self)`

Repairs any syntax errors still present in the input string.

Here is the call graph for this function:



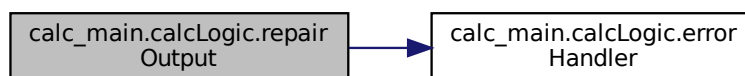
Here is the caller graph for this function:



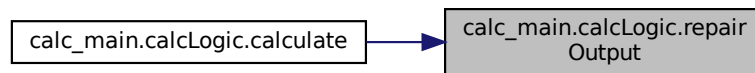
7.1.3.27 repairOutput() `def calc_main.calcLogic.repairOutput (`
`self)`

Repairs the format of the output.

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.28 sColor() `def calc_main.calcLogic.sColor (`
 `self,`
 `dark)`

Function for UI color change (dark/white)

7.1.4 Member Data Documentation

7.1.4.1 bin_ops `calc_main.calcLogic.bin_ops`

List of basic binary operands ('+', '-', '/', '*', '^')

7.1.4.2 dec_p `calc_main.calcLogic.dec_p`

Bool that keeps track of whether a decimal point can be placed.

7.1.4.3 error `calc_main.calcLogic.error`

Bool that keeps track of whether an error has occurred.

7.1.4.4 md_text `calc_main.calcLogic.md_text`

Text displayed on the main display.

7.1.4.5 memory `calc_main.calcLogic.memory`

Variable stores a string that can later be loaded to the main display.

7.1.4.6 open_par `calc_main.calcLogic.open_par`

Number of open parentheses.

7.1.4.7 parentheses `calc_main.calcLogic.parentheses`

List of parenthesis characters ('(',')')

7.1.4.8 result `calc_main.calcLogic.result`

Bool that informs us about whether a result from previous calculation is currently being shown.

7.1.4.9 sd_text `calc_main.calcLogic.sd_text`

Text displayed on the secondary (upper) display.

7.1.4.10 un_ops `calc_main.calcLogic.un_ops`

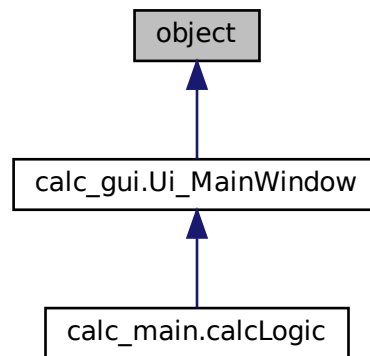
List of "unary" operands (root, '!') (although root is more of a binary operand it behaves more like factorial)

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

- [calc_main.py](#)

7.2 object Class Reference

Inheritance diagram for object:



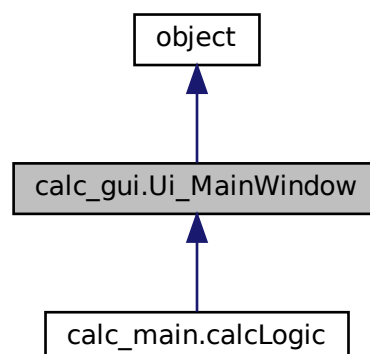
The documentation for this class was generated from the following file:

- [calc_gui.py](#)

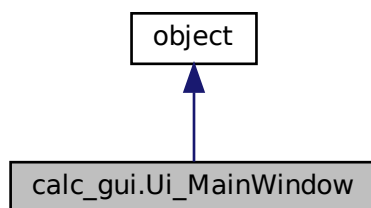
7.3 calc_gui.Ui_MainWindow Class Reference

Class responsible for the proper look of GUI of the calculator.

Inheritance diagram for calc_gui.Ui_MainWindow:



Collaboration diagram for `calc_gui.Ui_MainWindow`:



Public Member Functions

- def `setUpUi` (self, MainWindow)
- def `retranslateUi` (self, MainWindow)

Public Attributes

- `centralwidget`
- `frame`

The frame of the calculator.

- `n_button_0`
- `n_button_1`
- `n_button_2`
- `n_button_3`
- `n_button_4`
- `n_button_5`
- `n_button_6`
- `n_button_7`
- `n_button_8`
- `n_button_9`
- `button_plus`
- `button_minus`
- `button_multiply`
- `button_divide`
- `button_fact`
- `button_root`
- `button_sroot`
- `n_button_sign`
- `n_button_h_dot`
- `n_button_comma`
- `button_parent_o`
- `button_parent_c`
- `button_ms`
- `button_mc`
- `button_ml`
- `button_random`
- `button_delete`

- [button_ce](#)
- [h_equal_button](#)
- [equal_button](#)
- [main_display](#)
- [h_display](#)
- [line](#)
- [line_2](#)
- [menubar](#)
- [menuMenu](#)
- [actionDark_mode](#)
- [actionHelp](#)
- [pWin](#)
- [textPwin](#)
- [IName](#)
- [IVersion](#)
- [actionInformation](#)

7.3.1 Detailed Description

Class responsible for the proper look of GUI of the calculator.

7.3.2 Member Function Documentation

7.3.2.1 `retranslateUi()` `def calc_gui.Ui_MainWindow.retranslateUi (`
 self,
 MainWindow)

7.3.2.2 `setupUi()` `def calc_gui.Ui_MainWindow.setupUi (`
 self,
 MainWindow)

7.3.3 Member Data Documentation

7.3.3.1 `actionDark_mode` `calc_gui.Ui_MainWindow.actionDark_mode`

7.3.3.2 `actionHelp` `calc_gui.Ui_MainWindow.actionHelp`

7.3.3.3 actionInformation `calc_gui.Ui_MainWindow.actionInformation`

7.3.3.4 button_ce `calc_gui.Ui_MainWindow.button_ce`

7.3.3.5 button_delete `calc_gui.Ui_MainWindow.button_delete`

7.3.3.6 button_divide `calc_gui.Ui_MainWindow.button_divide`

7.3.3.7 button_fact `calc_gui.Ui_MainWindow.button_fact`

7.3.3.8 button_mc `calc_gui.Ui_MainWindow.button_mc`

7.3.3.9 button_minus `calc_gui.Ui_MainWindow.button_minus`

7.3.3.10 button_ml `calc_gui.Ui_MainWindow.button_ml`

7.3.3.11 button_ms `calc_gui.Ui_MainWindow.button_ms`

7.3.3.12 button_multiply `calc_gui.Ui_MainWindow.button_multiply`

7.3.3.13 button_parent_c `calc_gui.Ui_MainWindow.button_parent_c`

7.3.3.14 `button_parent_o` `calc_gui.Ui_MainWindow.button_parent_o`

7.3.3.15 `button_plus` `calc_gui.Ui_MainWindow.button_plus`

7.3.3.16 `button_random` `calc_gui.Ui_MainWindow.button_random`

7.3.3.17 `button_root` `calc_gui.Ui_MainWindow.button_root`

7.3.3.18 `button_sroot` `calc_gui.Ui_MainWindow.button_sroot`

7.3.3.19 `centralwidget` `calc_gui.Ui_MainWindow.centralwidget`

7.3.3.20 `equal_button` `calc_gui.Ui_MainWindow.equal_button`

7.3.3.21 `frame` `calc_gui.Ui_MainWindow.frame`

The frame of the calculator.

7.3.3.22 `h_display` `calc_gui.Ui_MainWindow.h_display`

7.3.3.23 `h_equal_button` `calc_gui.Ui_MainWindow.h_equal_button`

7.3.3.24 `line` `calc_gui.Ui_MainWindow.line`

7.3.3.25 line_2 `calc_gui.Ui_MainWindow.line_2`

7.3.3.26 lName `calc_gui.Ui_MainWindow.lName`

7.3.3.27 lVersion `calc_gui.Ui_MainWindow.lVersion`

7.3.3.28 main_display `calc_gui.Ui_MainWindow.main_display`

7.3.3.29 menubar `calc_gui.Ui_MainWindow.menubar`

7.3.3.30 menuMenu `calc_gui.Ui_MainWindow.menuMenu`

7.3.3.31 n_button_0 `calc_gui.Ui_MainWindow.n_button_0`

7.3.3.32 n_button_1 `calc_gui.Ui_MainWindow.n_button_1`

7.3.3.33 n_button_2 `calc_gui.Ui_MainWindow.n_button_2`

7.3.3.34 n_button_3 `calc_gui.Ui_MainWindow.n_button_3`

7.3.3.35 n_button_4 `calc_gui.Ui_MainWindow.n_button_4`

7.3.3.36 `n_button_5` `calc_gui.Ui_MainWindow.n_button_5`

7.3.3.37 `n_button_6` `calc_gui.Ui_MainWindow.n_button_6`

7.3.3.38 `n_button_7` `calc_gui.Ui_MainWindow.n_button_7`

7.3.3.39 `n_button_8` `calc_gui.Ui_MainWindow.n_button_8`

7.3.3.40 `n_button_9` `calc_gui.Ui_MainWindow.n_button_9`

7.3.3.41 `n_button_comma` `calc_gui.Ui_MainWindow.n_button_comma`

7.3.3.42 `n_button_h_dot` `calc_gui.Ui_MainWindow.n_button_h_dot`

7.3.3.43 `n_button_sign` `calc_gui.Ui_MainWindow.n_button_sign`

7.3.3.44 `pWin` `calc_gui.Ui_MainWindow.pWin`

7.3.3.45 `textPwin` `calc_gui.Ui_MainWindow.textPwin`

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

- [calc_gui.py](#)

8 File Documentation

8.1 1000_num.txt File Reference

8.2 100_num.txt File Reference

8.3 10_num.txt File Reference

8.4 calc_gui.py File Reference

The interface of the calculator.

Classes

- class [calc_gui.Ui_MainWindow](#)
Class responsible for the proper look of GUI of the calculator.

Namespaces

- [calc_gui](#)

8.4.1 Detailed Description

The interface of the calculator.

Author

Martin Talajka

Date

19.4.2021

8.5 calc_main.py File Reference

The main logic of the calculator.

Classes

- class [calc_main.calcLogic](#)
Class responsible for the logical aspect of the calculator.

Namespaces

- [calc_main](#)

Variables

- `calc_main.app` = `QtWidgets.QApplication(sys.argv)`
- `calc_main.MainWindow` = `QtWidgets.QMainWindow()`
- `calc_main.ui` = `calcLogic()`

8.5.1 Detailed Description

The main logic of the calculator.

Author

Martin Talajka

Ondrej Kováč

Date

21.4.2021

8.6 mathlib.py File Reference

Math library defining math operations (+,-,*,/,!,^,nth_root, RNG)

Namespaces

- `mathlib`

Functions

- def `mathlib.add` (a, b)
Sum of 2 numbers.
- def `mathlib.sub` (a, b)
Subtraction of 2 numbers.
- def `mathlib.mul` (a, b)
Product of 2 numbers.
- def `mathlib.div` (a, b)
Division of 2 numbers.
- def `mathlib.fact` (n)
Factorial of a whole number.
- def `mathlib.exp` (b, n)
*Exponentiation of a base *b* by an exponent *n*.*
- def `mathlib.root` (n, x)
*Finding *n* th root of number *x*.*
- def `mathlib.rng` (x)
Producing a random number in a given range.

8.6.1 Detailed Description

Math library defining math operations (+,-,*,/,!,^,nth_root, RNG)

Author

Ondrej Kováč

Martin Talajka (optimization)

Date

13.4.2021

Index

- `__init__`
 - `calc_main.calcLogic`, 11
 - `1000_num.txt`, 34
 - `100_num.txt`, 34
 - `10_num.txt`, 34
- `aBinOp`
 - `calc_main.calcLogic`, 11
- `actionDark_mode`
 - `calc_gui.Ui_MainWindow`, 29
- `actionHelp`
 - `calc_gui.Ui_MainWindow`, 29
- `actionInformation`
 - `calc_gui.Ui_MainWindow`, 29
- `add`
 - `mathlib`, 4
- `aDecPoint`
 - `calc_main.calcLogic`, 11
- `aNum`
 - `calc_main.calcLogic`, 11
- `aParenthesis`
 - `calc_main.calcLogic`, 12
- `app`
 - `calc_main`, 3
- `aUnOp`
 - `calc_main.calcLogic`, 12
- `backSpace`
 - `calc_main.calcLogic`, 12
- `bin_ops`
 - `calc_main.calcLogic`, 25
- `button_ce`
 - `calc_gui.Ui_MainWindow`, 30
- `button_delete`
 - `calc_gui.Ui_MainWindow`, 30
- `button_divide`
 - `calc_gui.Ui_MainWindow`, 30
- `button_fact`
 - `calc_gui.Ui_MainWindow`, 30
- `button_mc`
 - `calc_gui.Ui_MainWindow`, 30
- `button_minus`
 - `calc_gui.Ui_MainWindow`, 30
- `button_ml`
 - `calc_gui.Ui_MainWindow`, 30
- `button_ms`
 - `calc_gui.Ui_MainWindow`, 30
- `button_multiply`
 - `calc_gui.Ui_MainWindow`, 30
- `button_parent_c`
 - `calc_gui.Ui_MainWindow`, 30
- `button_parent_o`
 - `calc_gui.Ui_MainWindow`, 30
- `button_plus`
 - `calc_gui.Ui_MainWindow`, 31
- `button_random`
 - `calc_gui.Ui_MainWindow`, 31
- `button_root`
 - `calc_gui.Ui_MainWindow`, 31
- `button_sroot`
 - `calc_gui.Ui_MainWindow`, 31

- `calc_gui`, 3
- `calc_gui.py`, 34
- `calc_gui.Ui_MainWindow`, 27
 - `actionDark_mode`, 29
 - `actionHelp`, 29
 - `actionInformation`, 29
 - `button_ce`, 30
 - `button_delete`, 30
 - `button_divide`, 30
 - `button_fact`, 30
 - `button_mc`, 30
 - `button_minus`, 30
 - `button_ml`, 30
 - `button_ms`, 30
 - `button_multiply`, 30
 - `button_parent_c`, 30
 - `button_parent_o`, 30
 - `button_plus`, 31
 - `button_random`, 31
 - `button_root`, 31
 - `button_sroot`, 31
 - `centralwidget`, 31
 - `equal_button`, 31
 - `frame`, 31
 - `h_display`, 31
 - `h_equal_button`, 31
 - `line`, 31
 - `line_2`, 31
 - `lName`, 32
 - `lVersion`, 32
 - `main_display`, 32
 - `menubar`, 32
 - `menuMenu`, 32
 - `n_button_0`, 32
 - `n_button_1`, 32
 - `n_button_2`, 32
 - `n_button_3`, 32
 - `n_button_4`, 32
 - `n_button_5`, 32
 - `n_button_6`, 33
 - `n_button_7`, 33
 - `n_button_8`, 33
 - `n_button_9`, 33
 - `n_button_comma`, 33
 - `n_button_h_dot`, 33
 - `n_button_sign`, 33
 - `pWin`, 33
 - `retranslateUi`, 29
 - `setupUi`, 29

- textPwin, 33
- calc_main, 3
 - app, 3
 - MainWindow, 4
 - ui, 4
- calc_main.calcLogic, 8
 - __init__, 11
 - aBinOp, 11
 - aDecPoint, 11
 - aNum, 11
 - aParenthesis, 12
 - aUnOp, 12
 - backSpace, 12
 - bin_ops, 25
 - calcExp, 12
 - calcFact, 13
 - calcMulDiv, 14
 - calcPlusMin, 14
 - calcRnd, 15
 - calcRoot, 16
 - calculate, 17
 - calculateString, 18
 - changeSign, 19
 - clearEverything, 19
 - dec_p, 25
 - error, 25
 - errorHandler, 20
 - findAllPositions, 20
 - findParPairs, 21
 - getLeftOperand, 21
 - getRightOperand, 22
 - md_text, 25
 - memClear, 23
 - memLoad, 23
 - memory, 25
 - memSet, 23
 - oPDF_g, 23
 - open_par, 26
 - parentheses, 26
 - repairInput, 24
 - repairOutput, 24
 - result, 26
 - sColor, 25
 - sd_text, 26
 - un_ops, 26
- calc_main.py, 34
- calcExp
 - calc_main.calcLogic, 12
- calcFact
 - calc_main.calcLogic, 13
- calcMulDiv
 - calc_main.calcLogic, 14
- calcPlusMin
 - calc_main.calcLogic, 14
- calcRnd
 - calc_main.calcLogic, 15
- calcRoot
 - calc_main.calcLogic, 16
- calculate
 - calc_main.calcLogic, 17
- calculateString
 - calc_main.calcLogic, 18
- centralwidget
 - calc_gui.Ui_MainWindow, 31
- changeSign
 - calc_main.calcLogic, 19
- clearEverything
 - calc_main.calcLogic, 19
- dec_p
 - calc_main.calcLogic, 25
- div
 - mathlib, 4
- equal_button
 - calc_gui.Ui_MainWindow, 31
- error
 - calc_main.calcLogic, 25
- errorHandler
 - calc_main.calcLogic, 20
- exp
 - mathlib, 5
- fact
 - mathlib, 5
- findAllPositions
 - calc_main.calcLogic, 20
- findParPairs
 - calc_main.calcLogic, 21
- frame
 - calc_gui.Ui_MainWindow, 31
- getLeftOperand
 - calc_main.calcLogic, 21
- getRightOperand
 - calc_main.calcLogic, 22
- h_display
 - calc_gui.Ui_MainWindow, 31
- h_equal_button
 - calc_gui.Ui_MainWindow, 31
- line
 - calc_gui.Ui_MainWindow, 31
- line_2
 - calc_gui.Ui_MainWindow, 31
- lName
 - calc_gui.Ui_MainWindow, 32
- lVersion
 - calc_gui.Ui_MainWindow, 32
- main_display
 - calc_gui.Ui_MainWindow, 32
- MainWindow
 - calc_main, 4
- mathlib, 4
 - add, 4
 - div, 4

- exp, [5](#)
- fact, [5](#)
- mul, [6](#)
- rng, [6](#)
- root, [7](#)
- sub, [7](#)
- mathlib.py, [35](#)
- md_text
 - calc_main.calcLogic, [25](#)
- memClear
 - calc_main.calcLogic, [23](#)
- memLoad
 - calc_main.calcLogic, [23](#)
- memory
 - calc_main.calcLogic, [25](#)
- memSet
 - calc_main.calcLogic, [23](#)
- menubar
 - calc_gui.Ui_MainWindow, [32](#)
- menuMenu
 - calc_gui.Ui_MainWindow, [32](#)
- mul
 - mathlib, [6](#)
- n_button_0
 - calc_gui.Ui_MainWindow, [32](#)
- n_button_1
 - calc_gui.Ui_MainWindow, [32](#)
- n_button_2
 - calc_gui.Ui_MainWindow, [32](#)
- n_button_3
 - calc_gui.Ui_MainWindow, [32](#)
- n_button_4
 - calc_gui.Ui_MainWindow, [32](#)
- n_button_5
 - calc_gui.Ui_MainWindow, [32](#)
- n_button_6
 - calc_gui.Ui_MainWindow, [33](#)
- n_button_7
 - calc_gui.Ui_MainWindow, [33](#)
- n_button_8
 - calc_gui.Ui_MainWindow, [33](#)
- n_button_9
 - calc_gui.Ui_MainWindow, [33](#)
- n_button_comma
 - calc_gui.Ui_MainWindow, [33](#)
- n_button_h_dot
 - calc_gui.Ui_MainWindow, [33](#)
- n_button_sign
 - calc_gui.Ui_MainWindow, [33](#)
- object, [27](#)
- oPDF_g
 - calc_main.calcLogic, [23](#)
- open_par
 - calc_main.calcLogic, [26](#)
- parentheses
 - calc_main.calcLogic, [26](#)
- pWin
 - calc_gui.Ui_MainWindow, [33](#)
- repairInput
 - calc_main.calcLogic, [24](#)
- repairOutput
 - calc_main.calcLogic, [24](#)
- result
 - calc_main.calcLogic, [26](#)
- retranslateUi
 - calc_gui.Ui_MainWindow, [29](#)
- rng
 - mathlib, [6](#)
- root
 - mathlib, [7](#)
- sColor
 - calc_main.calcLogic, [25](#)
- sd_text
 - calc_main.calcLogic, [26](#)
- setupUi
 - calc_gui.Ui_MainWindow, [29](#)
- sub
 - mathlib, [7](#)
- textPwin
 - calc_gui.Ui_MainWindow, [33](#)
- ui
 - calc_main, [4](#)
- un_ops
 - calc_main.calcLogic, [26](#)