

Team_Not_Found_Calc

Generated by Doxygen 1.13.2

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 calc_lib Namespace Reference	9
5.1.1 Detailed Description	9
5.1.2 Function Documentation	9
5.1.2.1 absolute()	9
5.1.2.2 add()	10
5.1.2.3 div()	10
5.1.2.4 expon()	10
5.1.2.5 factorial()	10
5.1.2.6 fib()	10
5.1.2.7 mul()	11
5.1.2.8 sqr()	11
5.1.2.9 sub()	11
5.2 mockup Namespace Reference	11
5.2.1 Function Documentation	12
5.2.1.1 backspace()	12
5.2.1.2 calculate()	12
5.2.1.3 call_factorial()	12
5.2.1.4 call_fibonacci()	12
5.2.1.5 call_power()	12
5.2.1.6 call_root()	12
5.2.1.7 clear()	12
5.2.1.8 insert_char()	13
5.2.1.9 key_handler()	13
5.2.1.10 show_help()	13
5.2.2 Variable Documentation	13
5.2.2.1 btn	13
5.2.2.2 buttons	13
5.2.2.3 col	13
5.2.2.4 column	14
5.2.2.5 colspanspan	14
5.2.2.6 entry	14

5.2.2.7 padx	14
5.2.2.8 pady	14
5.2.2.9 root	14
5.2.2.10 row	14
5.3 profiling Namespace Reference	14
5.3.1 Detailed Description	15
5.3.2 Variable Documentation	15
5.3.2.1 count	15
5.3.2.2 numCount	15
5.3.2.3 nums	15
5.3.2.4 prum	15
5.3.2.5 prumExp	15
5.4 test_calc_lib Namespace Reference	15
6 Class Documentation	17
6.1 test_calc_lib.TestMyMathLib Class Reference	17
6.1.1 Detailed Description	18
6.1.2 Member Function Documentation	18
6.1.2.1 test_abs()	18
6.1.2.2 test_add()	18
6.1.2.3 test_div()	18
6.1.2.4 test_div_zero()	18
6.1.2.5 test_expon()	18
6.1.2.6 test_factorial()	18
6.1.2.7 test_factorial_value_err()	19
6.1.2.8 test_fib()	19
6.1.2.9 test_mul()	19
6.1.2.10 test_sqr()	19
6.1.2.11 test_sub()	19
6.1.2.12 test_value()	19
7 File Documentation	21
7.1 calc_lib.py File Reference	21
7.2 mockup.py File Reference	21
7.3 profiling.py File Reference	22
7.4 test_calc_lib.py File Reference	22
7.4.1 Detailed Description	22
Index	23

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

calc_lib	9
mockup	11
profiling	14
test_calc_lib	15

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

```
unittest.TestCase
test_calc_lib.TestMyMathLib . . . . . 17
```

Chapter 3

Class Index

3.1 Class List

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

test_calc_lib.TestMyMathLib	
Obsahuje testovací metody pro různé matematické operace	17

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

calc_lib.py	21
mockup.py	21
profiling.py	22
test_calc_lib.py	
Unit testy pro matematické funkce v knihovně calc_lib	22

Chapter 5

Namespace Documentation

5.1 calc_lib Namespace Reference

Functions

- [add](#) (num1, num2)
- [sub](#) (num1, num2)
- [mul](#) (num1, num2)
- [div](#) (num1, num2)
- [factorial](#) (num1)
- [expon](#) (num1, num2)
- [sqr](#) (num1, num2)
- [absolute](#) (num1)
- [fib](#) (num1)

5.1.1 Detailed Description

```
@file calc_lib.py
@brief knihovna matematických funkcí
@details jsou volány funkce které načítají jedno nebo dvě čísla a ověřuje jejich integritu s tím že vrací
@author Adam Kadlec
```

5.1.2 Function Documentation

5.1.2.1 absolute()

```
calc_lib.absolute (
    num1)
```

```
@brief Vrábí absolutní hodnotu čísla.
@param num1 Číslo (int nebo float).
@return Vrací absolutní hodnotu, nebo chybovou hlášku (float nebo str).
```


5.1.2.2 add()

```
calc_lib.add (  
    num1,  
    num2)  
  
@brief Sčítá dvě čísla.  
@param num1 První číslo (int nebo float).  
@param num2 Druhé číslo (int nebo float).  
@return Vrací součet dvou čísel, nebo chybovou hlášku (int, float nebo str).
```

5.1.2.3 div()

```
calc_lib.div (  
    num1,  
    num2)  
  
@brief Dělí dvě čísla.  
@param num1 Dělenec (int nebo float).  
@param num2 Dělitel (int nebo float).  
@return Vrací podíl dvou čísel, nebo chybovou hlášku (float nebo str).
```

5.1.2.4 expon()

```
calc_lib.expon (  
    num1,  
    num2)  
  
@brief Umocní číslo.  
@param num1 Základ (int nebo float).  
@param num2 Exponent (int nebo float).  
@return Vrací výsledek umocnění, nebo chybovou hlášku (float nebo str).
```

5.1.2.5 factorial()

```
calc_lib.factorial (  
    num1)  
  
@brief Vypočítá faktoriál čísla.  
@param num1 Celé číslo (int).  
@return Vrací faktoriál čísla, nebo chybovou hlášku (int nebo str).
```

5.1.2.6 fib()

```
calc_lib.fib (  
    num1)  
  
@brief Vypočítá Fibonacciho číslo na dané pozici.  
@param num1 Pořadí v posloupnosti (int).  
@return Vrací číslo z Fibonacciho posloupnosti, nebo chybovou hlášku (int nebo str).
```

5.1.2.7 mul()

```
calc_lib.mul (  
    num1,  
    num2)  
  
@brief Násobí dvě čísla.  
@param num1 První číslo (int nebo float).  
@param num2 Druhé číslo (int nebo float).  
@return Vrací součin dvou čísel, nebo chybovou hlášku (int, float nebo str).
```

5.1.2.8 sqr()

```
calc_lib.sqr (  
    num1,  
    num2)  
  
@brief Vypočítá odmocninu čísla.  
@param num1 Odmocňované číslo (int nebo float).  
@param num2 Kolikátá odmocnina (int nebo float).  
@return Vrací výsledek odmocnění, nebo chybovou hlášku (float nebo str).
```

5.1.2.9 sub()

```
calc_lib.sub (  
    num1,  
    num2)  
  
@brief Odečítá dvě čísla.  
@param num1 První číslo (int nebo float).  
@param num2 Druhé číslo (int nebo float).  
@return Vrací rozdíl dvou čísel, nebo chybovou hlášku (int, float nebo str).
```

5.2 mockup Namespace Reference

Functions

- [insert_char](#) (char)
- [clear](#) ()
- [backspace](#) ()
- [calculate](#) ()
- [call_factorial](#) ()
- [call_power](#) ()
- [call_root](#) ()
- [call_fibonacci](#) ()
- [show_help](#) ()
- [key_handler](#) (event)

Variables

- `root` = `tk.Tk()`
- `entry` = `tk.Entry(root, width=25, font=("Arial", 16), justify='right')`
- `row` = 1
- `column`
- `columnspan`
- `padx`
- `pady`
- list `buttons`
- int `col` = 0
- `btn` = `tk.Button(root, text=text, width=5, height=2, command=command)`

5.2.1 Function Documentation

5.2.1.1 `backspace()`

```
mockup.backspace ()
```

5.2.1.2 `calculate()`

```
mockup.calculate ()
```

5.2.1.3 `call_factorial()`

```
mockup.call_factorial ()
```

5.2.1.4 `call_fibonacci()`

```
mockup.call_fibonacci ()
```

5.2.1.5 `call_power()`

```
mockup.call_power ()
```

5.2.1.6 `call_root()`

```
mockup.call_root ()
```

5.2.1.7 `clear()`

```
mockup.clear ()
```

5.2.1.8 insert_char()

```
mockup.insert_char (
    char)
```

5.2.1.9 key_handler()

```
mockup.key_handler (
    event)
```

5.2.1.10 show_help()

```
mockup.show_help ()
```

5.2.2 Variable Documentation

5.2.2.1 btn

```
mockup.btn = tk.Button(root, text=text, width=5, height=2, command=command)
```

5.2.2.2 buttons

```
list mockup.buttons
```

Initial value:

```
00001 = [
00002     ("7", lambda: insert_char("7")),
00003     ("8", lambda: insert_char("8")),
00004     ("9", lambda: insert_char("9")),
00005     ("/", lambda: insert_char("/")),
00006     ("4", lambda: insert_char("4")),
00007     ("5", lambda: insert_char("5")),
00008     ("6", lambda: insert_char("6")),
00009     ("*", lambda: insert_char("*")),
00010     ("1", lambda: insert_char("1")),
00011     ("2", lambda: insert_char("2")),
00012     ("3", lambda: insert_char("3")),
00013     ("-", lambda: insert_char("-")),
00014     ("0", lambda: insert_char("0")),
00015     (".", lambda: insert_char(".")),
00016     ("+", lambda: insert_char("+")),
00017     ("=", calculate),
00018     ("C", clear),
00019     (" ", backspace),
00020     ("!", call_factorial),
00021     ("^", lambda: insert_char("^")),
00022     (" ", lambda: insert_char(" ")),
00023     ("^=", call_power),
00024     ("=", call_root),
00025     ("Fib", call_fibonacci),
00026     ("?", show_help)
00027 ]
```

5.2.2.3 col

```
int mockup.col = 0
```

5.2.2.4 column

```
mockup.column
```

5.2.2.5 colspan

```
mockup.colspan
```

5.2.2.6 entry

```
mockup.entry = tk.Entry(root, width=25, font=("Arial", 16), justify='right')
```

5.2.2.7 padx

```
mockup.padx
```

5.2.2.8 pady

```
mockup.pady
```

5.2.2.9 root

```
mockup.root = tk.Tk()
```

5.2.2.10 row

```
int mockup.row = 1
```

5.3 profiling Namespace Reference

Variables

- list `nums` = []
- int `numCount` = 0
načítá čísla ze stdin a kontroluje zda se jedná o číslo.
- int `count` = 0
počítá celkovou hodnotu všech čísel
- `prum` = `div(numCount,count)`
počítá kolik čísel načte
- int `prumExp` = 0
počítá průměr

5.3.1 Detailed Description

```
@file profiling.py
@author Adam Kadlec
@brief Výpočet výběrové směrodatné odchylky ze standardního vstupu pomocí vlastní matematické knihovny.
@details Program načítá čísla ze standardního vstupu, vypočítá aritmetický průměr a následně výběrovou směrodatnou odchylku.

@section usage Usage
Spuštění programu s čísly zadanými na standardním vstupu. Program načítá čísla, počítá průměr a následně výběrovou směrodatnou odchylku.

@param nums Seznam čísel, která se načítají ze vstupu.
@param numCount Součet všech čísel načtených ze vstupu.
@param count Počet načtených čísel.

@return Vrací výběrovou směrodatnou odchylku pomocí výpisu na stdout.
```

5.3.2 Variable Documentation

5.3.2.1 count

```
profiling.count = 0
```

počítá celkovou hodnotu všech čísel

5.3.2.2 numCount

```
profiling.numCount = 0
```

načítá čísla ze stdin a kontroluje zda se jedná o číslo.

ukládá čísla do listu nums

5.3.2.3 nums

```
list profiling.nums = []
```

5.3.2.4 prum

```
profiling.prum = div(numCount, count)
```

počítá kolik čísel načte

5.3.2.5 prumExp

```
profiling.prumExp = 0
```

počítá průměr

prochází list hodnot a umocňuje je a potom sčítá

5.4 test_calc_lib Namespace Reference

Classes

- class [TestMyMathLib](#)

Obsahuje testovací metody pro různé matematické operace.

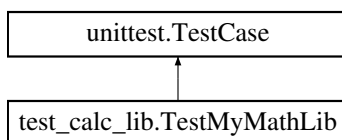
Chapter 6

Class Documentation

6.1 test_calc_lib.TestMyMathLib Class Reference

Obsahuje testovací metody pro různé matematické operace.

Inheritance diagram for test_calc_lib.TestMyMathLib:



Public Member Functions

- [test_add](#) (self)
Testy pro funkci add (sčítání).
- [test_sub](#) (self)
Testy pro funkci sub (odčítání).
- [test_mul](#) (self)
Testy pro funkci mul (násobení).
- [test_div](#) (self)
Testy pro funkci div (dělení).
- [test_div_zero](#) (self)
Test dělení nulou.
- [test_factorial](#) (self)
Testy pro funkci factorial.
- [test_factorial_value_err](#) (self)
Testy pro factorial s neplatnými vstupy.
- [test_expon](#) (self)
Testy pro funkci expon (mocnění).
- [test_sqr](#) (self)
Testy pro funkci sqr (odmocnina).
- [test_fib](#) (self)
Testy pro Fibonacciho posloupnost.
- [test_abs](#) (self)
Testy pro absolutní hodnotu.
- [test_value](#) (self)
Testy validace vstupních typů (např.

6.1.1 Detailed Description

Obsahuje testovací metody pro různé matematické operace.

6.1.2 Member Function Documentation

6.1.2.1 test_abs()

```
test_calc_lib.TestMyMathLib.test_abs (  
    self)
```

Testy pro absolutní hodnotu.

6.1.2.2 test_add()

```
test_calc_lib.TestMyMathLib.test_add (  
    self)
```

Testy pro funkci add (sčítání).

6.1.2.3 test_div()

```
test_calc_lib.TestMyMathLib.test_div (  
    self)
```

Testy pro funkci div (dělení).

6.1.2.4 test_div_zero()

```
test_calc_lib.TestMyMathLib.test_div_zero (  
    self)
```

Test dělení nulou.

6.1.2.5 test_expon()

```
test_calc_lib.TestMyMathLib.test_expon (  
    self)
```

Testy pro funkci expon (mocnění).

6.1.2.6 test_factorial()

```
test_calc_lib.TestMyMathLib.test_factorial (  
    self)
```

Testy pro funkci factorial.

6.1.2.7 test_factorial_value_err()

```
test_calc_lib.TestMyMathLib.test_factorial_value_err (
    self)
```

Testy pro factorial s neplatnými vstupy.

6.1.2.8 test_fib()

```
test_calc_lib.TestMyMathLib.test_fib (
    self)
```

Testy pro Fibonacciho posloupnost.

6.1.2.9 test_mul()

```
test_calc_lib.TestMyMathLib.test_mul (
    self)
```

Testy pro funkci mul (násobení).

6.1.2.10 test_sqr()

```
test_calc_lib.TestMyMathLib.test_sqr (
    self)
```

Testy pro funkci sqr (odmocnina).

6.1.2.11 test_sub()

```
test_calc_lib.TestMyMathLib.test_sub (
    self)
```

Testy pro funkci sub (odčítání).

6.1.2.12 test_value()

```
test_calc_lib.TestMyMathLib.test_value (
    self)
```

Testy validace vstupních typů (např.

string místo čísla).

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

- [test_calc_lib.py](#)

Chapter 7

File Documentation

7.1 `calc_lib.py` File Reference

Namespaces

- namespace [calc_lib](#)

Functions

- [calc_lib.add](#) (num1, num2)
- [calc_lib.sub](#) (num1, num2)
- [calc_lib.mul](#) (num1, num2)
- [calc_lib.div](#) (num1, num2)
- [calc_lib.factorial](#) (num1)
- [calc_lib.expon](#) (num1, num2)
- [calc_lib.sqr](#) (num1, num2)
- [calc_lib.absolute](#) (num1)
- [calc_lib.fib](#) (num1)

7.2 `mockup.py` File Reference

Namespaces

- namespace [mockup](#)

Functions

- [mockup.insert_char](#) (char)
- [mockup.clear](#) ()
- [mockup.backspace](#) ()
- [mockup.calculate](#) ()
- [mockup.call_factorial](#) ()
- [mockup.call_power](#) ()
- [mockup.call_root](#) ()
- [mockup.call_fibonacci](#) ()
- [mockup.show_help](#) ()
- [mockup.key_handler](#) (event)

Variables

- `mockup.root` = `tk.Tk()`
- `mockup.entry` = `tk.Entry(root, width=25, font=("Arial", 16), justify='right')`
- `mockup.row` = 1
- `mockup.column`
- `mockup.columnspan`
- `mockup.padx`
- `mockup.pady`
- list `mockup.buttons`
- int `mockup.col` = 0
- `mockup.btn` = `tk.Button(root, text=text, width=5, height=2, command=command)`

7.3 profiling.py File Reference

Namespaces

- namespace `profiling`

Variables

- list `profiling.num`s = []
- int `profiling.numCount` = 0
načítá čísla ze stdin a kontroluje zda se jedná o číslo.
- int `profiling.count` = 0
počítá celkovou hodnotu všech čísel
- `profiling.prum` = `div(numCount,count)`
počítá kolik čísel načte
- int `profiling.prumExp` = 0
počítá průměr

7.4 test_calc_lib.py File Reference

Unit testy pro matematické funkce v knihovně `calc_lib`.

Classes

- class `test_calc_lib.TestMyMathLib`
Obsahuje testovací metody pro různé matematické operace.

Namespaces

- namespace `test_calc_lib`

7.4.1 Detailed Description

Unit testy pro matematické funkce v knihovně `calc_lib`.

Tento soubor obsahuje rozsáhlé testy pro funkce jako sčítání, odčítání, násobení, dělení, faktoriál, mocniny, odmocniny, Fibonacciho posloupnost, absolutní hodnotu a validaci vstupů.

Index

- absolute
 - [calc_lib, 9](#)
- add
 - [calc_lib, 9](#)
- backspace
 - [mockup, 12](#)
- btn
 - [mockup, 13](#)
- buttons
 - [mockup, 13](#)
- [calc_lib, 9](#)
 - [absolute, 9](#)
 - [add, 9](#)
 - [div, 10](#)
 - [expon, 10](#)
 - [factorial, 10](#)
 - [fib, 10](#)
 - [mul, 10](#)
 - [sqr, 11](#)
 - [sub, 11](#)
- [calc_lib.py, 21](#)
- calculate
 - [mockup, 12](#)
- call_factorial
 - [mockup, 12](#)
- call_fibonacci
 - [mockup, 12](#)
- call_power
 - [mockup, 12](#)
- call_root
 - [mockup, 12](#)
- clear
 - [mockup, 12](#)
- col
 - [mockup, 13](#)
- column
 - [mockup, 13](#)
- columnspan
 - [mockup, 14](#)
- count
 - [profiling, 15](#)
- div
 - [calc_lib, 10](#)
- entry
 - [mockup, 14](#)
- expon
 - [calc_lib, 10](#)
- factorial
 - [calc_lib, 10](#)
- fib
 - [calc_lib, 10](#)
- insert_char
 - [mockup, 12](#)
- key_handler
 - [mockup, 13](#)
- mockup, 11
 - [backspace, 12](#)
 - [btn, 13](#)
 - [buttons, 13](#)
 - [calculate, 12](#)
 - [call_factorial, 12](#)
 - [call_fibonacci, 12](#)
 - [call_power, 12](#)
 - [call_root, 12](#)
 - [clear, 12](#)
 - [col, 13](#)
 - [column, 13](#)
 - [columnspan, 14](#)
 - [entry, 14](#)
 - [insert_char, 12](#)
 - [key_handler, 13](#)
 - [padx, 14](#)
 - [pady, 14](#)
 - [root, 14](#)
 - [row, 14](#)
 - [show_help, 13](#)
- [mockup.py, 21](#)
- mul
 - [calc_lib, 10](#)
- numCount
 - [profiling, 15](#)
- nums
 - [profiling, 15](#)
- padx
 - [mockup, 14](#)
- pady
 - [mockup, 14](#)
- profiling, 14
 - [count, 15](#)
 - [numCount, 15](#)
 - [nums, 15](#)

- prum, [15](#)
 - prumExp, [15](#)
- profiling.py, [22](#)
- prum
 - profiling, [15](#)
- prumExp
 - profiling, [15](#)
- root
 - mockup, [14](#)
- row
 - mockup, [14](#)
- show_help
 - mockup, [13](#)
- sqr
 - calc_lib, [11](#)
- sub
 - calc_lib, [11](#)
- test_abs
 - test_calc_lib.TestMyMathLib, [18](#)
- test_add
 - test_calc_lib.TestMyMathLib, [18](#)
- test_calc_lib, [15](#)
- test_calc_lib.py, [22](#)
- test_calc_lib.TestMyMathLib, [17](#)
 - test_abs, [18](#)
 - test_add, [18](#)
 - test_div, [18](#)
 - test_div_zero, [18](#)
 - test_expon, [18](#)
 - test_factorial, [18](#)
 - test_factorial_value_err, [18](#)
 - test_fib, [19](#)
 - test_mul, [19](#)
 - test_sqr, [19](#)
 - test_sub, [19](#)
 - test_value, [19](#)
- test_div
 - test_calc_lib.TestMyMathLib, [18](#)
- test_div_zero
 - test_calc_lib.TestMyMathLib, [18](#)
- test_expon
 - test_calc_lib.TestMyMathLib, [18](#)
- test_factorial
 - test_calc_lib.TestMyMathLib, [18](#)
- test_factorial_value_err
 - test_calc_lib.TestMyMathLib, [18](#)
- test_fib
 - test_calc_lib.TestMyMathLib, [19](#)
- test_mul
 - test_calc_lib.TestMyMathLib, [19](#)
- test_sqr
 - test_calc_lib.TestMyMathLib, [19](#)
- test_sub
 - test_calc_lib.TestMyMathLib, [19](#)
- test_value
 - test_calc_lib.TestMyMathLib, [19](#)