

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 Detailed Description	12
5.2.2 Variable Documentation	12
5.2.2.1 app	12
5.2.2.2 root	12
5.3 stddev Namespace Reference	12
5.3.1 Detailed Description	12
5.3.2 Function Documentation	13
5.3.2.1 calculate_stddev()	13
5.3.2.2 run_profiled_calculations()	13
5.3.3 Variable Documentation	13
5.3.3.1 INPUT_SIZE	13
5.4 test_calc_lib Namespace Reference	13
6 Class Documentation	15
6.1 mockup.Calculator Class Reference	15
6.1.1 Detailed Description	16
6.1.2 Constructor & Destructor Documentation	16
6.1.2.1 __init__()	16

6.1.3 Member Function Documentation	16
6.1.3.1 append_number()	16
6.1.3.2 backspace()	16
6.1.3.3 calculate_absolute()	16
6.1.3.4 calculate_factorial()	17
6.1.3.5 calculate_fibonacci()	17
6.1.3.6 clear()	17
6.1.3.7 create_button()	17
6.1.3.8 create_buttons()	17
6.1.3.9 create_menu()	18
6.1.3.10 display_error()	18
6.1.3.11 evaluate()	18
6.1.3.12 negate()	18
6.1.3.13 prepare_operation()	18
6.1.3.14 setup_keyboard_bindings()	18
6.1.3.15 show_about()	19
6.1.3.16 show_help()	19
6.1.3.17 update_display()	19
6.1.4 Member Data Documentation	19
6.1.4.1 buttons_frame	19
6.1.4.2 current_expression	19
6.1.4.3 display_frame	19
6.1.4.4 expecting_second_number	19
6.1.4.5 expression_label	20
6.1.4.6 first_number	20
6.1.4.7 just_evaluated	20
6.1.4.8 pending_operation	20
6.1.4.9 result	20
6.1.4.10 result_label	20
6.1.4.11 root	20
6.1.4.12 second_number	20
6.1.4.13 total_expression	21
6.2 test_calc_lib.TestMyMathLib Class Reference	21
6.2.1 Detailed Description	22
6.2.2 Member Function Documentation	22
6.2.2.1 test_abs()	22
6.2.2.2 test_add()	22
6.2.2.3 test_div()	22
6.2.2.4 test_div_zero()	22
6.2.2.5 test_expon()	22
6.2.2.6 test_factorial()	22
6.2.2.7 test_factorial_value_err()	23

6.2.2.8 test_fib()	23
6.2.2.9 test_mul()	23
6.2.2.10 test_sqr()	23
6.2.2.11 test_sub()	23
6.2.2.12 test_value()	23
7 File Documentation	25
7.1 calc_lib.py File Reference	25
7.2 mockup.py File Reference	25
7.3 stddev.py File Reference	26
7.4 test_calc_lib.py File Reference	26
7.4.1 Detailed Description	26
Index	27

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

calc_lib	9
mockup	11
stddev	12
test_calc_lib	13

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

mockup.Calculator	15
unittest.TestCase	
test_calc_lib.TestMyMathLib	21

Chapter 3

Class Index

3.1 Class List

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

mockup.Calculator	15
test_calc_lib.TestMyMathLib Obsahuje testovací metody pro různé matematické operace	21

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

calc_lib.py	25
mockup.py	25
stddev.py	26
test_calc_lib.py	
Unit testy pro matematické funkce v knihovně calc_lib	26

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 Vrací 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

Classes

- class [Calculator](#)

Variables

- [root](#) = tk.Tk()
- [app](#) = [Calculator](#)(root)

5.2.1 Detailed Description

```
@file calculator.py
@brief GUI kalkulačka s běžnými a pokročilými matematickými funkcemi.

@details Tento modul implementuje grafickou kalkulačku pomocí Tkinteru.
Poskytuje základní aritmetické operace, stejně jako pokročilé funkce
jako faktoriály, exponenty, Fibonacciho posloupnost a absolutní hodnoty.
Kalkulačka podporuje zadávání jak myši, tak klávesnicí.

@author William Denis Tihelka
@date 30. dubna 2025
@version 1.0
```

5.2.2 Variable Documentation

5.2.2.1 app

```
mockup.app = Calculator(root)
```

5.2.2.2 root

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

5.3 stddev Namespace Reference

Functions

- `calculate_stddev()`
Funkce pro výpočet výběrové směrodatné odchylky ze vstupu.
- `run_profiled_calculations()`
Spuštění výpočtu s měřením výkonnosti pomocí cProfile.

Variables

- `int INPUT_SIZE = 10`

5.3.1 Detailed Description

```
@file profiling.py
@author Adam Kadlec
@brief Výpočet výběrové směrodatné odchylky s náhodně generovanými vstupy a profilováním.
@details Tento skript načítá číselné hodnoty ze standardního vstupu, počítá výběrovou směrodatnou odchylku pomocí
a zároveň měří výkon pomocí profileru cProfile.
```

5.3.2 Function Documentation

5.3.2.1 calculate_stddev()

```
stddev.calculate_stddev ()
```

Funkce pro výpočet výběrové směrodatné odchylky ze vstupu.

Načítá čísla ze stdin, počítá jejich součet, průměr a průměr čtverců, poté aplikuje vzorec pro směrodatnou odchylku.

Returns

Výsledek (směrodatná odchylka) se vypíše na standardní výstup.

5.3.2.2 run_profiled_calculations()

```
stddev.run_profiled_calculations ()
```

Spuštění výpočtu s měřením výkonnosti pomocí cProfile.

Profiluje funkci calculate_stddev a ukládá výstup do souboru profiling_output.txt.

5.3.3 Variable Documentation

5.3.3.1 INPUT_SIZE

```
int stddev.INPUT_SIZE = 10
```

5.4 test_calc_lib Namespace Reference

Classes

- class [TestMyMathLib](#)

Obsahuje testovací metody pro různé matematické operace.

Chapter 6

Class Documentation

6.1 mockup.Calculator Class Reference

Public Member Functions

- `__init__` (self, root)
- `create_menu` (self)
- `show_help` (self)
- `show_about` (self)
- `setup_keyboard_bindings` (self)
- `create_buttons` (self)
- `create_button` (self, text, row, column, colspan=1, rowspan=1, command=None, bg="#e0e0e0")
- `append_number` (self, number)
- `prepare_operation` (self, operation)
- `clear` (self)
- `backspace` (self)
- `calculate_factorial` (self)
- `calculate_fibonacci` (self)
- `calculate_absolute` (self)
- `evaluate` (self)
- `display_error` (self, error_code)
- `negate` (self)
- `update_display` (self, evaluated=False)

Public Attributes

- `root` = root
- str `current_expression` = ""
- str `total_expression` = ""
- str `pending_operation` = None
- `first_number` = None
- `second_number` = None
- `result` = None
- bool `expecting_second_number` = False
- bool `just_evaluated` = False
- `display_frame` = tk.Frame(self.root, height=100, bg="#f0f0f0")
- `expression_label`
- `result_label`
- `buttons_frame` = tk.Frame(self.root, bg="#f0f0f0")

6.1.1 Detailed Description

```
@class Calculator
@brief Grafická kalkulačka s podporou základních a pokročilých matematických operací.

@details Tato třída implementuje kalkulačku s grafickým uživatelským rozhraním pomocí Tkinteru.
Poskytuje standardní aritmetické operace, pokročilé funkce jako faktoriál a Fibonacciho posloupnost
a podporuje jak klikání na tlačítka, tak zadávání z klávesnice.
```

6.1.2 Constructor & Destructor Documentation

6.1.2.1 `__init__()`

```
mockup.Calculator.__init__ (
    self,
    root)

@brief Inicializace grafického rozhraní kalkulačky.

@details Nastaví hlavní okno, zobrazovací oblasti, tlačítka a vazby na klávesnici.

@param root Hlavní okno Tkinteru.
```

6.1.3 Member Function Documentation

6.1.3.1 `append_number()`

```
mockup.Calculator.append_number (
    self,
    number)

@brief Přidání číslíce nebo desetinné čárky do aktuálního výrazu.

@details Zpracovává logiku pro různé stavy: zahájení nového výpočtu,
zadávání prvního čísla nebo zadávání druhého čísla.

@param number Číslíce nebo desetinná čárka, která se má přidat.
```

6.1.3.2 `backspace()`

```
mockup.Calculator.backspace (
    self)
```

6.1.3.3 `calculate_absolute()`

```
mockup.Calculator.calculate_absolute (
    self)
```

6.1.3.4 calculate_factorial()

```
mockup.Calculator.calculate_factorial (
    self)
```

6.1.3.5 calculate_fibonacci()

```
mockup.Calculator.calculate_fibonacci (
    self)
```

6.1.3.6 clear()

```
mockup.Calculator.clear (
    self)
```

6.1.3.7 create_button()

```
mockup.Calculator.create_button (
    self,
    text,
    row,
    column,
    colspanspan = 1,
    rowspan = 1,
    command = None,
    bg = "#e0e0e0")
```

@brief Vytvoření jednotlivého tlačítka v kalkulačce.

@param text Text, který se zobrazí na tlačítku.

@param row Pozice řádku v mřížce.

@param column Pozice sloupce v mřížce.

@param colspanspan Počet sloupců, které tlačítko zabírá (výchozí: 1).

@param rowspan Počet řádků, které tlačítko zabírá (výchozí: 1).

@param command Funkce, která se zavolá po kliknutí na tlačítko (výchozí: None).

@param bg Barva pozadí tlačítka (výchozí: "#e0e0e0").

@return Objekt vytvořeného tlačítka.

6.1.3.8 create_buttons()

```
mockup.Calculator.create_buttons (
    self)
```

@brief Vytvoření a umístění všech tlačítek kalkulačky.

@details Nastaví rozložení všech tlačítek kalkulačky, včetně čísel, operátorů a speciálních funkcí jako Fibonacciho posloupnost a absolutní hodnota.

6.1.3.9 create_menu()

```
mockup.Calculator.create_menu (  
    self)
```

6.1.3.10 display_error()

```
mockup.Calculator.display_error (  
    self,  
    error_code)
```

6.1.3.11 evaluate()

```
mockup.Calculator.evaluate (  
    self)
```

6.1.3.12 negate()

```
mockup.Calculator.negate (  
    self)
```

6.1.3.13 prepare_operation()

```
mockup.Calculator.prepare_operation (  
    self,  
    operation)
```

@brief Nastavení binární operace mezi dvěma čísly.

@details Uloží první číslo a typ operace, aktualizuje displej a připraví kalkulačku na zadání druhého čísla.

@param operation Řetězec reprezentující operaci ("add", "sub" atd.).

6.1.3.14 setup_keyboard_bindings()

```
mockup.Calculator.setup_keyboard_bindings (  
    self)
```

@brief Nastavení klávesových zkratk pro operace kalkulačky.

@details Mapuje klávesy na funkce kalkulačky:

- Číselné klávesy (0-9) pro zadávání čísl
- Operátory (+, -, *, /, ^) pro základní operace
- Funkční klávesy (!, a, f) pro faktoriál, absolutní hodnotu a Fibonacciho posloupnost
- Řídící klávesy (Enter, Backspace, Escape) pro vyhodnocení, mazání a vymazání

6.1.3.15 show_about()

```
mockup.Calculator.show_about (  
    self)
```

6.1.3.16 show_help()

```
mockup.Calculator.show_help (  
    self)
```

6.1.3.17 update_display()

```
mockup.Calculator.update_display (  
    self,  
    evaluated = False)
```

@brief Aktualizace displeje kalkulačky aktuálními hodnotami.

@details Aktualizuje jak displej pro výraz (ukazuje výpočet), tak i displej pro výsledek (ukazuje aktuální číslo).

@param *evaluated* Boolean určující, zda byl výraz právě vyhodnocen (výchozí: False).

6.1.4 Member Data Documentation

6.1.4.1 buttons_frame

```
mockup.Calculator.buttons_frame = tk.Frame(self.root, bg="#f0f0f0")
```

6.1.4.2 current_expression

```
str mockup.Calculator.current_expression = ""
```

6.1.4.3 display_frame

```
mockup.Calculator.display_frame = tk.Frame(self.root, height=100, bg="#f0f0f0")
```

6.1.4.4 expecting_second_number

```
bool mockup.Calculator.expecting_second_number = False
```

6.1.4.5 expression_label

mockup.Calculator.expression_label

Initial value:

```
= tk.Label(  
    self.display_frame,  
    text="",  
    font=("Arial", 14),  
    anchor="e",  
    bg="#f0f0f0",  
    fg="#505050"  
)
```

6.1.4.6 first_number

mockup.Calculator.first_number = None

6.1.4.7 just_evaluated

bool mockup.Calculator.just_evaluated = False

6.1.4.8 pending_operation

str mockup.Calculator.pending_operation = None

6.1.4.9 result

mockup.Calculator.result = None

6.1.4.10 result_label

mockup.Calculator.result_label

Initial value:

```
= tk.Label(  
    self.display_frame,  
    text="0",  
    font=("Arial", 24, "bold"),  
    anchor="e",  
    bg="#f0f0f0"  
)
```

6.1.4.11 root

mockup.Calculator.root = root

6.1.4.12 second_number

mockup.Calculator.second_number = None

6.1.4.13 total_expression

```
str mockup.Calculator.total_expression = ""
```

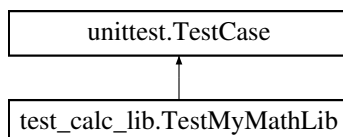
The documentation for this class was generated from the following file:

- [mockup.py](#)

6.2 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.

6.2.1 Detailed Description

Obsahuje testovací metody pro různé matematické operace.

6.2.2 Member Function Documentation

6.2.2.1 test_abs()

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

Testy pro absolutní hodnotu.

6.2.2.2 test_add()

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

Testy pro funkci add (sčítání).

6.2.2.3 test_div()

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

Testy pro funkci div (dělení).

6.2.2.4 test_div_zero()

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

Test dělení nulou.

6.2.2.5 test_expon()

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

Testy pro funkci expon (mocnění).

6.2.2.6 test_factorial()

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

Testy pro funkci factorial.

6.2.2.7 test_factorial_value_err()

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

Testy pro factorial s neplatnými vstupy.

6.2.2.8 test_fib()

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

Testy pro Fibonacciho posloupnost.

6.2.2.9 test_mul()

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

Testy pro funkci mul (násobení).

6.2.2.10 test_sqr()

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

Testy pro funkci sqr (odmocnina).

6.2.2.11 test_sub()

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

Testy pro funkci sub (odčítání).

6.2.2.12 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

Classes

- class `mockup.Calculator`

Namespaces

- namespace `mockup`

Variables

- `mockup.root` = `tk.Tk()`
- `mockup.app` = `Calculator(root)`

7.3 stddev.py File Reference

Namespaces

- namespace [stddev](#)

Functions

- [stddev.calculate_stddev](#) ()
Funkce pro výpočet výběrové směrodatné odchylky ze vstupu.
- [stddev.run_profiled_calculations](#) ()
Spuštění výpočtu s měřením výkonnosti pomocí cProfile.

Variables

- int [stddev.INPUT_SIZE](#) = 10

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

- `__init__`
 - `mockup.Calculator`, 16
- `absolute`
 - `calc_lib`, 9
- `add`
 - `calc_lib`, 9
- `app`
 - `mockup`, 12
- `append_number`
 - `mockup.Calculator`, 16
- `backspace`
 - `mockup.Calculator`, 16
- `buttons_frame`
 - `mockup.Calculator`, 19
- `calc_lib`, 9
 - `absolute`, 9
 - `add`, 9
 - `div`, 10
 - `expon`, 10
 - `factorial`, 10
 - `fib`, 10
 - `mul`, 10
 - `sqr`, 11
 - `sub`, 11
- `calc_lib.py`, 25
- `calculate_absolute`
 - `mockup.Calculator`, 16
- `calculate_factorial`
 - `mockup.Calculator`, 16
- `calculate_fibonacci`
 - `mockup.Calculator`, 17
- `calculate_stddev`
 - `stddev`, 13
- `clear`
 - `mockup.Calculator`, 17
- `create_button`
 - `mockup.Calculator`, 17
- `create_buttons`
 - `mockup.Calculator`, 17
- `create_menu`
 - `mockup.Calculator`, 17
- `current_expression`
 - `mockup.Calculator`, 19
- `display_error`
 - `mockup.Calculator`, 18
- `display_frame`
 - `mockup.Calculator`, 19
- `div`
 - `calc_lib`, 10
- `evaluate`
 - `mockup.Calculator`, 18
- `expecting_second_number`
 - `mockup.Calculator`, 19
- `expon`
 - `calc_lib`, 10
- `expression_label`
 - `mockup.Calculator`, 19
- `factorial`
 - `calc_lib`, 10
- `fib`
 - `calc_lib`, 10
- `first_number`
 - `mockup.Calculator`, 20
- `INPUT_SIZE`
 - `stddev`, 13
- `just_evaluated`
 - `mockup.Calculator`, 20
- `mockup`, 11
 - `app`, 12
 - `root`, 12
- `mockup.Calculator`, 15
 - `__init__`, 16
 - `append_number`, 16
 - `backspace`, 16
 - `buttons_frame`, 19
 - `calculate_absolute`, 16
 - `calculate_factorial`, 16
 - `calculate_fibonacci`, 17
 - `clear`, 17
 - `create_button`, 17
 - `create_buttons`, 17
 - `create_menu`, 17
 - `current_expression`, 19
 - `display_error`, 18
 - `display_frame`, 19
 - `evaluate`, 18
 - `expecting_second_number`, 19
 - `expression_label`, 19
 - `first_number`, 20
 - `just_evaluated`, 20
 - `negate`, 18
 - `pending_operation`, 20

- prepare_operation, [18](#)
- result, [20](#)
- result_label, [20](#)
- root, [20](#)
- second_number, [20](#)
- setup_keyboard_bindings, [18](#)
- show_about, [18](#)
- show_help, [19](#)
- total_expression, [20](#)
- update_display, [19](#)
- mockup.py, [25](#)
- mul
 - calc_lib, [10](#)
- negate
 - mockup.Calculator, [18](#)
- pending_operation
 - mockup.Calculator, [20](#)
- prepare_operation
 - mockup.Calculator, [18](#)
- result
 - mockup.Calculator, [20](#)
- result_label
 - mockup.Calculator, [20](#)
- root
 - mockup, [12](#)
 - mockup.Calculator, [20](#)
- run_profiled_calculations
 - stddev, [13](#)
- second_number
 - mockup.Calculator, [20](#)
- setup_keyboard_bindings
 - mockup.Calculator, [18](#)
- show_about
 - mockup.Calculator, [18](#)
- show_help
 - mockup.Calculator, [19](#)
- sqr
 - calc_lib, [11](#)
- stddev, [12](#)
 - calculate_stddev, [13](#)
 - INPUT_SIZE, [13](#)
 - run_profiled_calculations, [13](#)
- stddev.py, [26](#)
- sub
 - calc_lib, [11](#)
- test_abs
 - test_calc_lib.TestMyMathLib, [22](#)
- test_add
 - test_calc_lib.TestMyMathLib, [22](#)
- test_calc_lib, [13](#)
- test_calc_lib.py, [26](#)
- test_calc_lib.TestMyMathLib, [21](#)
 - test_abs, [22](#)
 - test_add, [22](#)
 - test_div, [22](#)
 - test_div_zero, [22](#)
 - test_expon, [22](#)
 - test_factorial, [22](#)
 - test_factorial_value_err, [22](#)
 - test_fib, [23](#)
 - test_mul, [23](#)
 - test_sqr, [23](#)
 - test_sub, [23](#)
 - test_value, [23](#)
- test_div
 - test_calc_lib.TestMyMathLib, [22](#)
- test_div_zero
 - test_calc_lib.TestMyMathLib, [22](#)
- test_expon
 - test_calc_lib.TestMyMathLib, [22](#)
- test_factorial
 - test_calc_lib.TestMyMathLib, [22](#)
- test_factorial_value_err
 - test_calc_lib.TestMyMathLib, [22](#)
- test_fib
 - test_calc_lib.TestMyMathLib, [23](#)
- test_mul
 - test_calc_lib.TestMyMathLib, [23](#)
- test_sqr
 - test_calc_lib.TestMyMathLib, [23](#)
- test_sub
 - test_calc_lib.TestMyMathLib, [23](#)
- test_value
 - test_calc_lib.TestMyMathLib, [23](#)
- total_expression
 - mockup.Calculator, [20](#)
- update_display
 - mockup.Calculator, [19](#)