

HugoCalc

Generated by Doxygen 1.9.1



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui	.....	??
----	-------	----



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

```
QMainWindow
  MainWindow . . . . . ??
testing::Test
  Advanced_math_test . . . . . ??
  Basic_math_test . . . . . ??
```



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Advanced_math_test</a>	Slouzi hlavne k rozeznani testu Advanced (slozitejsich) funkci matematicke knihovny . . . . .	??
<a href="#">Basic_math_test</a>	Slouzi hlavne k rozeznani testu Basic (jednoduzsich) funkci matematicke knihovny . . . . .	??
<a href="#">MainWindow</a>	Class pro UI projektu . . . . .	??





## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">mathlib.cpp</a>	Definice funkci matematicke knihovny pro kalkulacka . . . . .	??
<a href="#">mathlib.h</a>	Deklarace funkci matematicke knihovny pro kalkulacka . . . . .	??
<a href="#">gui/main.cpp</a>	Driver program pro UI . . . . .	??
<a href="#">gui/mainwindow.cpp</a>	Implementace UI funkcí, Vytvořeno pomocí QT editoru . . . . .	??
<a href="#">gui/mainwindow.h</a>	Definice UI funkcí, Vytvořeno pomocí QT editoru . . . . .	??
<a href="#">profiling/profiling.cpp</a>	Výpočes směrodatné odchylky, který slouží k profilingu matematické knihovny . . . . .	??
<a href="#">test/test_mathlib.cpp</a>	Testy pro matematickou knihovnu zapomoci googletest. kazdy test zahrnuje test na spravnost vysledku a chybove inputy (desetinne cisla, zaporne cisla a zaporne desetinne cisla) . . . . .	??



## Chapter 5

# Namespace Documentation

### 5.1 Ui Namespace Reference



## Chapter 6

# Class Documentation

### 6.1 Advanced\_math\_test Class Reference

Slouzi hlavne k rozeznani testu Advanced (slozitejsich) funkci matematicke knihovny.

Inheritance diagram for Advanced\_math\_test:

### 6.2 Basic\_math\_test Class Reference

Slouzi hlavne k rozeznani testu Basic (jednoduzsich) funkci matematicke knihovny.

Inheritance diagram for Basic\_math\_test:

Collaboration diagram for Basic\_math\_test:

#### 6.2.1 Detailed Description

Slouzi hlavne k rozeznani testu Basic (jednoduzsich) funkci matematicke knihovny.

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

- test/[test\\_mathlib.cpp](#)

### 6.3 MainWindow Class Reference

Class pro UI projektu.

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:

Collaboration diagram for MainWindow:

## Public Member Functions

- [MainWindow](#) (QWidget \*parent=nullptr)  
*Konstruktor [MainWindow](#).*
- [~MainWindow](#) ()  
*Destruktor [MainWindow](#).*
- void [result](#) ()  
*do primary dá výsledek požadovaného výpočtu podle operation*
- void [reset](#) ()  
*vynuluje primary, secondary a operand pole*
- void [handleOperation](#) (QString operand)  
*dá do pole operand parametr QString operand*
- void [on\\_button\\_0\\_clicked](#) ()  
*při kliknutí na tlačítko 0 přidá do pole primary 0*
- void [on\\_button\\_1\\_clicked](#) ()  
*při kliknutí na tlačítko 1 přidá do pole primary 1*
- void [on\\_button\\_2\\_clicked](#) ()  
*při kliknutí na tlačítko 2 přidá do pole primary 2*
- void [on\\_button\\_3\\_clicked](#) ()  
*při kliknutí na tlačítko 3 přidá do pole primary 3*
- void [on\\_button\\_4\\_clicked](#) ()  
*při kliknutí na tlačítko 4 přidá do pole primary 4*
- void [on\\_button\\_5\\_clicked](#) ()  
*při kliknutí na tlačítko 5 přidá do pole primary 5*
- void [on\\_button\\_6\\_clicked](#) ()  
*při kliknutí na tlačítko 6 přidá do pole primary 6*
- void [on\\_button\\_7\\_clicked](#) ()  
*při kliknutí na tlačítko 7 přidá do pole primary 7*
- void [on\\_button\\_8\\_clicked](#) ()  
*při kliknutí na tlačítko 8 přidá do pole primary 8*
- void [on\\_button\\_9\\_clicked](#) ()  
*při kliknutí na tlačítko 9 přidá do pole primary 9*
- void [on\\_button\\_ac\\_clicked](#) ()  
*při kliknutí na tlačítko AC vynuluje pole primary, secondary a operand*
- void [on\\_button\\_del\\_clicked](#) ()  
*při kliknutí na tlačítko del smaže poslední znak z primary*
- void [on\\_button\\_times\\_ten\\_clicked](#) ()  
*při kliknutí na tlačítko 10x přidá 2 nuly na konec pole primary (popřípadě přidá desetinnou tečku)*
- void [on\\_button\\_plus\\_clicked](#) ()  
*při kliknutí na tlačítko + nastaví pole operand na +*
- void [on\\_button\\_minus\\_clicked](#) ()  
*při kliknutí na tlačítko - nastaví pole operand na -*
- void [on\\_button\\_times\\_clicked](#) ()  
*při kliknutí na tlačítko \* nastaví pole operand na \**
- void [on\\_button\\_div\\_clicked](#) ()  
*při kliknutí na tlačítko / nastaví pole operand na /*
- void [on\\_button\\_equals\\_clicked](#) ()  
*při kliknutí na tlačítko = nastaví pole primary na výsledek požadované operace*
- void [on\\_button\\_negate\\_clicked](#) ()  
*při kliknutí na tlačítko +/- vymění znaménko čísla*
- void [on\\_button\\_decimal\\_clicked](#) ()

- při kliknutí na tlačítko `.` do pole `primary` přidá desetinnou tečku  
 • void [on\\_button\\_y\\_exp\\_x\\_clicked](#) ()
- při kliknutí na tlačítko `^` nastaví pole `operand` na `^`  
 • void [on\\_button\\_factorial\\_clicked](#) ()
- při kliknutí na tlačítko `!` nastaví pole `operand` na `!`  
 • void [on\\_pushButton\\_clicked](#) ()
- při kliknutí na tlačítko `?` otevře nové okno `help`  
 • void [on\\_exitButton\\_clicked](#) ()
- při kliknutí na tlačítko `exit` v `help` zavře okno `help`  
 • void [on\\_button\\_1\\_over\\_x\\_clicked](#) ()
- při kliknutí na tlačítko `1/x` nastaví pole `operand` na `1/x`  
 • void [on\\_button\\_exp\\_x\\_root\\_y\\_clicked](#) ()
- při kliknutí na tlačítko  nastaví pole `operand` na

## Protected Member Functions

- void [keyPressEvent](#) (QKeyEvent \*event)  
*handluje input z klávesnice*

## Private Attributes

- Ui::MainWindow \* [ui](#)
- QDialog \* [dialog](#)

### 6.3.1 Detailed Description

Class pro UI projektu.

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 MainWindow()

```
MainWindow::MainWindow (
    QWidget * parent = nullptr )
```

Konstruktor [MainWindow](#).

References [ui](#).

### 6.3.2.2 ~MainWindow()

```
MainWindow::~MainWindow ( )
```

Destruktor [MainWindow](#).

References [ui](#).

## 6.3.3 Member Function Documentation

### 6.3.3.1 handleOperation()

```
void MainWindow::handleOperation (
    QString operand )
```

dá do pole operand parametr QString operand

#### Parameters

in	<i>operand</i>	string který chceme nastavit do pole parametr
----	----------------	---

References [ui](#).

Referenced by [keyPressEvent\(\)](#), [on\\_button\\_1\\_over\\_x\\_clicked\(\)](#), [on\\_button\\_div\\_clicked\(\)](#), [on\\_button\\_exp\\_x\\_root\\_y\\_clicked\(\)](#), [on\\_button\\_factorial\\_clicked\(\)](#), [on\\_button\\_minus\\_clicked\(\)](#), [on\\_button\\_plus\\_clicked\(\)](#), [on\\_button\\_times\\_clicked\(\)](#), and [on\\_button\\_y\\_exp\\_x\\_clicked\(\)](#).

### 6.3.3.2 keyPressEvent()

```
void MainWindow::keyPressEvent (
    QKeyEvent * event ) [protected]
```

handluje input z klávesnice

#### Parameters

in	<i>event</i>	input z klávesnice
----	--------------	--------------------

References [allowedInputs](#), [allowedOperations](#), [handleOperation\(\)](#), [reset\(\)](#), [result\(\)](#), and [ui](#).

### 6.3.3.3 on\_button\_0\_clicked()

```
void MainWindow::on_button_0_clicked ( )
```



při kliknutí na tlačítko 0 přidá do pole primary 0

References ui.

#### 6.3.3.4 on\_button\_1\_clicked()

```
void MainWindow::on_button_1_clicked ( )
```

při kliknutí na tlačítko 1 přidá do pole primary 1

References ui.

#### 6.3.3.5 on\_button\_1\_over\_x\_clicked()

```
void MainWindow::on_button_1_over_x_clicked ( )
```

při kliknutí na tlačítko 1/x nastaví pole operand na 1/x

References handleOperation(), and result().

#### 6.3.3.6 on\_button\_2\_clicked()

```
void MainWindow::on_button_2_clicked ( )
```

při kliknutí na tlačítko 2 přidá do pole primary 2

References ui.

#### 6.3.3.7 on\_button\_3\_clicked()

```
void MainWindow::on_button_3_clicked ( )
```

při kliknutí na tlačítko 3 přidá do pole primary 3

References ui.

#### 6.3.3.8 on\_button\_4\_clicked()

```
void MainWindow::on_button_4_clicked ( )
```

při kliknutí na tlačítko 4 přidá do pole primary 4

References ui.

#### 6.3.3.9 on\_button\_5\_clicked()

```
void MainWindow::on_button_5_clicked ( )
```

při kliknutí na tlačítko 5 přidá do pole primary 5

References ui.

#### 6.3.3.10 on\_button\_6\_clicked()

```
void MainWindow::on_button_6_clicked ( )
```

při kliknutí na tlačítko 6 přidá do pole primary 6

References ui.

#### 6.3.3.11 on\_button\_7\_clicked()

```
void MainWindow::on_button_7_clicked ( )
```

při kliknutí na tlačítko 7 přidá do pole primary 7

References ui.

#### 6.3.3.12 on\_button\_8\_clicked()

```
void MainWindow::on_button_8_clicked ( )
```

při kliknutí na tlačítko 8 přidá do pole primary 8

References ui.

**6.3.3.13 on\_button\_9\_clicked()**

```
void MainWindow::on_button_9_clicked ( )
```

při kliknutí na tlačítko 9 přidá do pole primary 9

References ui.

**6.3.3.14 on\_button\_ac\_clicked()**

```
void MainWindow::on_button_ac_clicked ( )
```

při kliknutí na tlačítko AC vynuluje pole primary, secondary a operand

References reset().

**6.3.3.15 on\_button\_decimal\_clicked()**

```
void MainWindow::on_button_decimal_clicked ( )
```

při kliknutí na tlačítko . do pole primary přidá desetinnou tečku

References ui.

**6.3.3.16 on\_button\_del\_clicked()**

```
void MainWindow::on_button_del_clicked ( )
```

při kliknutí na tlačítko del umaže poslední znak z primary

References ui.

**6.3.3.17 on\_button\_div\_clicked()**

```
void MainWindow::on_button_div_clicked ( )
```

při kliknutí na tlačítko / nastaví pole operand na /

References handleOperation().

#### 6.3.3.18 on\_button\_equals\_clicked()

```
void MainWindow::on_button_equals_clicked ( )
```

při kliknutí na tlačítko = nastaví pole primary na výsledek požadované operace

References result().

#### 6.3.3.19 on\_button\_exp\_x\_root\_y\_clicked()

```
void MainWindow::on_button_exp_x_root_y_clicked ( )
```

při kliknutí na tlačítko nastaví pole operand na

References handleOperation().

#### 6.3.3.20 on\_button\_factorial\_clicked()

```
void MainWindow::on_button_factorial_clicked ( )
```

při kliknutí na tlačítko ! nastaví pole operand na !

References handleOperation(), and result().

#### 6.3.3.21 on\_button\_minus\_clicked()

```
void MainWindow::on_button_minus_clicked ( )
```

při kliknutí na tlačítko - nastaví pole operand na -

References handleOperation().

#### 6.3.3.22 on\_button\_negate\_clicked()

```
void MainWindow::on_button_negate_clicked ( )
```

při kliknutí na tlačítko +/- vymění znaménko čísla

References ui.

**6.3.3.23 on\_button\_plus\_clicked()**

```
void MainWindow::on_button_plus_clicked ( )
```

při kliknutí na tlačítko + nastaví pole operand na +

References `handleOperation()`.

**6.3.3.24 on\_button\_times\_clicked()**

```
void MainWindow::on_button_times_clicked ( )
```

při kliknutí na tlačítko \* nastaví pole operand na \*

References `handleOperation()`.

**6.3.3.25 on\_button\_times\_ten\_clicked()**

```
void MainWindow::on_button_times_ten_clicked ( )
```

při kliknutí na tlačítko 10x přidá 2 nuly na konec pole primary (popřípadě přidá desetinnou tečku)

References `ui`.

**6.3.3.26 on\_button\_y\_exp\_x\_clicked()**

```
void MainWindow::on_button_y_exp_x_clicked ( )
```

při kliknutí na tlačítko ^ nastaví pole operand na ^

References `handleOperation()`.

**6.3.3.27 on\_exitButton\_clicked()**

```
void MainWindow::on_exitButton_clicked ( )
```

při kliknutí na tlačítko exit v help zavře okno help

References `dialog`.

Referenced by `on_pushButton_clicked()`.

### 6.3.3.28 on\_pushButton\_clicked()

```
void MainWindow::on_pushButton_clicked ( )
```

při kliknutí na tlačítko ? otevře nové okno help

References dialog, and on\_exitButton\_clicked().

### 6.3.3.29 reset()

```
void MainWindow::reset ( )
```

vynuluje primary, secondary a operand pole

References ui.

Referenced by keyPressEvent(), on\_button\_ac\_clicked(), and result().

### 6.3.3.30 result()

```
void MainWindow::result ( )
```

do primary dá výsledek požadovaného výpočtu podle operation

References k\_deleno(), k\_faktorial(), k\_krat(), k\_minus(), k\_mocnina(), k\_obracena\_hodnota(), k\_odmocnina(), k\_plus(), reset(), and ui.

Referenced by keyPressEvent(), on\_button\_1\_over\_x\_clicked(), on\_button\_equals\_clicked(), and on\_button\_factorial\_clicked().

## 6.3.4 Member Data Documentation

### 6.3.4.1 dialog

```
QDialog* MainWindow::dialog [private]
```

Referenced by on\_exitButton\_clicked(), and on\_pushButton\_clicked().

### 6.3.4.2 ui

```
Ui::MainWindow* MainWindow::ui [private]
```

Referenced by handleOperation(), keyPressEvent(), MainWindow(), on\_button\_0\_clicked(), on\_button\_1\_clicked(), on\_button\_2\_clicked(), on\_button\_3\_clicked(), on\_button\_4\_clicked(), on\_button\_5\_clicked(), on\_button\_6\_clicked(), on\_button\_7\_clicked(), on\_button\_8\_clicked(), on\_button\_9\_clicked(), on\_button\_decimal\_clicked(), on\_button\_del\_clicked(), on\_button\_negate\_clicked(), on\_button\_times\_ten\_clicked(), reset(), result(), and ~MainWindow().

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

- [gui/mainwindow.h](#)
- [gui/mainwindow.cpp](#)

## Chapter 7

# File Documentation

### 7.1 gui/main.cpp File Reference

Driver program pro UI.

```
#include "mainwindow.h"
#include <QApplication>
Include dependency graph for main.cpp:
```

#### Functions

- int [main](#) (int argc, char \*argv[ ])

#### 7.1.1 Detailed Description

Driver program pro UI.

Author

Adam Černoš

#### 7.1.2 Function Documentation

##### 7.1.2.1 main()

```
int main (
    int argc,
    char * argv[ ] )
```

## 7.2 gui/mainwindow.cpp File Reference

Implementace UI funkcí, Vytvořeno pomocí QT editoru.

```
#include "mainwindow.h"
#include "../mathlib.cpp"
#include "./ui_mainwindow.h"
#include <QMessageBox>
#include <stdexcept>
#include <string>
#include <QKeyEvent>
Include dependency graph for mainwindow.cpp:
```

### Variables

- QString [allowedInputs](#) []
- QString [allowedOperations](#) [] = {"^", "+", "-", "\*", "/"}

### 7.2.1 Detailed Description

Implementace UI funkcí, Vytvořeno pomocí QT editoru.

#### Author

Adam Černoš  
Patrik Mokruša

### 7.2.2 Variable Documentation

#### 7.2.2.1 allowedInputs

```
QString allowedInputs[]
```

##### Initial value:

```
= {"0", "1", "2", "3", "4", "5",  
   "6", "7", "8", "9", "."}
```

Referenced by MainWindow::keyPressEvent().

#### 7.2.2.2 allowedOperations

```
QString allowedOperations[] = {"^", "+", "-", "*", "/"}
```

Referenced by MainWindow::keyPressEvent().



## 7.3 gui/mainwindow.h File Reference

Definice UI funkcí, Vytvořeno pomocí QT editoru.

```
#include <QDialog>
#include <QLabel>
#include <QMainWindow>
#include <QWidget>
```

Include dependency graph for/mainwindow.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [MainWindow](#)  
*Class pro UI projektu.*

### Namespaces

- [Ui](#)

#### 7.3.1 Detailed Description

Definice UI funkcí, Vytvořeno pomocí QT editoru.

Author

Adam Černoš

## 7.4 mathlib.cpp File Reference

Definice funkcí matematické knihovny pro kalkulacka.

```
#include "mathlib.h"
#include <cmath>
#include <stdexcept>
#include <stdlib.h>
```

Include dependency graph for mathlib.cpp: This graph shows which files directly or indirectly include this file:

### Functions

- double [k\\_plus](#) (double base\_num, double add\_num)  
*k\_plus přičte add\_num k base\_num*
- double [k\\_minus](#) (double base\_num, double sub\_num)  
*k\_minus odečte sub\_num od base\_num*
- double [k\\_krat](#) (double base\_num, double times\_num)  
*k\_krat přinásobí tiems\_num k base\_num*
- double [k\\_deleno](#) (double base\_num, double div\_num)  
*k\_deleno vydělí base\_num s div\_num*
- double [k\\_faktorial](#) (double base\_num)  
*k\_faktorial vypocita faktorial*
- double [k\\_mocnina](#) (double base\_num, double mocnitel)  
*k\_mocnina umocní basenum\_num na  $base\_num$*
- double [k\\_odmocnina](#) (double base\_num, double odmocnitel)  
*k\_odmocnina base\_num odmocní odmocnitel -em*
- double [k\\_obracena\\_hodnota](#) (double base\_num)  
*k\_obracena\_hodnota obracena hodnota k base\_num ( $1/base\_num$ )*

### 7.4.1 Detailed Description

Definice funkci matematicke knihovny pro kalkulacka.

Author

Patrik Mokruša

### 7.4.2 Function Documentation

#### 7.4.2.1 k\_deleno()

```
double k_deleno (
    double base_num,
    double div_num )
```

k\_deleno vyděli base\_num s div\_num

##### Parameters

in	<i>base_num</i>	první číslo (před /)
in	<i>div_num</i>	druhé číslo (za /)

##### Returns

vrací vydelene cislo

##### Exceptions

<i>overflow_error</i>	při dělení nulou
-----------------------	------------------

Referenced by main(), MainWindow::result(), and TEST\_F().

#### 7.4.2.2 k\_faktorial()

```
double k_faktorial (
    double base_num )
```

k\_faktorial vypočítá faktoriál

##### Parameters

in	<i>base_num</i>	číslo ze kterého chceme faktoriál
----	-----------------	-----------------------------------

**Returns**

vraci faktorial

**Exceptions**

<i>out_of_range</i>	pri zadani zaporneho nebo desetinneho cisla
---------------------	---

Referenced by MainWindow::result(), and TEST\_F().

**7.4.2.3 k\_krat()**

```
double k_krat (
    double base_num,
    double times_num )
```

k\_krat prinasobi tiems\_num k base\_num

**Parameters**

in	<i>base_num</i>	prvni ciclo (pred *)
in	<i>times_num</i>	druhe cislo (za *)

**Returns**

vraci soucin cisel

Referenced by MainWindow::result(), and TEST\_F().

**7.4.2.4 k\_minus()**

```
double k_minus (
    double base_num,
    double sub_num )
```

k\_minus odecte sub\_num od base\_num

**Parameters**

in	<i>base_num</i>	prvni ciclo (pred -)
in	<i>sub_num</i>	druhe cislo (za -)

**Returns**

vraci rozdil cisel

Referenced by `main()`, `MainWindow::result()`, and `TEST_F()`.

#### 7.4.2.5 `k_mocnina()`

```
double k_mocnina (
    double base_num,
    double mocnitel )
```

`k_mocnina` umocni basenum\_num na  $^{\wedge}$  base\_num

##### Parameters

in	<i>base_num</i>	puvodni ciclo (pod mocninou)
in	<i>mocnitel</i>	cislo na (mocnitel)

##### Returns

vraci umocnene cislo

##### Exceptions

<i>out_of_range</i>	pri mocnenim zapornym cislem
---------------------	------------------------------

Referenced by `main()`, `MainWindow::result()`, and `TEST_F()`.

#### 7.4.2.6 `k_obracena_hodnota()`

```
double k_obracena_hodnota (
    double base_num )
```

`k_obracena_hodnota` obracena hodnota k base\_num ( $1/\text{base\_num}$ )

##### Parameters

in	<i>base_num</i>	cislo ke kteremu chceme obracenou hodnotu
----	-----------------	---

##### Returns

vraci obracenou hodnotu k cislu

##### Exceptions

<i>overflow_error</i>	pri zadani nuly (nelze delit nulou)
-----------------------	-------------------------------------

Referenced by `k_odmocnina()`, `MainWindow::result()`, and `TEST_F()`.

#### 7.4.2.7 `k_odmocnina()`

```
double k_odmocnina (
    double base_num,
    double odmocnitel )
```

`k_odmocnina base_num odmocni odmocnitel -em`

##### Parameters

in	<i>base_num</i>	puvodni cislo (pod odmocninou)
in	<i>times_num</i>	kolikata odmocnina (odmocnitel)

##### Returns

odmocnene cislo

##### Exceptions

<i>invalid_argument</i>	pri odmocnenim desetinnym nebo zapornym cislem
-------------------------	--

< pro klasickou druhou odmocninu

References `k_obracena_hodnota()`.

Referenced by `main()`, `MainWindow::result()`, and `TEST_F()`.

#### 7.4.2.8 `k_plus()`

```
double k_plus (
    double base_num,
    double add_num )
```

`k_plus` přičte `add_num` k `base_num`

##### Parameters

in	<i>base_num</i>	prvni cislo (pred +)
in	<i>add_num</i>	druhe cislo (za +)

##### Returns

vraci soucet cisel

Referenced by `main()`, `MainWindow::result()`, and `TEST_F()`.

## 7.5 mathlib.h File Reference

Deklarace funkci matematicke knihovny pro kalkulacka.

This graph shows which files directly or indirectly include this file:

### Functions

- double `k_plus` (double `base_num`, double `add_num`)  
*k\_plus přičte add\_num k base\_num*
- double `k_minus` (double `base_num`, double `sub_num`)  
*k\_minus odečte sub\_num od base\_num*
- double `k_krat` (double `base_num`, double `times_num`)  
*k\_krat prinasobi tiems\_num k base\_num*
- double `k_deleno` (double `base_num`, double `div_num`)  
*k\_deleno vydeli base\_num s div\_num*
- double `k_faktorial` (double `base_num`)  
*k\_faktorial vypocita faktorial*
- double `k_mocnina` (double `base_num`, double `mocnitel`)  
*k\_mocnina umocni basenum\_num na ^ base\_num*
- double `k_odmocnina` (double `base_num`, double `odmocnitel`)  
*k\_odmocnina base\_num odmocni odmocnitel -em*
- double `k_obracena_hodnota` (double `base_num`)  
*k\_obracena\_hodnota obracena hodnota k base\_num (1/base\_num)*

### 7.5.1 Detailed Description

Deklarace funkci matematicke knihovny pro kalkulacka.

Author

Patrik Mokruša

### 7.5.2 Function Documentation

#### 7.5.2.1 `k_deleno()`

```
double k_deleno (
    double base_num,
    double div_num )
```

`k_deleno` vydeli `base_num` s `div_num`

**Parameters**

in	<i>base_num</i>	prvni ciclo (pred /)
in	<i>div_num</i>	druhe cislo (za /)

**Returns**

vraci vydelene cislo

**Exceptions**

<i>overflow_error</i>	pri deleni nulou
-----------------------	------------------

Referenced by `main()`, `MainWindow::result()`, and `TEST_F()`.

**7.5.2.2 k\_faktorial()**

```
double k_faktorial (
    double base_num )
```

`k_faktorial` vypocita faktorial

**Parameters**

in	<i>base_num</i>	cislo ze ktoreho chceme faktorial
----	-----------------	-----------------------------------

**Returns**

vraci faktorial

**Exceptions**

<i>out_of_range</i>	pri zadani zaporneho nebo desetinneho cisla
---------------------	---

Referenced by `MainWindow::result()`, and `TEST_F()`.

**7.5.2.3 k\_krat()**

```
double k_krat (
    double base_num,
    double times_num )
```

`k_krat` prinasobi `times_num` k `base_num`

**Parameters**

in	<i>base_num</i>	prvni ciclo (pred *)
in	<i>times_num</i>	druhe cislo (za *)

**Returns**

vraci soucin cisel

Referenced by MainWindow::result(), and TEST\_F().

**7.5.2.4 k\_minus()**

```
double k_minus (
    double base_num,
    double sub_num )
```

k\_minus odecte sub\_num od base\_num

**Parameters**

in	<i>base_num</i>	prvni ciclo (pred -)
in	<i>sub_num</i>	druhe cislo (za -)

**Returns**

vraci rozdil cisel

Referenced by main(), MainWindow::result(), and TEST\_F().

**7.5.2.5 k\_mocnina()**

```
double k_mocnina (
    double base_num,
    double mocnitel )
```

k\_mocnina umocni basenum\_num na ^ base\_num

**Parameters**

in	<i>base_num</i>	puvodni ciclo (pod mocninou)
in	<i>mocnitel</i>	cislo na (mocnitel)



**Returns**

vraci umocnene cislo

**Exceptions**

<i>out_of_range</i>	pri mocnenim zapornym cislem
---------------------	------------------------------

Referenced by `main()`, `MainWindow::result()`, and `TEST_F()`.

**7.5.2.6 k\_obracena\_hodnota()**

```
double k_obracena_hodnota (  
    double base_num )
```

`k_obracena_hodnota` obracena hodnota k `base_num` ( $1/\text{base\_num}$ )

**Parameters**

in	<i>base_num</i>	cislo ke kteremu chceme obracenou hodnotu
----	-----------------	---

**Returns**

vraci obracenou hodnotu k cislu

**Exceptions**

<i>overflow_error</i>	pri zadani nuly (nelze delit nulou)
-----------------------	-------------------------------------

Referenced by `k_odmocnina()`, `MainWindow::result()`, and `TEST_F()`.

**7.5.2.7 k\_odmocnina()**

```
double k_odmocnina (  
    double base_num,  
    double odmocnitel )
```

`k_odmocnina` `base_num` odmocni `odmocnitel` -em

**Parameters**

in	<i>base_num</i>	puvodni cislo (pod odmocninou)
in	<i>times_num</i>	kolikata odmocnina (odmocnitel)

**Returns**

odmocnene cislo

**Exceptions**

<i>invalid_argument</i>	pri odmocnenim desetinnym nebo zapornym cisle
-------------------------	---

< pro klasickou druhou odmocninu

References k\_obracena\_hodnota().

Referenced by main(), MainWindow::result(), and TEST\_F().

**7.5.2.8 k\_plus()**

```
double k_plus (
    double base_num,
    double add_num )
```

k\_plus přičte add\_num k base\_num

**Parameters**

in	<i>base_num</i>	prvni cislo (pred +)
in	<i>add_num</i>	druhe cislo (za +)

**Returns**

vraci soucet cisel

Referenced by main(), MainWindow::result(), and TEST\_F().

**7.6 profiling/profiling.cpp File Reference**

Výpočet směrodatné odchylky, který slouží k profilingu matematické knihovny.

```
#include <iostream>
#include <vector>
#include "../mathlib.cpp"
Include dependency graph for profiling.cpp:
```

**Functions**

- int [main](#) ()

### 7.6.1 Detailed Description

Výpočes směrodatné odchylky, který slouží k profilingu matematické knihovny.

#### Author

Lukáš Jelínek

### 7.6.2 Function Documentation

#### 7.6.2.1 main()

```
int main ( )
```

References `k_deleno()`, `k_minus()`, `k_mocnina()`, `k_odmocnina()`, and `k_plus()`.

## 7.7 test/test\_mathlib.cpp File Reference

testy pro matematickou knihovnu zapomoci googletest. kazdy test zahrnuje test na spravnost vysledku a chybove inputy (desetinne cisla, zaporne cisla a zaporne desetinne cisla).

```
#include "../mathlib.cpp"
#include "gtest/gtest.h"
```

Include dependency graph for test\_mathlib.cpp:

### Classes

- class [Basic\\_math\\_test](#)  
*Slouzi hlavne k rozeznani testu Basic (jednoduzsich) funkci matematicke knihovny.*
- class [Advanced\\_math\\_test](#)  
*Slouzi hlavne k rozeznani testu Advanced (slozitejsich) funkci matematicke knihovny.*

### Functions

- [TEST\\_F](#) ([Basic\\_math\\_test](#), Plus)
- [TEST\\_F](#) ([Basic\\_math\\_test](#), Minus)
- [TEST\\_F](#) ([Basic\\_math\\_test](#), Krat)
- [TEST\\_F](#) ([Basic\\_math\\_test](#), Deleno)
- [TEST\\_F](#) ([Advanced\\_math\\_test](#), Faktorial)
- [TEST\\_F](#) ([Advanced\\_math\\_test](#), Mocnina)
- [TEST\\_F](#) ([Advanced\\_math\\_test](#), Odmocnina)
- [TEST\\_F](#) ([Advanced\\_math\\_test](#), Obracena\_hodnota)

### 7.7.1 Detailed Description

testy pro matematickou knihovnu zapomoci googletest. kazdy test zahrnuje test na spravnost vysledku a chybove inputy (desetinne cisla, zaporne cisla a zaporne desetinne cisla).

Author

Patrik Mokruša

### 7.7.2 Function Documentation

#### 7.7.2.1 TEST\_F() [1/8]

```
TEST_F (
    Advanced_math_test ,
    Faktorial )
```

References k\_faktorial().

#### 7.7.2.2 TEST\_F() [2/8]

```
TEST_F (
    Advanced_math_test ,
    Mocnina )
```

References k\_mocnina().

#### 7.7.2.3 TEST\_F() [3/8]

```
TEST_F (
    Advanced_math_test ,
    Obracena_hodnota )
```

References k\_obracena\_hodnota().

#### 7.7.2.4 TEST\_F() [4/8]

```
TEST_F (
    Advanced_math_test ,
    Odmocnina )
```

References k\_odmocnina().

#### 7.7.2.5 TEST\_F() [5/8]

```
TEST_F (
    Basic_math_test ,
    Deleno )
```

References k\_deleno().

#### 7.7.2.6 TEST\_F() [6/8]

```
TEST_F (
    Basic_math_test ,
    Krat )
```

References k\_krat().

#### 7.7.2.7 TEST\_F() [7/8]

```
TEST_F (
    Basic_math_test ,
    Minus )
```

References k\_minus().

#### 7.7.2.8 TEST\_F() [8/8]

```
TEST_F (
    Basic_math_test ,
    Plus )
```

References k\_plus().

