

Kalkulator

1

Wygenerowano przez Doxygen 1.8.14

Spis treści

1	Indeks hierarchiczny	1
1.1	Hierarchia klas	1
2	Indeks klas	3
2.1	Lista klas	3
3	Dokumentacja klas	5
3.1	Dokumentacja klasy Calculator	5
3.1.1	Opis szczegółowy	6
3.1.2	Dokumentacja konstruktora i destruktora	6
3.1.2.1	Calculator()	6
3.1.2.2	~Calculator()	6
3.1.3	Dokumentacja funkcji składowych	7
3.1.3.1	backspaceClicked	7
3.1.3.2	click_clear	7
3.1.3.3	click_minus	7
3.1.3.4	click_plus	7
3.1.3.5	click_podzielic	7
3.1.3.6	click_razy	7
3.1.3.7	click_rowne	7
3.1.3.8	clickDigit	8
3.1.3.9	clickPoint	8
3.1.3.10	dzialania	8
3.1.3.11	keyPressEvent	8

3.1.3.12 klik_plus_minus	8
3.1.3.13 logarytmy	8
3.1.3.14 nowe	8
3.1.3.15 pamiec	9
3.1.3.16 random	9
3.1.3.17 trygonometryczne	9
3.1.3.18 wersja_Kalkulatora	9
3.1.3.19 wersja_Kalkulatorakl	9
3.1.4 Dokumentacja atrybutów składowych	9
3.1.4.1 firstNum	9
3.1.4.2 tab	9
3.1.4.3 useristypingseconnumber	10
3.2 Dokumentacja klasy Complex	10
3.2.1 Opis szczegółowy	10
3.2.2 Dokumentacja konstruktora i destruktora	11
3.2.2.1 Complex()	11
3.2.2.2 ~Complex()	11
3.2.3 Dokumentacja funkcji składowych	11
3.2.3.1 close	11
3.2.3.2 dzialania	11
3.2.4 Dokumentacja atrybutów składowych	11
3.2.4.1 ze	11
3.3 Dokumentacja klasy Tomek	12
3.3.1 Opis szczegółowy	12
3.3.2 Dokumentacja konstruktora i destruktora	12
3.3.2.1 Tomek()	12
3.3.3 Dokumentacja funkcji składowych	13
3.3.3.1 clickBackspace	13
3.3.3.2 dzialania	13
3.3.3.3 keyPressEvent	14

3.3.4 Dokumentacja atrybutów składowych	14
3.3.4.1 jednostki	14
3.3.4.2 to	14
3.3.4.3 ui	14
3.4 Dokumentacja klasy Virtual	14
3.4.1 Opis szczegółowy	15
3.5 Dokumentacja klasy Zespolona	15
3.5.1 Opis szczegółowy	15
3.5.2 Dokumentacja konstruktora i destruktora	16
3.5.2.1 Zespolona() [1/3]	16
3.5.2.2 Zespolona() [2/3]	16
3.5.2.3 Zespolona() [3/3]	16
3.5.3 Dokumentacja funkcji składowych	16
3.5.3.1 operator*()	16
3.5.3.2 operator+()	16
3.5.3.3 operator-()	17
3.5.3.4 operator/()	17
3.5.3.5 operator==()	17
3.5.3.6 toString()	17
3.5.4 Dokumentacja atrybutów składowych	17
3.5.4.1 im	17
3.5.4.2 re	17
Indeks	19

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

QMainWindow	5
Calculator	
QWidget	
Complex	10
Tomek	12
Virtual	14
Complex	10
Tomek	12
Zespolona	15
Complex	10

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Calculator	5
Complex	10
Tomek	12
Virtual	14
Zespolona	15

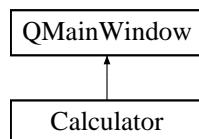
Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy Calculator

```
#include <calculator.h>
```

Diagram dziedziczenia dla Calculator



Metody publiczne

- [Calculator](#) (QWidget *parent=0)
- [~Calculator](#) ()

Atrybuty publiczne

- QMainWindow * p
- std::vector< QString > tab
- double firstNum
- bool useristypingseconnumber = false
- Ui::CalculatorClass ui

Sloty chronione

- void `clickDigit ()`
- void `clickPoint ()`
- void `klik_plus_minus ()`
- void `click_clear ()`
- void `click_rowne ()`
- void `click_podzielic ()`
- void `click_plus ()`
- void `click_minus ()`
- void `click_razy ()`
- void `backspaceClicked ()`
- void `pamiec ()`
- void `trygonometryczne ()`
- void `logarytmy ()`
- void `random ()`
- void `dzialania ()`
- void `wersja_Kalkulatora ()`
- void `wersja_Kalkulatorakl ()`
- void `nowe ()`
- void `keyPressEvent (QKeyEvent *event)`

Atrybuty chronione

- `Complex z`
- `Tomek t`

3.1.1 Opis szczegółowy

Główna klasa odpowiadająca za QMainWindow oraz podstawowe operacje kalkulatora naukowego oraz klasycznego. Tutaj tworzony jest obiekt klasy `Complex(liczby zespolone)`, oraz klasy `Tomek(konwersja jednostek)`, w metodzie "nowe()" po naciśnięciu odpowiedniego przycisku

3.1.2 Dokumentacja konstruktora i destruktora

3.1.2.1 Calculator()

```
Calculator::Calculator (
    QWidget * parent = 0 )
```

Konstruktor klasy `Calculator`

3.1.2.2 ~Calculator()

```
Calculator::~Calculator ( )
```

Destruktor klasy `Calculator`

3.1.3 Dokumentacja funkcji składowych

3.1.3.1 backspaceClicked

```
void Calculator::backspaceClicked ( ) [protected], [slot]
```

Usunięcie ostatniego chara z lineEdita.

3.1.3.2 click_clear

```
void Calculator::click_clear ( ) [protected], [slot]
```

Czyszczenie lineEdita.

3.1.3.3 click_minus

```
void Calculator::click_minus ( ) [protected], [slot]
```

Zaznaczenie przycisku dzielenie, oraz pobranie pierwszej liczby do obliczeń (firtsNum).

3.1.3.4 click_plus

```
void Calculator::click_plus ( ) [protected], [slot]
```

Zaznaczenie przycisku dzielenie, oraz pobranie pierwszej liczby do obliczeń (firtsNum).

3.1.3.5 click_podzielic

```
void Calculator::click_podzielic ( ) [protected], [slot]
```

Zaznaczenie przycisku dzielenie, oraz pobranie pierwszej liczby do obliczeń (firtsNum).

3.1.3.6 click_razy

```
void Calculator::click_razy ( ) [protected], [slot]
```

Zaznaczenie przycisku dzielenie, oraz pobranie pierwszej liczby do obliczeń (firtsNum).

3.1.3.7 click_rowne

```
void Calculator::click_rowne ( ) [protected], [slot]
```

Po naciśnięciu przycisku równa się sprawdzane jest który przycisk znaku działania jest naciśnięty, oraz pobraniu nowej liczby do obliczeń(secondNumber). Po sprawdzeniu tego wykonywane jest odpowiednie działanie i wysyłane do lineEdita.

3.1.3.8 clickDigit

```
void Calculator::clickDigit ( ) [protected], [slot]
```

Po nacisnieciu jednej z cyfr zostaje ona dodana do lineEdita.

3.1.3.9 clickPoint

```
void Calculator::clickPoint ( ) [protected], [slot]
```

Tutaj dodawany jest przecinek.

3.1.3.10 dzialania

```
void Calculator::dzialania ( ) [protected], [slot]
```

Tutaj wykonywane są działania w GUI przedstawione z niewiadomą "x" tj. odwrotność, pierwiastek, silnia, kwadrat, oraz sześciian liczby pobranej z lineEdita.

3.1.3.11 keyPressEvent

```
void Calculator::keyPressEvent (
    QKeyEvent * event ) [protected], [slot]
```

Obsługa klawiatury.

3.1.3.12 klik_plus_minus

```
void Calculator::klik_plus_minus ( ) [protected], [slot]
```

Zmienia znak liczby, oraz przedstawia wynik w procentach.

3.1.3.13 logarytmy

```
void Calculator::logarytmy ( ) [protected], [slot]
```

Obliczenie logarytmu o podstawie 10, oraz logarytmu naturalnego.

3.1.3.14 nowe

```
void Calculator::nowe ( ) [protected], [slot]
```

Dodanie dwóch dodatkowych modułów tj. do obliczania liczb zespolonych (Klasa [Complex](#)), oraz modułu do przeliczania jednostek długości w pierwszej potędze (Klasa [Tomek](#)).

3.1.3.15 pamiec

```
void Calculator::pamiec ( ) [protected], [slot]
```

Tutaj poprzez wykorzystanie std::vector<QString> tab; użytkownik dodaje sobie liczby do pamięci.

3.1.3.16 random

```
void Calculator::random ( ) [protected], [slot]
```

Wylosowanie liczby pseudolosowej i pokazanie jej w lineEdicie.

3.1.3.17 trygonometryczne

```
void Calculator::trygonometryczne ( ) [protected], [slot]
```

Wykorzystywana do obliczeń trygonometrycznych(sinus, cosinus, tangens).

3.1.3.18 wersja_Kalkulatora

```
void Calculator::wersja_Kalkulatora ( ) [protected], [slot]
```

Ustawienie wersji kalkulatora na wersje naukową.

3.1.3.19 wersja_Kalkulatorakl

```
void Calculator::wersja_Kalkulatorakl ( ) [protected], [slot]
```

Ustawienie wersji kalkulatora na wersje klasyczną.

3.1.4 Dokumentacja atrybutów składowych**3.1.4.1 firstNum**

```
double Calculator::firstNum
```

Tutaj wpisywana jest pierwsza liczba do obliczeń.

3.1.4.2 tab

```
std::vector<QString> Calculator::tab
```

Wektor, w którym przechowywane są elementy dodane do pamięci.

3.1.4.3 useristypingseconnumber

```
bool Calculator::useristypingseconnumber = false
```

Warunek do sprawdzania czy użytkownik wpisuje już drugą cyfrę.

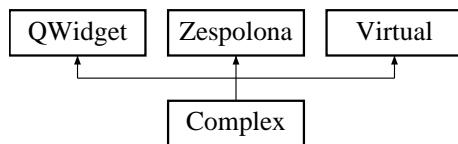
Dokumentacja dla tej klasy została wygenerowana z plików:

- Calculator/calculator.h
- Calculator/calculator.cpp

3.2 Dokumentacja klasy Complex

```
#include <complex.hpp>
```

Diagram dziedziczenia dla Complex



Sloty publiczne

- void [dzialania](#) () override
- void [close](#) () override

Metody publiczne

- [Complex](#) (QWidget *parent=Q_NULLPTR)
- [~Complex](#) ()

Atrybuty publiczne

- [Complex](#) * ze
- int [licznikcom](#) = 0
- Ui::Complex * [ui](#)

Dodatkowe Dziedziczone Składowe

3.2.1 Opis szczegółowy

Klasa odpowiedzialna za logikę dodatkowego modułu w kalkulatorze czyli wykonywania obliczeń na liczbach zespolonych

3.2.2 Dokumentacja konstruktora i destruktora

3.2.2.1 Complex()

```
Complex::Complex (   
    QWidget * parent = Q_NULLPTR )
```

Konstruktor klasy Complex

3.2.2.2 ~Complex()

```
Complex::~Complex ( )
```

Destruktor klasy Complex

3.2.3 Dokumentacja funkcji składowych

3.2.3.1 close

```
void Complex::close ( ) [override], [slot]
```

Niszczenie obiektu klasy Complex, po naciśnięciu przycisku zamknij.

3.2.3.2 dzialania

```
void Complex::dzialania ( ) [override], [slot]
```

Funkcja nadpisywana klasy bazowej Virtual. Tutaj wykonywane są wszystkie obliczenia na liczbach zespolonych

3.2.4 Dokumentacja atrybutów składowych

3.2.4.1 ze

```
Complex* Complex::ze
```

Wskaźnik na obiekt typu Complex wykorzystywany w klasie Calculator do dodania dodatkowego modułu do konwersji jednostek długości

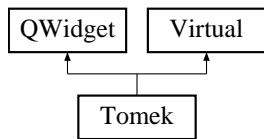
Dokumentacja dla tej klasy została wygenerowana z plików:

- Calculator/complex.hpp
- Calculator/complex.cpp

3.3 Dokumentacja klasy Tomek

```
#include <tomek.hpp>
```

Diagram dziedziczenia dla Tomek



Sloty publiczne

- void **close** () override
- void **dzialania** () override
- void **clickBackspace** ()
- void **keyPressEvent** (QKeyEvent *event)

Metody publiczne

- **Tomek** (QWidget *parent=Q_NULLPTR)

Atrybuty publiczne

- **Tomek** * **to** = nullptr

Atrybuty prywatne

- Ui::Tomek * **ui**
- std::map<QString, int> **jednostki**

3.3.1 Opis szczegółowy

Klasa odpowiedzialna za logike dodatkowego modułu w kalkulatorze czyli konwersji jednostek

3.3.2 Dokumentacja konstruktora i destruktora

3.3.2.1 Tomek()

```
Tomek::Tomek (
    QWidget * parent = Q_NULLPTR )
```

Konstruktor

3.3.3 Dokumentacja funkcji składowych

3.3.3.1 clickBackspace

```
void Tomek::clickBackspace ( ) [slot]
```

Usunięcie ostatniego chara z lineEdita

Przechowanie testu z lineEdita

3.3.3.2 dzialania

```
void Tomek::dzialania ( ) [override], [slot]
```

Tutaj na bazie drzewa utworzonego w prywatnych elementach klasy [Tomek](#), korzystając ze "Switch case" dokonywana jest konwersja jednostek

Jednostka z której użytkownik chce zamienić

Jednostka na którą użytkownik chce zamienić

Tutaj zapisywana jest liczba po obliczeniach zamiany

Tutaj wpisywana jest wartość po obliczeniach przekonwertowana na typ string

Tutaj wpisana jest zamieniona wartość string na typ QString w celu wpisania oblicznej wartości do lineEdita

Jednostka na jaką użytkownik chce zamienić wpisaną liczbę zamieniona już na jej odpowiednik w typie int

Jednostka z jakiej użytkownik chce zamienić wpisaną liczbę zamieniona już na jej odpowiednik w typie int

3.3.3.3 keyPressEvent

```
void Tomek::keyPressEvent (
    QKeyEvent * event ) [slot]
```

Obsługa klawiatury

3.3.4 Dokumentacja atrybutów składowych

3.3.4.1 jednostki

```
std::map<QString, int> Tomek::jednostki [private]
```

Tutaj przechowywane są wartości dla każdej jednostki wykorzystywane w konwersji jednostki typu string do int, w celu możliwości korzystania ze "switch case"

3.3.4.2 to

```
Tomek* Tomek::to = nullptr
```

Wskaźnik na obiekt typu **Tomek**, wykorzystywany w klasie **Calculator** w celu pokazania okna konwersji jednostek

3.3.4.3 ui

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

Wskaźnik na obiekt **Tomek**

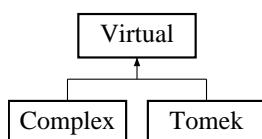
Dokumentacja dla tej klasy została wygenerowana z plików:

- Calculator/tomek.hpp
- Calculator/tomek.cpp

3.4 Dokumentacja klasy Virtual

```
#include <Virtual.h>
```

Diagram dziedziczenia dla Virtual



Metody publiczne

- virtual void **close** ()=0
- virtual void **dzialania** ()=0
 - wirtualna metoda close()

3.4.1 Opis szczegółowy

Wirtualna klasa bazowa klasy **Complex(liczby zespolone)**, oraz klasy **Tomek(zamiana jednostek)**

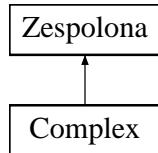
Dokumentacja dla tej klasy została wygenerowana z pliku:

- Calculator/Virtual.h

3.5 Dokumentacja klasy Zespolona

```
#include <Zespolone.h>
```

Diagram dziedziczenia dla Zespolona



Metody publiczne

- **Zespolona** (double a, double b)
- **Zespolona** ()
- **Zespolona** (**Zespolona** &a)
- **Zespolona operator+** (**Zespolona** b)
- **Zespolona operator-** (**Zespolona** b)
- **Zespolona operator*** (**Zespolona** b)
- **Zespolona operator/** (**Zespolona** b)
- bool **operator==** (const **Zespolona** &b)
- std::string **toString** ()

Atrybuty chronione

- double im
- double re

3.5.1 Opis szczegółowy

Klasa odpowiedzialna za działania na liczbach zespolonych, wykorzystywane jest tutaj przeciążenie operatorów, oraz strumień pamięciowy

3.5.2 Dokumentacja konstruktora i destruktora

3.5.2.1 Zespolona() [1/3]

```
Zespolona::Zespolona (
    double a,
    double b ) [inline]
```

Konstruktor liczby zespolonej

3.5.2.2 Zespolona() [2/3]

```
Zespolona::Zespolona ( ) [inline]
```

Konstruktor liczby zespolonej

3.5.2.3 Zespolona() [3/3]

```
Zespolona::Zespolona (
    Zespolona & a ) [inline]
```

Konstruktor kopiący

3.5.3 Dokumentacja funkcji składowych

3.5.3.1 operator*()

```
Zespolona Zespolona::operator* (
    Zespolona b ) [inline]
```

Operator mnożenia dwóch liczb zespolonych

3.5.3.2 operator+()

```
Zespolona Zespolona::operator+ (
    Zespolona b ) [inline]
```

Operator dodawania dwóch liczb zespolonych

3.5.3.3 operator-()

```
Zespolona Zespolona::operator- (
    Zespolona b ) [inline]
```

Operator odejmowania dwóch liczb zespolonych

3.5.3.4 operator/()

```
Zespolona Zespolona::operator/ (
    Zespolona b ) [inline]
```

Operator dzielenia dwóch liczb zespolonych

3.5.3.5 operator==()

```
bool Zespolona::operator== (
    const Zespolona & b ) [inline]
```

Operator porównania dwóch liczb zespolonych

3.5.3.6 toString()

```
std::string Zespolona::toString ( ) [inline]
```

Metoda zwracająca liczbę zespoloną w postaci stringa

3.5.4 Dokumentacja atrybutów składowych

3.5.4.1 im

```
double Zespolona::im [protected]
```

Część urojona liczby zespolonej

3.5.4.2 re

```
double Zespolona::re [protected]
```

Część rzeczywista liczby zespolonej

Dokumentacja dla tej klasy została wygenerowana z pliku:

- Calculator/Zespolone.h

Skorowidz

~Calculator
 Calculator, 6
~Complex
 Complex, 11

backspaceClicked
 Calculator, 7

Calculator, 5
 ~Calculator, 6
 backspaceClicked, 7
 Calculator, 6
 click_clear, 7
 click_minus, 7
 click_plus, 7
 click_podzielic, 7
 click_razy, 7
 click_rowne, 7
 clickDigit, 7
 clickPoint, 8
 dzialania, 8
 firstNum, 9
 keyPressEvent, 8
 klik_plus_minus, 8
 logarytmy, 8
 nowe, 8
 pamiec, 8
 random, 9
 tab, 9
 trygonometryczne, 9
 useristypingseconnumber, 9
 wersja_Kalkulatora, 9
 wersja_Kalkulatorakl, 9

click_clear
 Calculator, 7
click_minus
 Calculator, 7
click_plus
 Calculator, 7
click_podzielic
 Calculator, 7
click_razy
 Calculator, 7
click_rowne
 Calculator, 7
clickBackspace
 Tomek, 13
clickDigit
 Calculator, 7
clickPoint

Calculator, 8
close
 Complex, 11
Complex, 10
 ~Complex, 11
 close, 11
 Complex, 11
 dzialania, 11
 ze, 11

dzialania
 Calculator, 8
 Complex, 11
 Tomek, 13

firstNum
 Calculator, 9

im
 Zespolona, 17

jednostki
 Tomek, 14

keyPressEvent
 Calculator, 8
 Tomek, 13
 klik_plus_minus
 Calculator, 8

logarytmy
 Calculator, 8

nowe
 Calculator, 8

operator*
 Zespolona, 16
operator+
 Zespolona, 16
operator-
 Zespolona, 16
operator/
 Zespolona, 17
operator==
 Zespolona, 17

pamiec
 Calculator, 8

random

Calculator, 9
re
Zespolona, 17

tab
Calculator, 9

to
Tomek, 14

toString
Zespolona, 17

Tomek, 12

clickBackspace, 13
dzialania, 13
jednostki, 14
keyPressEvent, 13
to, 14
Tomek, 12
ui, 14

trygonometryczne
Calculator, 9

ui
Tomek, 14

useristypingseconnumber
Calculator, 9

Virtual, 14

wersja_Kalkulatora
Calculator, 9

wersja_Kalkulatorakl
Calculator, 9

ze
Complex, 11

Zespolona, 15

im, 17
operator*, 16
operator+, 16
operator-, 16
operator/, 17
operator==, 17
re, 17
toString, 17
Zespolona, 16