

# SymulacjaZbiornika

## 0.1

Wygenerowano przez Doxygen 1.8.6

Pn, 11 maj 2015 13:47:29



# Spis treści

<b>1</b>	<b>Wizualizacja rozkładu ciśnienia cieczy na podstawie symulacji komputerowej</b>	<b>1</b>
1.1	Opis projektu . . . . .	1
1.2	Funkcjonalności aplikacji . . . . .	1
<b>2</b>	<b>Indeks hierarchiczny</b>	<b>3</b>
2.1	Hierarchia klas . . . . .	3
<b>3</b>	<b>Indeks klas</b>	<b>5</b>
3.1	Lista klas . . . . .	5
<b>4</b>	<b>Indeks plików</b>	<b>7</b>
4.1	Lista plików . . . . .	7
<b>5</b>	<b>Dokumentacja klas</b>	<b>9</b>
5.1	Dokumentacja klasy Czasteczka . . . . .	9
5.1.1	Opis szczegółowy . . . . .	10
5.1.2	Dokumentacja konstruktora i destruktora . . . . .	10
5.1.2.1	Czasteczka . . . . .	10
5.1.3	Dokumentacja atrybutów składowych . . . . .	10
5.1.3.1	_Promien . . . . .	10
5.1.3.2	_RGB . . . . .	10
5.1.3.3	_x . . . . .	10
5.1.3.4	_y . . . . .	10
5.2	Dokumentacja klasy Kolor . . . . .	10
5.2.1	Opis szczegółowy . . . . .	11
5.2.2	Dokumentacja konstruktora i destruktora . . . . .	11
5.2.2.1	Kolor . . . . .	11
5.2.2.2	Kolor . . . . .	11
5.2.3	Dokumentacja atrybutów składowych . . . . .	11
5.2.3.1	_b . . . . .	11
5.2.3.2	_g . . . . .	12
5.2.3.3	_r . . . . .	12
5.3	Dokumentacja klasy OknoGlowne . . . . .	12

5.3.1	Opis szczegółowy . . . . .	13
5.3.2	Dokumentacja konstruktora i destruktora . . . . .	13
5.3.2.1	OknoGlowne . . . . .	13
5.3.3	Dokumentacja funkcji składowych . . . . .	13
5.3.3.1	GdyNapis . . . . .	13
5.4	Dokumentacja szablonu klasy <code>Vector&lt; TYPE &gt;</code> . . . . .	14
5.4.1	Opis szczegółowy . . . . .	15
5.4.2	Dokumentacja konstruktora i destruktora . . . . .	15
5.4.2.1	Vector . . . . .	15
5.4.2.2	Vector . . . . .	15
5.4.3	Dokumentacja funkcji składowych . . . . .	15
5.4.3.1	operator* . . . . .	15
5.4.3.2	operator*= . . . . .	15
5.4.3.3	operator+ . . . . .	15
5.4.3.4	operator+= . . . . .	16
5.4.3.5	operator- . . . . .	16
5.4.3.6	operator-= . . . . .	16
5.4.3.7	operator= . . . . .	16
5.4.3.8	operator[] . . . . .	16
5.4.3.9	operator[] . . . . .	16
5.4.4	Dokumentacja przyjaciół i funkcji związanych . . . . .	16
5.4.4.1	operator<< . . . . .	16
5.4.4.2	operator>> . . . . .	17
5.4.5	Dokumentacja atrybutów składowych . . . . .	17
5.4.5.1	_data . . . . .	17
5.4.5.2	_size . . . . .	17
5.5	Dokumentacja klasy Zbiornik . . . . .	17
5.5.1	Opis szczegółowy . . . . .	19
5.5.2	Dokumentacja konstruktora i destruktora . . . . .	19
5.5.2.1	Zbiornik . . . . .	19
5.5.3	Dokumentacja funkcji składowych . . . . .	19
5.5.3.1	GdyOdpowiedniCzas . . . . .	19
5.5.3.2	paintEvent . . . . .	19
5.5.3.3	RysujCzasteczke . . . . .	20
5.5.3.4	RysujZbiornik . . . . .	20
5.5.3.5	RysujZbiornikZCzasteczkami . . . . .	21
5.5.3.6	ZglosNapis . . . . .	21
5.5.4	Dokumentacja atrybutów składowych . . . . .	22
5.5.4.1	_Stoper . . . . .	22
5.5.4.2	Czasteczki . . . . .	22

<b>6 Dokumentacja plików</b>	<b>23</b>
6.1 Dokumentacja pliku ciec.cpp	23
6.1.1 Opis szczegółowy	24
6.1.2 Dokumentacja funkcji	24
6.1.2.1 main	24
6.1.3 Dokumentacja zmiennych	24
6.1.3.1 B	24
6.1.3.2 G	24
6.1.3.3 GRUBOSC	24
6.1.3.4 ODPOWIEDNI_CZAS	24
6.1.3.5 PODSTAWA	24
6.1.3.6 PROMIEN	24
6.1.3.7 R	24
6.1.3.8 WYSOKOSC	25
6.2 Dokumentacja pliku ciec.hh	25
6.2.1 Opis szczegółowy	26
6.3 Dokumentacja pliku czasteczka.hh	26
6.3.1 Opis szczegółowy	27
6.4 Dokumentacja pliku kolor.hh	27
6.4.1 Opis szczegółowy	27
6.5 Dokumentacja pliku moc_ciec.cpp	28
6.6 Dokumentacja pliku strona.dox	28
6.7 Dokumentacja pliku vector.hh	28
6.7.1 Dokumentacja funkcji	29
6.7.1.1 operator<<	29
6.7.1.2 operator>>	29
<b>Indeks</b>	<b>30</b>



## Rozdział 1

# Wizualizacja rozkładu ciśnienia cieczy na podstawie symulacji komputerowej

### Autor

Adam Balawender,  
Krzysztof Kwieciński,  
AiR, ARR, W4

### Data

11.05.2015

### Wersja

0.1

Aplikacja dotyczy komputerowej symulacji zachowania cieczy oraz wizualizacji jej stanu i rozkładu ciśnienia w zbiorniku z płynem.

## 1.1 Opis projektu

Symulacja będzie obejmowała ruch cieczy w przekroju 2D wybranego naczynia. Ciecz zostanie przedstawiona na płaszczyźnie jako zbiór oddziałujących ze sobą cząsteczek. Postaramy się, żeby jej zachowanie było możliwie zbliżone do rzeczywistego. Ruch płynu zostanie zamodelowany metodą numeryczną SPH (particle hydrodynamics - wygładzona hydrodynamika cząstek). Pozwoli to na realistyczne odwzorowanie zachowania cieczy. Możliwe będzie badanie cieczy o różnych parametrach, dlatego też modelowane będą jej właściwości fizyczne: gęstość i lepkość. Dodatkowo mierzone będzie ciśnienie cieczy i zostanie ono zwizualizowane jako odcień koloru płynu. Im będzie on ciemniejszy, tym wyższe ciśnienie będzie odzwierciedlał.

## 1.2 Funkcjonalności aplikacji

Najistotniejszymi funkcjonalnościami aplikacji będą:

- symulacja zachowania cieczy w zależności od zadanych warunków początkowych,
- item możliwość przededefiniowania parametrów cieczy (gęstości, lepkości),
- możliwość obserwacji wyniku symulacji (położenia cząsteczek i rozkładu ciśnień).





## Rozdział 2

# Indeks hierarchiczny

### 2.1 Hierarchia klas

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

Czasteczka . . . . .	9
Kolor . . . . .	10
QMainWindow	
OknoGlowne . . . . .	12
QWidget	
Zbiornik . . . . .	17
Vector< TYPE > . . . . .	14



## Rozdział 3

# Indeks klas

### 3.1 Lista klas

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

<a href="#">Czasteczka</a>		
	Klasa modelująca czasteczke . . . . .	9
<a href="#">Kolor</a>		
	Klasa modelująca kolor . . . . .	10
<a href="#">OknoGlowne</a>		
	Klasa modelujaca glowne okno aplikacji . . . . .	12
<a href="#">Vector&lt; TYPE &gt;</a>		14
<a href="#">Zbiornik</a>		
	Klasa modelująca zbiornik . . . . .	17



## Rozdział 4

# Indeks plików

### 4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

<a href="#">ciecz.cpp</a>	Zawiera definicje metod klasy <a href="#">Zbiornik</a> . . . . .	23
<a href="#">ciecz.hh</a>	Zawiera definicje klasy <a href="#">Zbiornik</a> i deklaracje jej metod . . . . .	25
<a href="#">czasteczka.hh</a>	Zawiera definicje klasy <a href="#">Kolor</a> oraz deklaracje jej metod . . . . .	26
<a href="#">kolor.hh</a>	Zawiera definicje klasy <a href="#">Kolor</a> oraz deklaracje jej metod . . . . .	27
<a href="#">moc_ciecz.cpp</a>	. . . . .	28
<a href="#">vector.hh</a>	Klasa przechowująca wektor Zawiera podstawowe metody pozwalające stworzyć odpowiedni obiekt, odczytywać i zapisywać dane do niego, wczytać i wydrukować na standardowych strumieniach, a także odjąć od niego inny wektor . . . . .	28



## Rozdział 5

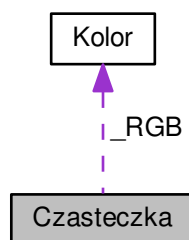
# Dokumentacja klas

### 5.1 Dokumentacja klasy Czasteczka

Klasa modelująca czasteczke.

```
#include <czasteczka.hh>
```

Diagram współpracy dla Czasteczka:



#### Metody publiczne

- **Czasteczka** (int x, int y, int r, const **Kolor** &rgb)  
*Konstruktor.*

#### Atrybuty publiczne

- int **\_x**  
*Atrybut opisujący położenie czasteczki na osi x.*
- int **\_y**  
*Atrybut opisujący położenie czasteczki na osi y.*
- int **\_Promien**  
*Atrybut określający promień czasteczki.*
- **Kolor** **\_RGB**  
*Atrybut opisujący kolor czasteczki.*

### 5.1.1 Opis szczegółowy

Klasa zawierająca podstawowe atrybuty czasteczki.

Definicja w linii 26 pliku czasteczka.hh.

### 5.1.2 Dokumentacja konstruktora i destruktora

#### 5.1.2.1 Czasteczka::Czasteczka ( int x, int y, int r, const Kolor & rgb ) [inline]

Konstruktor parametryczny, inicjalizujący czasteczkę podanymi własnościami.

Parametry

in	x	- położenie czasteczki na osi x
in	y	- położenie czasteczki na osi y
in	r	- promień czasteczki
in	rgb	- kolor czasteczki

Definicja w linii 38 pliku czasteczka.hh.

### 5.1.3 Dokumentacja atrybutów składowych

#### 5.1.3.1 int Czasteczka::\_Promien

Atrybut określający promień czasteczki.

Definicja w linii 63 pliku czasteczka.hh.

#### 5.1.3.2 Kolor Czasteczka::\_RGB

Atrybut opisujący kolor czasteczki.

Definicja w linii 70 pliku czasteczka.hh.

#### 5.1.3.3 int Czasteczka::\_x

Atrybut opisujący położenie czasteczki na osi x.

Definicja w linii 39 pliku czasteczka.hh.

#### 5.1.3.4 int Czasteczka::\_y

Atrybut opisujący położenie czasteczki na osi y.

Definicja w linii 56 pliku czasteczka.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [czasteczka.hh](#)

## 5.2 Dokumentacja klasy Kolor

Klasa modelująca kolor.

```
#include <kolor.hh>
```



## Metody publiczne

- **Kolor** (int r, int g, int b)  
*Konstruktor.*
- **Kolor** (const **Kolor** &rgb)  
*Konstruktor kopiujacy.*

## Atrybuty publiczne

- int **\_r**  
*Atrybut opisujacy wartosc odcieniu czerwonego.*
- int **\_g**  
*Atrybut opisujacy wartosc odcieniu zielonego.*
- int **\_b**  
*Atrybut opisujacy wartosc odcieniu niebieskiego.*

### 5.2.1 Opis szczegółowy

Klasa opisuje kolor w formacie RGB.

Definicja w linii 24 pliku kolor.hh.

### 5.2.2 Dokumentacja konstruktora i destruktora

#### 5.2.2.1 **Kolor::Kolor** ( int r, int g, int b ) [inline]

Konstruktor parametryczny, inicjalizujący kolor podanymi wartościami.

Parametry

in	<i>r</i>	- wartosc odcieniu czerwonego, [0, 255]
in	<i>g</i>	- wartosc odcieniu zielonego, [0, 255]
in	<i>b</i>	- wartosc odcieniu niebieskiego, [0, 255]

Definicja w linii 35 pliku kolor.hh.

#### 5.2.2.2 **Kolor::Kolor** ( const **Kolor** & rgb ) [inline]

Konstruktor kopiujacy.

Parametry

in	<i>rgb</i>	- obiekt do skopiowania
----	------------	-------------------------

Definicja w linii 43 pliku kolor.hh.

### 5.2.3 Dokumentacja atrybutów składowych

#### 5.2.3.1 int **Kolor::\_b**

Atrybut opisujacy wartosc odcieniu niebieskiego.

Definicja w linii 66 pliku kolor.hh.

#### 5.2.3.2 int Kolor::\_g

Atrybut opisujący wartość odcienia zielonego.

Definicja w linii 59 pliku kolor.hh.

#### 5.2.3.3 int Kolor::\_r

Atrybut opisujący wartość odcienia czerwonego.

Definicja w linii 43 pliku kolor.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [kolor.hh](#)

## 5.3 Dokumentacja klasy OknoGlowne

Klasa modelująca główne okno aplikacji.

```
#include <ciecz.hh>
```

Diagram dziedziczenia dla OknoGlowne

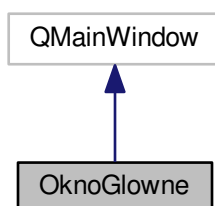
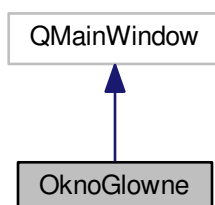


Diagram współpracy dla OknoGlowne:



## Sloty publiczne

- void `GdyNapis` (const QString &)

*Slot odpowiadający za wyświetlenie napisu po otrzymaniu sygnału.*

## Metody publiczne

- `OknoGlowne` (QWidget \*wRodzic=nullptr)

*Konstruktor.*

### 5.3.1 Opis szczegółowy

Dzięki tej klasie wyświetlane jest okno główne aplikacji.

Definicja w linii 137 pliku `ciecz.hh`.

### 5.3.2 Dokumentacja konstruktora i destruktor

#### 5.3.2.1 `OknoGlowne::OknoGlowne ( QWidget * wRodzic = nullptr )`

Konstruktor parametryczny.

Parametry

<code>in, out</code>	<code>wRodzic</code>	- wskaźnik na rodzica
----------------------	----------------------	-----------------------

Definicja w linii 129 pliku `ciecz.cpp`.

Oto graf wywołań dla tej funkcji:



### 5.3.3 Dokumentacja funkcji składowych

#### 5.3.3.1 `void OknoGlowne::GdyNapis ( const QString & Napis ) [slot]`

Odpowiada za wyświetlenie napisu na belce statusowej.

Parametry

<code>in</code>	<code>Napis</code>	- napis do wyświetlenia
-----------------	--------------------	-------------------------

Definicja w linii 149 pliku `ciecz.cpp`.

Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- [ciecz.hh](#)
- [ciecz.cpp](#)

## 5.4 Dokumentacja szablonu klasy `Vector< TYPE >`

```
#include <vector.hh>
```

### Metody publiczne

- `Vector` (const int size=0)  
*Konstruktor wektora.*
- `Vector` (const int &size, const TYPE \*data)  
*Konstruktor wektora z możliwością inicjowania danymi.*
- `TYPE & operator[]` (const int &i)  
*Operator dostępu do współrzędnych wektora.*
- const `TYPE operator[]` (const int &i) const  
*Operator dostępu do współrzędnych wektora.*
- `Vector operator-` (const `Vector` &vect) const  
*Operacja odejmowania wektorów.*
- `Vector & operator-=` (const `Vector` &vect)  
*Operacja odejmowania wektorów.*
- `Vector operator+` (const `Vector` &vect) const  
*Operacja dodawania wektorów.*
- `Vector & operator+=` (const `Vector` &vect)  
*Operacja dodawania wektorów.*
- `Vector operator*` (const long &scale) const  
*Operacja mnożenia wektora przez stałą*
- `Vector & operator*=` (const long &scale)  
*Operacja mnożenia wektora przez stałą*
- `Vector & operator=` (const `Vector` &vect)  
*Zmienia zawartość wektora.*

### Atrybuty prywatne

- `TYPE * _data`
- `int _size`

## Przyjaciele

- std::istream & **operator>>** (std::istream &, **Vector**< TYPE > &)

*Przeciążenie standardowego operatora wejściowego dla wektora. Wczytuje wektor. W sumie nie potrzebne, ale skoro przeciążam wyjście, to wejście też przeciążę. A co!*

- std::ostream & **operator<<** (std::ostream &, const **Vector**< TYPE > &)

*Przeciążenie standardowego operatora wyjściowego dla wektora. Wyświetla wektor.*

### 5.4.1 Opis szczegółowy

```
template<typename TYPE>class Vector< TYPE >
```

Definicja w linii 11 pliku vector.hh.

### 5.4.2 Dokumentacja konstruktora i destruktora

5.4.2.1 `template<typename TYPE > Vector< TYPE >::Vector ( const int size = 0 ) [inline]`

Parametry

in	<i>size</i>	- rozmiar, domyślnie SIZE
----	-------------	---------------------------

Definicja w linii 41 pliku vector.hh.

5.4.2.2 `template<typename TYPE > Vector< TYPE >::Vector ( const int & size, const TYPE * data )`

Parametry

in	<i>size</i>	- rozmiar
in	* <i>data</i>	- dane

### 5.4.3 Dokumentacja funkcji składowych

5.4.3.1 `template<typename TYPE > Vector Vector< TYPE >::operator* ( const long & scale ) const`

Zwraca

Zwraca wynik w postaci wektora

5.4.3.2 `template<typename TYPE > Vector& Vector< TYPE >::operator*= ( const long & scale )`

Zwraca

Zwraca wynik w postaci wektora

5.4.3.3 `template<typename TYPE > Vector Vector< TYPE >::operator+ ( const Vector< TYPE > & vect ) const`

Zwraca

Zwraca wynik w postaci wektora

**5.4.3.4** `template<typename TYPE > Vector& Vector< TYPE >::operator+=( const Vector< TYPE > & vect )`

Zwraca

Zwraca wynik w postaci wektora

**5.4.3.5** `template<typename TYPE > Vector Vector< TYPE >::operator- ( const Vector< TYPE > & vect ) const`

Zwraca

Zwraca wynik w postaci wektora

**5.4.3.6** `template<typename TYPE > Vector& Vector< TYPE >::operator-= ( const Vector< TYPE > & vect )`

Zwraca

Zwraca wynik w postaci wektora

**5.4.3.7** `template<typename TYPE > Vector& Vector< TYPE >::operator= ( const Vector< TYPE > & vect )`

Parametry

in	<a href="#">Vector</a>	- nowa zawartość wektora
----	------------------------	--------------------------

**5.4.3.8** `template<typename TYPE > TYPE& Vector< TYPE >::operator[] ( const int & i ) [inline]`

Parametry

in	<i>i</i>	- numer danej
----	----------	---------------

Zwraca

Referencja do odpowiedniego elementu

Definicja w linii 53 pliku vector.hh.

**5.4.3.9** `template<typename TYPE > const TYPE Vector< TYPE >::operator[] ( const int & i ) const [inline]`

Parametry

in	<i>i</i>	- numer danej
----	----------	---------------

Zwraca

Referencja do odpowiedniego elementu

Definicja w linii 59 pliku vector.hh.

## 5.4.4 Dokumentacja przyjaciół i funkcji związanych

**5.4.4.1** `template<typename TYPE > std::ostream& operator<< ( std::ostream & , const Vector< TYPE > & ) [friend]`

## Parametry

<i>in, out</i>	<i>out</i>	- strumień wyjściowy
<i>in</i>	<i>vect</i>	- wektor

## Zwraca

Zwraca referencję do zmodyfikowanego strumienia.

5.4.4.2 `template<typename TYPE > std::istream& operator>> ( std::istream & , Vector< TYPE > & ) [friend]`

## Parametry

<i>in, out</i>	<i>inp</i>	- strumień wejściowy
<i>out</i>	<i>vect</i>	- wektor

## Zwraca

Zwraca referencję do zmodyfikowanego strumienia.

## 5.4.5 Dokumentacja atrybutów składowych

5.4.5.1 `template<typename TYPE > TYPE* Vector< TYPE >::_data [private]`

Definicja w linii 34 pliku vector.hh.

5.4.5.2 `template<typename TYPE > int Vector< TYPE >::_size [private]`

Definicja w linii 35 pliku vector.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [vector.hh](#)

## 5.5 Dokumentacja klasy Zbiornik

Klasa modelująca zbiornik.

```
#include <ciecz.hh>
```

Diagram dziedziczenia dla Zbiornik

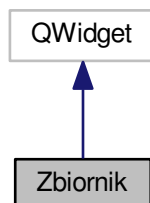
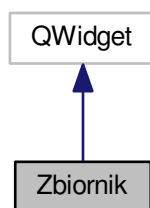


Diagram współpracy dla Zbiornik:



### Sloty publiczne

- void `GdyOdpowiedniCzas` ()  
*Slot odpowiadający za aktualizację danych. .*

### Sygnały

- void `ZglosNapis` (const QString &)  
*Sygnał zgłaszający napis.*

### Metody publiczne

- `Zbiornik` (QWidget \*wRodzic=nullptr)  
*Konstruktor.*
- virtual void `paintEvent` (QPaintEvent \*event)  
*Wirtualna metoda paintEvent wyrysowująca obiekt na ekranie.*
- void `RysujZbiornik` (QPainter &Rysownik, const int Podstawa, const int Wysokosc, const int Grubosc, const int x, const int y)  
*Metoda rysująca zbiornik.*
- void `RysujCzasteczke` (QPainter &Rysownik, const int Promien, const Kolor RGB, const double x, const double y)  
*Metoda rysująca czasteczkę.*
- void `RysujZbiornikZCzasteczkami` (QPainter &Rysownik)  
*Metoda rysująca zbiornik wraz z czasteczkami.*

### Atrybuty publiczne

- std::list< `Czasteczka` > `Czasteczki`  
*Lista wszystkich czasteczek.*

### Atrybuty prywatne

- QTimer `_Stoper`  
*Miernik czasu.*



### 5.5.1 Opis szczegółowy

Dzięki tej klasie możliwe jest wyrysowywanie na ekranie zbiornika.

Definicja w linii 34 pliku cieczeni.h.

### 5.5.2 Dokumentacja konstruktora i destruktor

#### 5.5.2.1 Zbiornik::Zbiornik ( QWidget \* wRodzic = nullptr )

Konstruktor parametryczny.

Parametry

in, out	wRodzic	- wskaźnik na rodzica
---------	---------	-----------------------

Definicja w linii 46 pliku cieczeni.cpp.

Oto graf wywołań dla tej funkcji:



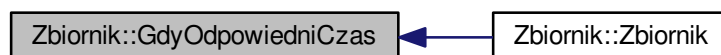
### 5.5.3 Dokumentacja funkcji składowych

#### 5.5.3.1 void Zbiornik::GdyOdpowiedniCzas ( ) [slot]

Odpowiada za uaktualnienie okienka w odpowiednich momentach.

Definicja w linii 111 pliku cieczeni.cpp.

Oto graf wywołań tej funkcji:



#### 5.5.3.2 void Zbiornik::paintEvent ( QPaintEvent \* event ) [virtual]

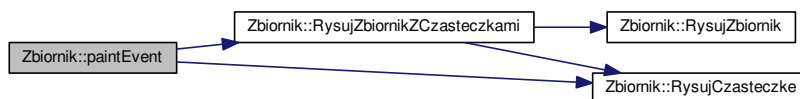
Odziedziczona wirtualna metoda paintEvent.

## Parametry

in, out	event	- wskaźnik obiekt klasy QPaintEvent
---------	-------	-------------------------------------

Definicja w linii 97 pliku cieczech.cpp.

Oto graf wywołań dla tej funkcji:



### 5.5.3.3 void Zbiornik::RysujCzasteczke ( QPainter & Rysownik, const int Promien, const Kolor RGB, const double x, const double y )

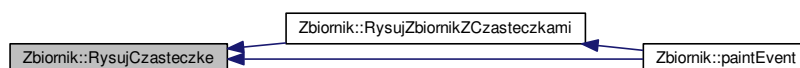
Rysuje czasteczke o zadanych parametrach

## Parametry

in, out	Rysownik	- referencja na obiekt klasy QPainter
in	Promien	- promien czasteczki
in	RGB	- kolor czasteczki w formacie RGB
in	x	- polozenie czasteczki na osi x
in	y	- polozenie czasteczki na osi y

Definicja w linii 74 pliku cieczech.cpp.

Oto graf wywoływań tej funkcji:



### 5.5.3.4 void Zbiornik::RysujZbiornik ( QPainter & Rysownik, const int Podstawa, const int Wysokosc, const int Grubosc, const int x, const int y )

Rysuje zbiornik o zadanych parametrach.

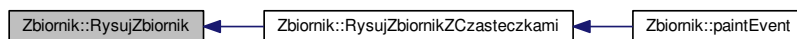
## Parametry

in, out	Rysownik	- referencja na obiekt klasy QPainter
in	Podstawa	- dlugosc podstawy zbiornika
in	Wysokosc	- wysokosc zbiornika
in	Grubosc	- grubosc wyrysowywanego odcinka
in	x	- polozenie lewego gornego punktu zbiornika na osi x

<code>in</code>	<code>y</code>	- położenie lewego gornego punktu zbiornika na osi y
-----------------	----------------	--

Definicja w linii 57 pliku `ciecz.cpp`.

Oto graf wywołań tej funkcji:



#### 5.5.3.5 void Zbiornik::RysujZbiornikZCzasteczkami ( QPainter & Rysownik )

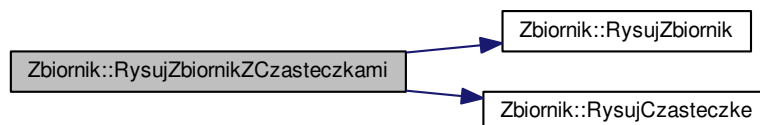
Rysuje zbiornik oraz czasteczki.

Parametry

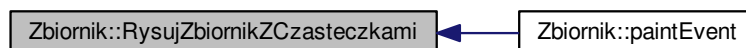
<code>in, out</code>	<code>Rysownik</code>	- referencja na obiekt klasy QPainter
----------------------	-----------------------	---------------------------------------

Definicja w linii 88 pliku `ciecz.cpp`.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



#### 5.5.3.6 void Zbiornik::ZglosNapis ( const QString & \_t1 ) [signal]

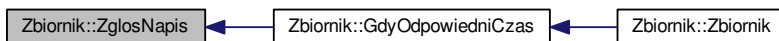
Sygnal zgłaszający napis do odpowiedniego slotu.

Parametry

in	<code>_t1</code>	- napis do zgłoszenia
----	------------------	-----------------------

Definicja w linii 99 pliku `moc_ciecz.cpp`.

Oto graf wywołań tej funkcji:



## 5.5.4 Dokumentacja atrybutów składowych

### 5.5.4.1 QTimer Zbiornik::\_Stoper [private]

Miernik czasu.

Definicja w linii 127 pliku `ciecz.hh`.

### 5.5.4.2 `std::list<Czasteczka>` Zbiornik::Czasteczki

Lista wszystkich czasteczek.

Definicja w linii 119 pliku `ciecz.hh`.

Dokumentacja dla tej klasy została wygenerowana z plików:

- [ciecz.hh](#)
- [moc\\_ciecz.cpp](#)
- [ciecz.cpp](#)

## Rozdział 6

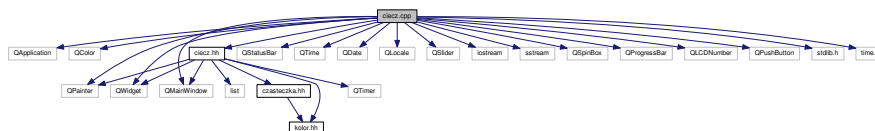
# Dokumentacja plików

### 6.1 Dokumentacja pliku cieczeni.cpp

Zawiera definicje metod klasy [Zbiornik](#).

```
#include <QApplication>
#include <QColor>
#include <QPainter>
#include <QStatusBar>
#include <QTime>
#include <QDate>
#include <QLocale>
#include <QWidget>
#include <QSlider>
#include <iostream>
#include <sstream>
#include <QMainWindow>
#include <QSpinBox>
#include <QProgressBar>
#include <QLCDNumber>
#include <QPushButton>
#include <stdlib.h>
#include <time.h>
#include "ciezeni.hh"
```

Wykres zależności załączania dla cieczeni.cpp:



### Funkcje

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

### Zmienne

- const int [PODSTAWA](#) = 200
- const int [WYSOKOSC](#) = 200

- `const int GRUBOSC = 3`
- `const int PROMIEN = 10`
- `const int R = 255`
- `const int G = 0`
- `const int B = 0`
- `const int ODPOWIEDNI_CZAS = 1000`

### 6.1.1 Opis szczegółowy

W pliku znajdują się:

- definicje konstruktorów, metod i przeciążeń klasy [Zbiornik](#).

Definicja w pliku [ciecz.cpp](#).

### 6.1.2 Dokumentacja funkcji

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

Definicja w linii 155 pliku [ciecz.cpp](#).

### 6.1.3 Dokumentacja zmiennych

#### 6.1.3.1 `const int B = 0`

Definicja w linii 42 pliku [ciecz.cpp](#).

#### 6.1.3.2 `const int G = 0`

Definicja w linii 41 pliku [ciecz.cpp](#).

#### 6.1.3.3 `const int GRUBOSC = 3`

Definicja w linii 38 pliku [ciecz.cpp](#).

#### 6.1.3.4 `const int ODPOWIEDNI_CZAS = 1000`

Definicja w linii 43 pliku [ciecz.cpp](#).

#### 6.1.3.5 `const int PODSTAWA = 200`

Definicja w linii 36 pliku [ciecz.cpp](#).

#### 6.1.3.6 `const int PROMIEN = 10`

Definicja w linii 39 pliku [ciecz.cpp](#).

#### 6.1.3.7 `const int R = 255`

Definicja w linii 40 pliku [ciecz.cpp](#).

6.1.3.8 `const int WYSOKOSC = 200`

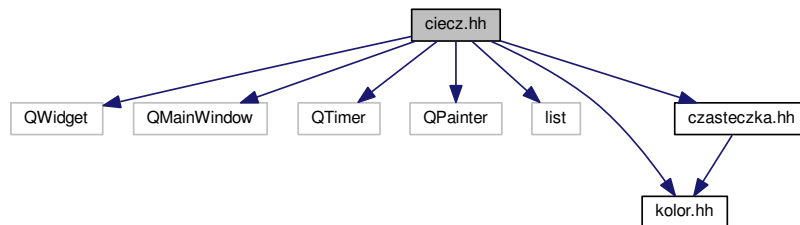
Definicja w linii 37 pliku ciecz.cpp.

## 6.2 Dokumentacja pliku ciecz.hh

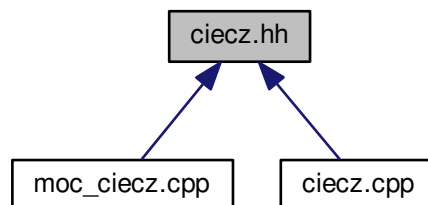
Zawiera definicje klasy [Zbiornik](#) i deklaracje jej metod.

```
#include <QWidget>
#include <QMainWindow>
#include <QTimer>
#include <QPainter>
#include <list>
#include "kolor.hh"
#include "czasteczka.hh"
```

Wykres zależności załączania dla ciecz.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



### Komponenty

- class [Zbiornik](#)  
*Klasa modelująca zbiornik.*
- class [OknoGlowne](#)  
*Klasa modelująca główne okno aplikacji.*

### 6.2.1 Opis szczegółowy

W pliku znajdują się:

- definicja klasy [Zbiornik](#) (modeluje pojęcie Zbiornika),
- deklaracje konstruktorów, metod i przeciążeń ww. klasy.

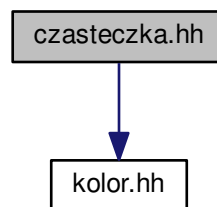
Definicja w pliku [ciecz.hh](#).

## 6.3 Dokumentacja pliku czasteczka.hh

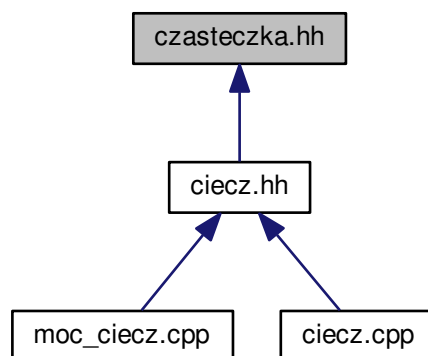
Zawiera definicje klasy [Kolor](#) oraz deklaracje jej metod.

```
#include "kolor.hh"
```

Wykres zależności załączania dla czasteczka.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



### Komponenty

- class [Czasteczka](#)



*Klasa modelująca czasteczke.*

### 6.3.1 Opis szczegółowy

W pliku znajdują się:

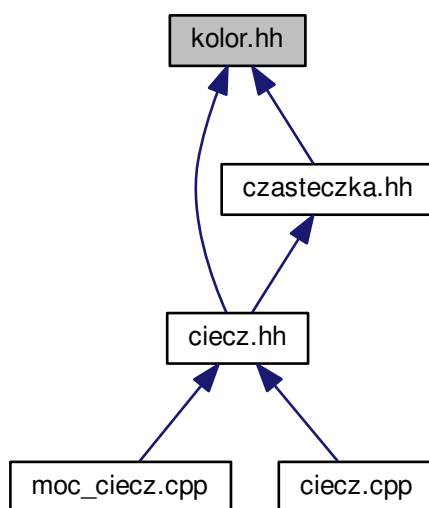
- definicja klasy `Czasteczka` (modeluje pojęcie czasteczki),
- definicje konstruktorów,
- deklaracje atrybutów.

Definicja w pliku `czasteczka.hh`.

## 6.4 Dokumentacja pliku kolor.hh

Zawiera definicje klasy `Kolor` oraz deklaracje jej metod.

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



### Komponenty

- class `Kolor`

*Klasa modelująca kolor.*

### 6.4.1 Opis szczegółowy

W pliku znajdują się:

- definicja klasy `Kolor` (modeluje pojęcie Koloru),

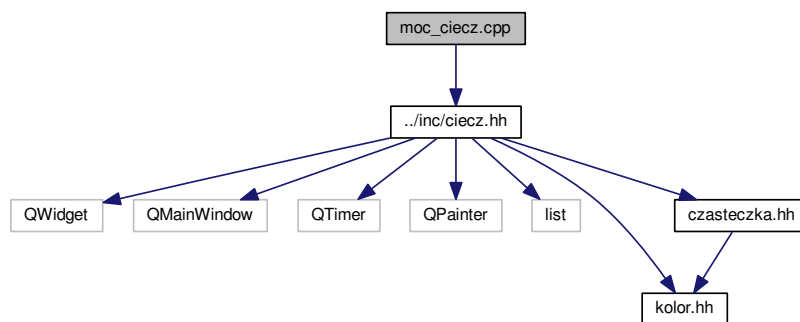
- definicje konstruktorów,
- deklaracje atrybutów.

Definicja w pliku [kolor.hh](#).

## 6.5 Dokumentacja pliku moc\_ciecz.cpp

```
#include "../inc/ciecz.hh"
```

Wykres zależności załączania dla moc\_ciecz.cpp:



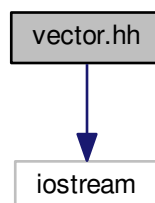
## 6.6 Dokumentacja pliku strona.dox

## 6.7 Dokumentacja pliku vector.hh

Klasa przechowująca wektor. Zawiera podstawowe metody pozwalające stworzyć odpowiedni obiekt, odczytywać i zapisywać dane do niego, wczytać i wydrukować na standardowych strumieniach, a także odjąć od niego inny wektor.

```
#include <iostream>
```

Wykres zależności załączania dla vector.hh:



## Komponenty

- class `Vector< TYPE >`
- class `Vector< TYPE >`

## Funkcje

- `template<typename TYPE >`  
`std::istream & operator>> (std::istream &, Vector< TYPE > &)`
- `template<typename TYPE >`  
`std::ostream & operator<< (std::ostream &, const Vector< TYPE > &)`

### 6.7.1 Dokumentacja funkcji

#### 6.7.1.1 `template<typename TYPE > std::ostream& operator<< ( std::ostream & , const Vector< TYPE > & )`

##### Parametry

<code>in, out</code>	<code>out</code>	- strumień wyjściowy
<code>in</code>	<code>vect</code>	- wektor

##### Zwraca

Zwraca referencję do zmodyfikowanego strumienia.

#### 6.7.1.2 `template<typename TYPE > std::istream& operator>> ( std::istream & , Vector< TYPE > & )`

##### Parametry

<code>in, out</code>	<code>inp</code>	- strumień wejściowy
<code>out</code>	<code>vect</code>	- wektor

##### Zwraca

Zwraca referencję do zmodyfikowanego strumienia.

# Skorowidz

`_Promien`  
    Czasteczka, 10  
`_RGB`  
    Czasteczka, 10  
`_Stoper`  
    Zbiornik, 22  
`_b`  
    Kolor, 11  
`_data`  
    Vector, 17  
`_g`  
    Kolor, 11  
`_r`  
    Kolor, 12  
`_size`  
    Vector, 17  
`_x`  
    Czasteczka, 10  
`_y`  
    Czasteczka, 10

**B**  
    ciecz.cpp, 24

ciecz.cpp, 23  
    B, 24  
    G, 24  
    GRUBOSC, 24  
    main, 24  
    ODPOWIEDNI\_CZAS, 24  
    PODSTAWA, 24  
    PROMIEN, 24  
    R, 24  
    WYSOKOSC, 24

ciecz.hh, 25  
Czasteczka, 9  
    Promien, 10  
    RGB, 10  
    x, 10  
    y, 10  
    Czasteczka, 10

czasteczka.hh, 26  
Czasteczki  
    Zbiornik, 22

**G**  
    ciecz.cpp, 24

GRUBOSC  
    ciecz.cpp, 24

GdyNapis

OknoGlowne, 13  
GdyOdpowiedniCzas  
    Zbiornik, 19

Kolor, 10  
    b, 11  
    g, 11  
    r, 12  
    Kolor, 11  
kolor.hh, 27

main  
    ciecz.cpp, 24  
moc\_ciecz.cpp, 28

ODPOWIEDNI\_CZAS  
    ciecz.cpp, 24

OknoGlowne, 12  
    GdyNapis, 13  
    OknoGlowne, 13  
    OknoGlowne, 13

operator<<  
    Vector, 16  
    vector.hh, 29

operator>>  
    Vector, 17  
    vector.hh, 29

operator\*  
    Vector, 15

operator\*=  
    Vector, 15

operator+  
    Vector, 15

operator+=  
    Vector, 15

operator-  
    Vector, 16

operator-=  
    Vector, 16

operator=  
    Vector, 16

PODSTAWA  
    ciecz.cpp, 24

PROMIEN  
    ciecz.cpp, 24

paintEvent  
    Zbiornik, 19

**R**

ciecz.cpp, [24](#)  
RysujCzasteczke  
    Zbiornik, [20](#)  
RysujZbiornik  
    Zbiornik, [20](#)  
RysujZbiornikZCzasteczkami  
    Zbiornik, [21](#)  
  
strona.dox, [28](#)  
  
Vector  
    \_data, [17](#)  
    \_size, [17](#)  
    operator<<, [16](#)  
    operator>>, [17](#)  
    operator\*, [15](#)  
    operator\*=[, 15](#)  
    operator+, [15](#)  
    operator+=, [15](#)  
    operator-, [16](#)  
    operator-=, [16](#)  
    operator=, [16](#)  
    Vector, [15](#)  
Vector< TYPE >, [14](#)  
vector.hh, [28](#)  
    operator<<, [29](#)  
    operator>>, [29](#)  
  
WYSOKOSC  
    ciecz.cpp, [24](#)  
  
Zbiornik, [17](#)  
    \_Stoper, [22](#)  
    Czasteczki, [22](#)  
    GdyOdpowiedniCzas, [19](#)  
    paintEvent, [19](#)  
    RysujCzasteczke, [20](#)  
    RysujZbiornik, [20](#)  
    RysujZbiornikZCzasteczkami, [21](#)  
    Zbiornik, [19](#)  
    ZglosNapis, [21](#)  
ZglosNapis  
    Zbiornik, [21](#)