

# Program do Algorytmow

## 1.1

Generated by Doxygen 1.8.4

Tue Mar 11 2014 21:28:49



# Contents

<b>1</b>	<b>Operacje na pliku tekstowym</b>	<b>1</b>
1.1	Działanie: . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	algorytm Class Reference . . . . .	7
4.1.1	Detailed Description . . . . .	8
4.1.2	Member Function Documentation . . . . .	8
4.1.2.1	dodaj_element . . . . .	8
4.1.2.2	dodaj_elementy . . . . .	8
4.1.2.3	odwroc_tablice . . . . .	8
4.1.2.4	operator= . . . . .	9
4.1.2.5	pobierz_czas . . . . .	9
4.1.2.6	sprawdz . . . . .	9
4.1.2.7	test . . . . .	9
4.1.2.8	wczytaj_dane . . . . .	10
4.1.2.9	wczytaj_dane_sprawdzajace . . . . .	10
4.1.2.10	wykonaj_obliczenia . . . . .	11
4.1.2.11	zamien_elementy . . . . .	11
4.1.3	Member Data Documentation . . . . .	11
4.1.3.1	elementy . . . . .	11
4.1.3.2	powtorzenia . . . . .	11
4.1.3.3	tab_danych . . . . .	11
4.1.3.4	tab_obliczone . . . . .	11
4.1.3.5	tab_sprawdzajace . . . . .	12
4.2	dane Class Reference . . . . .	12
4.2.1	Detailed Description . . . . .	12

4.2.2	Constructor & Destructor Documentation . . . . .	12
4.2.2.1	dane . . . . .	12
4.2.3	Member Function Documentation . . . . .	13
4.2.3.1	wylicz_odchylenie . . . . .	13
4.2.3.2	zapisz_do_csv . . . . .	13
4.2.4	Member Data Documentation . . . . .	13
4.2.4.1	czas_operacji . . . . .	13
4.2.4.2	odchylenie . . . . .	13
4.2.4.3	powtorzenia . . . . .	14
4.2.4.4	tab_czasow . . . . .	14
<b>5</b>	<b>File Documentation</b>	<b>15</b>
5.1	algorytm.cpp File Reference . . . . .	15
5.1.1	Detailed Description . . . . .	15
5.2	algorytm.h File Reference . . . . .	16
5.2.1	Detailed Description . . . . .	16
5.3	dane.cpp File Reference . . . . .	17
5.3.1	Detailed Description . . . . .	17
5.4	dane.h File Reference . . . . .	17
5.4.1	Detailed Description . . . . .	18
5.5	main.cpp File Reference . . . . .	18
5.5.1	Detailed Description . . . . .	19
5.5.2	Function Documentation . . . . .	19
5.5.2.1	main . . . . .	19
5.6	strona.dox File Reference . . . . .	20
	<b>Index</b>	<b>21</b>

# Chapter 1

## Operacje na pliku tekstowym

### Author

Radoslaw Chudy

### Date

03.03.2014

### Version

1.0

Program wykonuje operacje na danych wczytanych z pliku tekstowego.

### 1.1 Działanie:

Program ma za zadanie wykonanie operacji wczytania danych z pliku, następnie wykonywana jest operacja pomnożenia danego pliku przez liczbę 2. Uzyskane wyniki są porównywane z plikiem sprawdzającym. W czasie wykonywania operacji mnożenia mierzony jest czas wykonania tej operacji. Program pozwala na kilkukrotne wykonywanie tej operacji. Na koniec w programie wyliczone zostają dane podsumowujące tzn. liczba powtórzeń, czas danego powtórzenia i odchylenie standardowe. Te dane zostają zapisane w pliku formatu csv.



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">algorytm</a>	Modeluje pojecie algorytmu . . . . .	<a href="#">7</a>
<a href="#">dane</a>	Modeluje pojecie dane . . . . .	<a href="#">12</a>





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">algorytm.cpp</a>	Plik z definicjami metod dla klasy algorytm . . . . .	15
<a href="#">algorytm.h</a>	Definicja klasy algorytm . . . . .	16
<a href="#">dane.cpp</a>	Plik z definicjami metod dla klasy dane . . . . .	17
<a href="#">dane.h</a>	Definicja klasy dane . . . . .	17
<a href="#">main.cpp</a>	Funkcja main . . . . .	18



## Chapter 4

# Class Documentation

### 4.1 algorytm Class Reference

Modeluje pojecie algorytmu.

```
#include <algorytm.h>
```

#### Public Member Functions

- int \* [wczytaj\\_dane](#) ()  
*Metoda wczytujaca dane z pliku tesowego.*
- clock\_t [pobierz\\_czas](#) ()  
*Metoda pobierajaca czas.*
- int \* [wykonaj\\_obliczenia](#) ()  
*Metoda wykonaj obliczenia.*
- int \* [wczytaj\\_dane\\_sprawdzajace](#) ()  
*Metoda wczytujaca dane sprawdzajace z pliku tesowego.*
- bool [sprawdz](#) ()  
*Metoda Sprawdz.*
- void [zamien\\_elementy](#) (int \*tablica, int pierwszy, int drugi)  
*Metoda zamien elementy.*
- void [odwroc\\_tablice](#) (int \*tablica)  
*Metoda odwroc tablice.*
- int \* [dodaj\\_element](#) (int \*tablica, int element)  
*Metoda dodaj elementy.*
- int \* [dodaj\\_elementy](#) (int \*tab\_pierwsza, int \*tab\_druga)  
*Metoda dodaj elementy.*
- int \* [operator=](#) (int \*tab\_pierwsza)  
*Przeciazenie operatora przypisania.*
- int [test](#) (dane \*Info)  
*Metoda test.*

#### Public Attributes

- int [powtorzenia](#)  
*Pole powtorzenia.*
- int [elementy](#)

- Pole elementy.*
- int \* [tab\\_danych](#)  
*Wskaźnik tab\_danych.*
- int \* [tab\\_obliczone](#)  
*Wskaźnik tab\_obliczone.*
- int \* [tab\\_sprawdzajace](#)  
*Wskaźnik tab\_sprawdzajace.*

#### 4.1.1 Detailed Description

Klasa modeluje pojęcie algorytmu w skład którego wchodzi pola odpowiadające min. za ilość wczytanych elementów, powtórzenia algorytmów. Wśród metod klasy znajdują się min. wczytanie danych, pobranie czasu, zamiana elementów tablicy itd.

Definition at line 26 of file algorytm.h.

#### 4.1.2 Member Function Documentation

##### 4.1.2.1 int \* algorytm::dodaj\_element ( int \* *tablica*, int *element* )

Metoda dodaj element dodaje do zadanej tablicy jeden element o zadanej wartości na koniec tablicy przez co tablica zwiększa swój rozmiar o 1. Metoda zwraca wskaźnik na nową tablicę.

###### Parameters

in	<i>tablica</i>	- wskaźnik na tablicę do której zostanie dodany element.
in	<i>element</i>	- zmienna typu int która zostanie dodana do podanej tablicy.

###### Returns

Zwraca wskaźnik do nowej tablicy z dodanym elementem.

Definition at line 107 of file algorytm.cpp.

##### 4.1.2.2 int \* algorytm::dodaj\_elementy ( int \* *tab\_pierwsza*, int \* *tab\_druga* )

Metoda dodaje ze sobą dwie tablice tworząc jeden element. Jako argumenty metoda przyjmuje wskaźniki na tablice które mają być ze sobą dodane. Zwracany jest wskaźnik na nowo stworzoną tablicę złożoną z elementów obu tablic.

###### Parameters

in	<i>tab_pierwsza</i>	- tablica od której będzie rozpoczynała się nowo utworzona tablica.
in	<i>tab_druga</i>	- tablica która zostanie dopisana jako druga do nowo utworzonej tablicy.

###### Returns

Zwraca wskaźnik na nową tablicę.

Definition at line 117 of file algorytm.cpp.

##### 4.1.2.3 void algorytm::odwroc\_tablice ( int \* *tablica* )

Metoda odwroc tablicę wykonuje zamianę elementów tablicy w taki sposób iż pierwszy staje się ostatni, a ostatni pierwszym. Jako argument przyjmuje wskaźnik na tablicę którą chcemy odwrócić. Nie zwraca wartości.

## Parameters

<code>in</code>	<code>tablica</code>	- wskaźnik na tablice wartości int.
-----------------	----------------------	-------------------------------------

Definition at line 99 of file `algorytm.cpp`.

#### 4.1.2.4 `int * algorytm::operator= ( int * tab_pierwsza )`

Przeciążenie operatora przypisania pozwala na bezpośrednie przypisanie dwóch tablic ze sobą. Jako argument przyjmuje wskaźnik na tablicę, która zostanie przypisana. Zwraca natomiast wskaźnik na nową tablicę z przypisanymi elementami.

## Parameters

<code>in</code>	<code>tab_pierwsza</code>	- wskaźnik na tablicę, która będzie przypisana do drugiej
-----------------	---------------------------	---

## Returns

Zwraca wskaźnik na tę tablicę.

Definition at line 129 of file `algorytm.cpp`.

#### 4.1.2.5 `clock_t algorytm::pobierz_czas ( )`

Metoda służy do pobrania czasu i wykorzystywana jest do obliczenia czasu, w jakim wykonany został algorytm oraz każde jego powtórzenie. Zwraca aktualny czas.

## Returns

Zwraca zmienną typu `clock_t`.

Definition at line 44 of file `algorytm.cpp`.

#### 4.1.2.6 `bool algorytm::sprawdz ( )`

Metoda ta sprawdza poprawność wykonania operacji na tablicy danych poprzez porównanie każdego elementu `tab_obliczone` z odpowiadającym jej elementem `tab_sprawdzajace`. W przypadku poprawnego wykonania poprawnej operacji na `tab_danych` i porównaniu z `tab_sprawdzajace` metoda zwraca wartość `true`, gdy jakiś z elementów jest inny od odpowiednika, to zwracana jest wartość `false`.

## Returns

Zwraca wartość logiczną `true/false` w zależności od rezultatu.

Definition at line 74 of file `algorytm.cpp`.

#### 4.1.2.7 `int algorytm::test ( dane * Info )`

Metoda `test` jak nazwa wskazuje dokonuje poprawności działania założonego programu. Składa się z odpowiedniej sekwencji wywołania funkcji. Otwierane są tu pliki testowe i sprawdzające. Do obiektu `Info` są wysyłane odpowiednie informacje. Na koniec wywoływane są metody klasy `dane` dokonujące zapisu danych statystycznych do pliku `csv`. Jako argument przyjmuje wskaźnik do obiektu klasy `dane`. Zwraca wartość logiczną 0 po wykonaniu testu.

**Parameters**

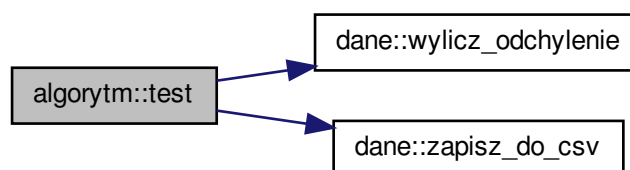
	<i>Info</i>	
<code>in</code>		- wskaźnik na obiekt klasy dane do którego zapisane będą wyniki przeprowadzonego testu.

**Returns**

Zwraca wartość logiczną typu `int` po wykonaniu testu.

Definition at line 138 of file `algorytm.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:

**4.1.2.8 int \* algorytm::wczytaj\_dane ( )**

Metoda ta wykonuje operacje wczytania pliku z danymi do tablicy `tab_danych` zawierającej dane `int`. rozmiar tablicy definiowany jest przez liczbę elementów określona przez pierwszą wartość wczytana z pliku. W przypadku niepowodzenia operacji otwarcia pliku użytkownik informowany jest o tym odpowiednim komunikatem. Metoda zwraca wskaźnik na tablicę z wczytanymi przez nią danymi.

**Returns**

Zwraca wskaźnik na `tab_danych`

Definition at line 26 of file `algorytm.cpp`.

**4.1.2.9 int \* algorytm::wczytaj\_dane\_sprawdzajace ( )**

Metoda ta wykonuje operacje wczytania pliku z danymi do tablicy `tab_sprawdzajace` zawierającej dane `int`. rozmiar tablicy definiowany jest przez liczbę elementów określona przez pierwszą wartość wczytana z pliku z danymi testowymi. W przypadku niepowodzenia operacji otwarcia pliku użytkownik informowany jest o tym odpowiednim komunikatem. Metoda zwraca wskaźnik na tablicę z wczytanymi przez nią danymi sprawdzającymi.

**Returns**

Zwraca wskaźnik na `tab_sprawdzajace`.

Definition at line 57 of file `algorytm.cpp`.

**4.1.2.10 int \* algorytm::wykonaj\_obliczenia ( )**

Metoda służy do wykonywania obliczeń na tablicy znajdującej się w polu klasy `Algorytm`. Funkcja zwraca wskaźnik na tablicę z danymi po wykonaniu operacji na nich.

**Returns**

Wskaźnik na `tab_obliczone`.

Definition at line 49 of file `algorytm.cpp`.

**4.1.2.11 void algorytm::zamien\_elementy ( int \* tablica, int pierwszy, int drugi )**

Metoda zamienia elementy dokonuje zamiany elementów o zadanych parametrach. Jako parametry metoda przyjmuje wskaźnik na tablicę na której chcemy wykonać operacje oraz dwa kolejne argumenty typu `int`, które podają numer elementu, które mają być ze sobą zamienione. Żadna wartość nie jest zwracana.

**Parameters**

<code>in</code>	<code>tablica</code>	- wskaźnik na tablicę, w której chcemy zamienić elementy.
<code>in</code>	<code>pierwszy</code>	- pierwszy z elementów tablicy, które chcemy zamienić.
<code>in</code>	<code>drugi</code>	- drugi z elementów tablicy, które chcemy ze sobą zamienić.

Definition at line 93 of file `algorytm.cpp`.

**4.1.3 Member Data Documentation****4.1.3.1 int algorytm::elementy**

Pole to odpowiedzialne jest za przechowywanie informacji mówiącej o ilości elementów, jaka zostanie wczytana do programu z pliku tekstowego.

Definition at line 42 of file `algorytm.h`.

**4.1.3.2 int algorytm::powtorzenia**

Pole to odpowiedzialne jest za przechowywanie informacji mówiącej o ilości powtórzeń algorytmu.

Definition at line 34 of file `algorytm.h`.

**4.1.3.3 int\* algorytm::tab\_danych**

Pole to odpowiedzialne jest za przechowywanie wskaźnika do alokowanej dynamicznie tablicy danych zawierającej elementy wczytane z pliku testowego.

Definition at line 51 of file `algorytm.h`.

**4.1.3.4 int\* algorytm::tab\_obliczone**

Pole to odpowiedzialne jest za przechowywanie wskaźnika do alokowanej dynamicznie tablicy danych zawierającej wartości elementów z tablicy danych przez określoną wartość.

Definition at line 60 of file `algorytm.h`.

#### 4.1.3.5 int\* algorytm::tab\_sprawdzajace

Pole to odpowiedzialne jest za przechowywanie wskaźnika do alokowanej dynamicznie tablicy danych zawierającej wartości elementów z tablicy danych przez określoną wartość.

Definition at line 69 of file algorytm.h.

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

- [algorytm.h](#)
- [algorytm.cpp](#)

## 4.2 dane Class Reference

Modeluje pojecie dane.

```
#include <dane.h>
```

### Public Member Functions

- [dane](#) (int [powtorzenia](#))  
*Konstruktor parametryczny.*
- void [wylicz\\_odchylenie](#) ()  
*Metoda wyliczajaca odchylenie standardowe.*
- void [zapisz\\_do\\_csv](#) ()  
*Metoda zapisujaca dane do pliku csv.*

### Public Attributes

- int [powtorzenia](#)  
*Pole powtorzenia.*
- clock\_t [czas\\_operacji](#)  
*Pole czas\_operacji.*
- double [odchylenie](#)  
*Pole odchylenie.*
- clock\_t \* [tab\\_czasow](#)  
*Pole \*tab\_czasow.*

#### 4.2.1 Detailed Description

Klasa modeluje pojecie dane. W tej klasie znajduja sie informacje dotyczace podsumowania pracy algorytmu. Polami tej klasy sa zmienne odpowiadajace za min. ilosc powtorzen algorytmu. Metody tej klasy wyliczaja odchylenie standardowe oraz zapisuja dane podsumowujace w pliku csv.

Definition at line 26 of file dane.h.

#### 4.2.2 Constructor & Destructor Documentation

##### 4.2.2.1 dane::dane ( int [powtorzenia](#) )

Konstruktor jako parametr przyjmuje wartosc powtorzen, aby zaalokowac pamiec potrzebna do przechowywania czasu kazdego powtorzenia algorytmu.

Definition at line 20 of file dane.cpp.



### 4.2.3 Member Function Documentation

#### 4.2.3.1 void dane::wylicz\_odchylenie ( )

Metoda ta wylicza wartosc odchylenia standardowego zestawu czasow zawartych w tablicy czasow `tab_czasow`. Metoda nie przyjmuje parametrow, ani nie zwraca zadnej wartosci.

Definition at line 23 of file `dane.cpp`.

Here is the caller graph for this function:

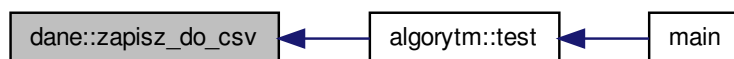


#### 4.2.3.2 void dane::zapisz\_do\_csv ( )

Metoda ta zapisuje informacje dotyczace wykonania sie algorytmu w pliku formatu csv. W tej metodzie tworzony jest odpowiedni obiekt klasy `fstream` do ktorego zapisywane sa informacje odnosnie ilosci powtorzen, czasu operacji, czasu powtorzen oraz odchylenie standardowe czasow.

Definition at line 37 of file `dane.cpp`.

Here is the caller graph for this function:



### 4.2.4 Member Data Documentation

#### 4.2.4.1 clock\_t dane::czas\_operacji

Pole to odpowiedzialne jest za przechowywanie informacji mowiacej w jakim czasie wykonano wszystkie powtorzenia algorytmu. Zmienna jest typu `clock_t`.

Definition at line 43 of file `dane.h`.

#### 4.2.4.2 double dane::odchylenie

Pole to odpowiedzialne jest za przechowywanie informacji mowiacej ile wynosi odchylenie standardowe z czasow wykonania algorytmu. Zmienna jest typu `double`.

Definition at line 51 of file `dane.h`.

#### 4.2.4.3 `int dane::powtorzenia`

Pole to odpowiedzialne jest za przechowywanie informacji mowiacej o ilosci powtorzen algorytmu. Zmienna jest typu `int`.

Definition at line 35 of file `dane.h`.

#### 4.2.4.4 `clock_t* dane::tab_czasow`

Pole to odpowiedzialne jest za przechowywanie czasow kazdego powtorzenia. Jest ono wskaznikiem na te tablice, która jest alokowana w sposob dynamiczny. Ilosc elementow okreslona jest przez ilosc powtorzen algorytmu.

Definition at line 60 of file `dane.h`.

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

- [dane.h](#)
- [dane.cpp](#)

## Chapter 5

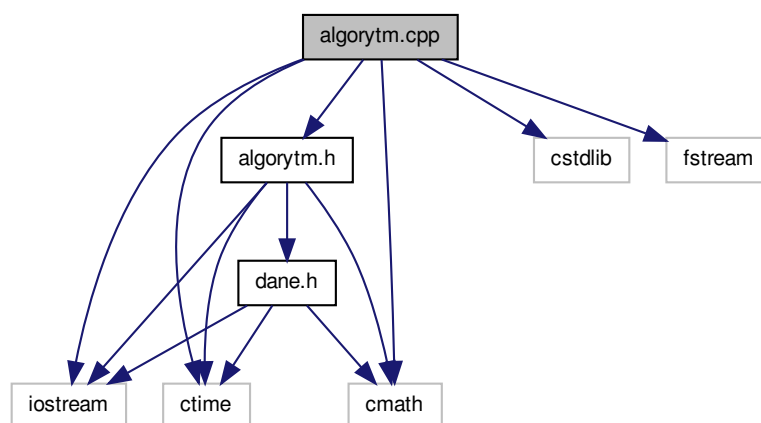
# File Documentation

### 5.1 algorytm.cpp File Reference

Plik z definicjami metod dla klasy algorytm.

```
#include <iostream>
#include <ctime>
#include <cstdlib>
#include <cmath>
#include <fstream>
#include "algorytm.h"
```

Include dependency graph for algorytm.cpp:



#### 5.1.1 Detailed Description

Plik zawiera definicje metod klasy `algorytm` oraz przeciążenie operatora przypisania. Zdefiniowane są tutaj metody odpowiadające za wczytanie pliku z danymi oraz pliku z danymi sprawdzającymi. Ponadto zdefiniowana są metody pobrania czasu, wykonania obliczeń algorytmu, sprawdzająca poprawność obliczeń oraz test algorytmu. W tym miejscu również znajdują się definicje operacji odwracania elementów tablicy, zamiany danych elementów, dodanie elementu do tablicy, dodanie dwóch tablic.

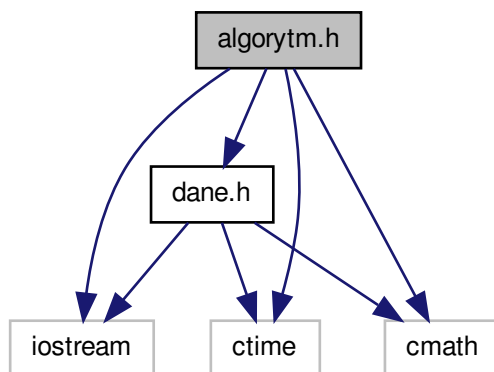
Definition in file [algorytm.cpp](#).

## 5.2 algorytm.h File Reference

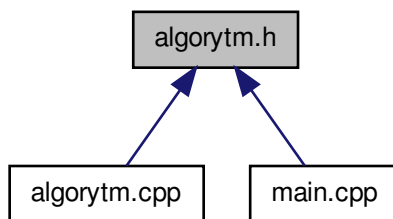
Definicja klasy `algorytm`.

```
#include <iostream>
#include <ctime>
#include <cmath>
#include "dane.h"
```

Include dependency graph for `algorytm.h`:



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



### Classes

- class `algorytm`  
*Modeluje pojecie algorytmu.*

### 5.2.1 Detailed Description

Plik zawiera definicje klasy `algorytm`, która wykonuje niezbędne operacje na plikach tekstowych.

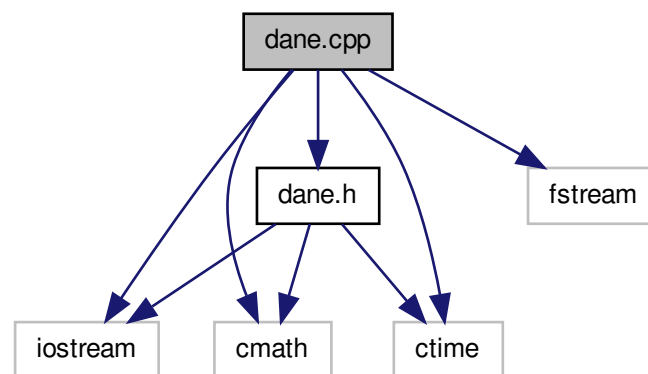
Definition in file [algorytm.h](#).

## 5.3 dane.cpp File Reference

Plik z definicjami metod dla klasy dane.

```
#include <iostream>
#include <fstream>
#include <cmath>
#include <ctime>
#include "dane.h"
```

Include dependency graph for dane.cpp:



### 5.3.1 Detailed Description

Plik zawiera definicje metod klasy dane tj. wyliczania wartosci odchylenia standardowego oraz zapisywania danych do pliku formatu csv. Ponadto zawiera definicje konstruktora parametrycznego w ktorym alokowana jest tablica do ktorej beda wpisywane czasy poszczegolnych powtorzen.

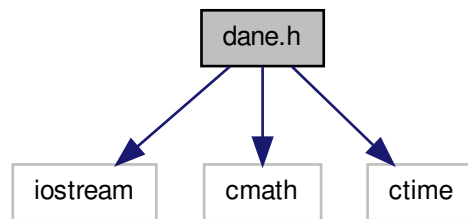
Definition in file [dane.cpp](#).

## 5.4 dane.h File Reference

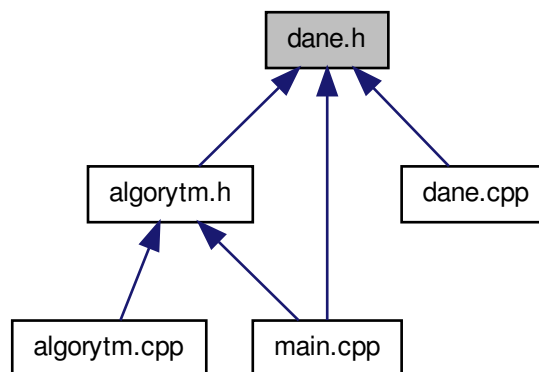
Definicja klasy dane.

```
#include <iostream>
#include <cmath>
#include <ctime>
```

Include dependency graph for dane.h:



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



## Classes

- class [dane](#)

*Modeluje pojecie dane.*

### 5.4.1 Detailed Description

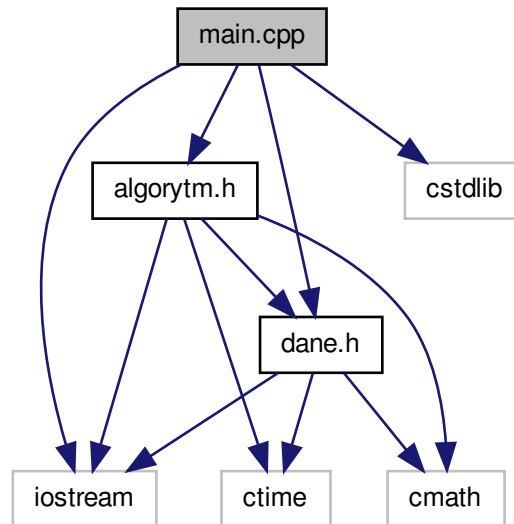
Plik zawiera definicje klasy dane w ktorej znajduja sie informacje dotyczace wykonania algorytmu z powtorzeniami.

Definition in file [dane.h](#).

## 5.5 main.cpp File Reference

Funkcja main.

```
#include <iostream>
#include <cstdlib>
#include "dane.h"
#include "algorytm.h"
Include dependency graph for main.cpp:
```



## Functions

- int [main](#) ()

### 5.5.1 Detailed Description

Plik zawiera glowna funkcje main w ktorej wywoływana jest metoda test obiektu algorytm. Ponadto w niej wczytywana jest do pola obiektu klasy algorytm liczba powtorzen wykonania algorytmu. Po wykonaniu wszystkich operacji funkcja zwraca zero.

#### Returns

Zwraca wartosc logiczna 0.

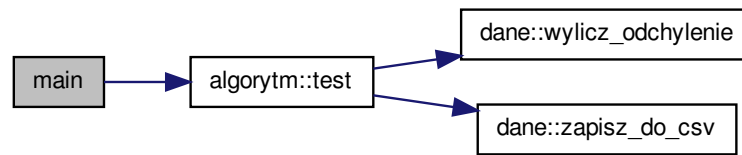
Definition in file [main.cpp](#).

### 5.5.2 Function Documentation

#### 5.5.2.1 int main ( )

Definition at line 20 of file main.cpp.

Here is the call graph for this function:



## 5.6 strona.dox File Reference



# Index

- algorytm, [7](#)
  - dodaj\_element, [8](#)
  - dodaj\_elementy, [8](#)
  - elementy, [11](#)
  - odwroc\_tablice, [8](#)
  - operator=, [9](#)
  - pobierz\_czas, [9](#)
  - powtorzenia, [11](#)
  - sprawdz, [9](#)
  - tab\_danych, [11](#)
  - tab\_obliczone, [11](#)
  - tab\_sprawdzajace, [11](#)
  - test, [9](#)
  - wczytaj\_dane, [10](#)
  - wczytaj\_dane\_sprawdzajace, [10](#)
  - wykonaj\_obliczenia, [11](#)
  - zamien\_elementy, [11](#)
- algorytm.cpp, [15](#)
- algorytm.h, [16](#)
- czas\_operacji
  - dane, [13](#)
- dane, [12](#)
  - czas\_operacji, [13](#)
  - dane, [12](#)
  - odchylenie, [13](#)
  - powtorzenia, [13](#)
  - tab\_czasow, [14](#)
  - wylicz\_odchylenie, [13](#)
  - zapisz\_do\_csv, [13](#)
- dane.cpp, [17](#)
- dane.h, [17](#)
- dodaj\_element
  - algorytm, [8](#)
- dodaj\_elementy
  - algorytm, [8](#)
- elementy
  - algorytm, [11](#)
- main
  - main.cpp, [19](#)
- main.cpp, [18](#)
- main, [19](#)
- odchylenie
  - dane, [13](#)
- odwroc\_tablice
  - algorytm, [8](#)
- operator=
  - algorytm, [9](#)
- pobierz\_czas
  - algorytm, [9](#)
- powtorzenia
  - algorytm, [11](#)
  - dane, [13](#)
- sprawdz
  - algorytm, [9](#)
- strona.dox, [20](#)
- tab\_czasow
  - dane, [14](#)
- tab\_danych
  - algorytm, [11](#)
- tab\_obliczone
  - algorytm, [11](#)
- tab\_sprawdzajace
  - algorytm, [11](#)
- test
  - algorytm, [9](#)
- wczytaj\_dane
  - algorytm, [10](#)
- wczytaj\_dane\_sprawdzajace
  - algorytm, [10](#)
- wykonaj\_obliczenia
  - algorytm, [11](#)
- wylicz\_odchylenie
  - dane, [13](#)
- zamien\_elementy
  - algorytm, [11](#)
- zapisz\_do\_csv
  - dane, [13](#)