PAMSI - P.Wilkosz

1.0

Wygenerowano przez Doxygen 1.7.6.1

Sun Mar 9 2014 20:38:42

Spis treści

Rozdział 1

Struktura katalogów

1.1 Katalogi

Ta struktura katalogów jest posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Rozdział 2

Indeks klas

2.1 Hierarchia klas

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

Rozdział 3

Indeks klas

3.1 Lista klas

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

Rozdział 4

Indeks plików

4.1 Lista plików

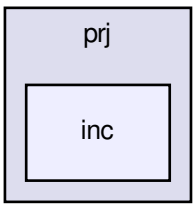
Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

Rozdział 5

Dokumentacja katalogów

5.1 Dokumentacja katalogu /home/pwilkosz/pamsi/lab1/prj/inc/

Directory dependency graph for /home/pwilkosz/pamsi/lab1/prj/inc/:



Pliki

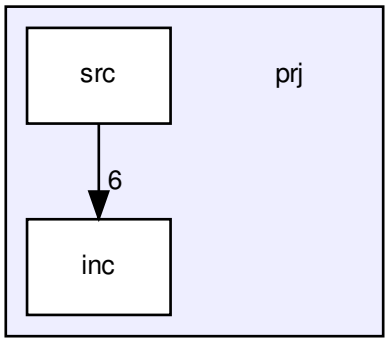
- plik algorytm.hh

  Definicja klas wykonujacych operacje na zestawie danych wejsciowych.
- plik operacje.hh
- plik statystyki.hh

  plik zawiera dekalracje funkcji odpowiedzialnych za przeprowadznaie statystyk

5.2 Dokumentacja katalogu /home/pwilkosz/pamsi/lab1/prj/

Directory dependency graph for /home/pwilkosz/pamsi/lab1/prj/:

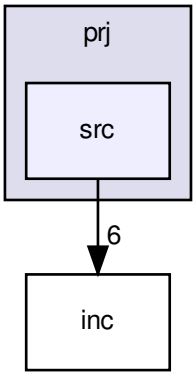


Katalogi

- katalog inc
- katalog src

5.3 Dokumentacja katalogu /home/pwilkosz/pamsi/lab1/prj/src/

Directory dependency graph for /home/pwilkosz/pamsi/lab1/prj/src/:



Pliki

- plik algorytm.cpp

  plik zawiera definicje metod klas zdefiniowanych w pliku algorytm.hh
- plik main.cpp

  plik glowny
- plik operacje.cpp
- plik statystyki.cpp

Rozdział 6

Dokumentacja klas

6.1 Dokumentacja klasy algorytm

Definicja klasy algorytm Jest to klasa bazowa, ktora ma za zadanie wczytac, przetworzyc i porownac dane z plikiem wzorcowym.

`#include <algorytm.hh>`

Diagram dziedziczenia dla algorytm

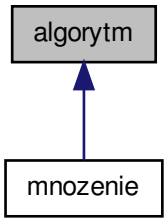
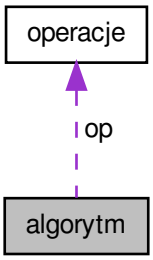


Diagram współpracy dla algorytm:



Metody publiczne

- algorytm (ifstream &plik1, ifstream &plik2, int N, int M)

  konstruktor kopiujacy - przekazuje informacje o nazwach plikow, ktore zapisywane sa do pol klasy
- void wykonaj ()

  funkcja dokonuje operacji na pliku wejsciowym, wywoluje metody odpowiedzialne za pomiar czasu oraz za porownanie wyniku operacji z plikiem wzorcowym
- bool wczytaj (ifstream &plik)

  Metoda wczytuje plik wejsciowy do tablicy `dane` *oraz do obiektu* `op` *klasy* `operacje`.
- bool wczytaj_wzor (ifstream &plik)

  Metoda wczytuje plik wzorcowy do tablicy `dane_wz`.
- virtual void przelicz ()

  Metoda odpowiada za przetworzenie danych wejsciowych zgodnie z zadanym algorytmem.
- bool porownaj ()

  porownuje przetworzony dane z danymi wzorcowymi
- int ile_danych ()
- float * jaki_czas ()
- float wlacz_zegar ()

  Metoda wlacza pomiar czasu poprzez wlaczenie funkcji `gettimeofday` *i przechowanie czasu w zmiennej* `start`.
- float wylacz_zegar ()

  Metoda wyacza pomiar czasu poprzez wlaczenie funkcji `gettimeofday` *i przechowanie czasu w zmiennej* `end`.
- void zapisz_do_csv ()

  Metoda zapisuje tablice `czas` *do pliku* `wyjscie.csv`.

Atrybuty publiczne

- float * czas

  zawiera wyniki dzialania algorytmu

Atrybuty chronione

- float * dane

  Tablica liczb wczytana z pliku.
- float * dane_wz

  tablica liczb zawartych w pliku wzorcowym
- int n

  ilosc danych w pliku
- int m

  ilosc powtorzen
- operacje op

  klasa zawierajaca tablice i metody do operacji na niej

6.1.1 Opis szczegółowy

Definicja klasy algorytm Jest to klasa bazowa, ktora ma za zadanie wczytac, przetworzyc i porownac dane z plikiem wzorcowym.

Definicja w linii 32 pliku algorytm.hh.

6.1.2 Dokumentacja konstruktora i destruktora

6.1.2.1 algorytm::algorytm (ifstream & *plik1*, ifstream & *plik2*, int *N*, int *M*) [inline]

konstruktor kopiujacy - przekazuje informacje o nazwach plikow, ktore zapisywane sa do pol klasy

Parametry

in	*plik1*	- plik wejsciowy
in	*plik2*	- plik wzorcowy
in	*N*	- ilosc danych wejsciowych
in	*M*	- ilosc powtorzen

Definicja w linii 74 pliku algorytm.hh.

6.1.3 Dokumentacja funkcji składowych

6.1.3.1 int algorytm::ile_danych ( )

Zwraca

ilosc liczb wejsciowych

Definicja w linii 31 pliku algorytm.cpp.

6.1.3.2 float * algorytm::jaki_czas ( )

Zwraca

tablica `czas` z danymi pomiarowymi czasu wykonywania algorytmu

Definicja w linii 34 pliku algorytm.cpp.

Oto graf wywoływań tej funkcji:

6.1.3.3 bool algorytm::porownaj ( )

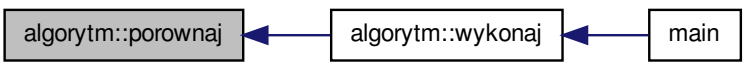
porownuje przetworzony dane z danymi wzorcowymi

Zwraca

true - gdy pliki zgodne false - w przeciwnym przypadku

Definicja w linii 68 pliku algorytm.cpp.

Oto graf wywoływań tej funkcji:



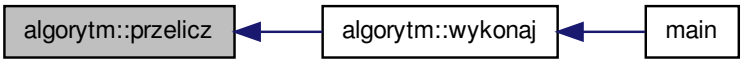
6.1.3.4 void algorytm::przelicz ( ) [virtual]

Metoda odpowiada za przetworzenie danych wejsciowych zgodnie z zadanym algorytmem.

Reimplementowana w mnozenie.

Definicja w linii 9 pliku algorytm.cpp.

Oto graf wywoływań tej funkcji:



6.1.3.5 bool algorytm::wczytaj (ifstream & *plik*)

Metoda wczytuje plik wejsciowy do tablicy `dane` oraz do obiektu `op` klasy `operacje`.

Parametry

`in`	*plik*	- strumien pliku wejsciowego

Definicja w linii 10 pliku algorytm.cpp.

6.1.3.6 bool algorytm::wczytaj_wzor (ifstream & *plik*)

Metoda wczytuje plik wzorcowy do tablicy `dane_wz`.

Parametry

`in`	*plik*	- strumien pliku wejsciowego

Definicja w linii 22 pliku algorytm.cpp.

6.1.3.7 float algorytm::wlacz_zegar ( )

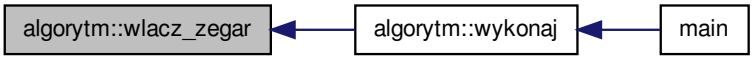
Metoda wlacza pomiar czasu poprzez wlaczenie funkcji `gettimeofday` i przechowanie czasu w zmiennej `start`.

Zwraca

start - zmienna pamietajaca czas poprzedzajacy wykonanie algorytmu

Definicja w linii 38 pliku algorytm.cpp.

Oto graf wywoływań tej funkcji:

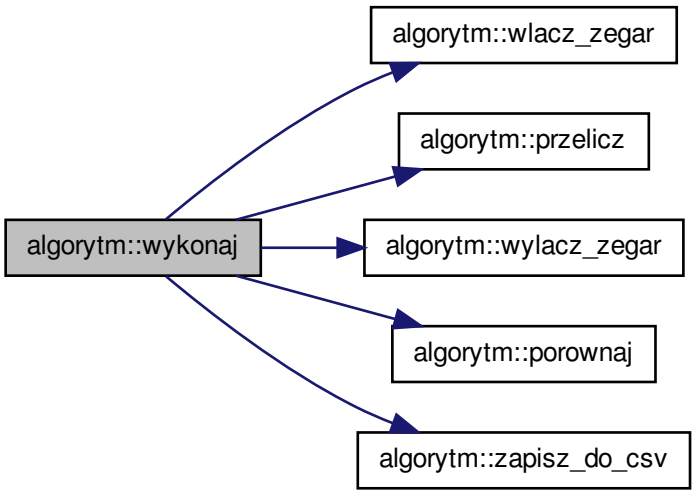


6.1.3.8 void algorytm::wykonaj ( )

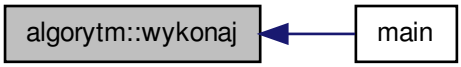
funkcja dokonuje operacji na pliku wejsciowym, wywoluje metody odpowiedzialne za pomiar czasu oraz za porownanie wyniku operacji z plikiem wzorcowym

Definicja w linii 53 pliku algorytm.cpp.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



6.1.3.9 float algorytm::wylacz_zegar ( )

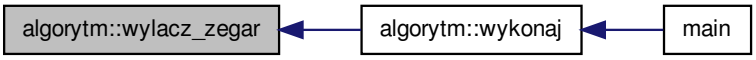
Metoda wyacza pomiar czasu poprzez wlaczenie funkcji `gettimeofday` i przechowanie czasu w zmiennej `end`.

Zwraca

end - zmienna pamietajaca czas poprzedzajacy wykonanie algorytmu

Definicja w linii 46 pliku algorytm.cpp.

Oto graf wywoływań tej funkcji:

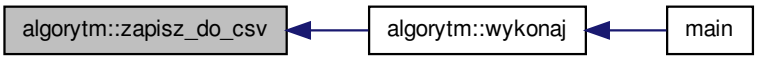


6.1.3.10 void algorytm::zapisz_do_csv ( )

Metoda zapisuje tablice `czas` do pliku `wyjscie.csv`.

Definicja w linii 73 pliku algorytm.cpp.

Oto graf wywoływań tej funkcji:



6.1.4 Dokumentacja atrybutów składowych

6.1.4.1 float* algorytm::czas

zawiera wyniki dzialania algorytmu

Definicja w linii 64 pliku algorytm.hh.

6.1.4.2 float* algorytm::dane [protected]

Tablica liczb wczytana z pliku.

Definicja w linii 40 pliku algorytm.hh.

6.1.4.3 float* algorytm::dane_wz [protected]

tablica liczb zawartych w pliku wzorcowym

Definicja w linii 45 pliku algorytm.hh.

6.1.4.4 int algorytm::m [protected]

ilosc powtorzen

Definicja w linii 55 pliku algorytm.hh.

6.1.4.5 int algorytm::n [protected]

ilosc danych w pliku

Definicja w linii 51 pliku algorytm.hh.

6.1.4.6 operacje algorytm::op [protected]

klasa zawierajaca tablice i metody do operacji na niej

Definicja w linii 59 pliku algorytm.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

- algorytm.hh
- algorytm.cpp

6.2 Dokumentacja klasy mnozenie

modeluje algorytm dokonujacy mnozenia kazdego elementu pliku wejsciowego przez 2

`#include <algorytm.hh>`

Diagram dziedziczenia dla mnozenie

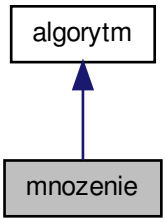
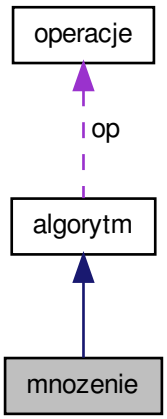


Diagram współpracy dla mnozenie:



Metody publiczne

- mnozenie (ifstream &plik1, ifstream &plik2, int N, int M)
- void przelicz ()

  wykonuje zalozony algorytm mnozenia elementow tablicy przez 2

6.2.1 Opis szczegółowy

modeluje algorytm dokonujacy mnozenia kazdego elementu pliku wejsciowego przez 2

Definicja w linii 125 pliku algorytm.hh.

6.2.2 Dokumentacja konstruktora i destruktora

6.2.2.1 mnozenie::mnozenie (ifstream & *plik1,* ifstream & *plik2,* int *N,* int *M*) [inline]

/brief konstruktor przekazuje do pol klasy informacje o nazwach pliku wejsciowego i wzorcowego

Parametry

in	*plik1*	- plik wejsciowy
in	*plik2*	- plik wzorcowy
in	*N*	- ilosc danych wejsciowych
in	*M*	- ilosc powtorzen

Definicja w linii 134 pliku algorytm.hh.

6.2.3 Dokumentacja funkcji składowych

6.2.3.1 void mnozenie::przelicz ( ) [virtual]

wykonuje zalozony algorytm mnozenia elementow tablicy przez 2

Reimplementowana z algorytm.

Definicja w linii 81 pliku algorytm.cpp.

Dokumentacja dla tej klasy została wygenerowana z plików:

- algorytm.hh
- algorytm.cpp

6.3 Dokumentacja klasy operacje

Klasa modeluje tablice z danymi i metody sluzace do operacji na niej.

`#include <operacje.hh>`

Metody publiczne

- operacje ()

  konstruktor bezparametryczny
- operacje (int N)

  konstruktor parametryczny - alokuje pamiec w dynamicznej tablicy `tab`
- bool zamien_elementy (int i, int j)

  Metoda zamienia 2 elementy tablicy.
- void odwroc_tablice ()

  metoda odwraca wszystkie elementy tablicy
- void dodaj_element (float e)

  metoda dodaje element do tablicy, alokujac dodatkowa pamiec
- void dodaj_elementy (float *tab2, int rozm)

  metoda dodaje elementy do tablicy
- void operator= (float *tab1)

  Przeciazenie operatora przypisania; przypisuje elementy tablicy `tab1` *do tablicy bedacej polem klasy.*

- bool operator== (float *tab1)

  Przeciazenie operatora porownania; metoda porownuje zawartosci dwoch tablic.

Atrybuty publiczne

- int n

  ilosc elementow w tablicy
- float * tab

  tablica z liczbami

6.3.1 Opis szczegółowy

Klasa modeluje tablice z danymi i metody sluzace do operacji na niej.

Definicja w linii 9 pliku operacje.hh.

6.3.2 Dokumentacja konstruktora i destruktora

6.3.2.1 operacje::operacje ( )

konstruktor bezparametryczny

6.3.2.2 operacje::operacje (int *N*) [inline]

konstruktor parametryczny - alokuje pamiec w dynamicznej tablicy `tab`

Parametry

in	*N*	- ilosc elementow w tablicy; parametr przypisywany do pola `n` w klasie, oraz alokuje pamiec o takim wlasnie rozmiarze

Definicja w linii 26 pliku operacje.hh.

6.3.3 Dokumentacja funkcji składowych

6.3.3.1 void operacje::dodaj_element (float *e*)

metoda dodaje element do tablicy, alokujac dodatkowa pamiec

Parametry

in	*e*	- element, ktory nalezy dolaczyc do tablicy

Definicja w linii 27 pliku operacje.cpp.

6.3.3.2 void operacje::dodaj_elementy (float * *tab2*, int *rozm*)

metoda dodaje elementy do tablicy

Parametry

in	*tab2*	- tablica, ktora nalezy dolaczyc
in	*rozm*	- rozmiar tablicy tab2

Definicja w linii 46 pliku operacje.cpp.

6.3.3.3 void operacje::odwroc_tablice ( )

metoda odwraca wszystkie elementy tablicy

Definicja w linii 12 pliku operacje.cpp.

6.3.3.4 void operacje::operator= (float * *tab1*)

Przeciazenie operatora przypisania; przypisuje elementy tablicy `tab1 do` tablicy bedacej polem klasy.

Parametry

in	*tab1*	- tablica, ktorej zawartosc przypisujemy

Definicja w linii 63 pliku operacje.cpp.

6.3.3.5 bool operacje::operator== (float * *tab1*)

Przeciazenie operatora porownania; metoda porownuje zawartosci dwoch tablic.

Parametry

in	*tab1*	- tablica, ktorej wartosci porownujemy

Zwraca

true - gdy zawartsoc tablic jest identyczna false - w przeciwnym przypadku

Definicja w linii 69 pliku operacje.cpp.

6.3.3.6 bool operacje::zamien_elementy (int *i*, int *j*)

Metoda zamienia 2 elementy tablicy.

Parametry

in	*i*	- element tablicy
in	*j*	- element tablicy

Zwraca

true - gdy elementy nie wykraczaja poza zakres tablicy false - w przeciwnym przypadku

Definicja w linii 3 pliku operacje.cpp.

6.3.4 Dokumentacja atrybutów składowych

6.3.4.1 int operacje::n

ilosc elementow w tablicy

Definicja w linii 14 pliku operacje.hh.

6.3.4.2 float* operacje::tab

tablica z liczbami

Definicja w linii 17 pliku operacje.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

- operacje.hh
- operacje.cpp

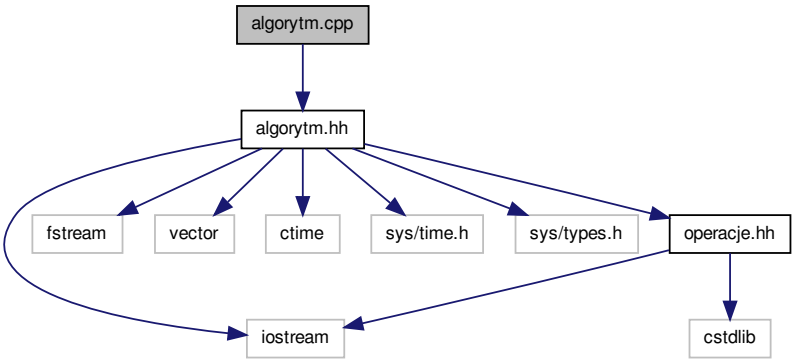
Rozdział 7

Dokumentacja plików

7.1 Dokumentacja pliku algorytm.cpp

plik zawiera definicje metod klas zdefiniowanych w pliku algorytm.hh

`#include "algorytm.hh"` Wykres zależności załączania dla algorytm.cpp:



7.1.1 Opis szczegółowy

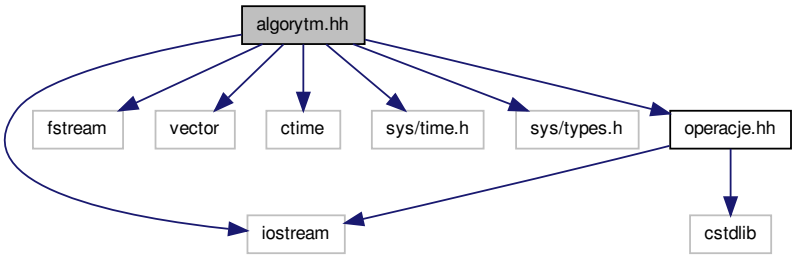
plik zawiera definicje metod klas zdefiniowanych w pliku algorytm.hh

Definicja w pliku algorytm.cpp.

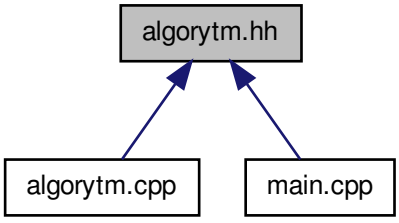
7.2 Dokumentacja pliku algorytm.hh

Definicja klas wykonujacych operacje na zestawie danych wejsciowych.

`#include <iostream>` `#include <fstream>` `#include <vector>` `#include <ctime>` `#include <sys/time.h>` `#include <sys/types.h>` `#include "operacje.hh"` Wykres zależności załączania dla algorytm.hh:



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



Komponenty

- class algorytm

  Definicja klasy algorytm Jest to klasa bazowa, ktora ma za zadanie wczytac, przetworzyc i porownac dane z plikiem wzorcowym.

- class mnozenie

  modeluje algorytm dokonujacy mnozenia kazdego elementu pliku wejsciowego przez 2

7.2.1 Opis szczegółowy

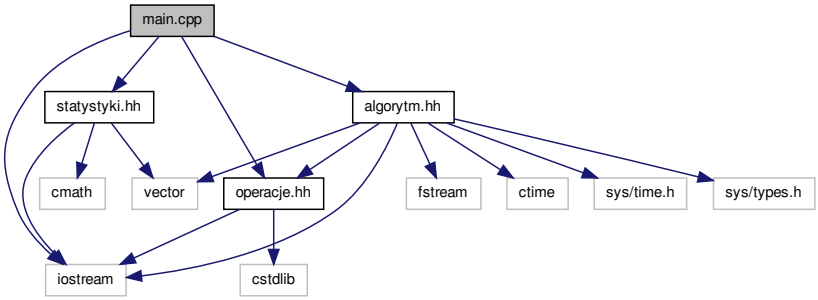
Definicja klas wykonujacych operacje na zestawie danych wejsciowych.

Definicja w pliku algorytm.hh.

7.3 Dokumentacja pliku main.cpp

plik glowny

`#include <iostream>` `#include "algorytm.hh"` `#include "statystyki.-hh"` `#include "operacje.hh"` Wykres zależności załączania dla main.cpp:



Funkcje

- int main ()

7.3.1 Opis szczegółowy

plik glowny

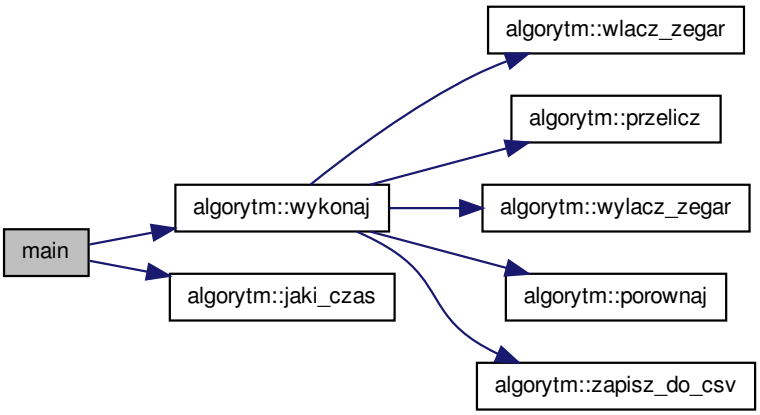
Definicja w pliku main.cpp.

7.3.2 Dokumentacja funkcji

7.3.2.1 int main ( )

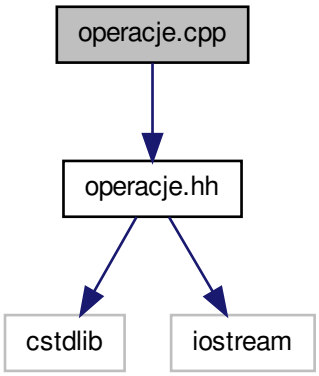
Definicja w linii 11 pliku main.cpp.

Oto graf wywołań dla tej funkcji:



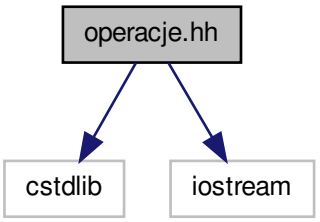
7.4 Dokumentacja pliku operacje.cpp

`#include "operacje.hh"` Wykres zależności załączania dla operacje.cpp:

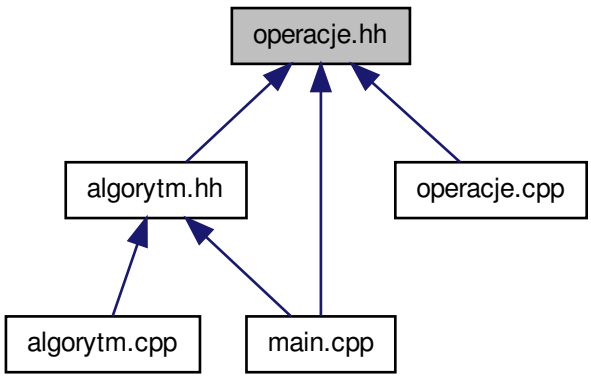


7.5 Dokumentacja pliku operacje.hh

`#include <cstdlib>` `#include <iostream>` Wykres zależności załączania dla operacje.hh:



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



Komponenty

- class operacje

  Klasa modeluje tablice z danymi i metody sluzace do operacji na niej.

Definicje

- #define ROZMIAR 9

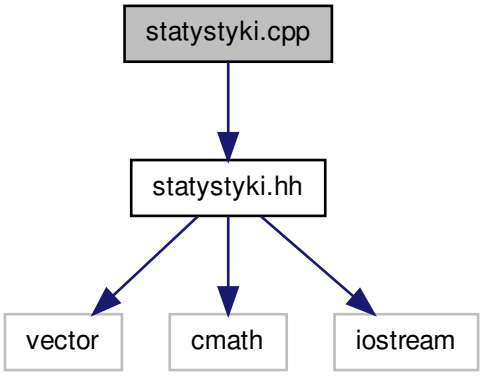
7.5.1 Dokumentacja definicji

7.5.1.1 #define ROZMIAR 9

Definicja w linii 3 pliku operacje.hh.

7.6 Dokumentacja pliku statystyki.cpp

`#include "statystyki.hh"` Wykres zależności załączania dla statystyki.cpp:



Funkcje

- float srednia (vector< float > &tab)

  funckja oblicza wartosc srednia
- float odchylenie_standardowe (float srednia, vector< float > &tab)

  funckja oblicza odchylenie standardowe

7.6.1 Dokumentacja funkcji

7.6.1.1 float odchylenie_standardowe (float *srednia,* vector< float > & *tab*)

funckja oblicza odchylenie standardowe

Parametry

tab	- kontener zawierajacy czasy wykonania algorytmu
srednia	- wartosc srednia

Zwraca

odchylenie standardowe

Definicja w linii 16 pliku statystyki.cpp.

7.6.1.2 float srednia (vector< float > & *tab*)

funckja oblicza wartosc srednia

Parametry

tab	- kontener zawierajacy czasy wykonania algorytmu

Zwraca

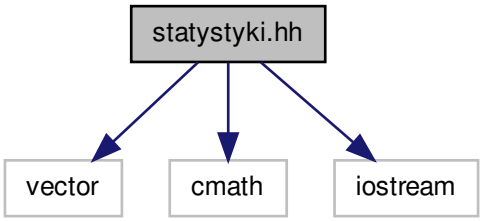
wartosc srednia

Definicja w linii 3 pliku statystyki.cpp.

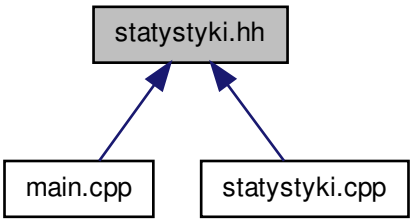
7.7 Dokumentacja pliku statystyki.hh

plik zawiera dekalracje funkcji odpowiedzialnych za przeprowadznaie statystyk

```
#include <vector> #include <cmath> #include <iostream>
```
Wykres zależności załączania dla statystyki.hh:



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



Funkcje

- float srednia (vector< float > &tab)

  funckja oblicza wartosc srednia
- float odchylenie_standardowe (float srednia, vector< float > &tab)

  funckja oblicza odchylenie standardowe

7.7.1 Opis szczegółowy

plik zawiera dekalracje funkcji odpowiedzialnych za przeprowadznaie statystyk

Definicja w pliku statystyki.hh.

7.7.2 Dokumentacja funkcji

7.7.2.1 float odchylenie_standardowe (float *srednia,* vector< float > & *tab*)

funckja oblicza odchylenie standardowe

Parametry

tab	- kontener zawierajacy czasy wykonania algorytmu
srednia	- wartosc srednia

Zwraca

odchylenie standardowe

Definicja w linii 16 pliku statystyki.cpp.

7.7.2.2 float srednia (vector< float > & *tab*)

funckja oblicza wartosc srednia

Parametry

tab	- kontener zawierajacy czasy wykonania algorytmu

Zwraca

wartosc srednia

Definicja w linii 3 pliku statystyki.cpp.

7.8 Dokumentacja pliku strona.dox