

Projekt_znajomi
1.01

Generated by Doxygen 1.8.6

Wed Jun 4 2014 20:21:48

Contents

1	Wyszukiwanie znajomości	1
1.1	Najważniejsze cechy	1
1.2	Opis algorytmu	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	Graf Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	Graf	8
4.1.2.2	~Graf	8
4.1.3	Member Function Documentation	8
4.1.3.1	czy_bylo	8
4.1.3.2	czy_poprawne	9
4.1.3.3	czy_spojny	9
4.1.3.4	dijkstry	9
4.1.3.5	domyslny_plik	10
4.1.3.6	dowolny_plik	10
4.1.3.7	generuj_liste	11
4.1.3.8	losowe_dijkstry	11
4.1.3.9	przerob_z_fb	12
4.1.3.10	przypisz_indeks	12
4.1.3.11	stworz_liste_z_pliku	12
4.1.3.12	stworz_wierzcholki	12
4.1.3.13	testowy_plik	12
4.1.3.14	wyswietl	13
4.1.4	Member Data Documentation	13

4.1.4.1	E	13
4.1.4.2	lista_sasiadujaca	13
4.1.4.3	V	13
4.1.4.4	wierzcholki	13
4.2	por Struct Reference	14
4.2.1	Detailed Description	14
4.2.2	Member Function Documentation	14
4.2.2.1	operator()	14
4.3	Wezel Struct Reference	14
4.3.1	Detailed Description	15
4.3.2	Member Data Documentation	15
4.3.2.1	g	15
4.3.2.2	nr	15
4.4	Wierzcholek Class Reference	15
4.4.1	Detailed Description	15
4.4.2	Constructor & Destructor Documentation	16
4.4.2.1	Wierzcholek	16
4.4.3	Member Function Documentation	16
4.4.3.1	operator=	16
4.4.4	Member Data Documentation	16
4.4.4.1	sasiad	16
4.4.4.2	waga	16
5	File Documentation	17
5.1	/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/doc/pages/strona.dox File Reference	17
5.2	/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/inc/Dijkstry.hh File Reference	17
5.2.1	Detailed Description	18
5.3	/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/src/Dijkstry.cpp File Reference	18
5.3.1	Detailed Description	19
5.4	/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/src/main.cpp File Reference	19
5.4.1	Detailed Description	19
5.4.2	Function Documentation	20
5.4.2.1	main	20
Index		21

Chapter 1

Wyszukiwanie znajomości

Author

Paweł Żurek

Date

04.06.2014

Version

1.01

Program umożliwia znalezienie połączenia między zadanymi osobami.

1.1 Najważniejsze cechy

Program potrafi przerobić plik z odpowiednio uporządkowanymi danymi. W pliku tym muszą być zawarte obok siebie dwa numeru wierzchołków. Świadczy to o ich połączeniu. Program na podstawie tego tworzy wierzchołki i połączenia między nimi. W aktualnej wersji programu wierzchołkami są zwykłe liczby. Związane jest to z trudnością zdobycia prawdziwych imion i nazwisk. Pojedyncza liczba całkowita odpowiada jednej osobie. W programie jest to pojedynczy wierzchołek.

1.2 Opis algorytmu

Kiedy dane zostaną przerobione, program tworzy graf z wierzchołkami i połączeniami między nimi, każde o wadze tej samej i równej 1. Zakładam, że połączenie między każdą osobą ma tę samą wagę. Następnie program przeprowadza algorytm szukania ścieżki między dwoma podanymi osobami (wierzchołkami). Jako wyjście jest spis (wierzchołków) osób, które trzeba poznać aby poznać chcianą osobę.

Chapter 2

Class Index

2.1 Class List

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

Graf	Modeluje pojecie graf. Klasa sluzzy glownie do wykonania algorytmu wyszukiwania, czyli znalezienia polaczenia miedzy dwoma punktami	7
por	Struktura porownywania Struktura ta ma na celu ulatwienie dzialania algorytmu wyszukiwania (Dijkstry) Prownuje ona wartosci drogi miedzy dwoma wezlami	14
Wezel	Struktura Wezla Struktura ta ma zdefiniowane dwie zmienne nr oraz g, ktore odpowiadaja za przechowywanie numer wierzchołka oraz droge jaka juz przebyl od poczatku dzialania wyszukiwania	14
Wierzcholek	Definicje dla klasy Wierzcholek Struktura ta ma zdefiniowane dwie zmienne sasiad oraz waga. Przy wczytywaniu pliku dodajac wierzcholek dodajemy odrazu informacje o aktualnym sasiedzie oraz o wadze polaczenia miedzy wierzchołkiem i sasiadem	15

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/inc/ Dijkstry.hh	
Definicje funkcji dla klasy graf	17
/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/src/ Dijkstry.cpp	
Plik zawiera funkcje z klasy graf	18
/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/src/ main.cpp	
Plik zawiera funkcje main()	19

Chapter 4

Class Documentation

4.1 Graf Class Reference

Modeluje pojecie graf. Klasa sluzi glownie do wykonania algorytmu wyszukiwania, czyli znalezienia polaczenia miedzy dwoma punktami.

```
#include <Dijkstry.hh>
```

Public Member Functions

- [Graf](#) ()
Konstruktor klasy [Graf](#).
- [~Graf](#) ()
Destruktor klasy [Graf](#).
- void [generuj_liste](#) (int ilosc, int gestosc)
Funkcja generujaca [Graf](#).
- void [stworz_liste_z_pliku](#) (string nazwapliku)
Funkcja tworzaca [Graf](#) z pliku.
- void [stworz_wierzcholki](#) (string nazwapliku)
Funkcja tworzaca wierzcholki [Graf](#) u z pliku.
- void [przerob_z_fb](#) (string typ, string wyjscie)
Funkcja przerabiajaca plik z bazy .
- void [dowolny_plik](#) ()
Funkcja pomocnicza do main .
- void [domyslny_plik](#) ()
Funkcja pomocnicza do main .
- void [testowy_plik](#) ()
Funkcja pomocnicza do main .
- void [losowe_dijkstry](#) ()
Funkcja pomocnicza do main .
- bool [czy_spojny](#) ()
Funkcja pomocnicza.
- bool [czy_poprawne](#) (int a, int b)
Funkcja pomocnicza.
- bool [czy_bylo](#) (int v1, int v2)
Funkcja pomocnicza.
- int [przypisz_indeks](#) (int id)
Funkcja pomocnicza.

- int [dijkstra](#) (int a, int b)
Funkcja glowna programu wyszukujaca polaczenie.
- void [wyswietl](#) ()
Funkcja wyswietlajaca liste sasiedztwa grafu.

Public Attributes

- int [V](#)
Pole typu int, bedzie uzywane do przechowywania informacji o ilosci wierzchołkow.
- int [E](#)
Pole typu int, bedzie uzywane do przechowywania informacji o ilosci krawedzi (aktualnie nie uzywane w programie).
- vector< [Wierzcholek](#) > * [lista_sasiadujaca](#)
*Pole typu * vector, bedzie uzywane do przechowywania informacji o sasiadach dla poszczegolnych wierzchołkow.*
- int * [wierzcholki](#)
*Pole typu * int, bedzie uzywane jako wektor z prawdziwymi nazwami (id) wierzchołkow.*

4.1.1 Detailed Description

Modeluje pojecie graf. Klasa sluzi glownie do wykonania algorytmu wyszukiwania, czyli znalezienia polaczenia miedzy dwoma punktami.

Definition at line 76 of file Dijkstra.hh.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 [Graf::Graf](#) ()

Konstruktor klasy [Graf](#).

Konstruktor jest bezparametryczny. Jedyne jego zadanie to zainicjalizowanie zmiennych wewnetrznych klasy [Graf](#)

Definition at line 12 of file Dijkstra.cpp.

4.1.2.2 [Graf::~~Graf](#) ()

Destruktor klasy [Graf](#).

Destruktor jest bezparametryczny. Jedyne jego zadanie to usuniecie obiektow zaalokowanych dynamicznie

Definition at line 15 of file Dijkstra.cpp.

4.1.3 Member Function Documentation

4.1.3.1 [bool Graf::czy_bylo](#) (int v1, int v2)

Funkcja pomocnicza.

Funkcja sprawdza czy dane wierzcholki juz sa w bazie aby nie wpisywac ich ponownie

Parameters

v1	-> pole typu int z id drugiego wierzcholka
--------------------	--

<code>v2</code>	-> pole typu <code>int</code> z id pierwszego wierzchołka
-----------------	---

Returns

1 -> gdy wierzchołki są w bazie 0 -> gdy wierzchołków nie ma w bazie

Definition at line 127 of file `Dijkstry.cpp`.

4.1.3.2 bool Graf::czy_poprawne (int a, int b)

Funkcja pomocnicza.

Funkcja sprawdza wierzchołki znajdujące się w wektorze z wierzchołkami.

Parameters

<code>a</code>	-> pole typu <code>int</code> z id drugiego wierzchołka
<code>b</code>	-> pole typu <code>int</code> z id pierwszego wierzchołka

Returns

1 -> gdy wierzchołki są w bazie 0 -> gdy wierzchołków nie ma w bazie

Definition at line 175 of file `Dijkstry.cpp`.

4.1.3.3 bool Graf::czy_spojny ()

Funkcja pomocnicza.

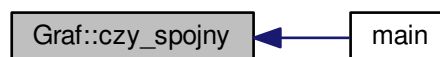
Funkcja sprawdza czy graf jest spójny.

Returns

1 -> gdy graf spójny 0 -> gdy graf nie spójny

Definition at line 290 of file `Dijkstry.cpp`.

Here is the caller graph for this function:

**4.1.3.4 int Graf::dijkstra (int a, int b)**

Funkcja główna programu wyszukująca połączenie.

Funkcja wyszukuje połączenie między podanymi wierzchołkami

Parameters

<i>a</i>	-> pole typu int z id wierzcholka z ktorego szukamy polaczenia
<i>b</i>	-> pole typu int z id wierzcholka do ktorego szukamy polaczenia

Returns

1 -> gdy wystapi blad 0 -> gdy bezblednie znajdzie polaczenie

Definition at line 194 of file Dijkstra.cpp.

Here is the caller graph for this function:



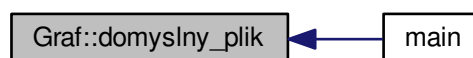
4.1.3.5 void Graf::domyslny_plik ()

Funkcja pomocnicza do main .

Funkcja jest tak naprawde tylko po to aby plik [main.cpp](#) byl bardziej przejrzysty Bierze domyslny plik (na chwile obecna jest to chyba plik fb.txt) i przerabia go, tworzy wierzcholki z niego i na koncu generuje polaczenia

Definition at line 36 of file Dijkstra.cpp.

Here is the caller graph for this function:



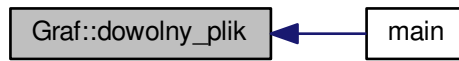
4.1.3.6 void Graf::dowolny_plik ()

Funkcja pomocnicza do main .

Funkcja jest tak naprawde tylko po to aby plik [main.cpp](#) byl bardziej przejrzysty Pyta o nazwe pliku, przerabia go, tworzy wierzcholki z niego i na koncu generuje polaczenia

Definition at line 22 of file Dijkstra.cpp.

Here is the caller graph for this function:



4.1.3.7 void Graf::generuj_liste (int *ilosc*, int *gestosc*)

Funkcja generująca [Graf](#).

Funkcja typu void, na podstawie parametrów tworzy graf o zadanej ilości :

Parameters

<i>ilosc</i>	-> wartosc typu int przechowuje informacje o ilości wierzchołków
<i>gestosc</i>	-> wartosc typu int przechowuje informacje o gęstości połączeń w grafie

Definition at line 337 of file Dijkstry.cpp.

Here is the caller graph for this function:



4.1.3.8 void Graf::losowe_dijkstry ()

Funkcja pomocnicza do main .

Funkcja jest tak naprawdę tylko po to aby plik [main.cpp](#) był bardziej przejrzysty. Losuje wierzchołki między którymi algorytm będzie szukał połączenia i następnie uruchamia go z tymi wartościami.

Definition at line 59 of file Dijkstry.cpp.

Here is the caller graph for this function:



4.1.3.9 void Graf::przerob_z_fb (string *typ*, string *wyjscie*)

Funkcja przerabiajaca plik z bazy .

Funkcja typu void, tworząca wierzchołki z pliku, gdzie sa podane tylko wierzchołki

Parameters

<i>typ</i>	-> wartosc typu string przechowuje nazwe pliku wejscowego, ktorego bedziemy obrabiac
<i>wyjscie</i>	-> wartosc typu string przechowuje nazwe pliku przerobionego (nowo stworzonego)

Definition at line 69 of file Dijkstry.cpp.

4.1.3.10 int Graf::przypisz_indeks (int *id*)

Funkcja pomocnicza.

Funkcja sluzy do przypisania indeksu wierzchołkowi po podanym id

Parameters

<i>id</i>	-> pole typu int z id wierzchołka
-----------	-----------------------------------

Returns

indeks -> mniejszy musi byc niz ilosc wierzchołkow

Definition at line 138 of file Dijkstry.cpp.

4.1.3.11 void Graf::stworz_liste_z_pliku (string *nazwapliku*)

Funkcja tworząca [Graf](#) z pliku.

Funkcja typu void, tworząca graf z pliku, gdzie sa podane polaczenia miedzy wierzchołkami (tylko)

Parameters

<i>nazwapliku</i>	-> wartosc typu string przechowuje nazwe pliku
-------------------	--

Definition at line 95 of file Dijkstry.cpp.

4.1.3.12 void Graf::stworz_wierzcholki (string *nazwapliku*)

Funkcja tworząca wierzchołki [Graf](#) u z pliku.

Funkcja typu void, tworząca wierzchołki z pliku, gdzie sa podane tylko wierzchołki

Parameters

<i>nazwapliku</i>	-> wartosc typu string przechowuje nazwe pliku
-------------------	--

Definition at line 150 of file Dijkstry.cpp.

4.1.3.13 void Graf::testowy_plik ()

Funkcja pomocnicza do main .

Funkcja jest tak naprawde tylko po to aby plik [main.cpp](#) byl bardziej przejrzysty Bierze plik testowy (test.txt) przerabia go, tworzy wierzchołki z niego i na koncu generuje polaczenia

Definition at line 47 of file Dijkstry.cpp.

Here is the caller graph for this function:



4.1.3.14 void Graf::wyswietl ()

Funkcja wyswietlajaca liste sasiedztwa grafu.

Funkcja typu void wyswietla aktualny stan grafu.

Definition at line 163 of file Dijkstry.cpp.

Here is the caller graph for this function:



4.1.4 Member Data Documentation

4.1.4.1 int Graf::E

Pole typu int, bedzie uzywane do przechowywania informacji o ilosci krawedzi (aktualnie nie uzywane w programie).

Definition at line 85 of file Dijkstry.hh.

4.1.4.2 vector<Wierzcholek>* Graf::lista_sasiadujaca

Pole typu * vector, bedzie uzywane do przechowywania informacji o sasiadach dla poszczegolnych wierzchoolkow.

Definition at line 89 of file Dijkstry.hh.

4.1.4.3 int Graf::V

Pole typu int, bedzie uzywane do przechowywania informacji o ilosci wierzchoolkow.

Definition at line 81 of file Dijkstry.hh.

4.1.4.4 int* Graf::wierzcholki

Pole typu * int, bedzie uzywane jako wektor z prawdziwymi nazwami (id) wierzchoolkow.

Definition at line 93 of file Dijkstry.hh.

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

- [/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/inc/Dijkstry.hh](#)
- [/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/src/Dijkstry.cpp](#)

4.2 por Struct Reference

Struktura porownywania Struktura ta ma na celu ulatwienie dzialania algorytmu wyszukiwania (Dijkstry) Prownuje ona wartosci drogi miedzy dwoma wezlami.

```
#include <Dijkstry.hh>
```

Public Member Functions

- bool [operator\(\)](#) (const [Wezel](#) &k1, const [Wezel](#) &k2)

4.2.1 Detailed Description

Struktura porownywania Struktura ta ma na celu ulatwienie dzialania algorytmu wyszukiwania (Dijkstry) Prownuje ona wartosci drogi miedzy dwoma wezlami.

Definition at line 244 of file Dijkstry.hh.

4.2.2 Member Function Documentation

4.2.2.1 bool [por::operator\(\)](#) (const [Wezel](#) & k1, const [Wezel](#) & k2) `[inline]`

Definition at line 245 of file Dijkstry.hh.

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

- [/home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/inc/Dijkstry.hh](#)

4.3 Wezel Struct Reference

Struktura Wezla Struktura ta ma zdefiniowane dwie zmienne nr oraz g, ktore odpowiadaja za przechowywanie numer wierzcholkana oraz droge jaka juz przebyl od poczatku dzialania wyszukiwania.

```
#include <Dijkstry.hh>
```

Public Attributes

- int [nr](#)
Pole typu int, bedzie uzywane do przechowywania id wierzcholka.
- int [g](#)
Pole typu int, bedzie uzywane do przechowywania informacji o przebytej drodze.

4.3.1 Detailed Description

Struktura Wezla Struktura ta ma zdefiniowane dwie zmienne nr oraz g, ktore odpowiadaja za przechowywanie numer wierzcholka oraz droge jaka juz przebyl od poczatku dzialania wyszukiwania.

Definition at line 60 of file Dijkstry.hh.

4.3.2 Member Data Documentation

4.3.2.1 int Wezel::g

Pole typu int, bedzie uzywane do przechowywania informacji o przebytej drodze.

Definition at line 68 of file Dijkstry.hh.

4.3.2.2 int Wezel::nr

Pole typu int, bedzie uzywane do przechowywania id wierzcholka.

Definition at line 64 of file Dijkstry.hh.

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

- /home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/inc/Dijkstry.hh

4.4 Wierzcholek Class Reference

Definicje dla klasy [Wierzcholek](#) Struktura ta ma zdefiniowane dwie zmienne sasiad oraz waga. Przy wczytywaniu pliku dodajac wierzcholek dodajemy odrazu informacje o aktualnym sasiedzie oraz o wadze polaczenia miedzy wierzcholkiem i sasiadem.

```
#include <Dijkstry.hh>
```

Public Member Functions

- [Wierzcholek](#) ()
Konstruktor klasy [Wierzcholek](#).
- [Wierzcholek](#) & operator= ([Wierzcholek](#) const &c1)
Funkcja przeciazajaca operator '='.

Public Attributes

- int [sasiad](#)
Pole typu int, bedzie uzywane do numeru id sasiada dla danego wierzcholka.
- int [waga](#)
Pole typu int, bedzie uzywane do przechowywania informacji o wadze polaczenia.

4.4.1 Detailed Description

Definicje dla klasy [Wierzcholek](#) Struktura ta ma zdefiniowane dwie zmienne sasiad oraz waga. Przy wczytywaniu pliku dodajac wierzcholek dodajemy odrazu informacje o aktualnym sasiedzie oraz o wadze polaczenia miedzy wierzcholkiem i sasiadem.

Definition at line 29 of file Dijkstry.hh.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Wierzcholek::Wierzcholek () [inline]

Konstruktor klasy [Wierzcholek](#).

Konstruktor jest bezparametryczny. Wykonuje wyzerowanie wszystkich składowych wewnętrznych klasy

Definition at line 46 of file Dijkstry.hh.

4.4.3 Member Function Documentation

4.4.3.1 Wierzcholek & Wierzcholek::operator= (Wierzcholek const & c1)

Funkcja przeciażająca operator '='.

Definition at line 281 of file Dijkstry.cpp.

4.4.4 Member Data Documentation

4.4.4.1 int Wierzcholek::sasiad

Pole typu int, będzie używane do numeru id sąsiada dla danego wierzchołka.

Definition at line 34 of file Dijkstry.hh.

4.4.4.2 int Wierzcholek::waga

Pole typu int, będzie używane do przechowywania informacji o wadze połączenia.

Definition at line 38 of file Dijkstry.hh.

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

- /home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/inc/Dijkstry.hh
- /home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/src/Dijkstry.cpp

Chapter 5

File Documentation

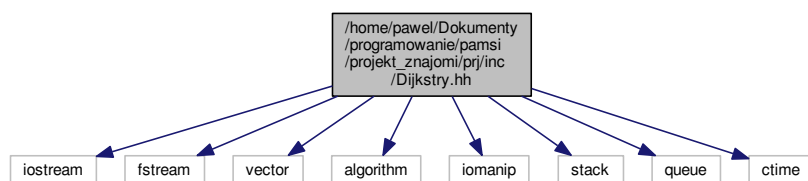
5.1 /home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/doc/pages/strona.dox File Reference

5.2 /home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/inc/Dijkstry.hh File Reference

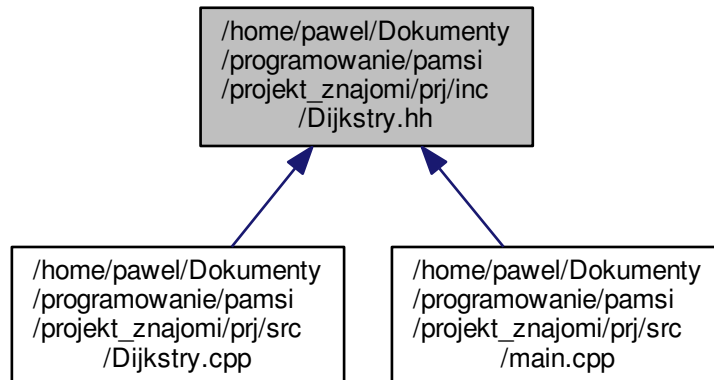
Definicje funkcji dla klasy graf.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <stack>
#include <queue>
#include <ctime>
```

Include dependency graph for Dijkstry.hh:



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



Classes

- class [Wierzcholek](#)

Definicje dla klasy [Wierzcholek](#) Struktura ta ma zdefiniowane dwie zmienne sasiad oraz waga. Przy wczytywaniu pliku dodajac wierzcholek dodajemy odrazu informacje o aktualnym sasiedzie oraz o wadze polaczenia miedzy wierzchołkiem i sasiadem.

- struct [Wezel](#)

Struktura Wezla Struktura ta ma zdefiniowane dwie zmienne nr oraz g, ktore odpowiadaja za przechowywanie numer wierzchołkana oraz droge jaka juz przebyl od poczatku dzialania wyszukiwania.

- class [Graf](#)

Modeluje pojecie graf. Klasa sluzzy glownie do wykonania algorytmu wyszukiwania, czyli znalezienia polaczenia miedzy dwoma punktami.

- struct [por](#)

Struktura porownywania Struktura ta ma na celu ulatwienie dzialania algorytmu wyszukiwania (Dijkstry) Prownuje ona wartosci drogi miedzy dwoma wezlami.

5.2.1 Detailed Description

Definicje funkcji dla klasy graf.

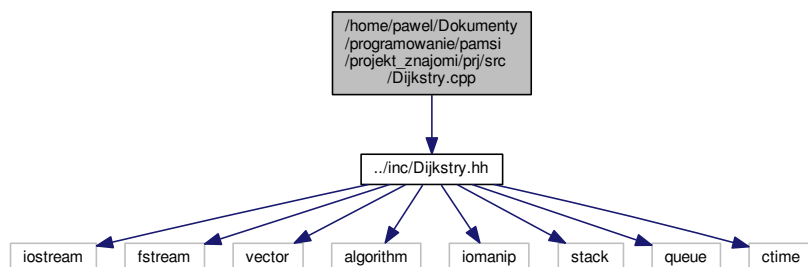
Definition in file [Dijkstry.hh](#).

5.3 /home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/src/Dijkstry.cpp File Reference

Plik zawiera funkcje z klasy graf.

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

Include dependency graph for Dijkstry.cpp:



5.3.1 Detailed Description

Plik zawiera funkcje z klasy graf.

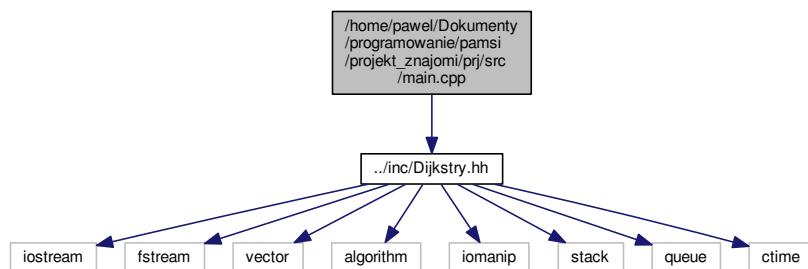
Definition in file [Dijkstry.cpp](#).

5.4 /home/pawel/Dokumenty/programowanie/pamsi/projekt_znajomi/prj/src/main.cpp File Reference

Plik zawiera funkcje [main\(\)](#)

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

Include dependency graph for main.cpp:



Functions

- int [main](#) ()

5.4.1 Detailed Description

Plik zawiera funkcje [main\(\)](#)

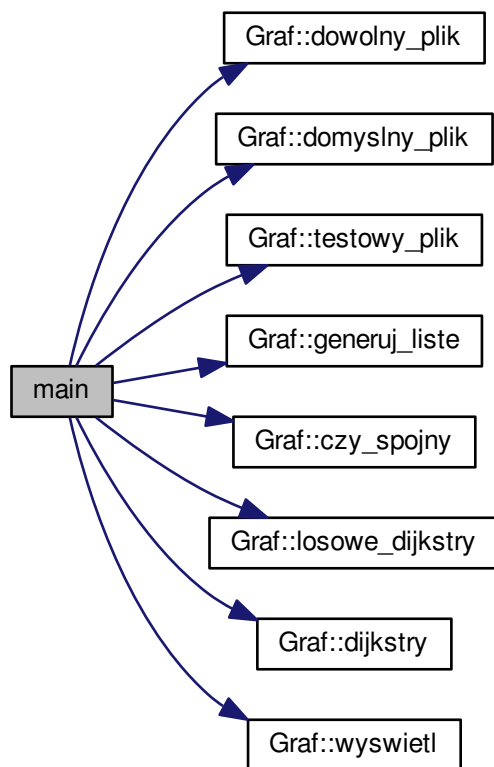
Definition in file [main.cpp](#).

5.4.2 Function Documentation

5.4.2.1 `int main ()`

Definition at line 11 of file main.cpp.

Here is the call graph for this function:



Index

- ~Graf
 - Graf, 8
 - /home/pawel/Dokumenty/programowanie/pamsi/projekt-_znajomi/prj/doc/pages/strona.dox, 17
 - /home/pawel/Dokumenty/programowanie/pamsi/projekt-_znajomi/prj/inc/Dijkstry.hh, 17
 - /home/pawel/Dokumenty/programowanie/pamsi/projekt-_znajomi/prj/src/Dijkstry.cpp, 18
 - /home/pawel/Dokumenty/programowanie/pamsi/projekt-_znajomi/prj/src/main.cpp, 19
- czy_bylo
 - Graf, 8
- czy_poprawne
 - Graf, 9
- czy_spojny
 - Graf, 9
- dijkstry
 - Graf, 9
- domyslny_plik
 - Graf, 10
- dowolny_plik
 - Graf, 10
- E
 - Graf, 13
- g
 - Wezel, 15
- generuj_liste
 - Graf, 11
- Graf, 7
 - ~Graf, 8
 - czy_bylo, 8
 - czy_poprawne, 9
 - czy_spojny, 9
 - dijkstry, 9
 - domyslny_plik, 10
 - dowolny_plik, 10
 - E, 13
 - generuj_liste, 11
 - Graf, 8
 - lista_sasiadujaca, 13
 - losowe_dijkstry, 11
 - przerob_z_fb, 11
 - przypisz_indeks, 12
 - stworz_liste_z_pliku, 12
 - stworz_wierzcholki, 12
 - testowy_plik, 12
 - V, 13
 - wierzcholki, 13
 - wyswietl, 13
- lista_sasiadujaca
 - Graf, 13
- losowe_dijkstry
 - Graf, 11
- main
 - main.cpp, 20
- main.cpp
 - main, 20
- nr
 - Wezel, 15
- operator()
 - por, 14
- operator=
 - Wierzcholek, 16
- por, 14
 - operator(), 14
- przerob_z_fb
 - Graf, 11
- przypisz_indeks
 - Graf, 12
- sasiad
 - Wierzcholek, 16
- stworz_liste_z_pliku
 - Graf, 12
- stworz_wierzcholki
 - Graf, 12
- testowy_plik
 - Graf, 12
- V
 - Graf, 13
- waga
 - Wierzcholek, 16
- Wezel, 14
 - g, 15
 - nr, 15
- Wierzcholek, 15
 - operator=, 16
 - sasiad, 16
 - waga, 16

Wierzcholek, [16](#)
wierzchołki
Graf, [13](#)
wyswietl
Graf, [13](#)