

Wyszukiwanie cykli w grafie

Wygenerowano przez Doxygen 1.8.13

Spis treści

1	Indeks klas	1
1.1	Lista klas	1
2	Indeks plików	3
2.1	Lista plików	3
3	Dokumentacja klas	5
3.1	Dokumentacja struktury ojciec	5
3.1.1	Opis szczegółowy	5
3.1.2	Dokumentacja atrybutów składowych	5
3.1.2.1	dane	6
3.1.2.2	nastepny	6
3.1.2.3	odwiedzony	6
3.1.2.4	synowie	6
3.2	Dokumentacja struktury wierzcholek	6
3.2.1	Opis szczegółowy	7
3.2.2	Dokumentacja atrybutów składowych	7
3.2.2.1	dane	7
3.2.2.2	nastepny	7

4 Dokumentacja plików	9
4.1 Dokumentacja pliku funkcje.cpp	9
4.1.1 Dokumentacja funkcji	10
4.1.1.1 DFS()	10
4.1.1.2 dodaj_dane()	10
4.1.1.3 dodaj_do_listy()	10
4.1.1.4 dodaj_ojca()	11
4.1.1.5 dodaj_syna()	11
4.1.1.6 usun() [1/2]	11
4.1.1.7 usun() [2/2]	12
4.1.1.8 wczytaj_graf()	12
4.1.1.9 wyczyszc_odwiedzone()	12
4.1.1.10 wypisz_graf()	13
4.1.1.11 wyszukaj_ojca()	13
4.2 Dokumentacja pliku main.cpp	13
Indeks	15

Rozdział 1

Indeks klas

1.1 Lista klas

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

ojciec	5
wierzcholek	6

Rozdział 2

Indeks plików

2.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

funkcje.cpp	9
funkcje.h	??
main.cpp	13
struktury.h	??

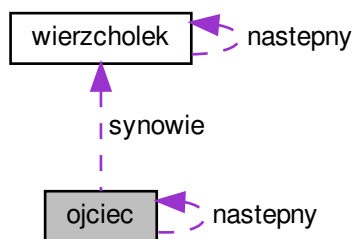
Rozdział 3

Dokumentacja klas

3.1 Dokumentacja struktury ojciec

```
#include <struktury.h>
```

Diagram współpracy dla ojciec:



Atrybuty publiczne

- int `dane`
- bool `odwiedzony`
- `wierzcholek` * `synowie`
- `ojciec` * `nastepny`

3.1.1 Opis szczegółowy

Struktura reprezentująca graf. Jest to lista list wierzchołków (synów) wychodzących z danego ojca

3.1.2 Dokumentacja atrybutów składowych

3.1.2.1 dane

```
int ojciec::dane
```

wierzchołek

3.1.2.2 następny

```
ojciec* ojciec::nastepny
```

wskaźnik na następnego ojca

3.1.2.3 odwiedzony

```
bool ojciec::odwiedzony
```

czy został odwiedzony

3.1.2.4 synowie

```
wierzcholek* ojciec::synowie
```

wskaźnik na listę synów

Dokumentacja dla tej struktury została wygenerowana z pliku:

- struktury.h

3.2 Dokumentacja struktury wierzcholek

```
#include <struktury.h>
```

Diagram współpracy dla wierzcholek:



Atrybuty publiczne

- int dane
- wierzcholek * nastepny

3.2.1 Opis szczegółowy

Lista trzymająca kolejne wierzchołki wychodzące ze wspólnego ojca.

3.2.2 Dokumentacja atrybutów składowych

3.2.2.1 dane

```
int wierzcholek::dane
```

wierzchołek

3.2.2.2 następny

```
wierzcholek* wierzcholek::nastepny
```

wskaźnik do następnego wierzchołka

Dokumentacja dla tej struktury została wygenerowana z pliku:

- struktury.h

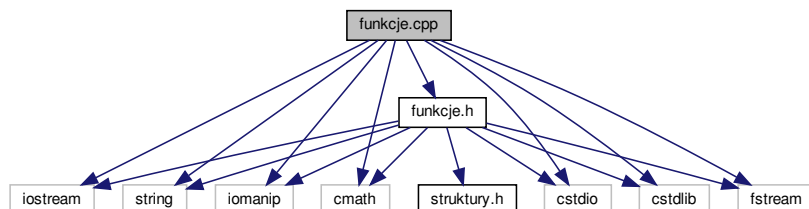
Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku funkcje.cpp

```
#include <iostream>
#include <string>
#include <iomanip>
#include <cmath>
#include <cstdio>
#include <cstdlib>
#include <fstream>
#include "funkcje.h"
```

Wykres zależności załączania dla funkcje.cpp:



Funkcje

- `ojciec * wyszukaj_ojca (ojciec *grafHead, int ojciec_dane)`
- `void dodaj_ojca (ojciec *&grafHead, int ojciec_dane)`
- `void dodaj_syna (ojciec *&ojciecSyna, int synDane)`
- `void dodaj_dane (int ojciecDane, int synDane, ojciec *&grafHead)`
- `void wczytaj_graf (std::string plik, ojciec *&start_grafu)`
- `void wypisz_graf (ojciec *grafHead)`
- `void dodaj_do_listy (ojciec *&listaHead, wierzcholek *&cykl, int iterator)`
- `void wypisz_stos_od_tylu (wierzcholek *stos, std::ofstream *plikWy)`
- `void dodaj_do_stosu (wierzcholek *&stos, int wartosc)`
- `void zdejmij_ze_stosu (wierzcholek *&stos)`
- `void DFS (ojciec *&grafHead, int poczatek, wierzcholek *&lista, int pierwszy, std::ofstream *plikWy, int &counter)`
- `void wyczyszc_odwiedzone (ojciec *&grafHead)`
- `void usun (ojciec *&grafHead)`
- `void usun (wierzcholek *&grafHead)`

4.1.1 Dokumentacja funkcji

4.1.1.1 DFS()

```
void DFS (
    ojciec *& grafHead,
    int poczatek,
    wierzcholek *& lista,
    int pierwszy,
    std::ofstream * plikWy,
    int & counter )
```

Funkcja będąca implementacją algorytmu DFS na listach powieszanych

Parametry

<i>grafHead</i>	graf, którym przechodzimy
<i>poczatek</i>	początkowy wierzchołek do przechodzenia grafu
<i>lista</i>	pomocnicza lista do trzymania stosu wierzchołków
<i>pierwszy</i>	początkowy wierzchołek, do którego ścieżki szukamy
<i>plikWy</i>	plik wyjściowy
<i>counter</i>	licznik cykli

4.1.1.2 dodaj_dane()

```
void dodaj_dane (
    int ojciecDane,
    int synDane,
    ojciec *& grafHead )
```

Funkcja dodająca syna do listy ojca

Parametry

<i>ojciecDane</i>	wartość ojca
<i>synDane</i>	wartość syna
<i>grafHead</i>	adres grafu

4.1.1.3 dodaj_do_listy()

```
void dodaj_do_listy (
    ojciec *& listaHead,
```

```
wierzcholek *& cykl,  
int iterator )
```

Funkcja dodająca listę z cyklem do listy głównej, zawierającej wszystkie cykle

Parametry

<i>listHead</i>	lista, do której dodajemy cykle
-----------------	---------------------------------

4.1.1.4 dodaj_ojca()

```
void dodaj_ojca (  
    ojciec *& grafHead,  
    int ojciecDane )
```

Funkcja dodająca ojca do grafu

Parametry

<i>grafHead</i>	graf, w którym dodajemy ojca
<i>ojciecDane</i>	wartość ojca, którą dodajemy do grafu

4.1.1.5 dodaj_syna()

```
void dodaj_syna (  
    ojciec *& ojciecSyna,  
    int synDane )
```

Funkcja dodająca syna do list adiacencji ojca

Parametry

<i>ojciecSyna</i>	adres ojca, do listy którego dodajemy syna
<i>synDane</i>	wartość syna, którego dodajemy do list ojca

4.1.1.6 usun() [1/2]

```
void usun (  
    ojciec *& grafHead )
```

Funkcja usuwająca graf

Parametry

<i>grafHead</i>	graf do usunięcia
-----------------	-------------------

4.1.1.7 `usun()` [2/2]

```
void usun (
    wierzcholek *& grafHead )
```

Funkcja usuwająca listę wierzchołków

Parametry

<i>grafHead</i>	lista do usunięcia
-----------------	--------------------

4.1.1.8 `wczytaj_graf()`

```
void wczytaj_graf (
    std::string plik,
    ojciec *& start_grafu )
```

Funkcja wczytująca graf z podanego pliku do pamięci programu.

Parametry

<i>plik</i>	nazwa pliku, z którego należy wczytać graf
<i>graf</i>	wskaźnik na listę, w której trzymanym będzie graf

4.1.1.9 `wyczyszc_odwiedzone()`

```
void wyczyszc_odwiedzone (
    ojciec *& grafHead )
```

Funkcja zerująca znacznik odwiedzony w grafie

Parametry

<i>grafHead</i>	graf
-----------------	------

4.1.1.10 wypisz_graf()

```
void wypisz_graf (
    ojciec * grafHead )
```

Funkcja wypisująca tablicę adiacencji (debugging)

Parametry

<i>grafHead</i>	graf do wypisania
-----------------	-------------------

4.1.1.11 wyszukaj_ojca()

```
ojciec* wyszukaj_ojca (
    ojciec * grafHead,
    int ojciecDane )
```

Funkcja wyszukująca adres ojca listy do której należy dodać kolejny wierzchołek.

Parametry

<i>grafHead</i>	graf w którym szukamy ojca
<i>ojciecDane</i>	wartość ojca, którego szukamy

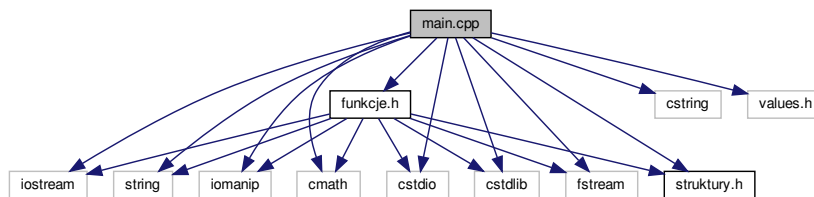
Zwraca

adres ojca, którego szukano

4.2 Dokumentacja pliku main.cpp

```
#include <iostream>
#include <string>
#include <iomanip>
#include <cmath>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <values.h>
#include <fstream>
#include "funkcje.h"
#include "struktury.h"
```

Wykres zależności załączania dla main.cpp:



Funkcje

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

Skorowidz

DFS
 funkcje.cpp, [10](#)

dane
 ojciec, [5](#)
 wierzcholek, [7](#)

dodaj_dane
 funkcje.cpp, [10](#)

dodaj_do_listy
 funkcje.cpp, [10](#)

dodaj_ojca
 funkcje.cpp, [11](#)

dodaj_syna
 funkcje.cpp, [11](#)

funkcje.cpp, [9](#)
 DFS, [10](#)
 dodaj_dane, [10](#)
 dodaj_do_listy, [10](#)
 dodaj_ojca, [11](#)
 dodaj_syna, [11](#)
 usun, [11](#), [12](#)
 wczytaj_graf, [12](#)
 wyczysc_odwiedzone, [12](#)
 wypisz_graf, [12](#)
 wyszukaj_ojca, [13](#)

main.cpp, [13](#)

nastepny
 ojciec, [6](#)
 wierzcholek, [7](#)

odwiedzony
 ojciec, [6](#)

ojciec, [5](#)
 dane, [5](#)
 nastepny, [6](#)
 odwiedzony, [6](#)
 synowie, [6](#)

synowie
 ojciec, [6](#)

usun
 funkcje.cpp, [11](#), [12](#)

wczytaj_graf
 funkcje.cpp, [12](#)

wierzcholek, [6](#)
 dane, [7](#)
 nastepny, [7](#)

wyczysc_odwiedzone
 funkcje.cpp, [12](#)

wypisz_graf
 funkcje.cpp, [12](#)

wyszukaj_ojca
 funkcje.cpp, [13](#)