

Kolorowy graf

1

Generated by Doxygen 1.9.6

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/Kolorowy graf/main.cpp	
File Reference	3
2.1.1 Function Documentation	3
2.1.1.1 main()	3
2.2 C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/Kolorowy graf/model.cpp	
File Reference	3
2.2.1 Function Documentation	4
2.2.1.1 czyMoznaPokolorowac()	4
2.2.1.2 czySasiedztwoPokolorowane()	5
2.2.1.3 obliczWykorzystaneKolory()	5
2.2.1.4 pokolorujGraf()	5
2.2.1.5 pokolorujSasiedztwo()	6
2.2.1.6 wczytajDaneZPliku()	6
2.2.1.7 wczytajParametry()	6
2.2.1.8 weryfikujPoprawnoscParametrow()	7
2.2.1.9 wyczyszcKolory()	7
2.2.1.10 zapiszWynikDoPliku()	8
2.3 C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/Kolorowy graf/model.h	
File Reference	8
2.3.1 Function Documentation	9
2.3.1.1 czyMoznaPokolorowac()	9
2.3.1.2 czySasiedztwoPokolorowane()	9
2.3.1.3 obliczWykorzystaneKolory()	10
2.3.1.4 pokolorujGraf()	10
2.3.1.5 pokolorujSasiedztwo()	10
2.3.1.6 wczytajDaneZPliku()	11
2.3.1.7 wczytajParametry()	11
2.3.1.8 weryfikujPoprawnoscParametrow()	11
2.3.1.9 wyczyszcKolory()	12
2.3.1.10 zapiszWynikDoPliku()	12
2.4 model.h	12
Index	15

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/Kolorowy graf/	main.cpp	. .	3
C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/Kolorowy graf/	model.cpp	.	3
C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/Kolorowy graf/	model.h	. .	8

Chapter 2

File Documentation

2.1 C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/main.cpp File Reference

```
#include <iostream>
#include "model.h"
```

Functions

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

2.1.1 Function Documentation

2.1.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

2.2 C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/model.cpp File Reference

```
#include "model.h"
#include <iostream>
#include <sstream>
#include <set>
```

Functions

- bool [weryfikujPoprawnoscParametrow](#) (int argc, char **argv)
Weryfikacja poprawności parametrów wejściowych aplikacji.
- void [wczytajParametry](#) (int argc, char **argv, std::string &plikWejscowy, std::string &plikWynikowy)
Wczytanie parametrów wywołania do zmiennych programu.
- std::vector< std::vector< int > > [wczytajDaneZPliku](#) (std::string nazwa)
Wczytywanie danych z podanego jako parametr pliku.
- void [pokolorujGraf](#) (int wierzcholek, std::vector< std::vector< int > > graf, std::vector< int > &kolory)
Główna funkcja odpowiedzialna za uruchomienie procesu kolorowania grafu.
- void [pokolorujSasiedztwo](#) (std::vector< std::vector< int > > graf, std::vector< int > sasiedztwo, std::vector< int > &kolory)
Funkcja rekurencyjna odpowiedzialna za kolorowanie grafu przechodząc po wszystkich sąsiadach wierzchołka. Funkcja wykonuje się do momentu aż wszystkie wierzchołki zostaną pokolorowane.
- bool [czyMoznaPokolorowac](#) (int wierzcholek, std::vector< std::vector< int > > graf, int kolor, std::vector< int > &kolory)
Funkcja sprawdzająca czy można pokolorować wskazany jako parametr wierzchołek, podanym jako parametr kolorem.
- bool [czySasiedztwoPokolorowane](#) (std::vector< int > sasiedztwo, std::vector< int > kolory)
Funkcja sprawdzająca czy całe sąsiedztwo danego wierzchołka ma już przydzielone swoje kolory.
- int [obliczWykorzystaneKolory](#) (std::vector< int > kolory)
Liczenie ilości wykorzystanych kolorów w grafie.
- void [wyczyszcKolory](#) (std::vector< int > kolory)
Czyszczenie tablicy kolorów.
- void [zapiszWynikDoPliku](#) (std::string nazwa, std::vector< std::vector< int > > graf, std::vector< int > kolory)
Zapisywanie wyników do pliku wyjściowego podanego jako parametr.

2.2.1 Function Documentation

2.2.1.1 czyMoznaPokolorowac()

```
bool czyMoznaPokolorowac (
    int wierzcholek,
    std::vector< std::vector< int > > graf,
    int kolor,
    std::vector< int > & kolory )
```

Funkcja sprawdzająca czy można pokolorować wskazany jako parametr wierzchołek, podanym jako parametr kolorem.

Parameters

<i>wierzcholek</i>	Numer wierzchołka
<i>graf</i>	Tablica reprezentująca graf wraz z jego połączeniami
<i>kolor</i>	Numer koloru
<i>kolory</i>	Tablica pokolorowanych wierzchołków

Returns

true - jeśli można pokolorować, false - jeśli nie można

2.2.1.2 czySasiedztwoPokolorowane()

```
bool czySasiedztwoPokolorowane (
    std::vector< int > sasiedztwo,
    std::vector< int > kolory )
```

Funkcja sprawdzająca czy całe sąsiedztwo danego wierzchołka ma już przydzielone swoje kolory.

Parameters

<i>sasiedztwo</i>	Sąsiedztwo wybranego wierzchołka
<i>kolory</i>	Tablica pokolorowanych wierzchołków

Returns

true - jeśli pokolorowane, false - jeśli niepokolorowane

2.2.1.3 obliczWykorzystaneKolory()

```
int obliczWykorzystaneKolory (
    std::vector< int > kolory )
```

Liczenie ilości wykorzystanych kolorów w grafie.

Parameters

<i>kolory</i>	Tablica pokolorowanych wierzchołków
---------------	-------------------------------------

Returns

Liczba wykorzystanych kolorów w grafie

2.2.1.4 pokolorujGraf()

```
void pokolorujGraf (
    int wierzcholek,
    std::vector< std::vector< int > > graf,
    std::vector< int > & kolory )
```

Główna funkcja odpowiedzialna za uruchomienie procesu kolorowania grafu.

Parameters

<i>wierzcholek</i>	Numer wierzchołka
<i>graf</i>	Tablica reprezentująca graf wraz z jego połączeniami
<i>kolory</i>	Tablica pokolorowanych wierzchołków

2.2.1.5 pokolorujSasiedztwo()

```
void pokolorujSasiedztwo (
    std::vector< std::vector< int > > graf,
    std::vector< int > sasiedztwo,
    std::vector< int > & kolory )
```

Funkcja rekurencyjna odpowiedzialna za kolorowanie grafu przechodząc po wszystkich sąsiadach wierzchołka. Funkcja wykonuje się do momentu aż wszystkie wierzchołki zostaną pokolorowane.

Parameters

<i>graf</i>	Tablica reprezentująca graf wraz z jego połączeniami
<i>sasiedztwo</i>	Sąsiedztwo wybranego wierzchołka
<i>kolory</i>	Tablica pokolorowanych wierzchołków

2.2.1.6 wczytajDaneZPliku()

```
std::vector< std::vector< int > > wczytajDaneZPliku (
    std::string plik )
```

Wczytywanie danych z podanego jako parametr pliku.

Parameters

<i>plik</i>	Nazwa pliku
-------------	-------------

Returns

Tablica reprezentująca graf wraz z jego połączeniami

2.2.1.7 wczytajParametry()

```
void wczytajParametry (
    int argc,
```

```
char ** argv,  
std::string & plikWejscowy,  
std::string & plikWynikowy )
```

Wczytanie parametrów wywołania do zmiennych programu.

Parameters

<i>argc</i>	Licznik parametrów
<i>argv</i>	Tablica wartości parametrów
<i>plikWejscowy</i>	Nazwa pliku wejściowego z parametrami grafu
<i>plikWynikowy</i>	Nazwa pliku wynikowego, do którego zostaną zapisane rezultaty działania programu

2.2.1.8 weryfikujPoprawnoscParametrow()

```
bool weryfikujPoprawnoscParametrow (  
    int argc,  
    char ** argv )
```

Weryfikacja poprawności parametrów wejściowych aplikacji.

Parameters

<i>argc</i>	Licznik parametrów
<i>argv</i>	Tablica wartości parametrów

Returns

Wynik weryfikacji poprawności parametrów

2.2.1.9 wyczyscKolory()

```
void wyczyscKolory (  
    std::vector< int > kolory )
```

Czyszczenie tablicy kolory.

Parameters

<i>kolory</i>	Tablica pokolorowanych wierzchołków
---------------	-------------------------------------

2.2.1.10 zapiszWynikDoPliku()

```
void zapiszWynikDoPliku (
    std::string nazwa,
    std::vector< std::vector< int > > graf,
    std::vector< int > kolory )
```

Zapisywanie wynikido pliku wyjściowego podanego jako parametr.

Parameters

<i>nazwa</i>	Nazwa pliku wyjściowego
<i>graf</i>	Tablica reprezentująca graf wraz z jego połączeniami
<i>kolory</i>	Tablica pokolorowanych wierzchołków

2.3 C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/model.h File Reference

```
#include <string>
#include <vector>
#include <fstream>
```

Functions

- bool [weryfikujPoprawnoscParametrow](#) (int argc, char **argv)
Weryfikacja poprawności parametrów wejściowych aplikacji.
- void [wczytajParametry](#) (int argc, char **argv, std::string &plikWejscowy, std::string &plikWynikowy)
Wczytanie parametrów wywołania do zmiennych programu.
- std::vector< std::vector< int > > [wczytajDaneZPliku](#) (std::string plik)
Wczytywanie danych z podanego jako parametr pliku.
- void [zapiszWynikDoPliku](#) (std::string nazwa, std::vector< std::vector< int > > graf, std::vector< int > kolory)
Zapisywanie wynikido pliku wyjściowego podanego jako parametr.
- void [pokolorujGraf](#) (int wierzcholek, std::vector< std::vector< int > > graf, std::vector< int > &kolory)
Główna funkcja odpowiedzialna za uruchomienie procesu kolorowania grafu.
- void [pokolorujSasiedztwo](#) (std::vector< std::vector< int > > graf, std::vector< int > sasiedztwo, std::vector< int > &kolory)
Funkcja rekurencyjna odpowiedzialna za kolorowanie grafu przechodząc po wszystkich sąsiadach wierzchołka. Funkcja wykonuje się do momentu aż wszystkie wierzchołki zostaną pokolorowane.
- bool [czyMoznaPokolorowac](#) (int wierzcholek, std::vector< std::vector< int > > graf, int kolor, std::vector< int > &kolory)
Funkcja sprawdzająca czy można pokolorować wskazany jako parametr wierzchołek, podanym jako parametr kolorem.
- bool [czySasiedztwoPokolorowane](#) (std::vector< int > sasiedztwo, std::vector< int > kolory)
Funkcja sprawdzająca czy całe sąsiedztwo danego wierzchołka ma już przydzielone swoje kolory.
- int [obliczWykorzystaneKolory](#) (std::vector< int > kolory)
Liczenie ilości wykorzystanych kolorów w grafie.
- void [wyczyscKolory](#) (std::vector< int > kolory)
Czyszczenie tablicy kolory.

2.3.1 Function Documentation

2.3.1.1 czyMoznaPokolorowac()

```
bool czyMoznaPokolorowac (
    int wierzcholek,
    std::vector< std::vector< int > > graf,
    int kolor,
    std::vector< int > & kolory )
```

Funkcja sprawdzająca czy można pokolorować wskazany jako parametr wierzchołek, podanym jako parametr kolorem.

Parameters

<i>wierzcholek</i>	Numer wierzchołka
<i>graf</i>	Tablica reprezentująca graf wraz z jego połączeniami
<i>kolor</i>	Numer koloru
<i>kolory</i>	Tablica pokolorowanych wierzchołków

Returns

true - jeśli można pokolorować, false - jeśli nie można

2.3.1.2 czySasiedztwoPokolorowane()

```
bool czySasiedztwoPokolorowane (
    std::vector< int > sasiedztwo,
    std::vector< int > kolory )
```

Funkcja sprawdzająca czy całe sąsiedztwo danego wierzchołka ma już przydzielone swoje kolory.

Parameters

<i>sasiedztwo</i>	Sąsiedztwo wybranego wierzchołka
<i>kolory</i>	Tablica pokolorowanych wierzchołków

Returns

true - jeśli pokolorowane, false - jeśli niepokolorowane

2.3.1.3 obliczWykorzystaneKolory()

```
int obliczWykorzystaneKolory (
    std::vector< int > kolory )
```

Liczenie ilości wykorzystanych kolorów w grafie.

Parameters

<i>kolory</i>	Tablica pokolorowanych wierzchołków
---------------	-------------------------------------

Returns

Liczba wykorzystanych kolorów w grafie

2.3.1.4 pokolorujGraf()

```
void pokolorujGraf (
    int wierzcholek,
    std::vector< std::vector< int > > graf,
    std::vector< int > & kolory )
```

Główna funkcja odpowiedzialna za uruchomienie procesu kolorowania grafu.

Parameters

<i>wierzcholek</i>	Numer wierzchołka
<i>graf</i>	Tablica reprezentująca graf wraz z jego połączeniami
<i>kolory</i>	Tablica pokolorowanych wierzchołków

2.3.1.5 pokolorujSasiedztwo()

```
void pokolorujSasiedztwo (
    std::vector< std::vector< int > > graf,
    std::vector< int > sasiedztwo,
    std::vector< int > & kolory )
```

Funkcja rekurencyjna odpowiedzialna za kolorowanie grafu przechodząc po wszystkich sąsiadach wierzchołka. Funkcja wykonuje się do momentu aż wszystkie wierzchołki zostaną pokolorowane.

Parameters

<i>graf</i>	Tablica reprezentująca graf wraz z jego połączeniami
<i>sasiedztwo</i>	Sąsiedztwo wybranego wierzchołka
<i>kolory</i>	Tablica pokolorowanych wierzchołków

2.3.1.6 wczytajDaneZPliku()

```
std::vector< std::vector< int > > wczytajDaneZPliku (
    std::string plik )
```

Wczytywanie danych z podanego jako parametr pliku.

Parameters

<i>plik</i>	Nazwa pliku
-------------	-------------

Returns

Tablica reprezentująca graf wraz z jego połączeniami

2.3.1.7 wczytajParametry()

```
void wczytajParametry (
    int argc,
    char ** argv,
    std::string & plikWejscowy,
    std::string & plikWynikowy )
```

Wczytanie parametrów wywołania do zmiennych programu.

Parameters

<i>argc</i>	Licznik parametrów
<i>argv</i>	Tablica wartości parametrów
<i>plikWejscowy</i>	Nazwa pliku wejściowego z parametrami grafu
<i>plikWynikowy</i>	Nazwa pliku wynikowego, do którego zostaną zapisane rezultaty działania programu

2.3.1.8 weryfikujPoprawnoscParametrow()

```
bool weryfikujPoprawnoscParametrow (
    int argc,
    char ** argv )
```

Weryfikacja poprawności parametrów wejściowych aplikacji.

Parameters

<i>argc</i>	Licznik parametrów
<i>argv</i>	Tablica wartości parametrów

Returns

Wynik weryfikacji poprawności parametrów

2.3.1.9 wyczyszcKolory()

```
void wyczyszcKolory (
    std::vector< int > kolory )
```

Czyszczenie tablicy kolory.

Parameters

<i>kolory</i>	Tablica pokolorowanych wierzchołków
---------------	-------------------------------------

2.3.1.10 zapiszWynikDoPliku()

```
void zapiszWynikDoPliku (
    std::string nazwa,
    std::vector< std::vector< int > > graf,
    std::vector< int > kolory )
```

Zapisywanie wynikido pliku wyjściowego podanego jako parametr.

Parameters

<i>nazwa</i>	Nazwa pliku wyjściowego
<i>graf</i>	Tablica reprezentująca graf wraz z jego połączeniami
<i>kolory</i>	Tablica pokolorowanych wierzchołków

2.4 model.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 #include <string>
00003 #include <vector>
00004 #include <fstream>
00011 bool weryfikujPoprawnoscParametrow(int argc, char** argv);
00019 void wczytajParametry(int argc, char** argv, std::string& plikWejscowy, std::string& plikWynikowy);
```



```
00025 std::vector<std::vector<int>> wczytajDaneZPliku(std::string plik);
00032 void zapiszWynikDoPliku(std::string nazwa, std::vector<std::vector<int>> graf, std::vector<int>
    kolory);
00039 void pokolorujGraf(int wierzcholek, std::vector<std::vector<int>> graf, std::vector<int>& kolory);
00047 void pokolorujSasiedztwo(std::vector<std::vector<int>> graf, std::vector<int> sasiedztwo,
    std::vector<int>& kolory);
00056 bool czyMoznaPokolorowac(int wierzcholek, std::vector<std::vector<int>> graf, int kolor,
    std::vector<int>& kolory);
00063 bool czySasiedztwoPokolorowane(std::vector<int> sasiedztwo, std::vector<int> kolory);
00069 int obliczWykorzystaneKolory(std::vector<int> kolory);
00074 void wyczyszcKolory(std::vector<int> kolory);
```


Index

C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/main.cpp, [3](#)
C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/model.cpp, [3](#)
C:/Users/SQ231/Downloads/TMP/Kolorowy graf/Kolorowy graf/Kolorowy graf/model.h, [8](#), [12](#)
czyMoznaPokolorowac
 model.cpp, [4](#)
 model.h, [9](#)
czySasiedztwoPokolorowane
 model.cpp, [5](#)
 model.h, [9](#)
main
 main.cpp, [3](#)
main.cpp
 main, [3](#)
model.cpp
 czyMoznaPokolorowac, [4](#)
 czySasiedztwoPokolorowane, [5](#)
 obliczWykorzystaneKolory, [5](#)
 pokolorujGraf, [5](#)
 pokolorujSasiedztwo, [6](#)
 wczytajDaneZPliku, [6](#)
 wczytajParametry, [6](#)
 weryfikujPoprawnoscParametrow, [7](#)
 wyczyscKolory, [7](#)
 zapiszWynikDoPliku, [7](#)
model.h
 czyMoznaPokolorowac, [9](#)
 czySasiedztwoPokolorowane, [9](#)
 obliczWykorzystaneKolory, [9](#)
 pokolorujGraf, [10](#)
 pokolorujSasiedztwo, [10](#)
 wczytajDaneZPliku, [11](#)
 wczytajParametry, [11](#)
 weryfikujPoprawnoscParametrow, [11](#)
 wyczyscKolory, [12](#)
 zapiszWynikDoPliku, [12](#)
obliczWykorzystaneKolory
 model.cpp, [5](#)
 model.h, [9](#)
pokolorujGraf
 model.cpp, [5](#)
 model.h, [10](#)
pokolorujSasiedztwo
 model.cpp, [6](#)
model.h, [10](#)
wczytajDaneZPliku
 model.cpp, [6](#)
 model.h, [11](#)
wczytajParametry
 model.cpp, [6](#)
 model.h, [11](#)
weryfikujPoprawnoscParametrow
 model.cpp, [7](#)
 model.h, [11](#)
wyczyscKolory
 model.cpp, [7](#)
 model.h, [12](#)
zapiszWynikDoPliku
 model.cpp, [7](#)
 model.h, [12](#)