ALOPS

Wygenerowano przez Doxygen 1.8.17

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 link	5
3.1.1 Opis szczegółowy	5
3.2 Dokumentacja struktury vertex	5
3.2.1 Opis szczegółowy	6
4 Dokumentacja plików	7
4.1 Dokumentacja pliku Funkcje.cpp	7
4.1.1 Dokumentacja funkcji	8
4.1.1.1 count()	8
4.1.1.2 delete_vertices_forend_r()	8
4.1.1.3 gauss()	8
4.1.1.4 generate_currents()	9
4.1.1.5 help()	9
4.1.1.6 makematrix()	9
4.1.1.7 sum_G()	9
4.1.2 Dokumentacja zmiennych	11
4.1.2.1 Internal_resistance	11
4.2 Dokumentacja pliku Funkcje.h	11
4.2.1 Dokumentacja funkcji	11
4.2.1.1 count()	11
4.2.1.2 delete_vertices_forend_r()	12
4.2.1.3 download()	12
4.2.1.4 gauss()	12
4.2.1.5 generate_currents()	13
4.2.1.6 help()	13
4.2.1.7 makematrix()	13
4.2.1.8 save()	14
4.2.1.9 sum_G()	14
4.3 Dokumentacja pliku main.cpp	14
4.4 Dokumentacja pliku Struktury.h	15

Indeks klas

1.1 Lista klas

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

link										
vertex	< Element listy połączęń/ gałęzie między wierzchołkami						 			5
TOITOX	< Wierzchołek grafu/ wezeł obwodu	 					 			5

2 Indeks klas

Indeks plików

2.1 Lista plików

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

Funkcje.cp	р					 																		7
Funkcje.h						 																		11
main.cpp .						 																		14
Strukturv.h						 										 								15

4 Indeks plików

Dokumentacja klas

3.1 Dokumentacja struktury link

< Element listy połączęń/ gałęzie między wierzchołkami

```
#include <Struktury.h>
```

Atrybuty publiczne

· int number

Indeks węzła będącego końcem tego połączenia.

link * nextlink

Wskaźnik na następny element listy.

· char type

Typ połączenia.

· long double value

Wartość wyrażona w jednostakch zalęznie od typu.

· unsigned int id

Indywidualne identyfikator pozwalający rozróżnić połączenia przu wypisywaniu.

· long double i

Prąd płynący przez gałąź

3.1.1 Opis szczegółowy

< Element listy połączęń/ gałęzie między wierzchołkami

Dokumentacja dla tej struktury została wygenerowana z pliku:

· Struktury.h

3.2 Dokumentacja struktury vertex

< Wierzchołek grafu/ węzeł obwodu

```
#include <Struktury.h>
```

6 Dokumentacja klas

Atrybuty publiczne

• int number

Nr węzłą

vertex * nextvertex

Wskaźnik na następny element listy węzłów.

link * phead

Wskaźnik na pierwszy element listy połączeń wychodzących z danego wierzchołka.

• long double potential

Potencjał węzła.

3.2.1 Opis szczegółowy

< Wierzchołek grafu/ węzeł obwodu

Dokumentacja dla tej struktury została wygenerowana z pliku:

· Struktury.h

Dokumentacja plików

4.1 Dokumentacja pliku Funkcje.cpp

```
#include <iostream>
#include <string>
#include <clocale>
#include <fstream>
#include <iomanip>
#include <cmath>
#include <vector>
#include "Funkcje.h"
```

Funkcje

- void delete_links_forend_r (link *&phead)
- void delete_vertices_forend_r (vertex *&phead)
- void print_I (link *phead, int begin)
- void add_v (vertex *&phead, int id)
- void add_I (vertex *&p, int begin, int end, char t, long double v, unsigned int id)
- void help ()
- bool download (string name, vertex *&a)
- long double sum_current_forcing (int nr, vertex *phead)
- long double **sum_G** (int first, int second, **vertex** *Phead)
- long double ** makematrix (vertex *phead)
- void generate_currents (vertex *Phead)
- bool gauss (long double **&Matrix, int n)
- int count (vertex *phead)
- bool is_readed (int nr, vertex *id)
- bool is_readed (int nr, vector< int > id)
- bool save (string name, vertex *phead)
- void swapp (vertex *phead1, vertex *phead2)
- void sort (vertex *Phead0)

Zmienne

• const long double Internal_resistance = 0.0000001

4.1.1 Dokumentacja funkcji

4.1.1.1 count()

Funkcja sprawdza liczbę elementów w liscie - wierzchołków w grafie.

Parametry

in	phead	Wskaźnik na pierwszy element listy
----	-------	------------------------------------

Zwraca

liczba elementów w liście

4.1.1.2 delete_vertices_forend_r()

Funkcja usuwa listę

Parametry

in, out phead Wskaźnik na początek list	, po usunięciu listy zwracany jest nullptr
---	--

4.1.1.3 gauss()

```
bool gauss ( \label{eq:long_double} \begin{tabular}{ll} $\operatorname{long_double}$ **& $\operatorname{\it Matrix}$, \\ & \operatorname{int} n \end{tabular}
```

Funkcja rozwiązuje układ równań liniowych

Parametry

in,out	Matrix	Tablica współczynników układu równanań linowych w postaci [n][n+1], rozwiązania zwaracane są w ostatniej kolumnie układu równań.
in	n	Liczba zmiennych w układzie

Zwraca

true gdy układ jest rozwiązywalny, false jeśli układ jest sprzeczny lub tożsamościowy

4.1.1.4 generate_currents()

Funkcja oblicza prądy płynące przez wszystkie połącznie miedzy wierzchołkami grafu zadekalrowanymi w liściee

Parametry

in phead Wskaźnik na początek	listy
-------------------------------	-------

4.1.1.5 help()

```
void help ( )
```

Funkcja wyświetla pomoc dla użytkownika

4.1.1.6 makematrix()

Funkcja tworzy, na podstawie grafu, macierz konduktancji oraz wymuszeń prądowych.

Parametry

in	phead	Wskaźnik na pierwszy element listy

Zwraca

Wskaźnik na dwuwymiarową tablicę

4.1.1.7 sum_G()

```
long double sum_G (
          int first,
```

```
int second,
  vertex * phead )
```

Funkcja oblicza sumę konduktancji miedzy dwoma zadanymi węzłami, lub sumę konduktancji podłączonych do danego węzła.

Parametry

in	first	Indeksy węzłów miedzy którymi ma być policzona suma konduktancji
in	second	Indeksy węzłów miedzy którymi ma być policzona suma konduktancji
in	phead	Wskaźnik na początek listy

4.1.2 Dokumentacja zmiennych

4.1.2.1 Internal resistance

```
const long double Internal_resistance = 0.0000001
```

W rzeczywistości nie istnieją idealne źródła napięciowe - bez rezystancji wewnętrznej - i dla takich trudnym jest zastosowanie metody potencjałów węzłowych. Rezystancje wewnętrzne źródeł rzeczywistych są rzędu dziesiątych części Ohma, zatem w obwodach w których zadeklarowna przezemnie wartość miałaby znaczenie nie możnaby pominąć nieidealności źródeł.

4.2 Dokumentacja pliku Funkcje.h

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

Funkcje

- bool download (std::string name, vertex *&phead)
- bool save (std::string name, vertex *phead)
- bool **gauss** (long double **&Matrix, int n)
- int count (vertex *phead)
- long double ** makematrix (vertex *phead)
- long double **sum_G** (int first, int second, **vertex** *phead)
- void generate_currents (vertex *phead)
- void delete_vertices_forend_r (vertex *&phead)
- void help ()

4.2.1 Dokumentacja funkcji

4.2.1.1 count()

```
int count (
     vertex * phead )
```

Funkcja sprawdza liczbę elementów w liscie - wierzchołków w grafie.

Parametry

in	phead	Wskaźnik na pierwszy element listy
----	-------	------------------------------------

Zwraca

liczba elementów w liście

4.2.1.2 delete_vertices_forend_r()

Funkcja usuwa listę

Parametry

in,out	phead	Wskaźnik na początek listy, po usunięciu listy zwracany jest nullptr
--------	-------	--

4.2.1.3 download()

Funkcja pobiera z pliku dane i dodaje elementy do listy.

Parametry

in	name	Ścieżka do pliku
in,out	phead	Wskaźnik na pierwszy element listy

Zwraca

true jeśli odczyt przebiegł bezbłędnie, false jeśi się nie powiódł

4.2.1.4 gauss()

```
bool gauss ( \label{eq:long_double} \mbox{long double **& $\it Matrix$,} \\ \mbox{int $\it n$} \mbox{)}
```

Funkcja rozwiązuje układ równań liniowych

Parametry

in,out	Matrix	Tablica współczynników układu równanań linowych w postaci [n][n+1], rozwiązania	
		zwaracane są w ostatniej kolumnie układu równań.	
in	n	Liczba zmiennych w układzie	

Zwraca

true gdy układ jest rozwiązywalny, false jeśli układ jest sprzeczny lub tożsamościowy

4.2.1.5 generate currents()

Funkcja oblicza prądy płynące przez wszystkie połącznie miedzy wierzchołkami grafu zadekalrowanymi w liściee

Parametry

i	.n	phead	Wskaźnik na początek listy
---	----	-------	----------------------------

4.2.1.6 help()

```
void help ( )
```

Funkcja wyświetla pomoc dla użytkownika

4.2.1.7 makematrix()

Funkcja tworzy, na podstawie grafu, macierz konduktancji oraz wymuszeń prądowych.

Parametry

in phead Wskaźnik na pierwszy element lis	ty	
---	----	--

Zwraca

Wskaźnik na dwuwymiarową tablicę

4.2.1.8 save()

```
bool save ( {\tt std::string} \ {\tt name,} {\tt vertex} \ * \ {\tt phead} \ )
```

Funkcja zapisuje do pliku analizę obwodu.

Parametry

in	name	Ścieżka do pliku	
in	phead	Wskaźnik na pierwszy element listy	

Zwraca

true jeśli zapis przebiegł bezbłędnie, false jeśi się nie powiódł

4.2.1.9 sum_G()

```
long double sum_G (
          int first,
          int second,
          vertex * phead )
```

Funkcja oblicza sumę konduktancji miedzy dwoma zadanymi węzłami, lub sumę konduktancji podłączonych do danego węzła.

Parametry

in	first	Indeksy węzłów miedzy którymi ma być policzona suma konduktancji
in	second	Indeksy węzłów miedzy którymi ma być policzona suma konduktancji
in	phead	Wskaźnik na początek listy

4.3 Dokumentacja pliku main.cpp

```
#include <string>
#include <clocale>
#include <iostream>
#include <fstream>
#include <iomanip>
#include "Funkcje.h"
#include "struktury.h"
```

Funkcje

• int main (int argc, char *argv[])

4.4 Dokumentacja pliku Struktury.h

Komponenty

- struct link
 - < Element listy połączęń/ gałęzie między wierzchołkami
- struct vertex
 - < Wierzchołek grafu/ węzeł obwodu