

## Sieć komputerowa

Wygenerowano przez Doxygen 1.8.17



<b>1 Indeks hierarchiczny</b>	<b>1</b>
1.1 Hierarchia klas	1
<b>2 Indeks klas</b>	<b>3</b>
2.1 Lista klas	3
<b>3 Indeks plików</b>	<b>5</b>
3.1 Lista plików	5
<b>4 Dokumentacja klas</b>	<b>7</b>
4.1 Dokumentacja klasy coordinates	7
4.1.1 Opis szczegółowy	8
4.1.2 Dokumentacja funkcji składowych	8
4.1.2.1 sendV()	8
4.1.2.2 sendW()	8
4.2 Dokumentacja struktury coorList	8
4.2.1 Opis szczegółowy	9
4.3 Dokumentacja klasy inhabitants	9
4.3.1 Opis szczegółowy	9
<b>5 Dokumentacja plików</b>	<b>11</b>
5.1 Dokumentacja pliku functions.cpp	11
5.1.1 Dokumentacja funkcji	12
5.1.1.1 addToCoorList()	12
5.1.1.2 addToTabVector()	12
5.1.1.3 checkArguments()	13
5.1.1.4 eraseList()	13
5.1.1.5 fillQueue()	13
5.1.1.6 howFar()	14
5.1.1.7 makeTheWeeb()	14
5.1.1.8 readCoorName()	15
5.1.1.9 readTabName()	15
5.2 Dokumentacja pliku functions.h	15
5.2.1 Dokumentacja funkcji	16
5.2.1.1 addToCoorList()	17
5.2.1.2 addToTabVector()	17
5.2.1.3 checkArguments()	17
5.2.1.4 eraseList()	18
5.2.1.5 fillQueue()	18
5.2.1.6 howFar()	19
5.2.1.7 makeTheWeeb()	19
5.2.1.8 readCoorName()	20
5.2.1.9 readTabName()	20
5.3 Dokumentacja pliku main.cpp	20

5.4 Dokumentacja pliku structs.h . . . . .	21
<b>Indeks</b>	<b>23</b>

# Rozdział 1

## Indeks hierarchiczny

### 1.1 Hierarchia klas

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

coorList . . . . .	8
inhabitants . . . . .	9
coordinates . . . . .	7



## Rozdział 2

# Indeks klas

### 2.1 Lista klas

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

<a href="#">coordinates</a>	7
<a href="#">coordList</a>	8
<a href="#">inhabitants</a>	9





## Rozdział 3

# Indeks plików

### 3.1 Lista plików

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

<a href="#">functions.cpp</a>	11
<a href="#">functions.h</a>	15
<a href="#">main.cpp</a>	20
<a href="#">structs.h</a>	21



## Rozdział 4

# Dokumentacja klas

### 4.1 Dokumentacja klasy coordinates

```
#include <structs.h>
```

Diagram dziedziczenia dla coordinates

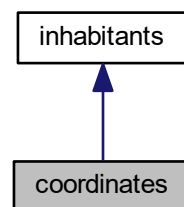
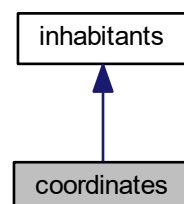


Diagram współpracy dla coordinates:



## Metody publiczne

- int [sendV](#) ()
- int [sendW](#) ()

### 4.1.1 Opis szczegółowy

Klasa przechowująca współrzędne mieszkańców - pochodna klasy inhabitants

### 4.1.2 Dokumentacja funkcji składowych

#### 4.1.2.1 sendV()

```
int coordinates::sendV ( ) [inline]
```

< metoda zwracająca współrzędną v

#### 4.1.2.2 sendW()

```
int coordinates::sendW ( ) [inline]
```

< metoda zwracająca współrzędną w

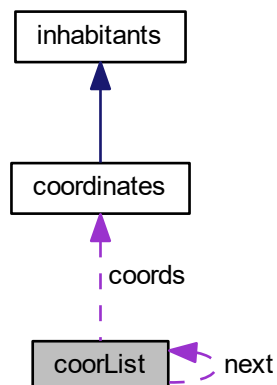
Dokumentacja dla tej klasy została wygenerowana z pliku:

- [structs.h](#)

## 4.2 Dokumentacja struktury coorList

```
#include <structs.h>
```

Diagram współpracy dla coorList:



## Atrybuty publiczne

- [coordinates](#) \* [coords](#)  
*wskaźnik na obiekt coordinations*
- [coorList](#) \* [next](#)  
*wskaźnik na następny element listy*

### 4.2.1 Opis szczegółowy

Lista przechowująca wskaźniki na obiekty coordinations

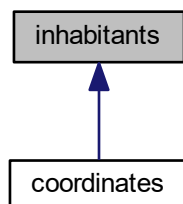
Dokumentacja dla tej struktury została wygenerowana z pliku:

- [structs.h](#)

## 4.3 Dokumentacja klasy inhabitants

```
#include <structs.h>
```

Diagram dziedziczenia dla inhabitants



### 4.3.1 Opis szczegółowy

Klasa przechowująca index, oraz nazwisko mieszkańców

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [structs.h](#)



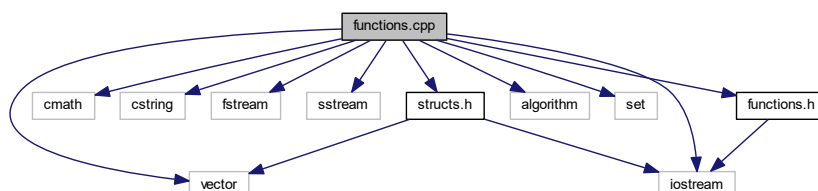
## Rozdział 5

# Dokumentacja plików

### 5.1 Dokumentacja pliku functions.cpp

```
#include <iostream>
#include <cmath>
#include <cstring>
#include <fstream>
#include <sstream>
#include <vector>
#include <algorithm>
#include <set>
#include "structs.h"
#include "functions.h"
```

Wykres zależności załączania dla functions.cpp:



### Funkcje

- int [checkArguments](#) (int argc, char \*\*&argv, std::string &coorName, std::string &tabName, std::string &outName)
- void [readCoorName](#) (const std::string &coorName, [coorList](#) \*&head)
- void [addToCoorList](#) ([coorList](#) \*&head, int \_index, std::string \_surname, int \_v, int \_w)
- void [readTabName](#) (const std::string &tabName, std::vector< std::pair< int, int >> &tabVector)
- void [addToTabVector](#) (std::vector< std::pair< int, int >> &tabVector, int \_x, int \_y)
- void [fillQueue](#) (std::vector< std::pair< int, std::pair< int, int >> > &Q, [coorList](#) \*cHead, std::vector< std::pair< int, int >> tabVector)
- int [howFar](#) ([coordinates](#) \*mainElement, [coordinates](#) \*secondaryElement)
- void [makeTheWeeb](#) (std::vector< std::set< int >> &sets, std::vector< std::pair< int, std::pair< int, int >> > &Q, const std::string &outName)
- void [eraseList](#) ([coorList](#) \*cHead)

## 5.1.1 Dokumentacja funkcji

### 5.1.1.1 addToCoorList()

```
void addToCoorList (
    coorList *& head,
    int _index,
    std::string _surname,
    int _v,
    int _w )
```

Funkcja tworząca obiekt coordinates i dodająca go do listy

#### Parametry

<i>head</i>	głowa listy
<i>_index</i>	dana index do utworzenia obiektu
<i>_surname</i>	nazwisko
<i>_v</i>	współrzędna v
<i>_w</i>	współrzędna w

#### Zwraca

void

### 5.1.1.2 addToTabVector()

```
void addToTabVector (
    std::vector< std::pair< int, int >> & tabVector,
    int _x,
    int _y )
```

Funkcja dodająca element do vectora z niemożliwymi połączeniami

#### Parametry

<i>tabVector</i>	vector par intów
<i>_x</i>	mieszkanie 1
<i>_y</i>	mieszkanie 2

#### Zwraca

void



### 5.1.1.3 checkArguments()

```
int checkArguments (
    int argc,
    char **& argv,
    std::string & coorName,
    std::string & tabName,
    std::string & outName )
```

Funkcja sprawdza argumenty podane w konsoli przy uruchomieniu programu

#### Parametry

<i>argc</i>	ilość wyrazów pobrana z konsoli
<i>argv</i>	wyrazy pobrane z konsoli
<i>coorName</i>	nazwa pliku ze współrzędnymi
<i>tabName</i>	nazwa pliku z niemożliwymi połączeniami
<i>outName</i>	nazwa pliku wyjściowego

#### Zwraca

int

### 5.1.1.4 eraseList()

```
void eraseList (
    coorList * cHead )
```

Funkcja zwalniająca pamięć z listy wskaźników na obiekty

#### Parametry

<i>cHead</i>	wskaźnik na kolejne elementy listy
--------------	------------------------------------

#### Zwraca

void

### 5.1.1.5 fillQueue()

```
void fillQueue (
    std::vector< std::pair< int, std::pair< int, int > > > & Q,
    coorList * cHead,
    std::vector< std::pair< int, int >> tabVector )
```

Funkcja uzupełniająca i sortująca kolejkę Q wykorzystywaną przez algorytm kruskala

## Parametry

<i>Q</i>	vector par int par intów reprezentujący odległość i mieszkania między którymi możemy zrobić połączenie
<i>cHead</i>	głowa listy na kolejne obiekty coordinates

## Zwraca

void

## 5.1.1.6 howFar()

```
int howFar (
    coordinates * mainElement,
    coordinates * secondaryElement )
```

Funkcja licząca odległość między mieszkaniami idąc po prostych

## Parametry

<i>mainElement</i>	wskaźnik na obiekt 1 mieszkania
<i>secondaryElement</i>	wskaźnik na obiekt 2 mieszkania

## Zwraca

int

## 5.1.1.7 makeTheWeeb()

```
void makeTheWeeb (
    std::vector< std::set< int > > & sets,
    std::vector< std::pair< int, std::pair< int, int > > > & Q,
    const std::string & outName )
```

Funkcja wpisująca do pliku kolejne połączenia między mieszkaniami wraz z sumaryczną długością kabli przy każdym nowym połączeniu

## Parametry

<i>sets</i>	vector zbiorów mieszkań dzięki którym eliminujemy pętle
<i>Q</i>	vector par int par intów reprezentujący odległość i mieszkania między którymi możemy zrobić połączenie
<i>outName</i>	nazwa pliku wyjściowego

Zwraca

void

#### 5.1.1.8 readCoorName()

```
void readCoorName (
    const std::string & coorName,
    coorList *& head )
```

Funkcja wczytująca plik ze współzrędnymi do listy

Parametry

<i>coorName</i>	nazwa pliku
<i>head</i>	głowa listy na współzrędnym mieszkańców (początkowo nullptr)

Zwraca

void

#### 5.1.1.9 readTabName()

```
void readTabName (
    const std::string & tabName,
    std::vector< std::pair< int, int >> & tabVector )
```

Funkcja wczytująca plik z niemożliwymi połączeniami wpisującą je do wektora par intów i sortującą go malejąco

Parametry

<i>tabName</i>	nazwa pliku z niemożliwymi połączeniami
<i>tabVector</i>	wektor z parą intów niemożliwych do połączenia mieszkań

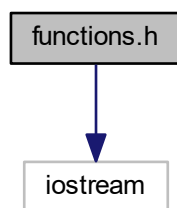
Zwraca

void

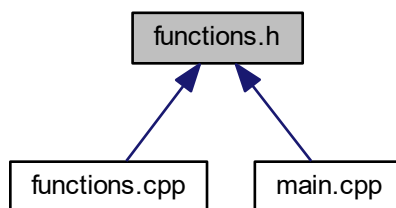
## 5.2 Dokumentacja pliku functions.h

```
#include <iostream>
```

Wykres zależności załączania dla functions.h:



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



## Funkcje

- int `checkArguments` (int argc, char \*\*&argv, std::string &coordName, std::string &tabName, std::string &outName)
- void `readCoordName` (const std::string &coordName, `coordList` \*&head)
- void `addToCoordList` (`coordList` \*&head, int \_index, std::string \_surname, int \_v, int \_w)
- void `readTabName` (const std::string &tabName, std::vector< std::pair< int, int >> &tabVector)
- void `addToTabVector` (std::vector< std::pair< int, int >> &tabVector, int \_x, int \_y)
- void `fillQueue` (std::vector< std::pair< int, std::pair< int, int >> > &Q, `coordList` \*cHead, std::vector< std::pair< int, int >> tabVector)
- int `howFar` (`coordinates` \*mainElement, `coordinates` \*secondaryElement)
- void `makeTheWeeb` (std::vector< std::set< int >> &sets, std::vector< std::pair< int, std::pair< int, int >> > &Q, const std::string &outName)
- void `eraseList` (`coordList` \*cHead)

### 5.2.1 Dokumentacja funkcji

### 5.2.1.1 addToCoorList()

```
void addToCoorList (
    coorList *& head,
    int _index,
    std::string _surname,
    int _v,
    int _w )
```

Funkcja tworząca obiekt coordinates i dodająca go do listy

#### Parametry

<i>head</i>	głowa listy
<i>_index</i>	dana index do utworzenia obiektu
<i>_surname</i>	nazwisko
<i>_v</i>	współrzędna v
<i>_w</i>	współrzędna w

#### Zwraca

void

### 5.2.1.2 addToTabVector()

```
void addToTabVector (
    std::vector< std::pair< int, int >> & tabVector,
    int _x,
    int _y )
```

Funkcja dodająca element do vectora z niemożliwymi połączeniami

#### Parametry

<i>tabVector</i>	vector par intów
<i>_x</i>	mieszkanie 1
<i>_y</i>	mieszkanie 2

#### Zwraca

void

### 5.2.1.3 checkArguments()

```
int checkArguments (
    int argc,
```

```
char **& argv,  
std::string & coorName,  
std::string & tabName,  
std::string & outName )
```

Funkcja sprawdza argumenty podane w konsoli przy uruchomieniu programu

#### Parametry

<i>argc</i>	ilość wyrazów pobrana z konsoli
<i>argv</i>	wyrazy pobrane z konsoli
<i>coorName</i>	nazwa pliku ze współrzędnymi
<i>tabName</i>	nazwa pliku z niemożliwymi połączeniami
<i>outName</i>	nazwa pliku wyjściowego

#### Zwraca

int

#### 5.2.1.4 eraseList()

```
void eraseList (  
    coorList * cHead )
```

Funkcja zwalnająca pamięć z listy wskaźników na obiekty

#### Parametry

<i>cHead</i>	wskaźnik na kolejne elementy listy
--------------	------------------------------------

#### Zwraca

void

#### 5.2.1.5 fillQueue()

```
void fillQueue (  
    std::vector< std::pair< int, std::pair< int, int > > > & Q,  
    coorList * cHead,  
    std::vector< std::pair< int, int >> tabVector )
```

Funkcja zapelniająca i sortująca kolejkę Q wykorzystywaną przez algorytm kruskala

## Parametry

<i>Q</i>	vector par int par intów reprezentujący odległość i mieszkania między którymi możemy zrobić połączenie
<i>cHead</i>	głowa listy na kolejne obiekty coordinates

## Zwraca

void

## 5.2.1.6 howFar()

```
int howFar (
    coordinates * mainElement,
    coordinates * secondaryElement )
```

Funkcja licząca odległość między mieszkaniami idąc po prostych

## Parametry

<i>mainElement</i>	wskaźnik na obiekt 1 mieszkania
<i>secondaryElement</i>	wskaźnik na obiekt 2 mieszkania

## Zwraca

int

## 5.2.1.7 makeTheWeeb()

```
void makeTheWeeb (
    std::vector< std::set< int > > & sets,
    std::vector< std::pair< int, std::pair< int, int > > > & Q,
    const std::string & outName )
```

Funkcja wpisująca do pliku kolejne połączenia między mieszkaniami wraz z sumaryczną długością kabli przy każdym nowym połączeniu

## Parametry

<i>sets</i>	vector zbiorów mieszkań dzięki którym eliminujemy pętle
<i>Q</i>	vector par int par intów reprezentujący odległość i mieszkania między którymi możemy zrobić połączenie
<i>outName</i>	nazwa pliku wyjściowego

**Zwraca**

void

**5.2.1.8 readCoorName()**

```
void readCoorName (
    const std::string & coorName,
    coorList *& head )
```

Funkcja wczytująca plik ze współrzędnymi do listy

**Parametry**

<i>coorName</i>	nazwa pliku
<i>head</i>	głowa listy na współrzędne mieszkańców (początkowo nullptr)

**Zwraca**

void

**5.2.1.9 readTabName()**

```
void readTabName (
    const std::string & tabName,
    std::vector< std::pair< int, int >> & tabVector )
```

Funkcja wczytująca plik z niemożliwymi połączeniami wpisująca je do wektora par intów i sortująca go malejąco

**Parametry**

<i>tabName</i>	nazwa pliku z niemożliwymi połączeniami
<i>tabVector</i>	vector z parą intów niemożliwych do połączenia mieszkań

**Zwraca**

void

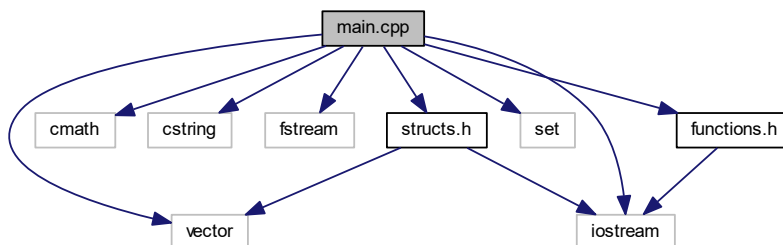
**5.3 Dokumentacja pliku main.cpp**

```
#include <iostream>
#include <cmath>
#include <cstring>
```



```
#include <fstream>
#include <vector>
#include <set>
#include "structs.h"
#include "functions.h"
```

Wykres zależności załączania dla main.cpp:



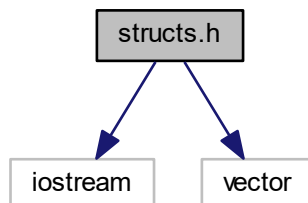
## Funkcje

- int **main** (int argc, char \*\*argv)

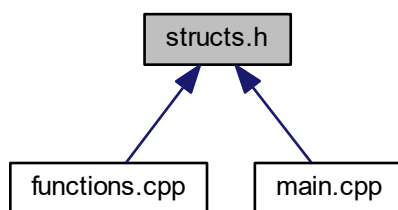
## 5.4 Dokumentacja pliku structs.h

```
#include <iostream>
#include <vector>
```

Wykres zależności załączania dla structs.h:



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



## Komponenty

- class `inhabitants`
- class `coordinates`
- struct `coorList`

# Indeks

- addToCoorList
  - functions.cpp, [12](#)
  - functions.h, [16](#)
- addToTabVector
  - functions.cpp, [12](#)
  - functions.h, [17](#)
- checkArguments
  - functions.cpp, [12](#)
  - functions.h, [17](#)
- coordinates, [7](#)
  - sendV, [8](#)
  - sendW, [8](#)
- coorList, [8](#)
- eraseList
  - functions.cpp, [13](#)
  - functions.h, [18](#)
- fillQueue
  - functions.cpp, [13](#)
  - functions.h, [18](#)
- functions.cpp, [11](#)
  - addToCoorList, [12](#)
  - addToTabVector, [12](#)
  - checkArguments, [12](#)
  - eraseList, [13](#)
  - fillQueue, [13](#)
  - howFar, [14](#)
  - makeTheWeeb, [14](#)
  - readCoorName, [15](#)
  - readTabName, [15](#)
- functions.h, [15](#)
  - addToCoorList, [16](#)
  - addToTabVector, [17](#)
  - checkArguments, [17](#)
  - eraseList, [18](#)
  - fillQueue, [18](#)
  - howFar, [19](#)
  - makeTheWeeb, [19](#)
  - readCoorName, [20](#)
  - readTabName, [20](#)
- howFar
  - functions.cpp, [14](#)
  - functions.h, [19](#)
- inhabitants, [9](#)
- main.cpp, [20](#)
- makeTheWeeb
  - functions.cpp, [14](#)
  - functions.h, [19](#)
- readCoorName
  - functions.cpp, [15](#)
  - functions.h, [20](#)
- readTabName
  - functions.cpp, [15](#)
  - functions.h, [20](#)
- sendV
  - coordinates, [8](#)
- sendW
  - coordinates, [8](#)
- structs.h, [21](#)