

Spedycja

Wygenerowano przez Doxygen 1.8.14

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 droga	5
3.1.1	Opis szczegółowy	5
3.1.2	Dokumentacja atrybutów składowych	6
3.1.2.1	pdroga	6
3.1.2.2	pmiasto	6
3.1.2.3	trasa	6
3.2	Dokumentacja struktury miasto	6
3.2.1	Opis szczegółowy	7
3.2.2	Dokumentacja atrybutów składowych	7
3.2.2.1	miastaobok	8
3.2.2.2	nazwamiasta	8
3.2.2.3	odleglosc_od_centrali	8
3.2.2.4	odwiedzony	8
3.2.2.5	pmiasto	8
3.2.2.6	pMiastoPoprzednie	8

4 Dokumentacja plików	9
4.1 Dokumentacja pliku ConsoleApplication1.cpp	9
4.1.1 Dokumentacja definicji	10
4.1.1.1 debug	10
4.1.2 Dokumentacja funkcji	10
4.1.2.1 main()	10
4.2 Dokumentacja pliku funkcje.cpp	11
4.2.1 Dokumentacja funkcji	11
4.2.1.1 Dijkstra()	11
4.2.1.2 help()	12
4.2.1.3 sprawdz_argumenty()	13
4.2.1.4 stworz_droga()	13
4.2.1.5 stworz_miasto()	14
4.2.1.6 usun()	15
4.2.1.7 usun_drogi()	15
4.2.1.8 usun_miasta()	16
4.2.1.9 wczytajzPliku()	17
4.2.1.10 wypisz_droga()	17
4.2.1.11 wypisz_miasta()	18
4.2.1.12 wypisz_miasto()	19
4.2.1.13 wypisz_wynik()	19
4.3 Dokumentacja pliku funkcje.h	20
4.3.1 Dokumentacja funkcji	21
4.3.1.1 Dijkstra()	22
4.3.1.2 help()	22
4.3.1.3 sprawdz_argumenty()	23
4.3.1.4 stworz_droga()	24
4.3.1.5 stworz_miasto()	24
4.3.1.6 usun()	25
4.3.1.7 usun_drogi()	26
4.3.1.8 usun_miasta()	26
4.3.1.9 wczytajzPliku()	27
4.3.1.10 wypisz_droga()	28
4.3.1.11 wypisz_miasta()	28
4.3.1.12 wypisz_miasto()	29
4.3.1.13 wypisz_wynik()	30
4.4 Dokumentacja pliku pch.cpp	31
4.5 Dokumentacja pliku pch.h	31
4.6 Dokumentacja pliku struktury.h	32

Rozdział 1

Indeks klas

1.1 Lista klas

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

droga	5
miasto	6

Rozdział 2

Indeks plików

2.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

ConsoleApplication1.cpp	9
funkcje.cpp	11
funkcje.h	20
pch.cpp	31
pch.h	31
struktury.h	32

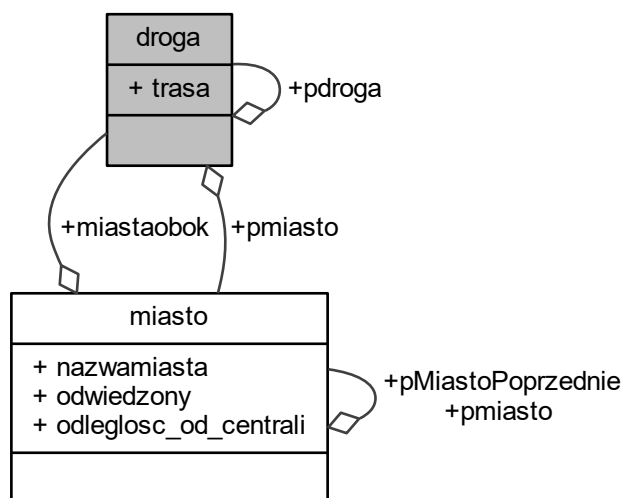
Rozdział 3

Dokumentacja klas

3.1 Dokumentacja struktury droga

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

Diagram współpracy dla droga:



Atrybuty publiczne

- int [trasa](#)
- droga * [pdroga](#)
- miasto * [pmiasto](#)

3.1.1 Opis szczegółowy

Struktura droga

Parametry

<i>trasa</i>	odleglosc miedzy miastami
<i>pdroga</i>	wskaznik na nastepna droge
<i>pmiasto</i>	wskaznik na odpowiednie miasto

3.1.2 Dokumentacja atrybutów składowych

3.1.2.1 pdroga

```
droga* droga::pdroga
```

3.1.2.2 pmiasto

```
miasto* droga::pmiasto
```

3.1.2.3 trasa

```
int droga::trasa
```

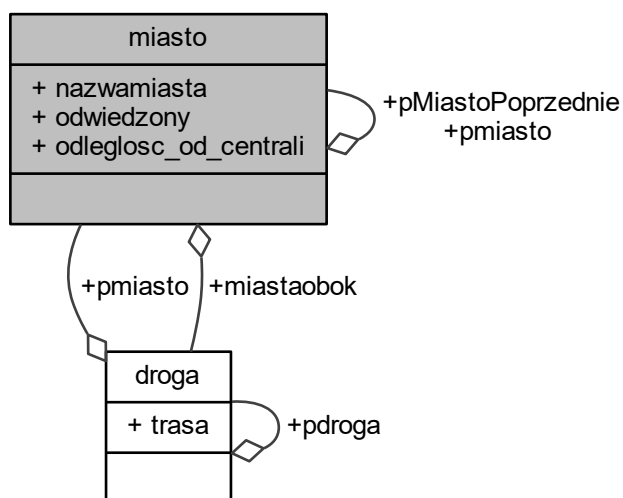
Dokumentacja dla tej struktury została wygenerowana z pliku:

- [struktury.h](#)

3.2 Dokumentacja struktury miasto

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

Diagram współpracy dla miasto:



Atrybuty publiczne

- `std::string` `nazwamiasta`
- `miasto` * `pmiasto`
- `droga` * `miastaobok`
- `bool` `odwiedzony`
- `int` `odleglosc_od_centrali`
- `miasto` * `pMiastoPoprzednie`

3.2.1 Opis szczegółowy

Struktura miasto

Parametry

<i>nazwamiasta</i>	nazwa miasta
<i>pmiasto</i>	wskaznik na nastepne miasto
<i>miastaobok</i>	wskaznik na pierwszy element list drog
<i>odwiedzony</i>	true, gdy miasto zostalo odwiedzone, false – nie zostalo odwiedzone
<i>odleglosc_od_centrali</i>	odleglosc od centrali
<i>pMiastoPoprzednie</i>	wskaznik na miasto poprzednie

3.2.2 Dokumentacja atrybutów składowych

3.2.2.1 miastaobok

[droga*](#) miasto::miastaobok

3.2.2.2 nazwamiasta

std::string miasto::nazwamiasta

3.2.2.3 odleglosc_od_centrali

int miasto::odleglosc_od_centrali

3.2.2.4 odwiedzony

bool miasto::odwiedzony

3.2.2.5 pmiasto

[miasto*](#) miasto::pmiasto

3.2.2.6 pMiastoPoprzednie

[miasto*](#) miasto::pMiastoPoprzednie

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [struktury.h](#)

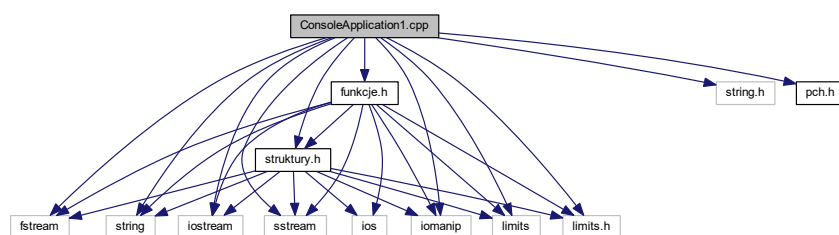
Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku ConsoleApplication1.cpp

```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <iomanip>
#include <limits>
#include <limits.h>
#include "string.h"
#include "pch.h"
#include "struktury.h"
#include "funkcje.h"
```

Wykres zależności załączania dla ConsoleApplication1.cpp:



Definicje

- `#define debug(x) std::cerr << "(" << __LINE__ << ")" << #x << " == " << (x) << std::endl;`

Funkcje

- `int main (int ile, char **params)`

4.1.1 Dokumentacja definicji

4.1.1.1 debug

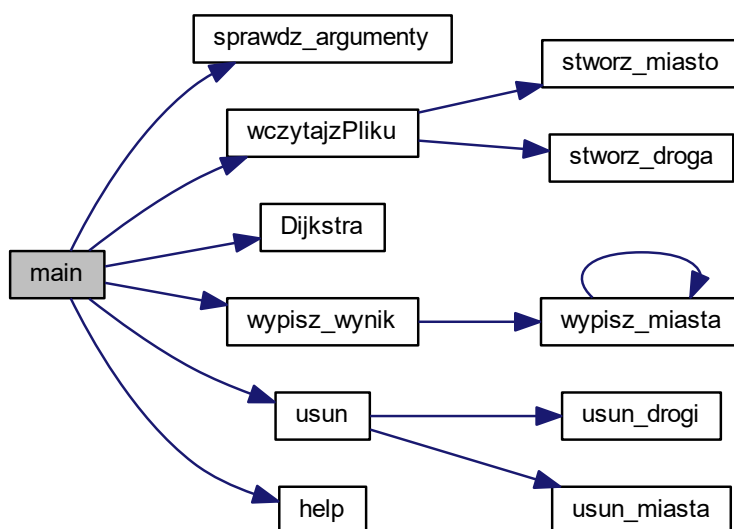
```
#define debug(  
    x ) std::cerr << "(" << __LINE__ << " ) " << #x << " == " << (x) << std::endl;  
::endl;
```

4.1.2 Dokumentacja funkcji

4.1.2.1 main()

```
int main (  
    int ile,  
    char ** params )
```

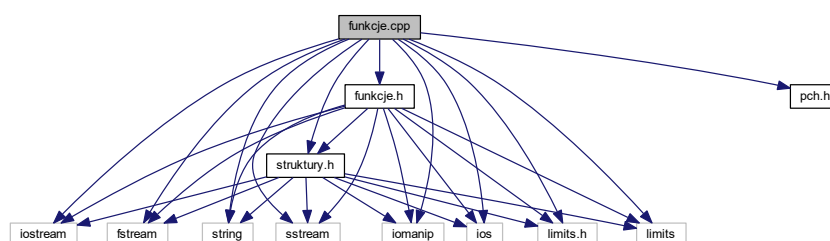
Oto graf wywołań dla tej funkcji:



4.2 Dokumentacja pliku funkcje.cpp

```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <limits>
#include <iomanip>
#include <ios>
#include <limits.h>
#include "pch.h"
#include "struktury.h"
#include "funkcje.h"
```

Wykres zależności załączania dla funkcje.cpp:



Funkcje

- `miasto * stworz_miasto (miasto *pHead_miasto, const std::string &nowanazwa)`
- `void stworz_droga (int kilometry, miasto *&nowe_miasto1, miasto *&nowe_miasto2)`
- `void wypisz_droga (droga *pHead_droga)`
- `void wypisz_miasto (miasto *pHead)`
- `bool Dijkstra (const std::string &startowy, miasto *pHead)`
- `void wypisz_miasta (miasto *pHead, std::ostream &wyjscie)`
- `void wypisz_wynik (miasto *pHead, const std::string &wyjscie)`
- `bool sprawdz_argumenty (int ile, char **params, std::string &wejscie, std::string &wyjscie, std::string &start)`
- `void wczytajzPliku (const std::string &wejscie, miasto *&pGlowa)`
- `void usun_drogi (miasto *pmiasto)`
- `void usun_miasta (miasto *&pHead)`
- `void usun (miasto *glowa_miasta)`
- `void help (int ile, char **params)`

4.2.1 Dokumentacja funkcji

4.2.1.1 Dijkstra()

```
bool Dijkstra (
    const std::string & startowy,
    miasto * pHead )
```

Funkcja, wykorzystująca algorytm Dijkstry do znalezienia najkrótszych dróg z miasta startowego do pozostałych miast

Parametry

<i>startowy</i>	miasto, od ktorego beda rozpoczynac sie wszystkie trasy
<i>pHead</i>	wskaznik na pierwszy element listy

Zwraca

Funkcja zwraca true, gdy miasto startowe bylo w liscie i false, gdy nie bylo w liscie

Oto graf wywoływań tej funkcji:

**4.2.1.2 help()**

```
void help (
    int ile,
    char ** params )
```

Funkcja wyswietla pomoc

Parametry

<i>ile</i>	ilosc argumentow
<i>params</i>	tablica argumentow

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywoływań tej funkcji:



4.2.1.3 sprawdz_argumenty()

```
bool sprawdz_argumenty (
    int ile,
    char ** params,
    std::string & wejscie,
    std::string & wyjscie,
    std::string & start )
```

Funkcja sprawdza argumenty wywołania programu

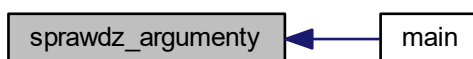
Parametry

	<i>ile</i>	ilosc argumentow
	<i>params</i>	tablica argumentow
out	<i>wejscie</i>	plik wejscowy
out	<i>wyjscie</i>	plik wyjscowy
out	<i>start</i>	miasto startowe

Zwraca

Funkcja zwraca true, gdy podane argumenty byly poprawne i false, gdy byly bledne

Oto graf wywoływań tej funkcji:



4.2.1.4 stworz_droga()

```
void stworz_droga (
    int kilometry,
    miasto *& nowe_miasto1,
    miasto *& nowe_miasto2 )
```

Funkcja dodaje nowe elementy do listy drParametry *kilometry* odlegosc pomiedzy miastami

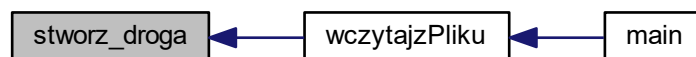
nowe_miasto1 wskaznik na miasto poczatkowe

nowe_miasto2 wskaznik na miasto docelowe

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywoływań tej funkcji:

**4.2.1.5 stworz_miasto()**

```
miasto* stworz_miasto (
    miasto *& pHead,
    const std::string & nowanazwa )
```

Funkcja dodaje nowe miasto do listy

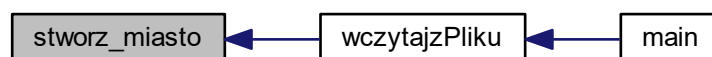
Parametry

<i>pHead</i>	wskaznik na pierwszy element listy
<i>nowanazwa</i>	nazwa miasta dodawanego do listy

Zwraca

Funkcja zwraca wskaznik na nowoutworzone miasto.

Oto graf wywoływań tej funkcji:



4.2.1.6 usun()

```
void usun (
    miasto * glowa_miasta )
```

Funkcja usuwa liste miast i drog

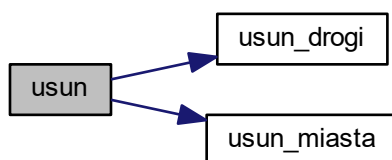
Parametry

<i>glowa_miasta</i>	wskaznik na pierwszy element listy miast
---------------------	--

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.2.1.7 usun_drogi()

```
void usun_drogi (
    miasto * pmiasto )
```

Funkcja usuwa listy drog wszystkich miast

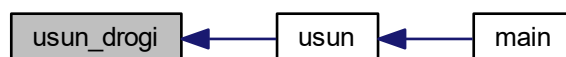
Parametry

<i>pmiasto</i>	wskaznik na kolejne miasto
----------------	----------------------------

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań tej funkcji:

**4.2.1.8 usun_miasta()**

```
void usun_miasta (
    miasto *& pHead )
```

Funkcja usuwa liste miast

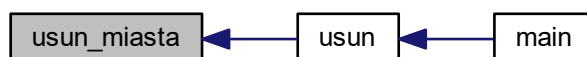
Parametry

<i>in, out</i>	<i>pHead</i>	wskaznik na pierwszy element listy
----------------	--------------	------------------------------------

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań tej funkcji:



4.2.1.9 wczytajzPliku()

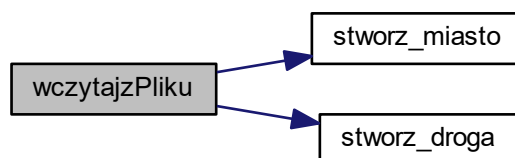
```
void wczytajzPliku (
    const std::string & wejście,
    miasto *& pGlowa )
```

Funkcja wczytuje dane (miasta oraz odleglosci) z pliku

Parametry

	<i>wejście</i>	nazwa pliku wejściowego
<i>in, out</i>	<i>pGlowa</i>	wskaznik na pierwszy element listy

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.2.1.10 wypisz_droga()

```
void wypisz_droga (
    droga * pHead_droga )
```

Funkcja wypisuje liste drog

Parametry

<i>pHead_droga</i>	wskaznik na pierwszy element listy
--------------------	------------------------------------

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań tej funkcji:

**4.2.1.11 wypisz_miasta()**

```
void wypisz_miasta (
    miasto * pHead,
    std::ostream & wyjście )
```

Funkcja wypisuje trase do wskazanego miasta

Parametry

<i>pHead</i>	wskaznik na pierwszy element listy
<i>wyjście</i>	nazwa pliku wyjściowego

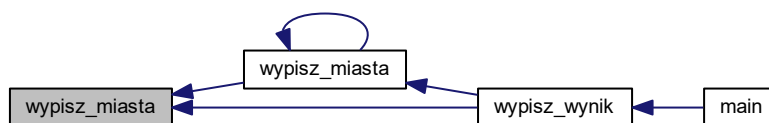
Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.2.1.12 wypisz_miasto()

```
void wypisz_miasto (
    miasto * pHead )
```

Funkcja wypisuje liste miast

Parametry

<i>pHead</i>	wskaznik na pierwszy element listy
--------------	------------------------------------

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań dla tej funkcji:



4.2.1.13 wypisz_wynik()

```
void wypisz_wynik (
    miasto * pHead,
    const std::string & wyjście )
```

Funkcja zapisuje wynik (wszystkie trasy do miast wraz z odleglosciami) do pliku wyjsciowego

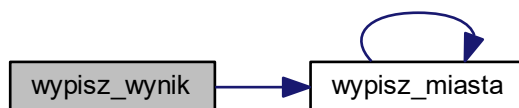
Parametry

<i>pHead</i>	wskaznik na pierwszy element listy
<i>wyjscie</i>	nazwa pliku wyjsciowego

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



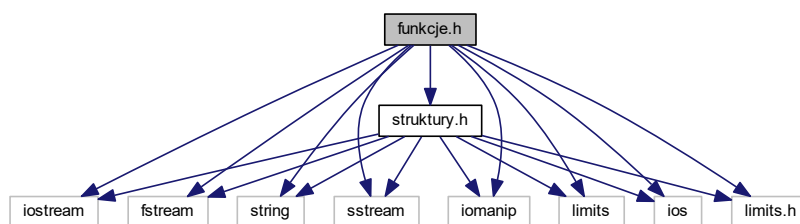
4.3 Dokumentacja pliku funkcje.h

```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <iomanip>
#include <limits>
#include <ios>
#include <limits.h>
```

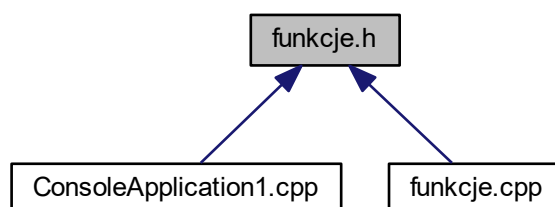


```
#include "struktury.h"
```

Wykres zależności załączania dla funkcje.h:



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



Funkcje

- `miasto * stworz_miasto (miasto *&pHead, const std::string &nowanazwa)`
- `void stworz_droga (int kilometry, miasto *&nowe_miasto1, miasto *&nowe_miasto2)`
- `void wypisz_miasto (miasto *pHead)`
- `void wypisz_droga (droga *pHead_droga)`
- `bool Dijkstra (const std::string &startowy, miasto *pHead)`
- `void wypisz_wynik (miasto *pHead, const std::string &wyjscie)`
- `void wypisz_miasta (miasto *pHead, std::ostream &wyjscie)`
- `bool sprawdz_argumenty (int ile, char **params, std::string &wejscie, std::string &wyjscie, std::string &start)`
- `void wczytajzPliku (const std::string &wejscie, miasto *&pGlowa)`
- `void usun_drogi (miasto *pmiasto)`
- `void usun_miasta (miasto *&pHead)`
- `void usun (miasto *glowa_miasta)`
- `void help (int ile, char **params)`

4.3.1 Dokumentacja funkcji

4.3.1.1 Dijkstra()

```
bool Dijkstra (
    const std::string & startowy,
    miasto * pHead )
```

Funkcja, wykorzystująca algorytm Dijkstry do znalezienia najkrótszych dróg z miasta startowego do pozostałych miast

Parametry

<i>startowy</i>	miasto, od którego będą rozpoczynać się wszystkie trasy
<i>pHead</i>	wskaznik na pierwszy element listy

Zwraca

Funkcja zwraca true, gdy miasto startowe było w liście i false, gdy nie było w liście

Oto graf wywołań tej funkcji:



4.3.1.2 help()

```
void help (
    int ile,
    char ** params )
```

Funkcja wyświetla pomoc

Parametry

<i>ile</i>	ilość argumentów
<i>params</i>	tablica argumentów

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywoływań tej funkcji:



4.3.1.3 sprawdz_argumenty()

```
bool sprawdz_argumenty (
    int ile,
    char ** params,
    std::string & wejscie,
    std::string & wyjscie,
    std::string & start )
```

Funkcja sprawdza argumenty wywołania programu

Parametry

	<i>ile</i>	ilosc argumentow
	<i>params</i>	tablica argumentow
out	<i>wejscie</i>	plik wejscowy
out	<i>wyjscie</i>	plik wyjsciowy
out	<i>start</i>	miasto startowe

Zwraca

Funkcja zwraca true, gdy podane argumenty byly poprawne i false, gdy byly bledne

Oto graf wywoływań tej funkcji:



4.3.1.4 stworz_droga()

```
void stworz_droga (
    int kilometry,
    miasto *& nowe_miasto1,
    miasto *& nowe_miasto2 )
```

Funkcja dodaje nowe elementy do listy drParametry *kilometry* odlegosc pomiedzy miastami

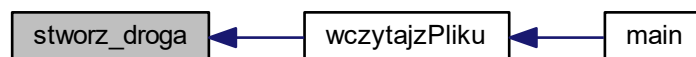
nowe_miasto1 wskaznik na miasto poczatkowe

nowe_miasto2 wskaznik na miasto docelowe

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywoływań tej funkcji:



4.3.1.5 stworz_miasto()

```
miasto* stworz_miasto (
    miasto *& pHead,
    const std::string & nowanazwa )
```

Funkcja dodaje nowe miasto do listy

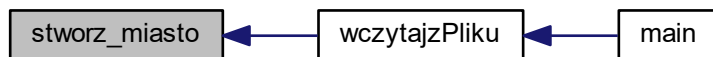
Parametry

<i>pHead</i>	wskaznik na pierwszy element listy
<i>nowanazwa</i>	nazwa miasta dodawanego do listy

Zwraca

Funkcja zwraca wskaźnik na nowoutworzone miasto.

Oto graf wywołań tej funkcji:

**4.3.1.6 usun()**

```
void usun (
    miasto * glowa_miasta )
```

Funkcja usuwa liste miast i drog

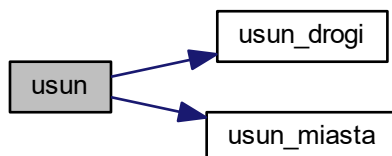
Parametry

<i>glowa_miasta</i>	wskaźnik na pierwszy element listy miast
---------------------	--

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań dla tej funkcji:



Oto graf wywołań tej funkcji:



4.3.1.7 usun_drogi()

```
void usun_drogi (
    miasto * pmiasto )
```

Funkcja usuwa listy dróg wszystkich miast

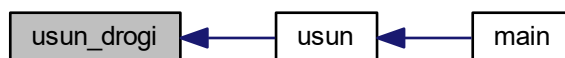
Parametry

<i>pmiasto</i>	wskaznik na kolejne miasto
----------------	----------------------------

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań tej funkcji:



4.3.1.8 usun_miasta()

```
void usun_miasta (
    miasto *& pHead )
```

Funkcja usuwa liste miast

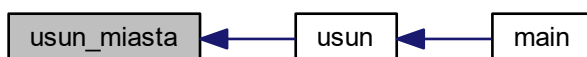
Parametry

in, out	<i>pHead</i>	wskaznik na pierwszy element listy
---------	--------------	------------------------------------

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań tej funkcji:



4.3.1.9 wczytajzPliku()

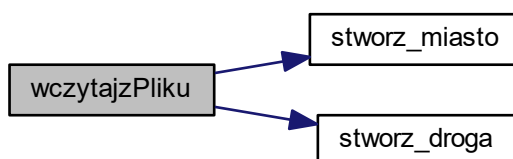
```
void wczytajzPliku (
    const std::string & wejście,
    miasto *& pGlowa )
```

Funkcja wczytuje dane (miasta oraz odleglosci) z pliku

Parametry

	<i>wejście</i>	nazwa pliku wejsciowego
in, out	<i>pGlowa</i>	wskaznik na pierwszy element listy

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.1.10 wypisz_droga()

```
void wypisz_droga (
    droga * pHead_droga )
```

Funkcja wypisuje liste drog

Parametry

<code>pHead_droga</code>	wskaznik na pierwszy element listy
--------------------------	------------------------------------

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywoływań tej funkcji:



4.3.1.11 wypisz_miasta()

```
void wypisz_miasta (
    miasto * pHead,
    std::ostream & wyjście )
```

Funkcja wypisuje trase do wskazanego miasta

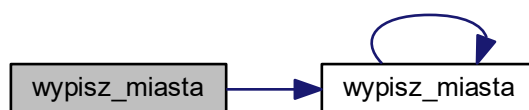
Parametry

<i>pHead</i>	wskaznik na pierwszy element listy
<i>wyjscie</i>	nazwa pliku wyjsciowego

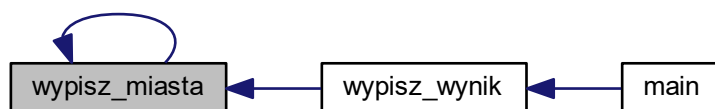
Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



4.3.1.12 wypisz_miasto()

```
void wypisz_miasto (
    miasto * pHead )
```

Funkcja wypisuje liste miast

Parametry

<i>pHead</i>	wskaznik na pierwszy element listy
--------------	------------------------------------

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań dla tej funkcji:

**4.3.1.13 wypisz_wynik()**

```
void wypisz_wynik (
    miasto * pHead,
    const std::string & wyjście )
```

Funkcja zapisuje wynik (wszystkie trasy do miast wraz z odleglosciami) do pliku wyjsciowego

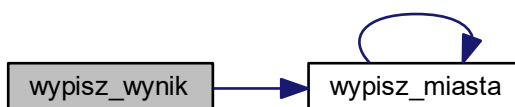
Parametry

<i>pHead</i>	wskaznik na pierwszy element listy
<i>wyjście</i>	nazwa pliku wyjsciowego

Zwraca

Funkcja nie zwraca niczego.

Oto graf wywołań dla tej funkcji:



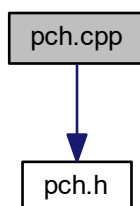
Oto graf wywoływań tej funkcji:



4.4 Dokumentacja pliku pch.cpp

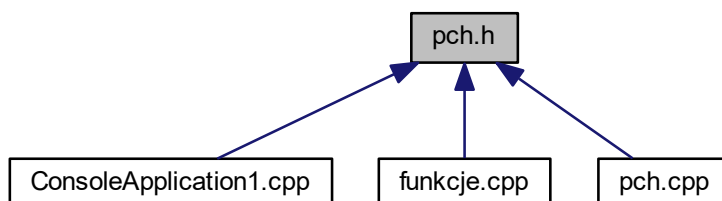
```
#include "pch.h"
```

Wykres zależności załączania dla pch.cpp:



4.5 Dokumentacja pliku pch.h

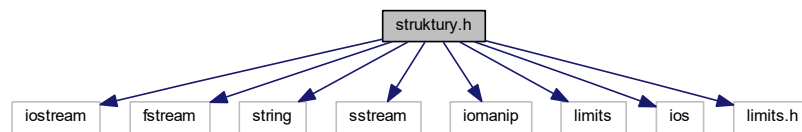
Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



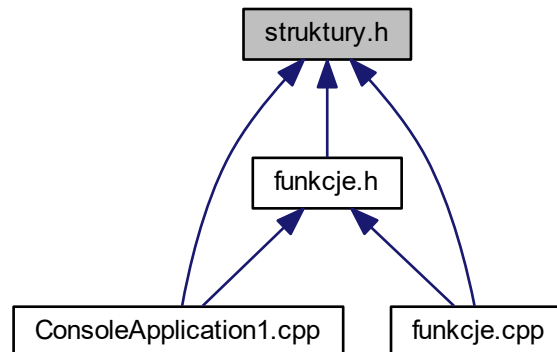
4.6 Dokumentacja pliku struktury.h

```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <iomanip>
#include <limits>
#include <ios>
#include <limits.h>
```

Wykres zależności załączania dla struktury.h:



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



Komponenty

- struct [miasto](#)
- struct [droga](#)