ADT

0.1

Generated by Doxygen 1.8.6

Sun May 15 2016 22:15:20

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

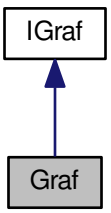
Chapter 4

Class Documentation

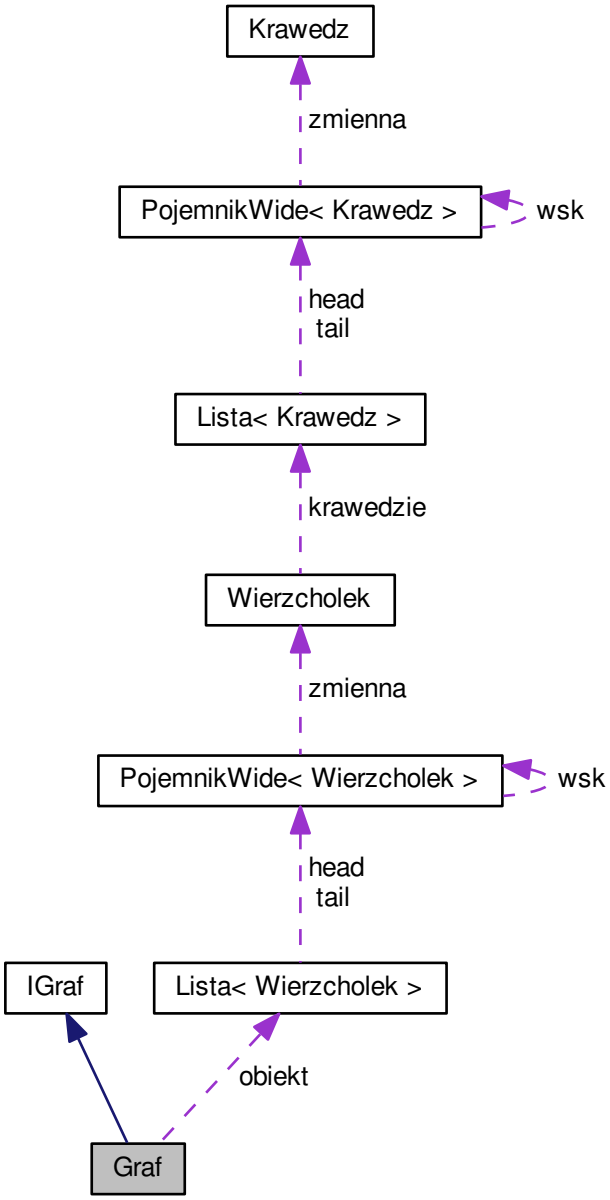
4.1 Graf Class Reference

`#include <Graf.hh>`

Inheritance diagram for Graf:



Collaboration diagram for Graf:



Public Member Functions

- ~Graf ()
- void DodajW (int pozycja)

  Dodawanie wierzcholka do grafu.
- bool DodajK (int poczatek, int koniec, int waga=1)

  Dodawanie krawedzi do grafu.
- bool UsunK (int poczatek, int koniec)
- bool UsunW (int pozycja)

- Wierzcholek & WezW (int indeks)
- int WyszukajW (int pozycja)
- void Wyswietl ()

Private Attributes

- Lista< Wierzcholek > obiekt

4.1.1 Detailed Description

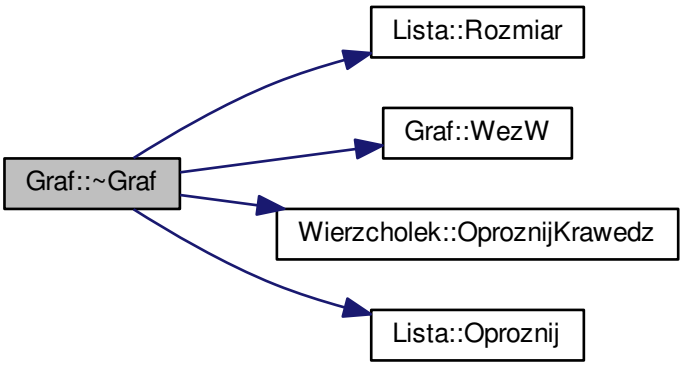
Definition at line 16 of file Graf.hh.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Graf::~Graf ( )

Definition at line 119 of file Graf.cpp.

Here is the call graph for this function:



4.1.3 Member Function Documentation

4.1.3.1 bool Graf::DodajK (int *poczatek*, int *koniec*, int *waga* = 1) [virtual]

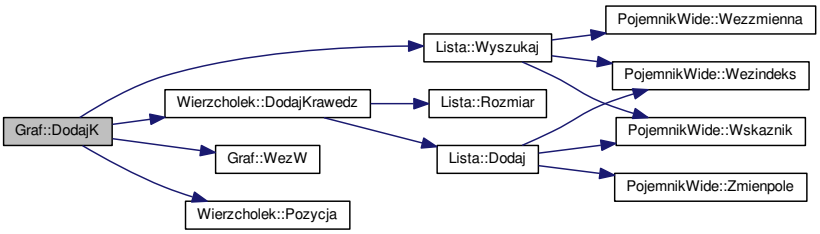
Parameters

in	*poczatek*	jeden z dwoch wierzcholkow, ktore laczy krawedz
in	*poczatek*	jeden z dwoch wierzcholkow, ktore laczy krawedz
in	*waga*	waga krawedzi

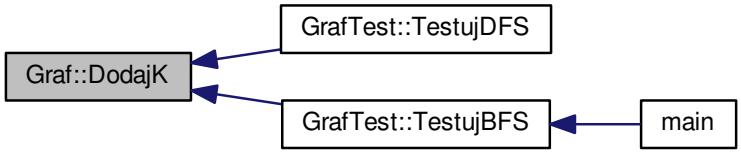
Implements IGraf.

Definition at line 39 of file Graf.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.3.2 void Graf::DodajW (int *pozycja*) [virtual]

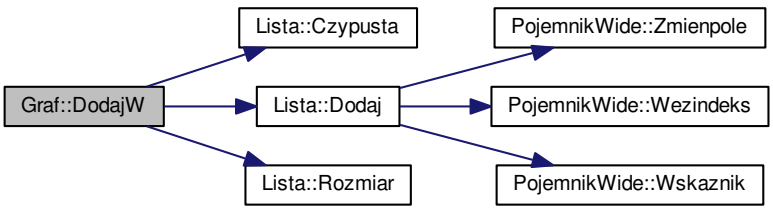
Parameters

in	*pozycja*	okresla numer identyfikujacy dany wierzcholek

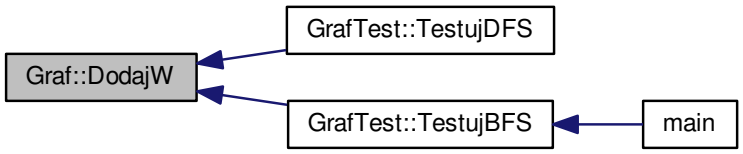
Implements IGraf.

Definition at line 9 of file Graf.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



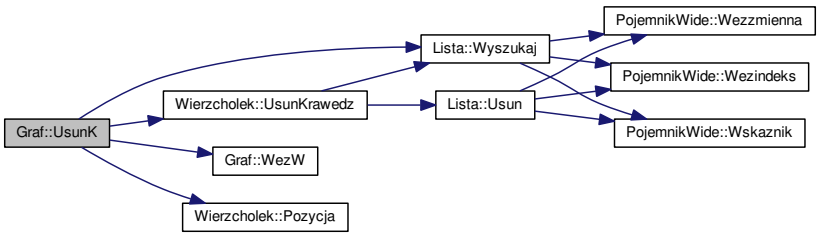
4.1.3.3 bool Graf::UsunK (int *poczatek*, int *koniec*) [virtual]

Usuwanie krawedzi z grafu

Implements IGraf.

Definition at line 63 of file Graf.cpp.

Here is the call graph for this function:



4.1.3.4 bool Graf::UsunW (int *pozycja*) [virtual]

Usuwa podany wierzcholek i przylegajace do niego krawedzie

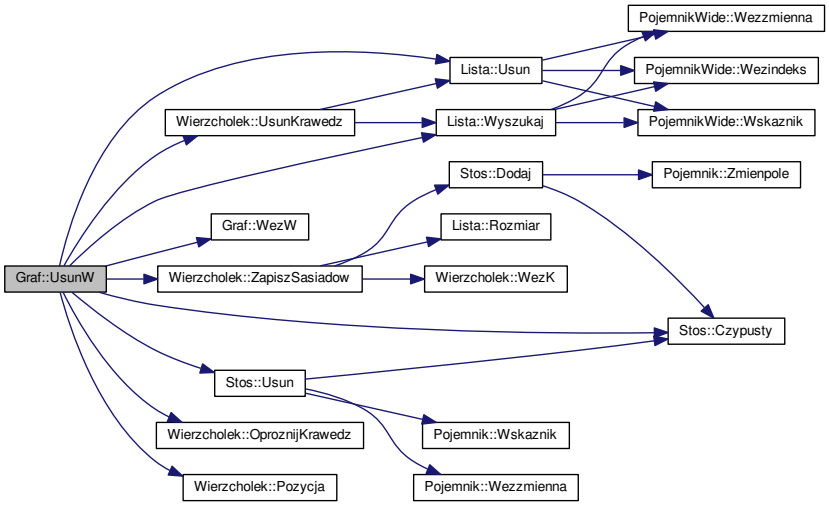
Parameters

in	*pozycja*	identyfikator wierzcholka do usuniecia

Implements IGraf.

Definition at line 89 of file Graf.cpp.

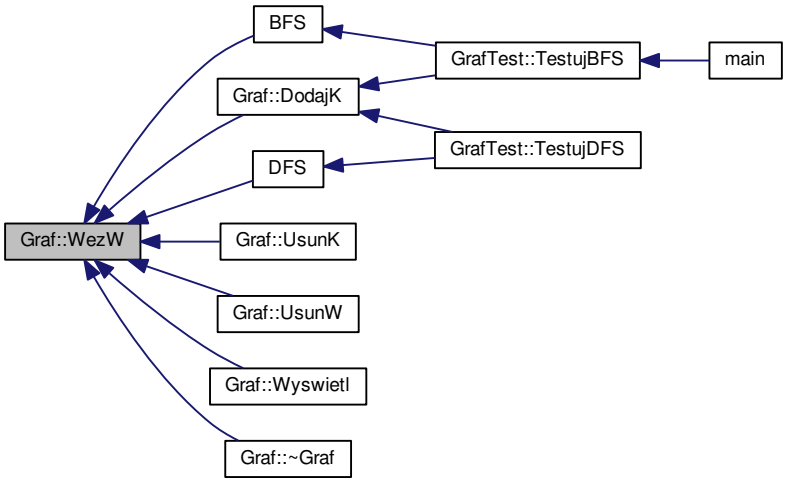
Here is the call graph for this function:



4.1.3.5 Wierzcholek& Graf::WezW (int *indeks*) [inline]

Definition at line 26 of file Graf.hh.

Here is the caller graph for this function:

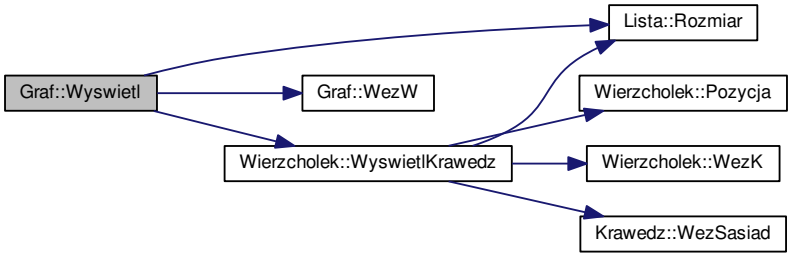


4.1.3.6 void Graf::Wyswietl ( )

Wyswietla Wszystkie wierzcholki i przylegajace do nich krawedzie

Definition at line 113 of file Graf.cpp.

Here is the call graph for this function:



4.1.3.7 int Graf::WyszukajW (int *pozycja*)

Wyszukuje dany wierzcholek grafu na podstawie jego identyfikatora

Parameters

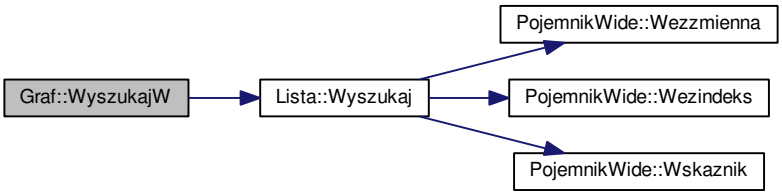
in	*pozycja*	identyfikator wierzcholka

Return values

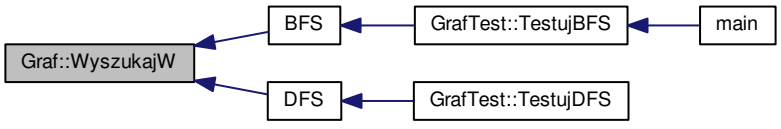
indeks	indeks, pod ktorym przechowywany jest dany wierzcholek w grafie

Definition at line 133 of file Graf.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.4 Member Data Documentation

4.1.4.1 Lista<Wierzcholek> Graf::obiekt [private]

Definition at line 17 of file Graf.hh.

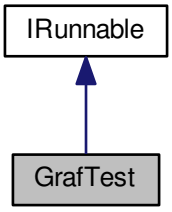
The documentation for this class was generated from the following files:

- Graf.hh
- Graf.cpp

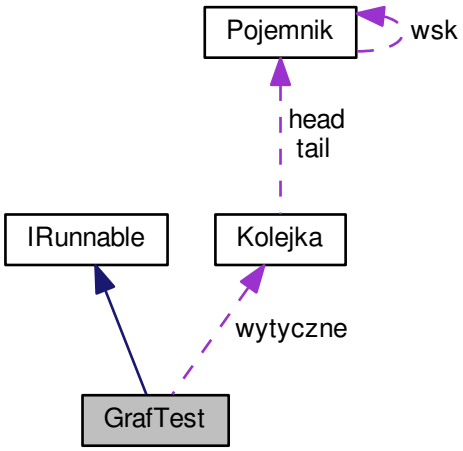
4.2 GrafTest Class Reference

`#include <GrafTest.hh>`

Inheritance diagram for GrafTest:



Collaboration diagram for GrafTest:



Public Member Functions

- bool Przygotuj (string nazwapliku)
- void TestujDFS ()

  Testowanie algorytmu DFS.
- void TestujBFS ()

  Testowanie algorytmu BFS.

Private Attributes

- Kolejka wytyczne

4.2.1 Detailed Description

Definition at line 19 of file GrafTest.hh.

4.2.2 Member Function Documentation

4.2.2.1 bool GrafTest::Przygotuj (string *nazwapliku*) [virtual]

Odczytuje jaka ilosc wierzcholkow ma byc zapisana w grafie i skladuje odczytane wartosci na stosie. Wartosci odczytywane sa z pliku. Przykladowa zawartosc pliku: 10 100 1000

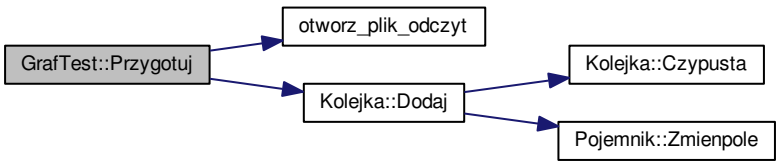
Parameters

in	*nazwapliku-*	nazwa pliku, w ktorum przechowywane sa dane na temat ilosci wierzcholkow do zapisania w grafie

Implements IRunnable.

Definition at line 38 of file GrafTest.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

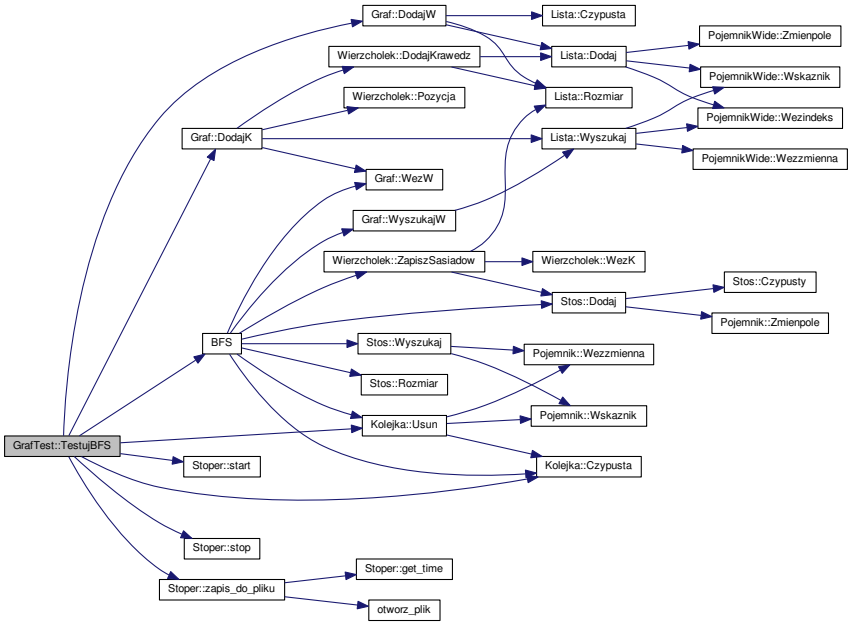
4.2.2.2 void GrafTest::TestujBFS ( ) [virtual]

Zapisuje odpowiednia ilosc elementow w grafie (odpowiednia-przechowywana w kolejce "wytyczne". Krawedzi jest 2x wiecej niz wierzcholkow, aby zwiekszyc prawdopodobienstwo spojnosci grafu), wywoluje algorytm BFS dla stworzonego grafu, mierzy czas jego dzialania i zapisuje go do pliku "czasy.dat"

Implements IRunnable.

Definition at line 97 of file GrafTest.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

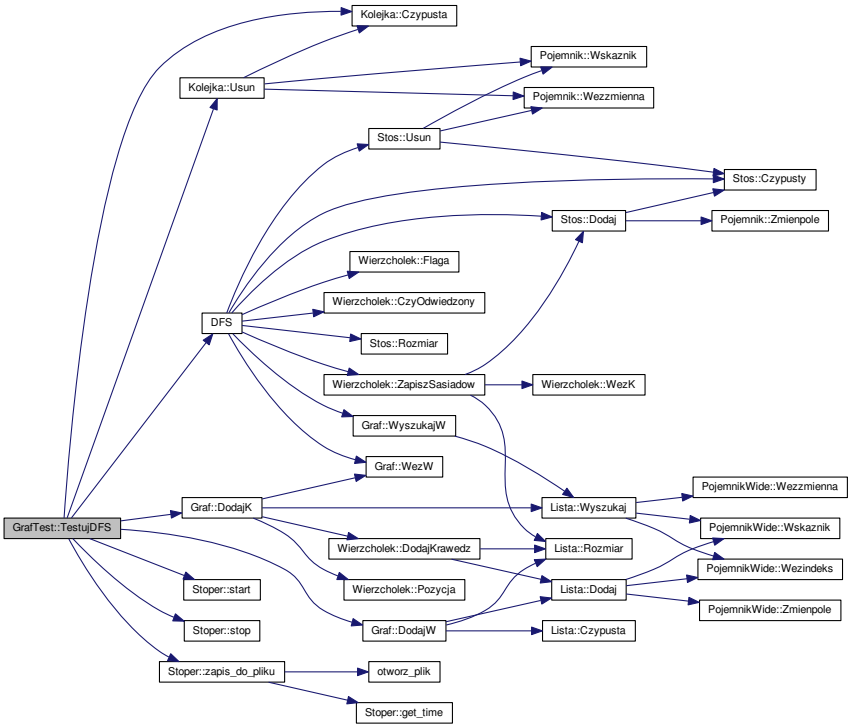
4.2.2.3 void GrafTest::TestujDFS ( ) [virtual]

Zapisuje odpowiednia ilosc elementow w grafie (odpowiednia-przechowywana w kolejce "wytyczne". Krawedzi jest 2x wiecej niz wierzcholkow, aby zwiekszyc prawdopodobienstwo spojnosci grafu), wywoluje algorytm DFS dla stworzonego grafu, mierzy czas jego dzialania i zapisuje go do pliku "czasy.dat"

Implements IRunnable.

Definition at line 60 of file GrafTest.cpp.

Here is the call graph for this function:



4.2.3 Member Data Documentation

4.2.3.1 Kolejka GrafTest::wytyczne [private]

Definition at line 20 of file GrafTest.hh.

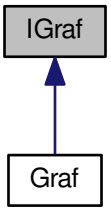
The documentation for this class was generated from the following files:

- GrafTest.hh
- GrafTest.cpp

4.3 IGraf Class Reference

```
#include <IGraf.hh>
```

Inheritance diagram for IGraf:



Public Member Functions

- virtual void DodajW (int pozycja)=0
- virtual bool DodajK (int poczatek, int koniec, int waga=1)=0
- virtual bool UsunK (int poczatek, int koniec)=0
- virtual bool UsunW (int pozycja)=0

4.3.1 Detailed Description

Definition at line 13 of file IGraf.hh.

4.3.2 Member Function Documentation

4.3.2.1 virtual bool IGraf::DodajK (int *poczatek*, int *koniec*, int *waga* = 1) [pure virtual]

Implemented in Graf.

4.3.2.2 virtual void IGraf::DodajW (int *pozycja*) [pure virtual]

Implemented in Graf.

4.3.2.3 virtual bool IGraf::UsunK (int *poczatek*, int *koniec*) [pure virtual]

Implemented in Graf.

4.3.2.4 virtual bool IGraf::UsunW (int *pozycja*) [pure virtual]

Implemented in Graf.

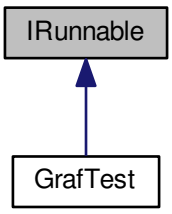
The documentation for this class was generated from the following file:

- IGraf.hh

4.4 IRunnable Class Reference

`#include <IRunnable.hh>`

Inheritance diagram for IRunnable:



Public Member Functions

- virtual bool Przygotuj (string nazwapliku)=0
- virtual void TestujDFS ()=0
- virtual void TestujBFS ()=0

4.4.1 Detailed Description

Definition at line 11 of file IRunnable.hh.

4.4.2 Member Function Documentation

4.4.2.1 virtual bool IRunnable::Przygotuj (string *nazwapliku*) [pure virtual]

Implemented in GrafTest.

4.4.2.2 virtual void IRunnable::TestujBFS ( ) [pure virtual]

Implemented in GrafTest.

4.4.2.3 virtual void IRunnable::TestujDFS ( ) [pure virtual]

Implemented in GrafTest.

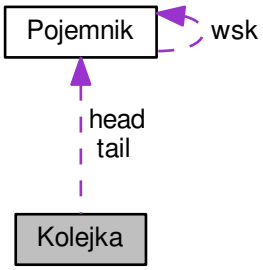
The documentation for this class was generated from the following file:

- IRunnable.hh

4.5 Kolejka Class Reference

`#include <Kolejka.hh>`

Collaboration diagram for Kolejka:



Public Member Functions

- ~Kolejka ()
- void Dodaj (int elem)
- int Usun ()
- int Wez ()
- int Rozmiar ()
- bool Czypusta ()
- void Oproznij ()
- void Wyswietl ()

Private Attributes

- Pojemnik * head =NULL
- Pojemnik * tail =NULL
- int rozmiar =0

4.5.1 Detailed Description

Definition at line 13 of file Kolejka.hh.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 Kolejka::~Kolejka ( ) [inline]

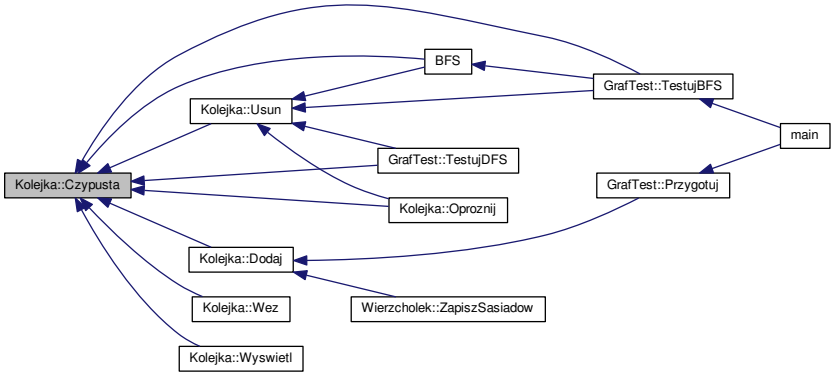
Definition at line 18 of file Kolejka.hh.

4.5.3 Member Function Documentation

4.5.3.1 bool Kolejka::Czypusta ( ) [inline]

Definition at line 23 of file Kolejka.hh.

Here is the caller graph for this function:



4.5.3.2 void Kolejka::Dodaj (int *elem*)

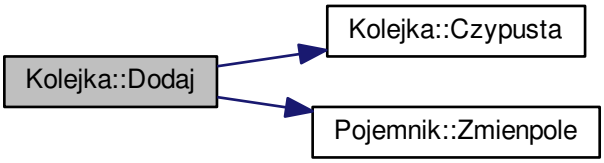
Dodaje element na koncu kolejki

Parameters

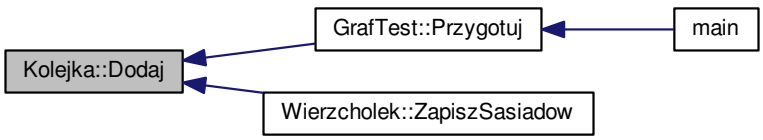
in	*elem*-	zmienna do przechowania

Definition at line 13 of file Kolejka.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

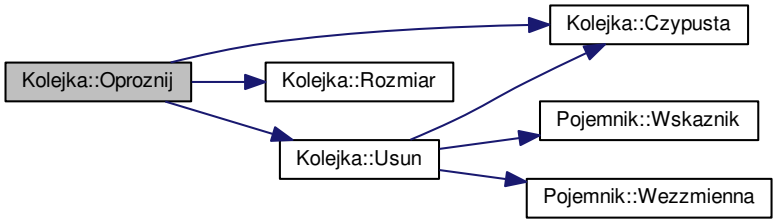


4.5.3.3 void Kolejka::Oproznij ( )

Usuwa wszystkie elementy kolejki

Definition at line 73 of file Kolejka.cpp.

Here is the call graph for this function:



4.5.3.4 int Kolejka::Rozmiar ( ) [inline]

Definition at line 22 of file Kolejka.hh.

Here is the caller graph for this function:

4.5.3.5 int Kolejka::Usun ( )

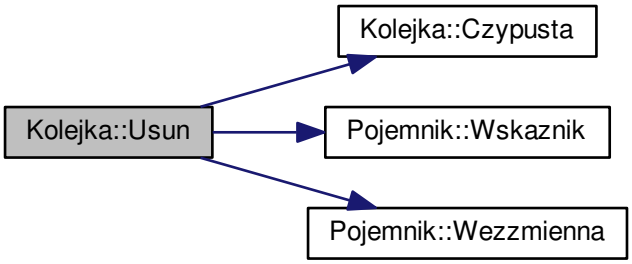
Usuwa element z poczatku kolejki i zwraca jego wartosc

Return values

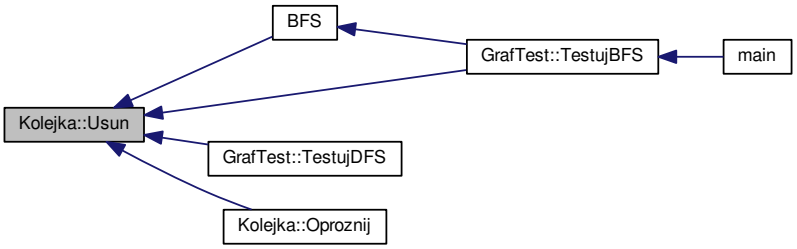
wartosc	usunietego elementu

Definition at line 37 of file Kolejka.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

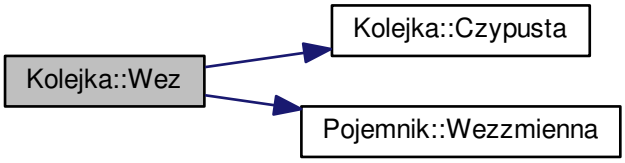


4.5.3.6 int Kolejka::Wez ( )

Zwraca wartosc pierwszego elementu w kolejce. Funkcja NIE sluzy do modyfikowania wartosci tego elementu

Definition at line 59 of file Kolejka.cpp.

Here is the call graph for this function:

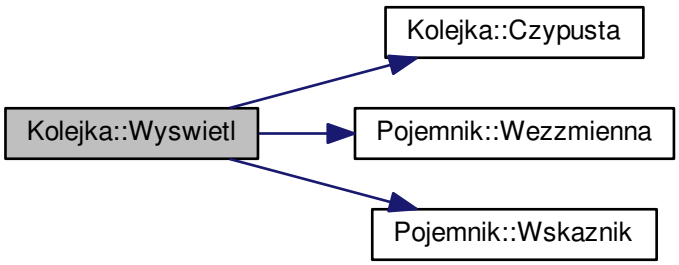


4.5.3.7 void Kolejka::Wyswietl ( )

Wyswietla wszystkie elementy kolejki od pierwszego do ostatniego

Definition at line 86 of file Kolejka.cpp.

Here is the call graph for this function:



4.5.4 Member Data Documentation

4.5.4.1 Pojemnik* Kolejka::head =NULL [private]

Definition at line 14 of file Kolejka.hh.

4.5.4.2 int Kolejka::rozmiar =0 [private]

Definition at line 16 of file Kolejka.hh.

4.5.4.3 Pojemnik* Kolejka::tail =NULL [private]

Definition at line 15 of file Kolejka.hh.

The documentation for this class was generated from the following files:

- Kolejka.hh
- Kolejka.cpp

4.6 Krawedz Class Reference

`#include <Krawedz.hh>`

Public Member Functions

- Krawedz ()
- Krawedz (int psasiad, int pwaga=1)
- int WezSasiad ()
- int WezWaga ()
- bool operator== (Krawedz druga)

Private Attributes

- int sasiad =0
- int waga =1

4.6.1 Detailed Description

Definition at line 12 of file Krawedz.hh.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Krawedz::Krawedz ( ) [inline]

Definition at line 18 of file Krawedz.hh.

4.6.2.2 Krawedz::Krawedz (int *psasiad*, int *pwaga* = 1) [inline]

Definition at line 19 of file Krawedz.hh.

4.6.3 Member Function Documentation

4.6.3.1 bool Krawedz::operator== (Krawedz *druga*)

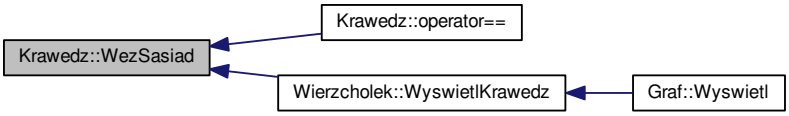
Definition at line 3 of file Krawedz.cpp.

Here is the call graph for this function:

4.6.3.2 int Krawedz::WezSasiad ( ) [inline]

Definition at line 20 of file Krawedz.hh.

Here is the caller graph for this function:



4.6.3.3 int Krawedz::WezWaga ( ) [inline]

Definition at line 21 of file Krawedz.hh.

4.6.4 Member Data Documentation

4.6.4.1 int Krawedz::sasiad =0 [private]

Definition at line 13 of file Krawedz.hh.

4.6.4.2 int Krawedz::waga =1 [private]

Definition at line 14 of file Krawedz.hh.

The documentation for this class was generated from the following files:

- Krawedz.hh
- Krawedz.cpp

4.7 Lista< typ > Class Template Reference

`#include <Lista.hh>`

Public Member Functions

- bool Dodaj (typ elem, int ind)
- typ Usun (int ind)
- typ & Wez (int ind)
- int Rozmiar ()
- bool Czypusta ()
- void Oproznij ()
- void Wyswietl ()
- int Wyszukaj (typ szukane)

Private Attributes

- PojemnikWide< typ > * head =NULL
- PojemnikWide< typ > * tail =NULL

4.7.1 Detailed Description

template<typename typ>class Lista< typ >

Definition at line 18 of file Lista.hh.

4.7.2 Member Function Documentation

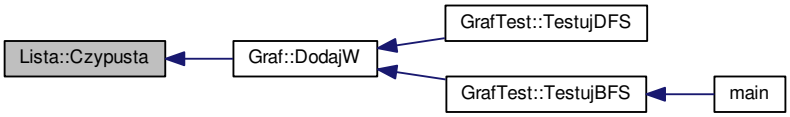
4.7.2.1 template<typename typ> bool Lista< typ >::Czypusta ( ) [inline]

Return values

true-	gdy lista jest pusta
false-	w przypadku przeciwnym

Definition at line 32 of file Lista.hh.

Here is the caller graph for this function:



4.7.2.2 template<typename typ> bool Lista< typ >::Dodaj (typ *elem,* int *ind*)

Funkcja przypisuje wartosc do przechowania elementowi typu "Pojemnik" i dodaje ten "Pojeminik" w DOWOLNYM miejscu listy czyli na koncu, poczatku badz wewnatrz listy

Parameters

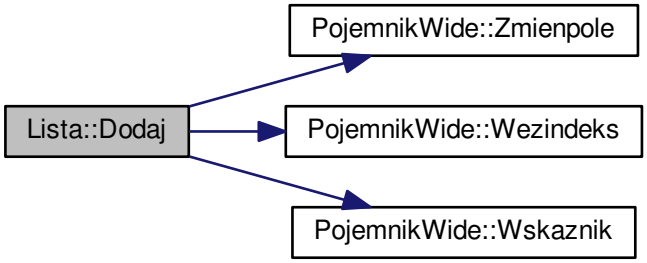
in	*elem-*	wartosc do przechowania
in	*index-*	indeks listy pod jakim bedzie przechowywany pojemnik ze zmienna

Return values

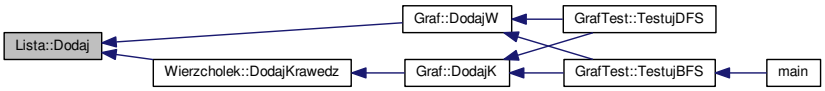
false-	gdy element ma byc wstawiony w nielogicznym miejscu, np-> wstawianie elementu o indeksie 100 kiedy lista ma aktualnie indeksy od 0 do 15
true-	gdy element wstawiono poprawnie do listy

Definition at line 53 of file Lista.hh.

Here is the call graph for this function:



Here is the caller graph for this function:

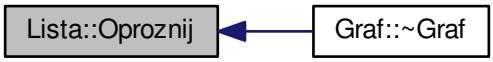


4.7.2.3 template<typename typ > void Lista< typ >::Oproznij ( )

Usuwa wszystkie elementy z listy

Definition at line 231 of file Lista.hh.

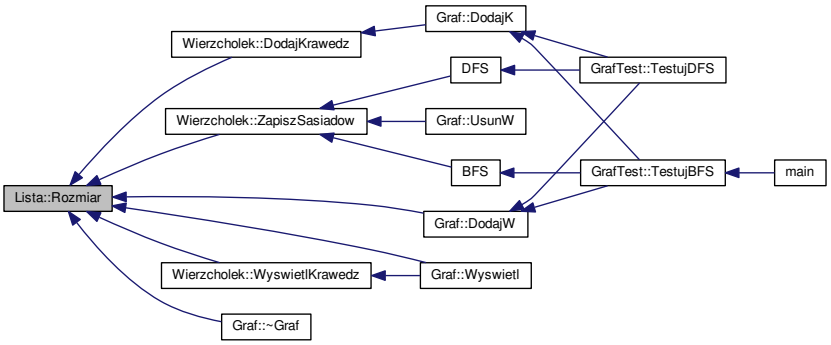
Here is the caller graph for this function:



4.7.2.4 template<typename typ> int Lista< typ >::Rozmiar ( ) [inline]

Definition at line 27 of file Lista.hh.

Here is the caller graph for this function:



4.7.2.5 template<typename typ > typ Lista< typ >::Usun (int *ind*)

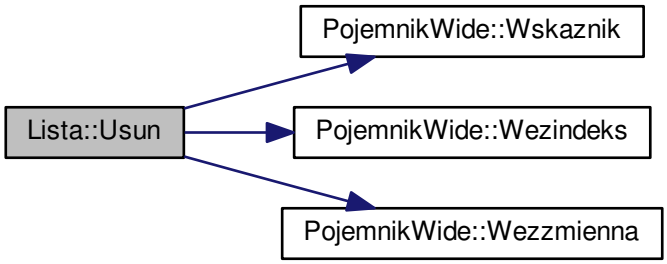
Usuwa element z Listy o zadanym indeksie i zwraca wartosc, ktora przechowywal

Parameters

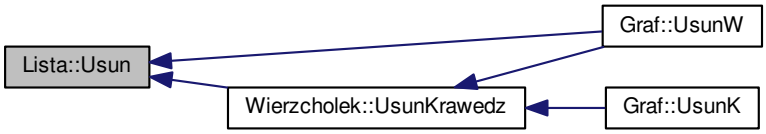
in	*ind-*	indeks elementu, ktory ma zostac usuniety z listy

Definition at line 146 of file Lista.hh.

Here is the call graph for this function:



Here is the caller graph for this function:



4.7.2.6 template<typename typ > typ & Lista< typ >::Wez (int *ind*)

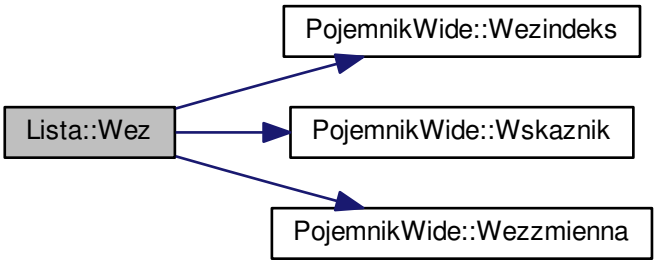
Zwraca wartość elementu o zadanym indeksie

Parameters

in	*ind-*	indeks poszukiwanego elementu

Definition at line 118 of file Lista.hh.

Here is the call graph for this function:

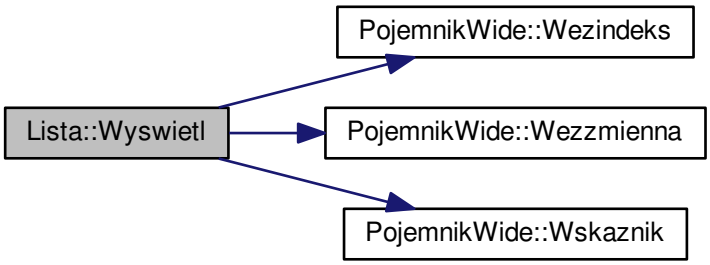


4.7.2.7 template<typename typ > void Lista< typ >::Wyswietl ( )

Wyswietla zawartosc listy na standardowe wyjscie

Definition at line 247 of file Lista.hh.

Here is the call graph for this function:



4.7.2.8 template<typename typ> int Lista< typ >::Wyszukaj (typ *szukane*)

Wyszukuje podany wyraz wsrod elementow listy

Parameters

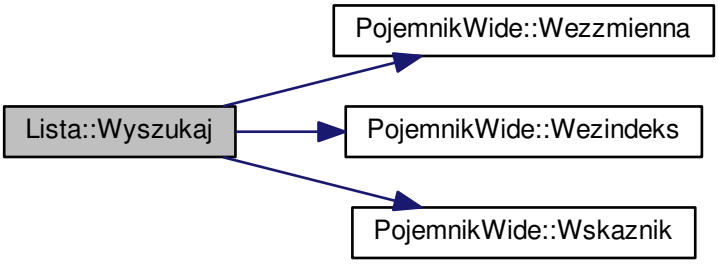
in	*szukane-*	szukany wyraz

Return values

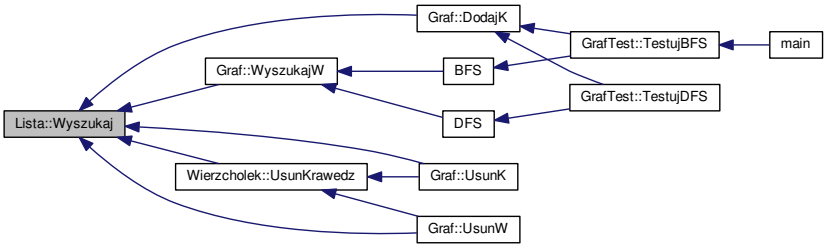
zwraca	numer indeksu elementu, ktory przechowuje szukany wyraz lub -1 w przypadku jego nieznalezienia

Definition at line 274 of file Lista.hh.

Here is the call graph for this function:



Here is the caller graph for this function:



4.7.3 Member Data Documentation

4.7.3.1 template<typename typ> PojemnikWide<typ>* Lista< typ >::head =NULL [private]

Definition at line 19 of file Lista.hh.

4.7.3.2 template<typename typ> PojemnikWide<typ>* Lista< typ >::tail =NULL [private]

Definition at line 20 of file Lista.hh.

The documentation for this class was generated from the following file:

- Lista.hh

4.8 Pojemnik Class Reference

```
#include <Pojemnik.hh>
```

Collaboration diagram for Pojemnik:

Public Member Functions

- void Zmienpole (double pom)
- int Wezzmienna ()
- Pojemnik ∗ Wskaznik ()

Public Attributes

- Pojemnik ∗ wsk =NULL

Private Attributes

- int zmienna =0

4.8.1 Detailed Description

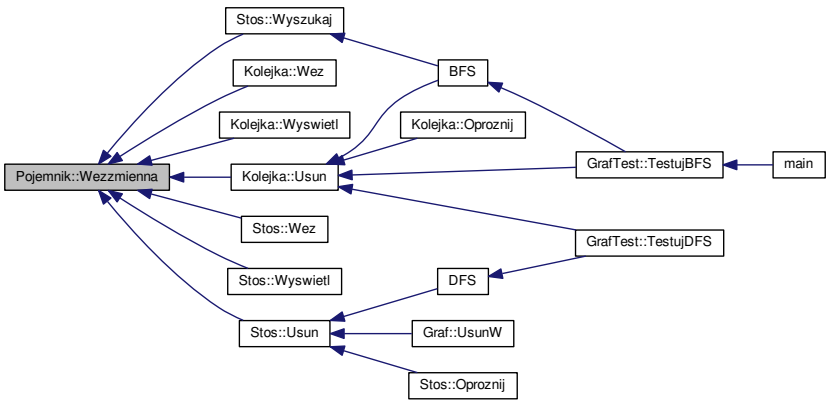
Definition at line 12 of file Pojemnik.hh.

4.8.2 Member Function Documentation

4.8.2.1 int Pojemnik::Wezzmienna ( ) [inline]

Definition at line 18 of file Pojemnik.hh.

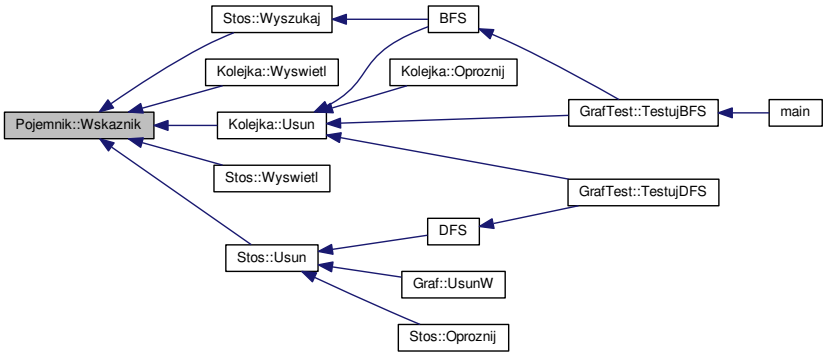
Here is the caller graph for this function:



4.8.2.2 Pojemnik* Pojemnik::Wskaznik ( ) [inline]

Definition at line 19 of file Pojemnik.hh.

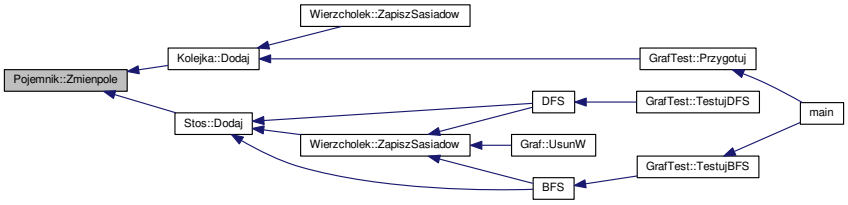
Here is the caller graph for this function:



4.8.2.3 void Pojemnik::Zmienpole (double *pom*) [inline]

Definition at line 17 of file Pojemnik.hh.

Here is the caller graph for this function:



4.8.3 Member Data Documentation

4.8.3.1 Pojemnik* Pojemnik::wsk =NULL

Definition at line 15 of file Pojemnik.hh.

4.8.3.2 int Pojemnik::zmienna =0 [private]

Definition at line 13 of file Pojemnik.hh.

The documentation for this class was generated from the following file:

- Pojemnik.hh

4.9 PojemnikWide< typ > Class Template Reference

`#include <PojemnikWide.hh>`

Public Member Functions

- void Zmienpole (typ pom)
- int & Wezindeks ()
- typ & Wezzmienna ()
- PojemnikWide< typ > * Wskaznik ()

Public Attributes

- PojemnikWide< typ > * wsk =NULL

Private Attributes

- typ zmienna
- int indeks =0

4.9.1 Detailed Description

template<typename typ>class PojemnikWide< typ >

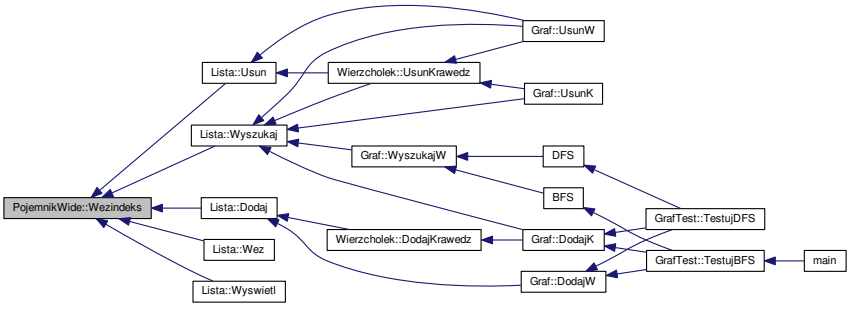
Definition at line 13 of file PojemnikWide.hh.

4.9.2 Member Function Documentation

4.9.2.1 template<typename typ> int& PojemnikWide< typ >::Wezindeks ( ) [inline]

Definition at line 20 of file PojemnikWide.hh.

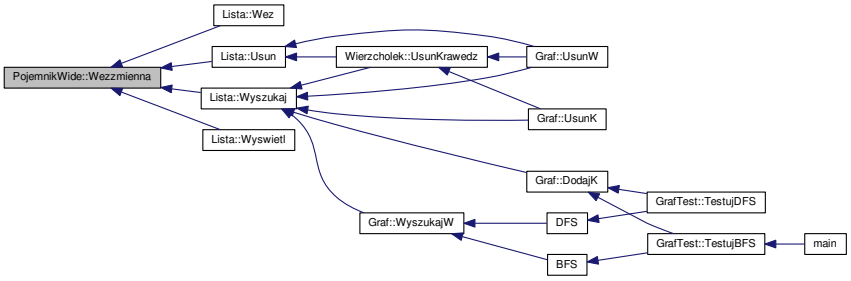
Here is the caller graph for this function:



4.9.2.2 template<typename typ> typ& PojemnikWide< typ >::Wezzmienna ( ) [inline]

Definition at line 21 of file PojemnikWide.hh.

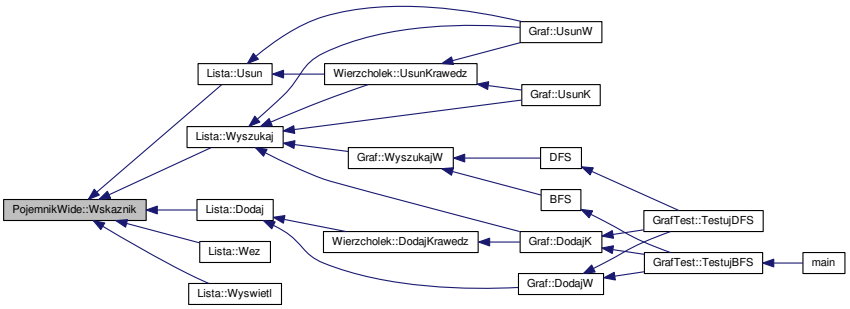
Here is the caller graph for this function:



4.9.2.3 template<typename typ> PojemnikWide<typ>* PojemnikWide< typ >::Wskaznik ( ) [inline]

Definition at line 22 of file PojemnikWide.hh.

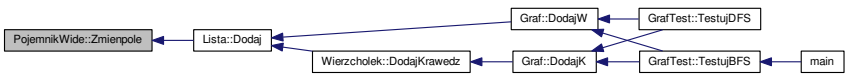
Here is the caller graph for this function:



4.9.2.4 template<typename typ> void PojemnikWide< typ >::Zmienpole (typ *pom*) [inline]

Definition at line 19 of file PojemnikWide.hh.

Here is the caller graph for this function:



4.9.3 Member Data Documentation

4.9.3.1 template<typename typ> int PojemnikWide< typ >::indeks =0 [private]

Definition at line 15 of file PojemnikWide.hh.

4.9.3.2 template<typename typ> PojemnikWide<typ>* PojemnikWide< typ >::wsk =NULL

Definition at line 17 of file PojemnikWide.hh.

4.9.3.3 template<typename typ> typ PojemnikWide< typ >::zmienna [private]

Definition at line 14 of file PojemnikWide.hh.

The documentation for this class was generated from the following file:

- PojemnikWide.hh

4.10 Stoper Class Reference

`#include <Stoper.hh>`

Public Member Functions

- void start ()
- void stop ()
- double get_time ()

  roznica czasowa
- bool zapis_do_pliku ()

  Zapis zmierzonego czasu do pliku.

Private Attributes

- timeval czas1
- timeval czas2

4.10.1 Detailed Description

Definition at line 11 of file Stoper.hh.

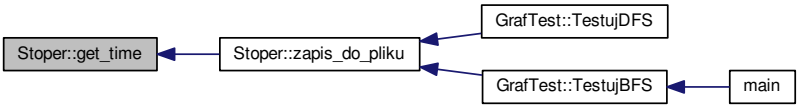
4.10.2 Member Function Documentation

4.10.2.1 double Stoper::get_time ( )

Zwraca roznice czasu miedzy "startem a "stopem". Wartosci wyrazone w mikrosekundach

Definition at line 9 of file Stoper.cpp.

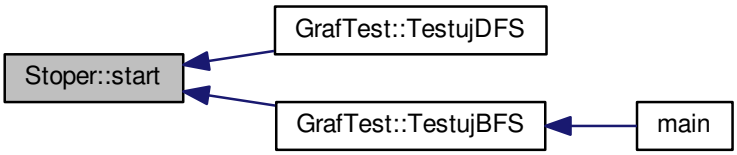
Here is the caller graph for this function:



4.10.2.2 void Stoper::start ( ) [inline]

Definition at line 16 of file Stoper.hh.

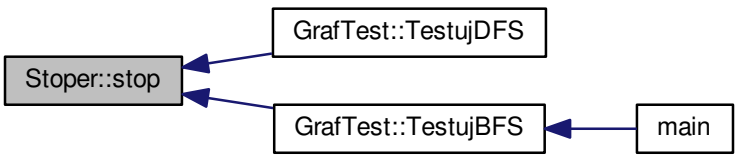
Here is the caller graph for this function:



4.10.2.3 void Stoper::stop ( ) [inline]

Definition at line 17 of file Stoper.hh.

Here is the caller graph for this function:

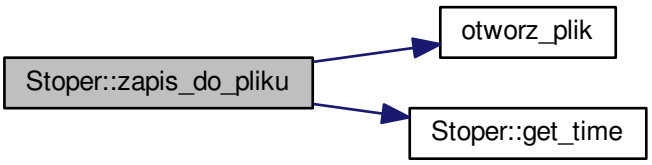


4.10.2.4 bool Stoper::zapis_do_pliku ( )

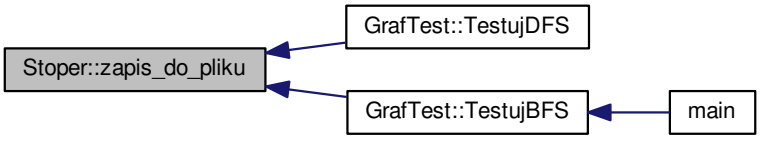
Wywolanie tej funkcji skutkuje dopisaniem do pliku "czasy.dat" ostatniej roznicy czasowej ("czas_stop"-"czas_start)
Wartosci wyrazone w sekundach

Definition at line 43 of file Stoper.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.10.3 Member Data Documentation

4.10.3.1 timeval Stoper::czas1 [private]

Definition at line 12 of file Stoper.hh.

4.10.3.2 timeval Stoper::czas2 [private]

Definition at line 13 of file Stoper.hh.

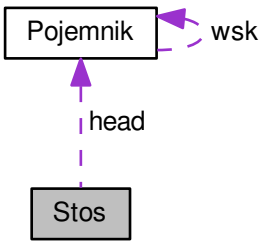
The documentation for this class was generated from the following files:

- Stoper.hh
- Stoper.cpp

4.11 Stos Class Reference

`#include <Stos.hh>`

Collaboration diagram for Stos:



Public Member Functions

- ~Stos ()
- void Dodaj (double elem)
- int Usun ()
- int Wez ()
- bool Czypusty ()
- int Rozmiar ()
- void Oproznij ()
- void Wyswietl ()
- bool Wyszukaj (int szukane)

Private Attributes

- Pojemnik * head =NULL
- int rozmiar =0

4.11.1 Detailed Description

Definition at line 13 of file Stos.hh.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 Stos::~Stos ( ) [inline]

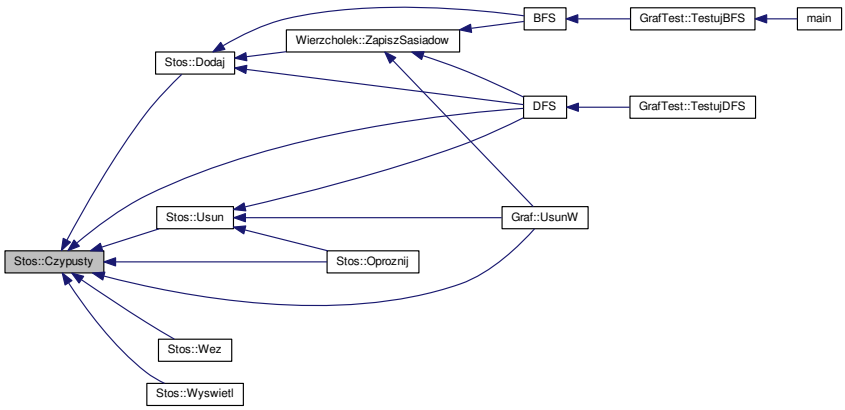
Definition at line 17 of file Stos.hh.

4.11.3 Member Function Documentation

4.11.3.1 bool Stos::Czypusty ( ) [inline]

Definition at line 21 of file Stos.hh.

Here is the caller graph for this function:



4.11.3.2 void Stos::Dodaj (double *elem*)

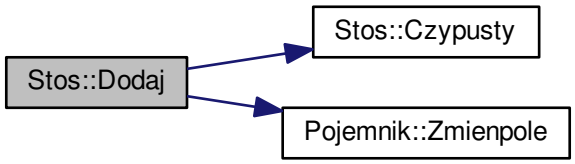
"Kladzie" element na Stos

Parameters

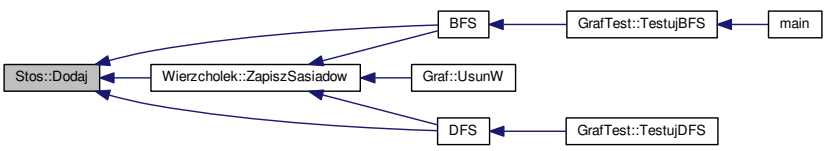
in	*elem*-	zmienna do przechowania

Definition at line 13 of file Stos.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

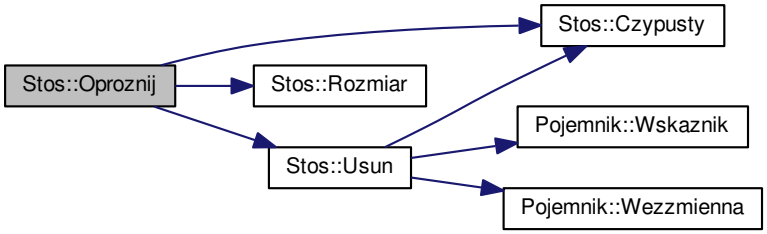


4.11.3.3 void Stos::Oproznij ( )

Usuwa wszystkie elementy stosu

Definition at line 71 of file Stos.cpp.

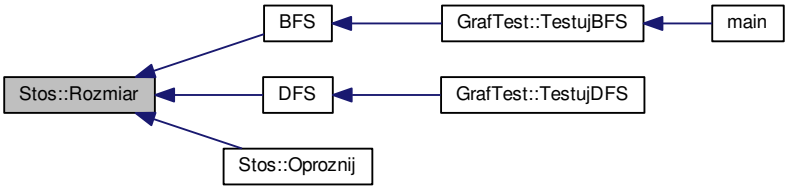
Here is the call graph for this function:



4.11.3.4 int Stos::Rozmiar ( ) [inline]

Definition at line 22 of file Stos.hh.

Here is the caller graph for this function:



4.11.3.5 int Stos::Usun ( )

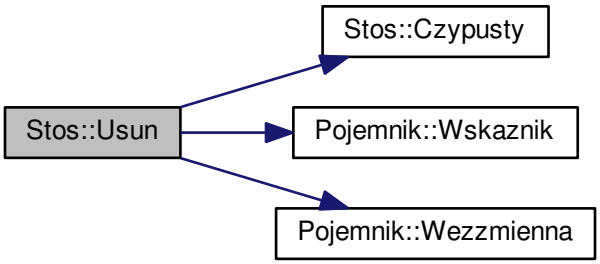
Usuwa element ze stosu

Return values

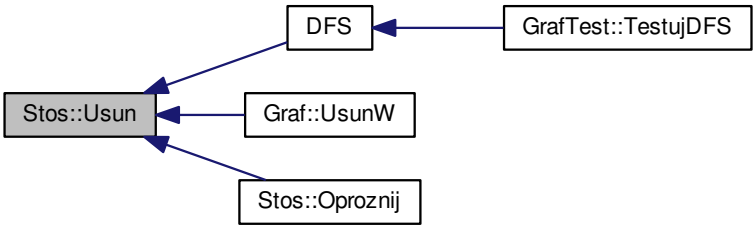
wartosc	usunietego elementu

Definition at line 35 of file Stos.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

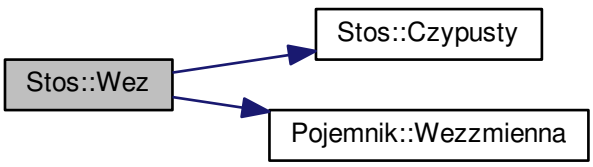


4.11.3.6 int Stos::Wez ( )

Zwraca wartosc elementu stosu, ktory jest "na wierzchu". Funkcja NIE sluzy do modyfikowania wartosci tego elementu

Definition at line 57 of file Stos.cpp.

Here is the call graph for this function:

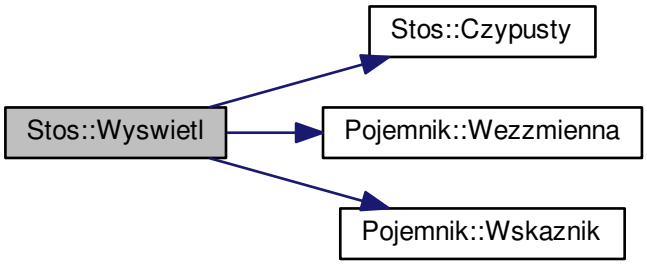


4.11.3.7 void Stos::Wyswietl ( )

Wyswietla wszystkie elementy stosu od "wierzcholka" do dolu

Definition at line 82 of file Stos.cpp.

Here is the call graph for this function:



4.11.3.8 bool Stos::Wyszukaj (int *szukane*)

Metoda sprawdzajaca czy dany element jest przechowywany w stosie

Parameters

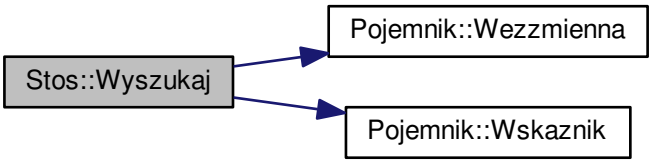
in	*szukane*	element, ktorego obecnosc w stosie jest sprawdzana

Return values

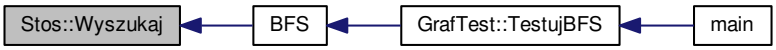
true	jesli szukany element jest w stosie
false	jesli stos nie przechowuje szukanego elementu

Definition at line 106 of file Stos.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.11.4 Member Data Documentation

4.11.4.1 Pojemnik* Stos::head =NULL [private]

Definition at line 14 of file Stos.hh.

4.11.4.2 int Stos::rozmiar =0 [private]

Definition at line 15 of file Stos.hh.

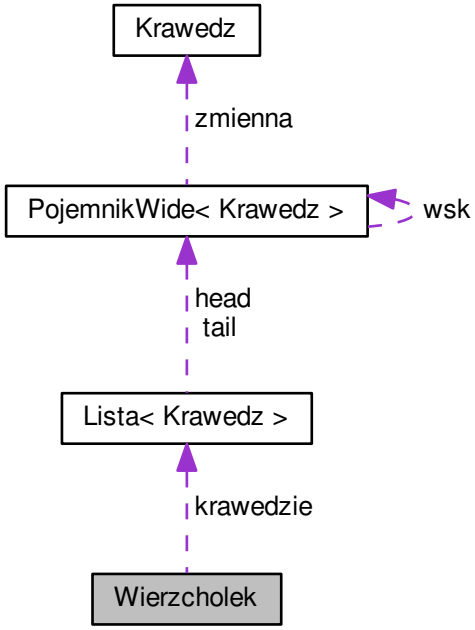
The documentation for this class was generated from the following files:

- Stos.hh
- Stos.cpp

4.12 Wierzcholek Class Reference

```
#include <Wierzcholek.hh>
```

Collaboration diagram for Wierzcholek:



Public Member Functions

- Wierzcholek ()
- Wierzcholek (int pnumer)
- int & Pozycja ()
- void DodajKrawedz (int sasiad, int waga=1)
- bool UsunKrawedz (int sasiad)
- Krawedz & WezK (int indeks)
- void WyswietlKrawedz ()
- void OproznijKrawedz ()
- void ZapiszSasiadow (Stos &sasiedzi)
- void ZapiszSasiadow (Kolejka &kolejka)
- void Flaga ()
- bool CzyOdwiedzony ()
- bool operator== (Wierzcholek drugi)

Private Attributes

- int numer =0
- Lista< Krawedz > krawedzie
- int flaga =0

4.12.1 Detailed Description

Definition at line 11 of file Wierzcholek.hh.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 Wierzcholek::Wierzcholek ( ) [inline]

Definition at line 17 of file Wierzcholek.hh.

4.12.2.2 Wierzcholek::Wierzcholek (int *pnumer*) [inline]

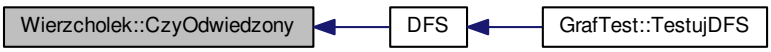
Definition at line 18 of file Wierzcholek.hh.

4.12.3 Member Function Documentation

4.12.3.1 bool Wierzcholek::CzyOdwiedzony ( ) [inline]

Definition at line 31 of file Wierzcholek.hh.

Here is the caller graph for this function:



4.12.3.2 void Wierzcholek::DodajKrawedz (int *sasiad,* int *waga* = 1)

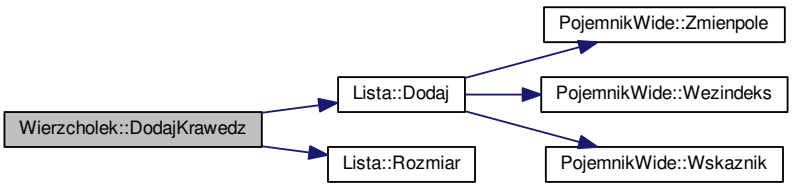
Przypisuje krawedz o zadanych parametrach do wierzcholka

Parameters

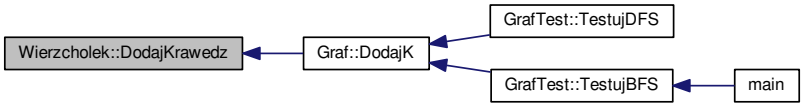
in	*sasiad*	zostanie dodana krawedz prowadzaca do tego sasiada
in	*waga*	waga krawedzi

Definition at line 9 of file Wierzcholek.cpp.

Here is the call graph for this function:



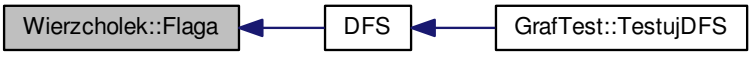
Here is the caller graph for this function:



4.12.3.3 void Wierzcholek::Flaga ( ) [inline]

Definition at line 30 of file Wierzcholek.hh.

Here is the caller graph for this function:



4.12.3.4 bool Wierzcholek::operator== (Wierzcholek *drugi*)

Przeciazenie porownania na potrzeby wyszukiwania wierzcholka w liscie wierzcholkow

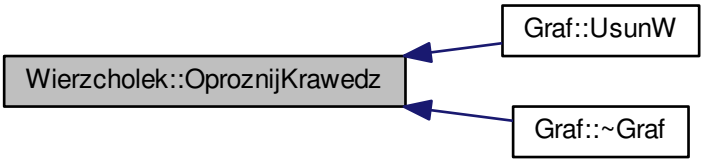
Definition at line 47 of file Wierzcholek.cpp.

Here is the call graph for this function:

4.12.3.5 void Wierzcholek::OproznijKrawedz ( ) [inline]

Definition at line 27 of file Wierzcholek.hh.

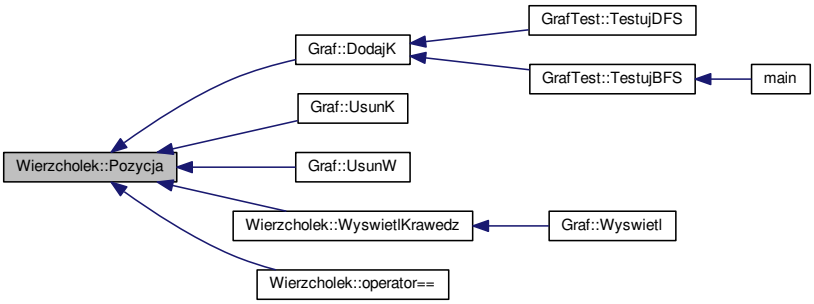
Here is the caller graph for this function:



4.12.3.6 int& Wierzcholek::Pozycja ( ) [inline]

Definition at line 19 of file Wierzcholek.hh.

Here is the caller graph for this function:



4.12.3.7 bool Wierzcholek::UsunKrawedz (int *sasiad*)

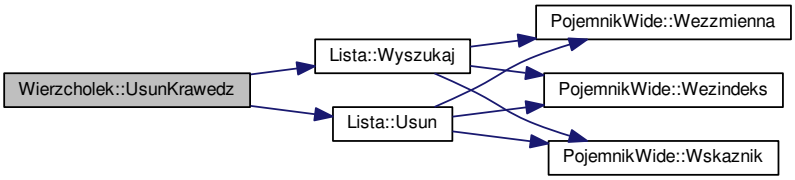
Usuwa podana krawedz przylegajaca do wierzcholka

Parameters

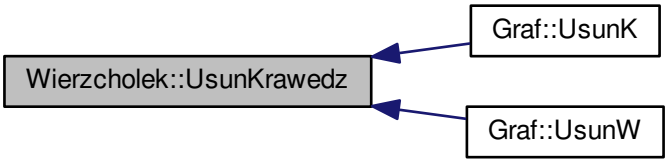
in	*sasiad*	krawedz prowadzaca do tego sasiada mamy usunac

Definition at line 19 of file Wierzcholek.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

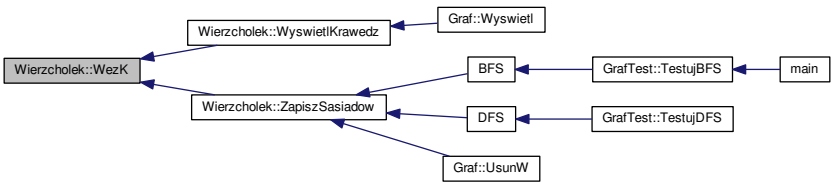


4.12.3.8 Krawedz& Wierzcholek::WezK (int *indeks*) [inline]

Daje dostep do i'tej krawedzi zapisanej na liscie wierzcholka

Definition at line 25 of file Wierzcholek.hh.

Here is the caller graph for this function:

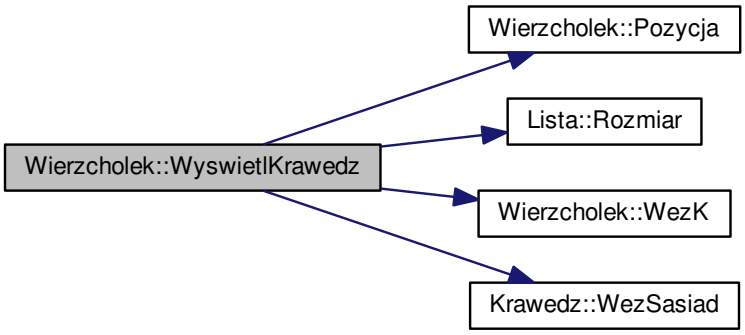


4.12.3.9 void Wierzcholek::WyswietlKrawedz ( )

Wyswietla wszystkie krawedzie przylegajace do danego wierzcholka

Definition at line 34 of file Wierzcholek.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

4.12.3.10 void Wierzcholek::ZapiszSasiadow (Stos & *stos*)

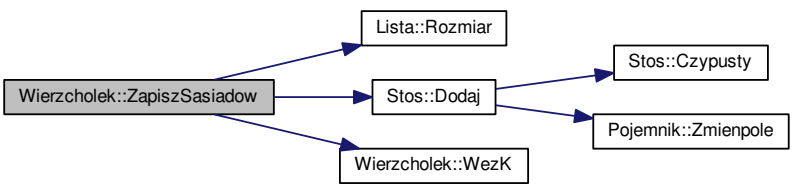
Zapisuje do stosu wszystkie sasiadujace wierzcholki

Parameters

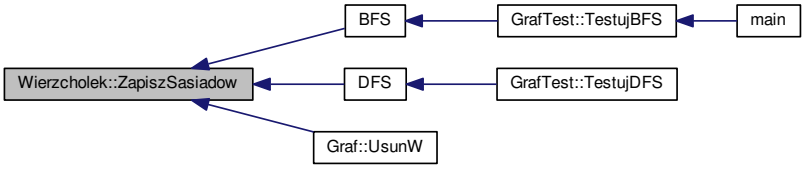
in	*stos*	w nim zapisane beda informacje o sasiadach

Definition at line 60 of file Wierzcholek.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.12.3.11 void Wierzcholek::ZapiszSasiadow (Kolejka & *kolejka*)

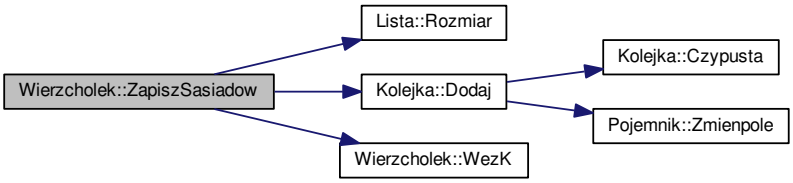
Zapisuje do kolejki wszystkie sasiadujace wierzcholki

Parameters

in	*kolejka*	w niej zapisane beda informacje o sasiadach

Definition at line 71 of file Wierzcholek.cpp.

Here is the call graph for this function:



4.12.4 Member Data Documentation

4.12.4.1 int Wierzcholek::flaga =0 [private]

Definition at line 14 of file Wierzcholek.hh.

4.12.4.2 Lista<Krawedz> Wierzcholek::krawedzie [private]

Definition at line 13 of file Wierzcholek.hh.

4.12.4.3 int Wierzcholek::numer =0 [private]

Definition at line 12 of file Wierzcholek.hh.

The documentation for this class was generated from the following files:

- Wierzcholek.hh
- Wierzcholek.cpp

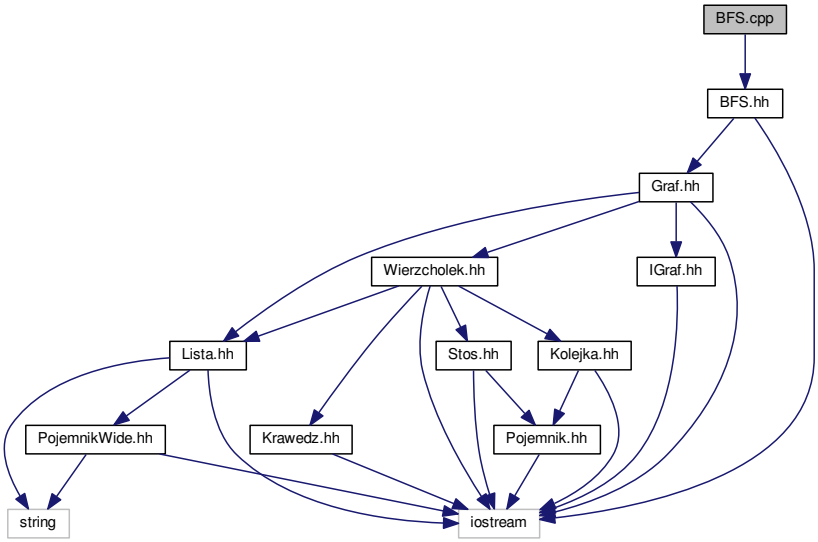
Chapter 5

File Documentation

5.1 BFS.cpp File Reference

```
#include "BFS.hh"
```

Include dependency graph for BFS.cpp:



Functions

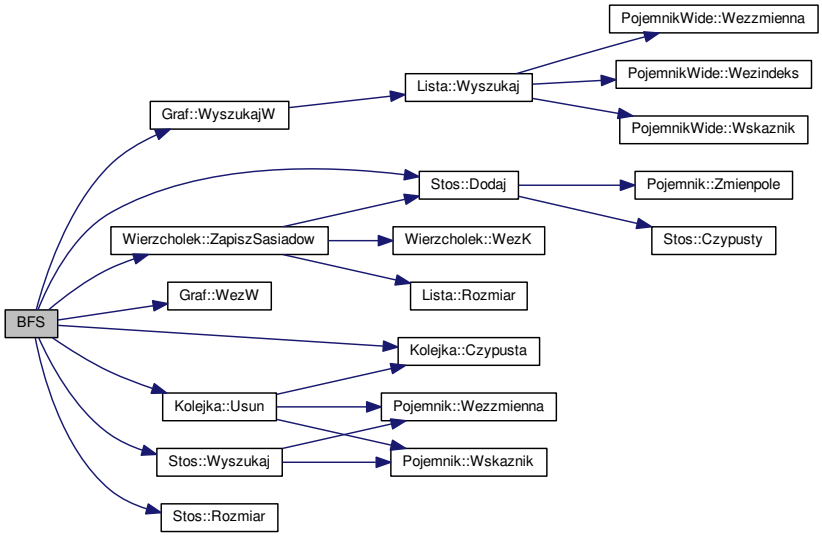
- int BFS (Graf &graf)

5.1.1 Function Documentation

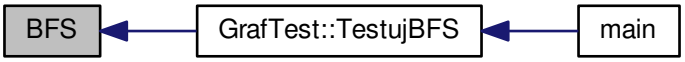
5.1.1.1 int BFS (Graf & *graf*)

Definition at line 3 of file BFS.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

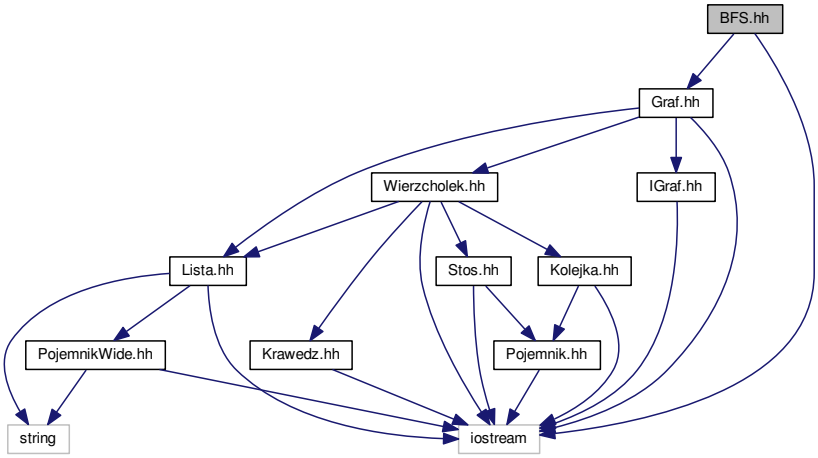


5.2 BFS.hh File Reference

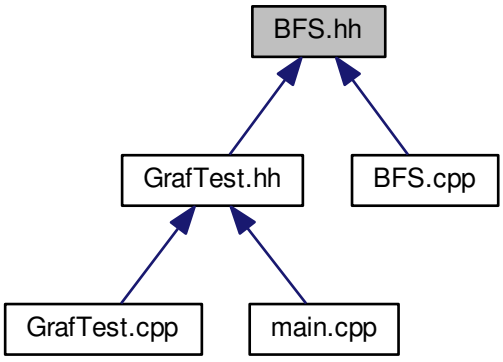
Implementacja funkcji BFS (Breadth-first search)

```
#include <iostream>
#include "Graf.hh"
```

Include dependency graph for BFS.hh:



This graph shows which files directly or indirectly include this file:



Functions

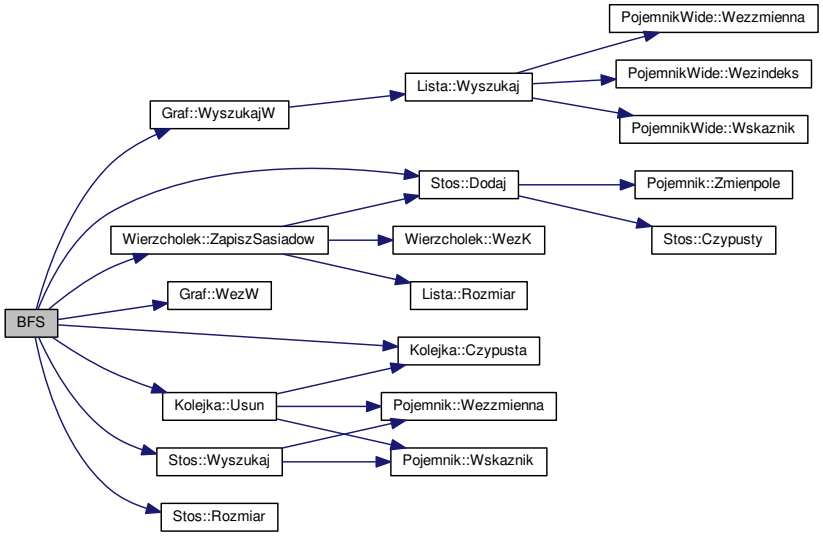
- int BFS (Graf &graf)

5.2.1 Function Documentation

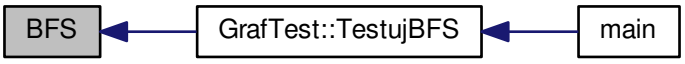
5.2.1.1 int BFS (Graf & *graf*)

Definition at line 3 of file BFS.cpp.

Here is the call graph for this function:



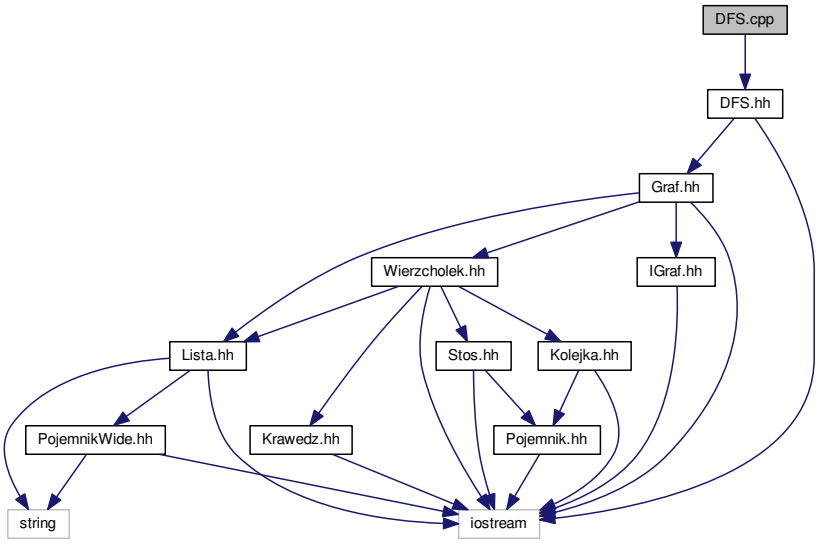
Here is the caller graph for this function:



5.3 DFS.cpp File Reference

```
#include "DFS.hh"
```

Include dependency graph for DFS.cpp:



Functions

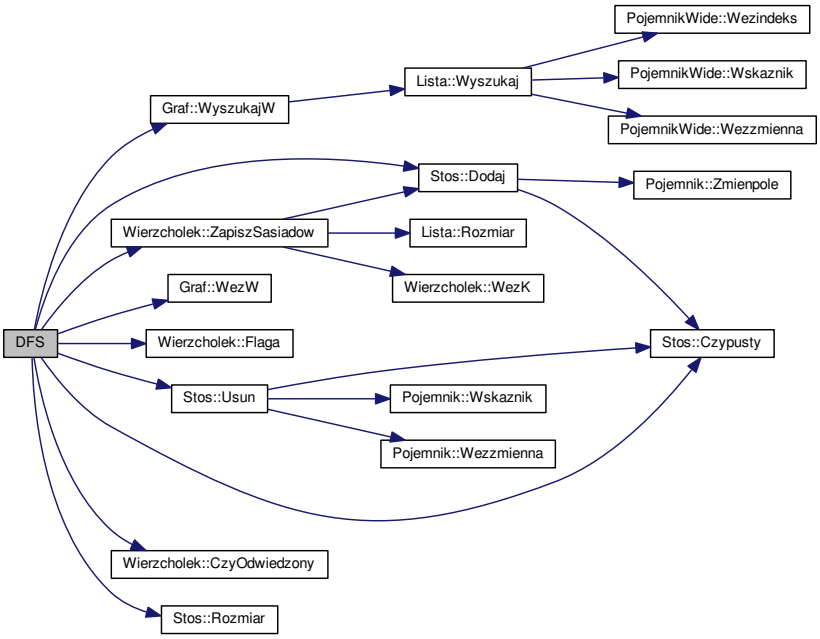
- int DFS (Graf &graf)

5.3.1 Function Documentation

5.3.1.1 int DFS (Graf & *graf*)

Definition at line 3 of file DFS.cpp.

Here is the call graph for this function:



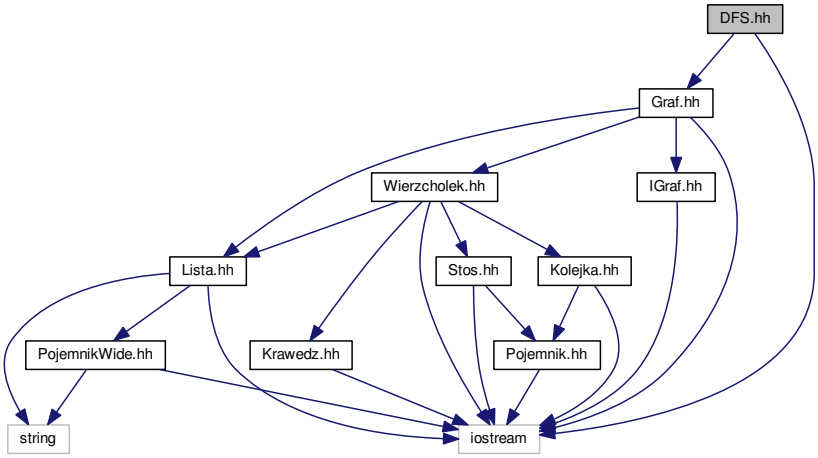
Here is the caller graph for this function:

5.4 DFS.hh File Reference

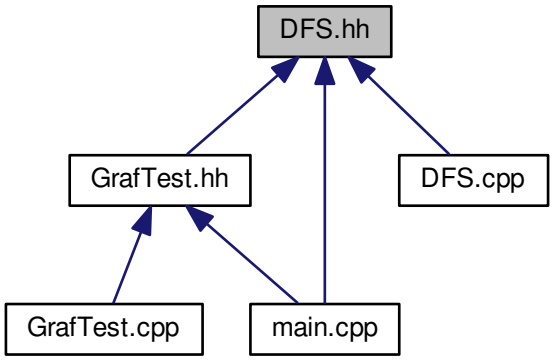
Implementacja funkcji DFS (Depth-first search)

```
#include <iostream>
#include "Graf.hh"
```

Include dependency graph for DFS.hh:



This graph shows which files directly or indirectly include this file:



Functions

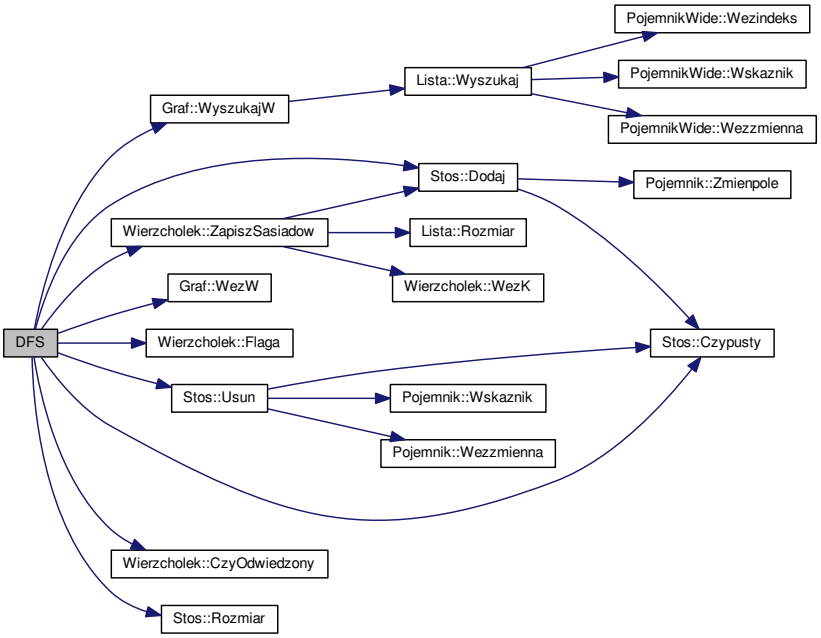
- int DFS (Graf &graf)

5.4.1 Function Documentation

5.4.1.1 int DFS (Graf & *graf*)

Definition at line 3 of file DFS.cpp.

Here is the call graph for this function:

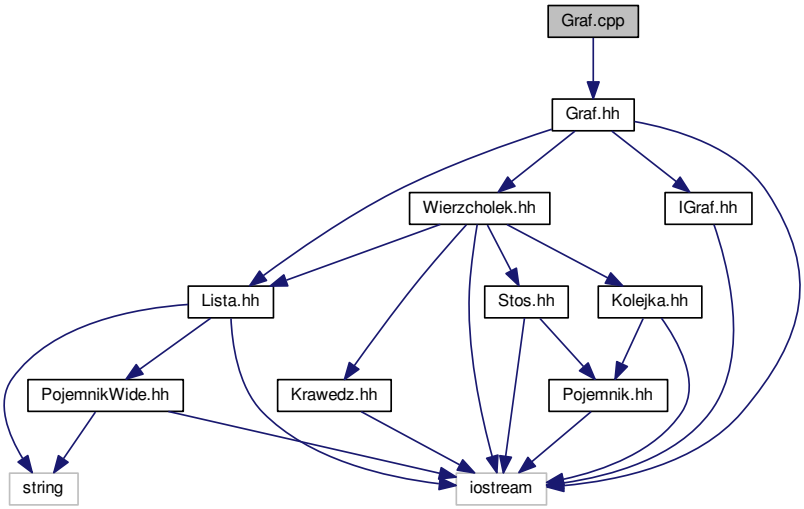


Here is the caller graph for this function:

5.5 Graf.cpp File Reference

```
#include "Graf.hh"
```

Include dependency graph for Graf.cpp:

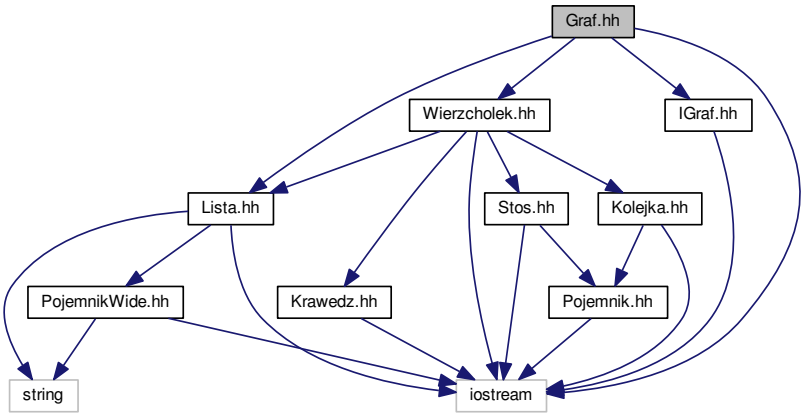


5.6 Graf.hh File Reference

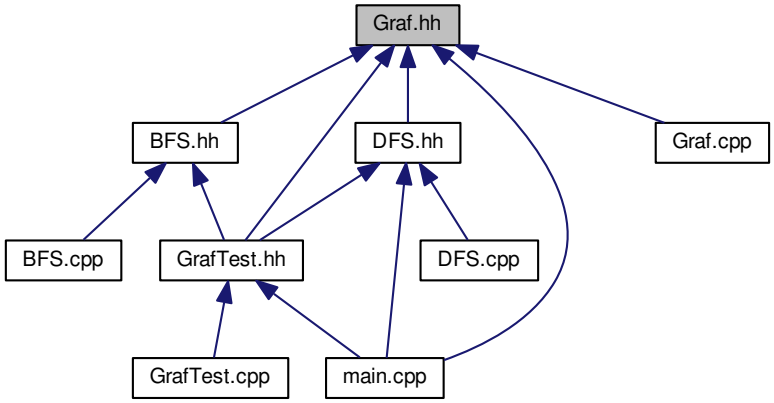
Implementacja grafu za pomoca listy sasiedztwa.

```
#include <iostream>
#include "Lista.hh"
#include "Wierzcholek.hh"
#include "IGraf.hh"
```

Include dependency graph for Graf.hh:



This graph shows which files directly or indirectly include this file:



Classes

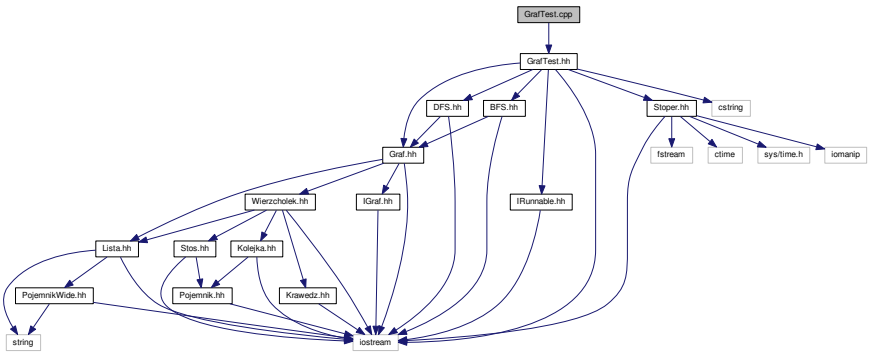
- class Graf

5.7 GrafTest.cpp File Reference

Definicja metod zwiazanych z "GrafTest".

```
#include "GrafTest.hh"
```

Include dependency graph for GrafTest.cpp:



Functions

- bool otworz_plik_odczyt (string nazwapom, fstream &StrmPlikowy)

  otwarcie pliku

5.7.1 Function Documentation

5.7.1.1 bool otworz_plik_odczyt (string *nazwapom,* fstream & *StrmPlikowy*)

Otwiera plik i tworzy strumien do odczytu

Parameters

in	*nazwapom-*	nazwa pliku, ktory ma zostac otwarty
in	*StrmPlikowy-*	Zapisywany jest w nim strumien skad bedziemy odczytywac dane

Definition at line 17 of file GrafTest.cpp.

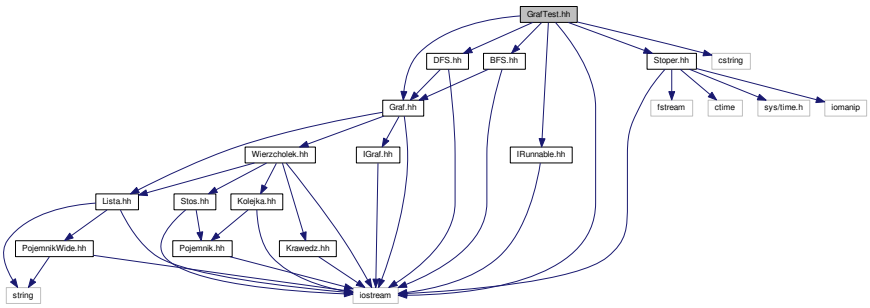
Here is the caller graph for this function:

5.8 GrafTest.hh File Reference

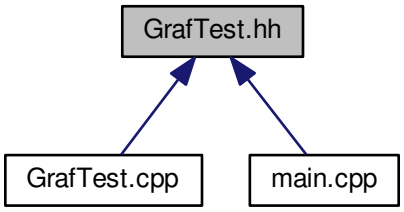
Implementacja klasy odpowiedzialnej za testowanie algorytmow DFS i BFS na grafie.

```
#include <iostream>
#include <cstring>
#include "Graf.hh"
#include "DFS.hh"
#include "BFS.hh"
#include "Stoper.hh"
#include "IRunnable.hh"
```

Include dependency graph for GrafTest.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class GrafTest

Functions

- bool otworz_plik_odczyt (string nazwapom, fstream &StrmPlikowy)

  otwarcie pliku

5.8.1 Function Documentation

5.8.1.1 bool otworz_plik_odczyt (string *nazwapom*, fstream & *StrmPlikowy*)

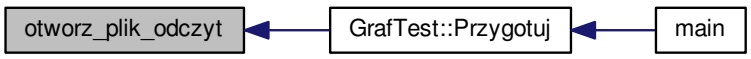
Otwiera plik i tworzy strumien do odczytu

Parameters

in	*nazwapom*-	nazwa pliku, ktory ma zostac otwarty
in	*StrmPlikowy*-	Zapisywany jest w nim strumien skad bedziemy odczytywac dane

Definition at line 17 of file GrafTest.cpp.

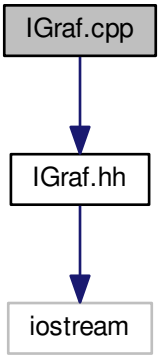
Here is the caller graph for this function:



5.9 IGraf.cpp File Reference

```
#include "IGraf.hh"
```

Include dependency graph for IGraf.cpp:

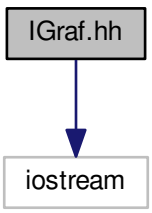


5.10 IGraf.hh File Reference

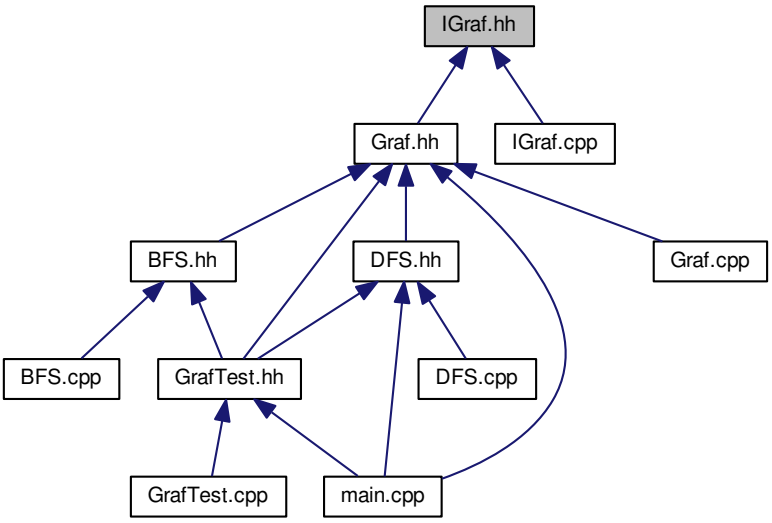
Interface Grafu.

```
#include <iostream>
```

Include dependency graph for IGraf.hh:



This graph shows which files directly or indirectly include this file:



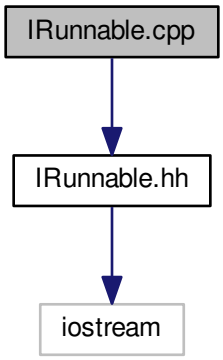
Classes

- class IGraf

5.11 IRunnable.cpp File Reference

```
#include "IRunnable.hh"
```

Include dependency graph for IRunnable.cpp:

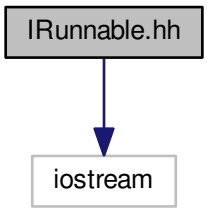


5.12 IRunnable.hh File Reference

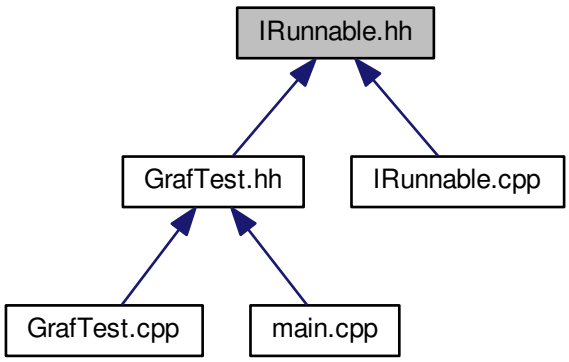
Interface testowania Grafu.

```
#include <iostream>
```

Include dependency graph for IRunnable.hh:



This graph shows which files directly or indirectly include this file:



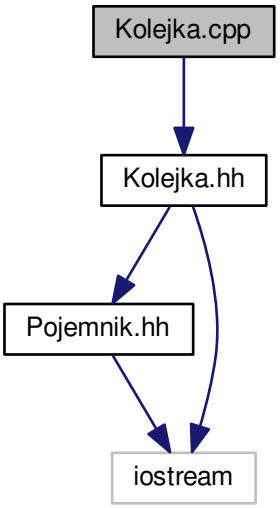
Classes

- class IRunnable

5.13 Kolejka.cpp File Reference

Definicja metod interface'u ADT- Kolejka.

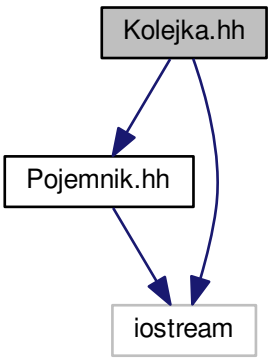
```
#include "Kolejka.hh"
```
Include dependency graph for Kolejka.cpp:



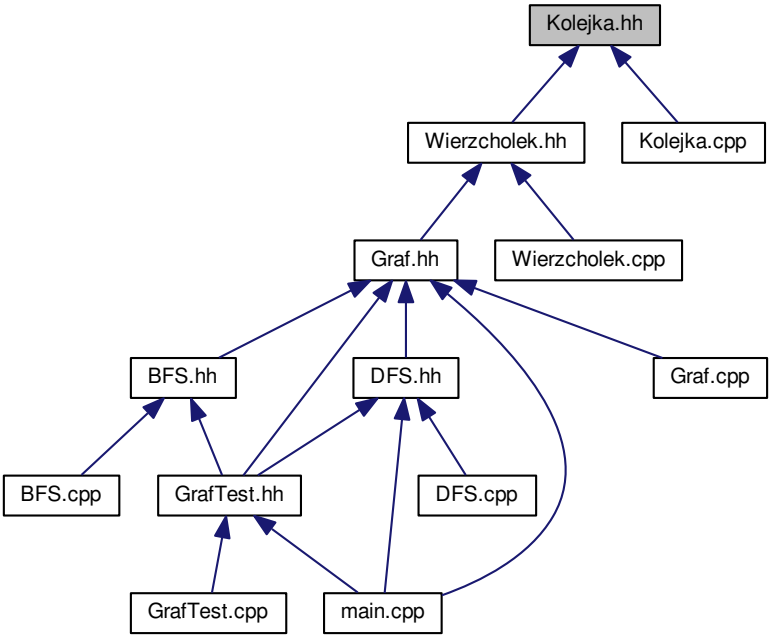
5.14 Kolejka.hh File Reference

interface abstrakcyjnego typu danych - Kolejka

```
#include "Pojemnik.hh"
#include <iostream>
```
Include dependency graph for Kolejka.hh:



This graph shows which files directly or indirectly include this file:



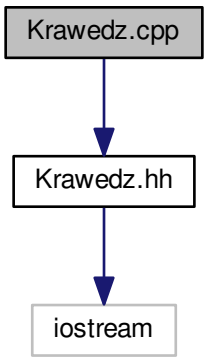
Classes

- class Kolejka

5.15 Krawedz.cpp File Reference

```
#include "Krawedz.hh"
```

Include dependency graph for Krawedz.cpp:

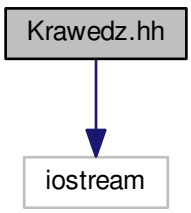


5.16 Krawedz.hh File Reference

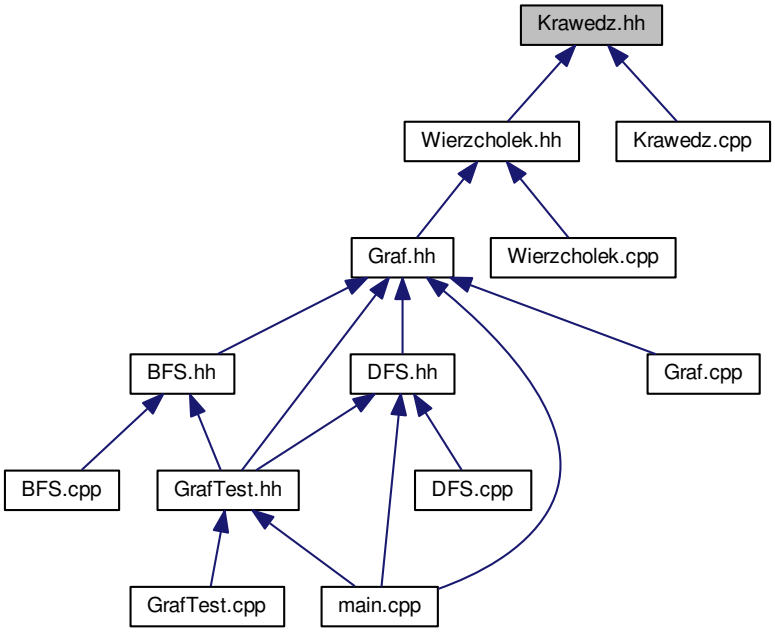
Implementacja krawedzi grafu.

```
#include <iostream>
```

Include dependency graph for Krawedz.hh:



This graph shows which files directly or indirectly include this file:



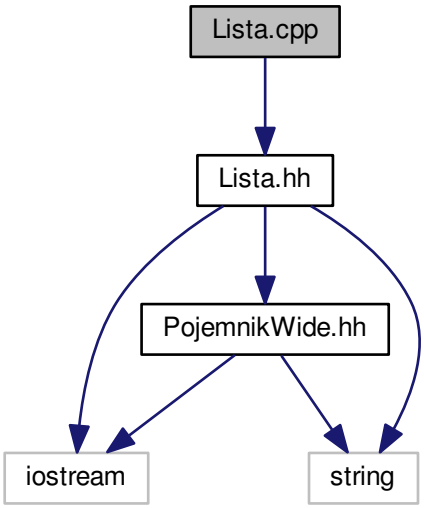
Classes

- class Krawedz

5.17 Lista.cpp File Reference

```
#include "Lista.hh"
```

Include dependency graph for Lista.cpp:

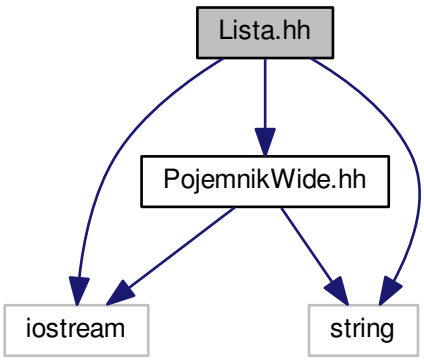


5.18 Lista.hh File Reference

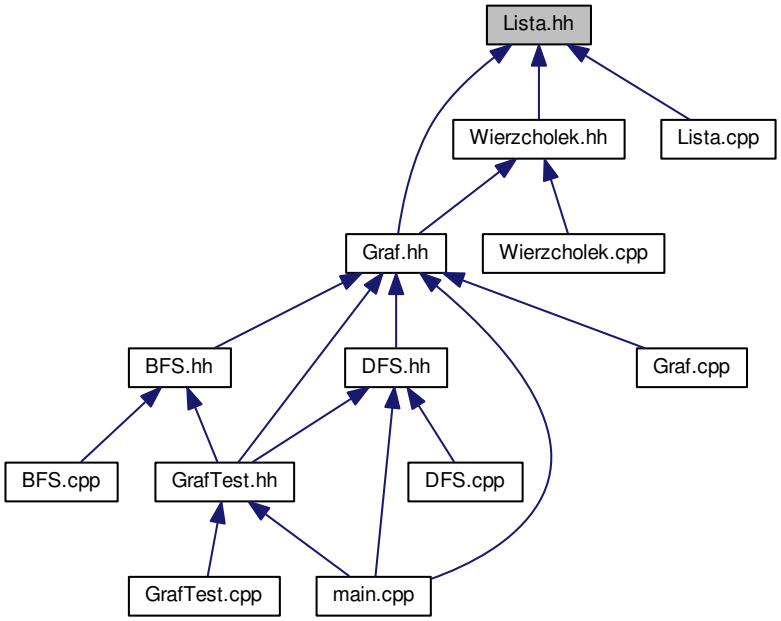
interface abstrakcyjnego typu danych - Lista

```
#include <iostream>
#include "PojemnikWide.hh"
#include <string>
```

Include dependency graph for Lista.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class Lista< typ >

5.18.1 Detailed Description

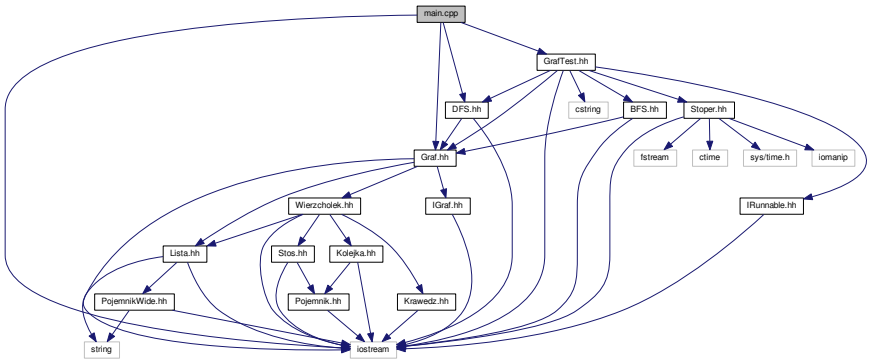
Elementy do Listy mozesz dodawac lub usuwac dowolnie czyli na poczatku, koncu badz wewnatrz listy

Definition in file Lista.hh.

5.19 main.cpp File Reference

```
#include <iostream>
#include "Graf.hh"
#include "DFS.hh"
#include "GrafTest.hh"
```

Include dependency graph for main.cpp:



Functions

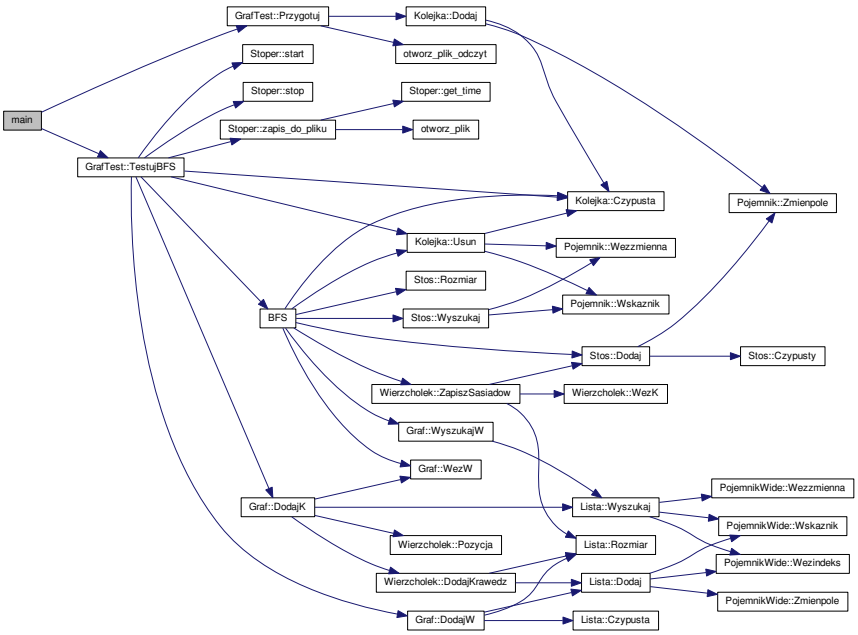
- int main ()

5.19.1 Function Documentation

5.19.1.1 int main ( )

Definition at line 9 of file main.cpp.

Here is the call graph for this function:

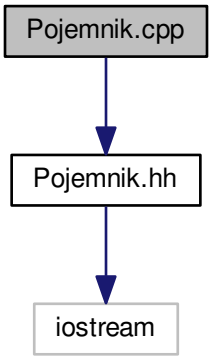


5.20 Pojemnik.cpp File Reference

Definicja metod pojedynczego elementu ADT (Kolejka, Stos)

```
#include "Pojemnik.hh"
```

Include dependency graph for Pojemnik.cpp:

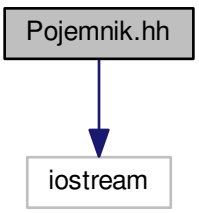


5.21 Pojemnik.hh File Reference

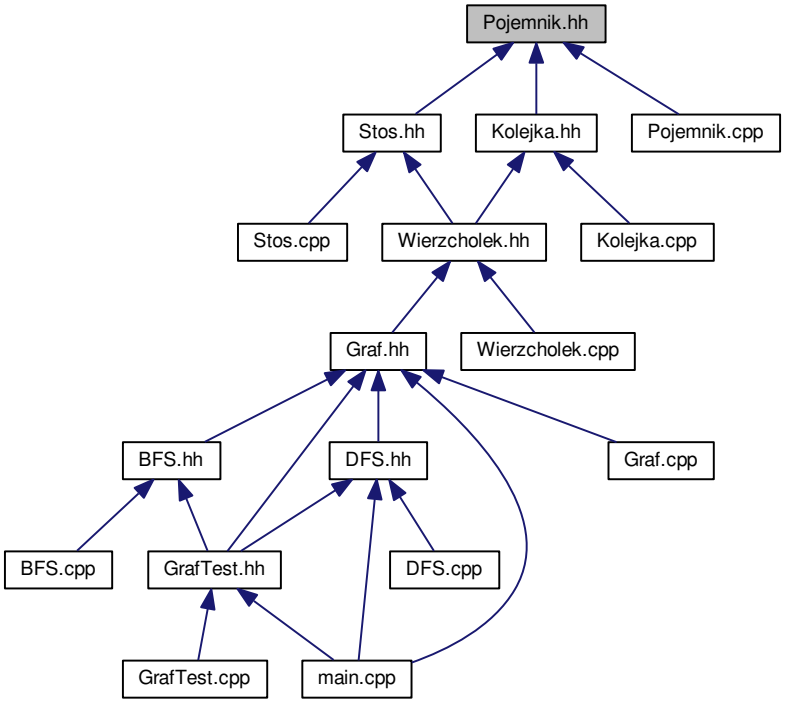
Pelni role pojedynczego elementu ADT (Kolejka, Stos)

```
#include <iostream>
```

Include dependency graph for Pojemnik.hh:



This graph shows which files directly or indirectly include this file:



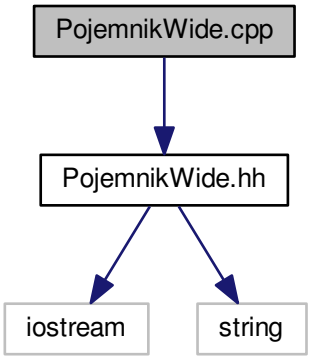
Classes

- class Pojemnik

5.22 PojemnikWide.cpp File Reference

Definicje metod pojedynczego elementu ADT (Lista)

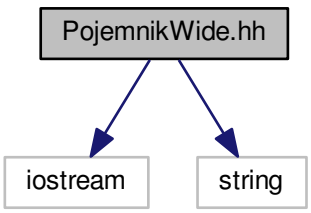
```
#include "PojemnikWide.hh"
```
Include dependency graph for PojemnikWide.cpp:



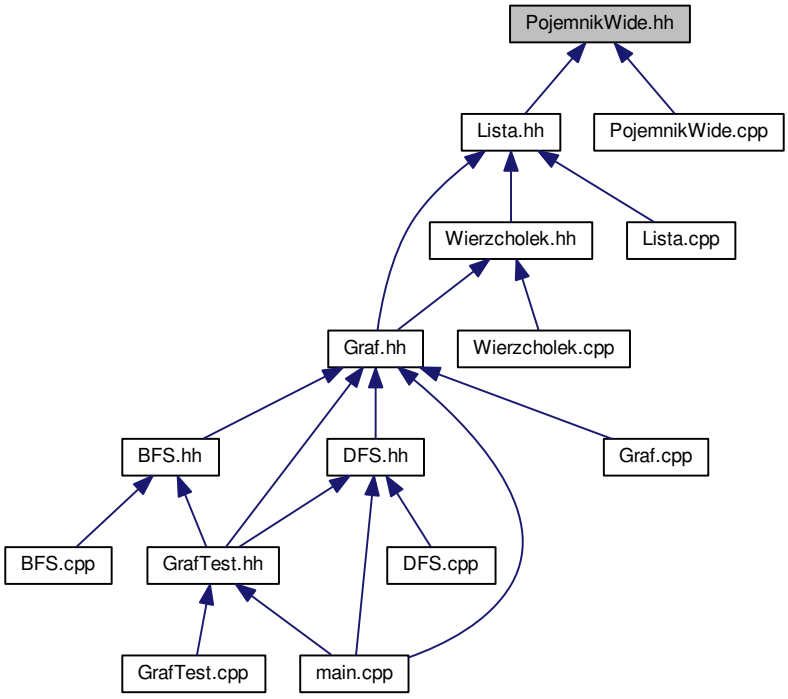
5.23 PojemnikWide.hh File Reference

Pelni role pojedynczego elementu ADT (Lista)

```
#include <iostream>
#include <string>
```
Include dependency graph for PojemnikWide.hh:



This graph shows which files directly or indirectly include this file:



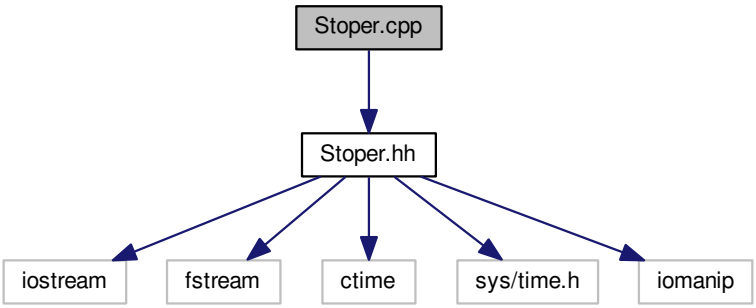
Classes

- class PojemnikWide< typ >

5.24 Stoper.cpp File Reference

```
#include "Stoper.hh"
```

Include dependency graph for Stoper.cpp:



Functions

- bool otworz_plik (string nazwapom, ofstream &StrmPlikowy)

  otwarcie pliku

5.24.1 Function Documentation

5.24.1.1 bool otworz_plik (string *nazwapom,* ofstream & *StrmPlikowy*)

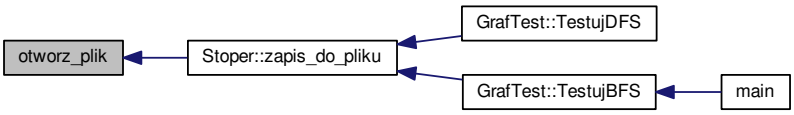
Otwiera plik i tworzy strumien do zapisywania UWAGA: PLIK OTWARTY W TRYBIE DOPISYWANIA

Parameters

in	*nazwapom-*	nazwa pliku, ktory ma zostac otwarty/utworzony
in	*StrmPlikowy-*	Zapisywany jest w nim strumien gdzie bedziemy zapisywac dane

Definition at line 23 of file Stoper.cpp.

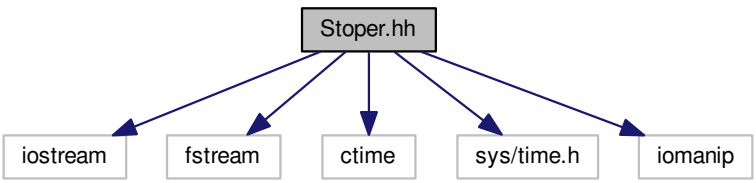
Here is the caller graph for this function:



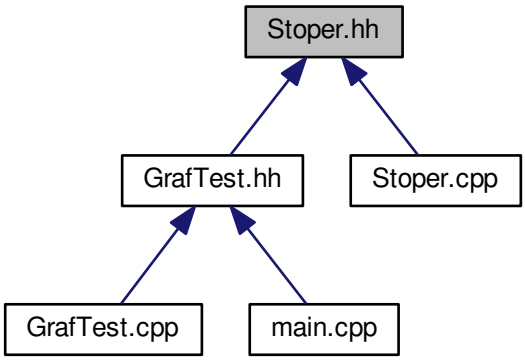
5.25 Stoper.hh File Reference

```
#include <iostream>
#include <fstream>
#include <ctime>
#include <sys/time.h>
#include <iomanip>
```

Include dependency graph for Stoper.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class Stoper

Functions

- bool otworz_plik (string nazwapom, ofstream &StrmPlikowy)

  otwarcie pliku

5.25.1 Function Documentation

5.25.1.1 bool otworz_plik (string *nazwapom*, ofstream & *StrmPlikowy*)

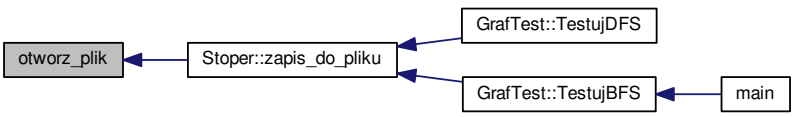
Otwiera plik i tworzy strumien do zapisywania UWAGA: PLIK OTWARTY W TRYBIE DOPISYWANIA

Parameters

in	*nazwapom*-	nazwa pliku, ktory ma zostac otwarty/utworzony
in	*StrmPlikowy*-	Zapisywany jest w nim strumien gdzie bedziemy zapisywac dane

Definition at line 23 of file Stoper.cpp.

Here is the caller graph for this function:

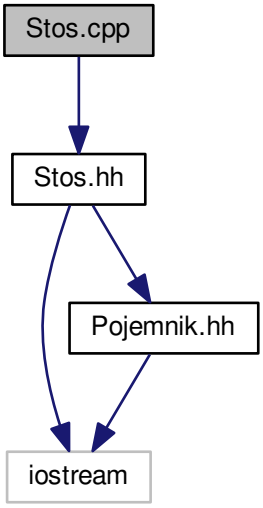


5.26 Stos.cpp File Reference

Definicja metod interface'u ADT- Stos.

```
#include "Stos.hh"
```

Include dependency graph for Stos.cpp:

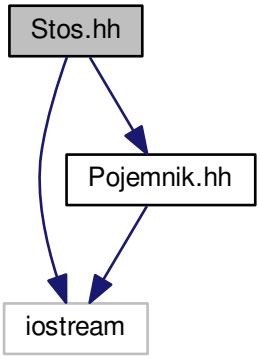


5.27 Stos.hh File Reference

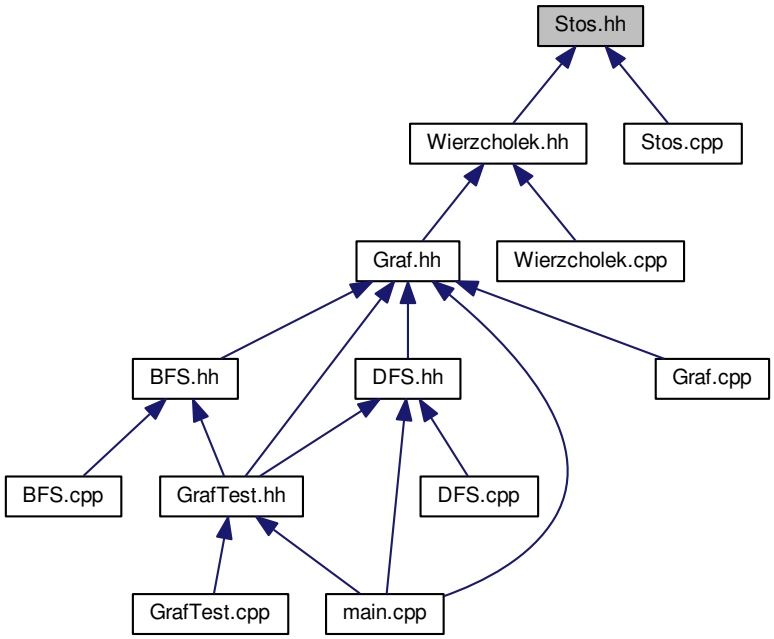
interface abstrakcyjnego typu danych - Stos

```
#include <iostream>
#include "Pojemnik.hh"
```

Include dependency graph for Stos.hh:



This graph shows which files directly or indirectly include this file:



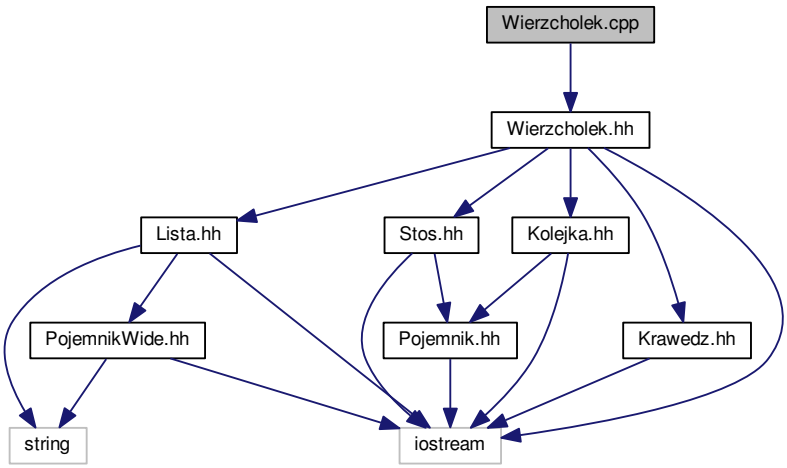
Classes

- class Stos

5.28 Wierzcholek.cpp File Reference

```
#include "Wierzcholek.hh"
```

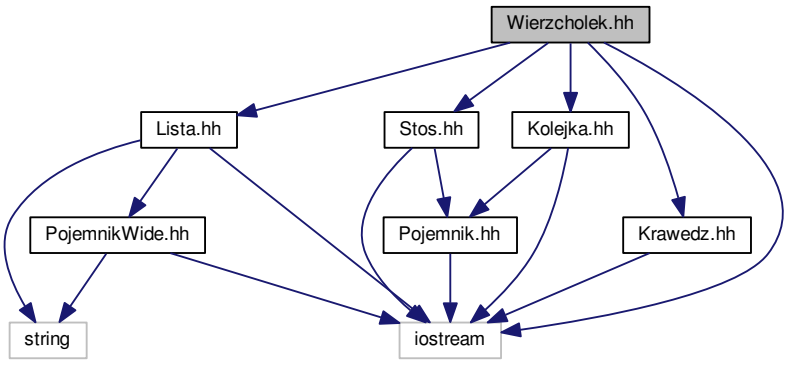
Include dependency graph for Wierzcholek.cpp:



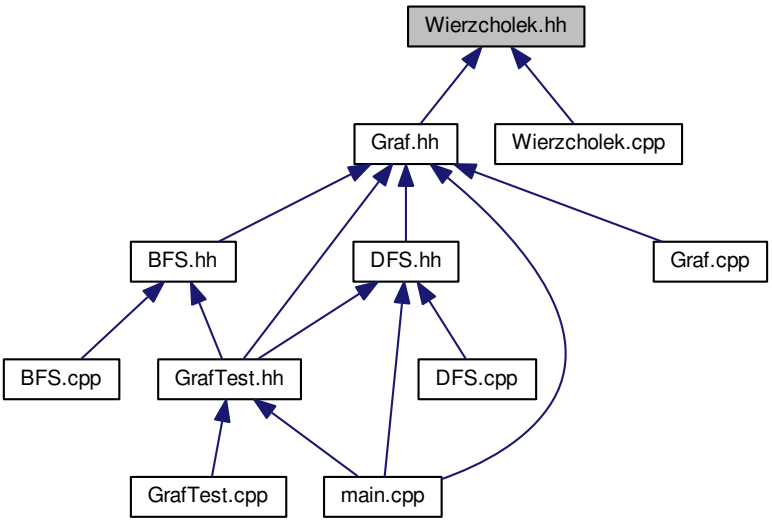
5.29 Wierzcholek.hh File Reference

```
#include <iostream>
#include "Lista.hh"
#include "Stos.hh"
#include "Kolejka.hh"
#include "Krawedz.hh"
```

Include dependency graph for Wierzcholek.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class Wierzcholek