

Lab6

Wygenerowano przez Doxygen 1.7.6.1

Tue Apr 22 2014 16:34:33



# Contents

<b>1</b>	<b>Struktura katalogów</b>	<b>1</b>
1.1	Katalogi . . . . .	1
<b>2</b>	<b>Indeks klas</b>	<b>3</b>
2.1	Lista klas . . . . .	3
<b>3</b>	<b>Indeks plików</b>	<b>5</b>
3.1	Lista plików . . . . .	5
<b>4</b>	<b>Dokumentacja katalogów</b>	<b>7</b>
4.1	Dokumentacja katalogu /home/martyna/pamsi2/lab6/prj/inc/ . . . . .	7
4.2	Dokumentacja katalogu /home/martyna/pamsi2/lab6/prj/ . . . . .	8
4.3	Dokumentacja katalogu /home/martyna/pamsi2/lab6/prj/src/ . . . . .	9
<b>5</b>	<b>Dokumentacja klas</b>	<b>11</b>
5.1	Dokumentacja szablonu klasy Drzewo< K, W > . . . . .	11
5.1.1	Opis szczegółowy . . . . .	11
5.1.2	Dokumentacja konstruktora i destruktoru . . . . .	12
5.1.2.1	Drzewo . . . . .	12
5.1.2.2	~Drzewo . . . . .	12
5.1.2.3	Drzewo . . . . .	12
5.1.3	Dokumentacja funkcji składowych . . . . .	12
5.1.3.1	dodaj . . . . .	12
5.1.3.2	dodaj . . . . .	12
5.1.3.3	nastepnik . . . . .	12
5.1.3.4	pobierzWartosc . . . . .	13
5.1.3.5	usun . . . . .	13

5.1.3.6	usun	13
5.1.4	Dokumentacja atrybutów składowych	13
5.1.4.1	korzen	13
5.1.4.2	liczbaElementow	14
5.2	Dokumentacja szablonu klasy Para< K, W >	14
5.2.1	Opis szczegółowy	14
5.2.2	Dokumentacja konstruktora i destruktor	14
5.2.2.1	Para	14
5.2.2.2	Para	14
5.2.2.3	~Para	15
5.2.2.4	Para	15
5.2.3	Dokumentacja funkcji składowych	15
5.2.3.1	pobierzKlucz	15
5.2.3.2	pobierzWartosc	16
5.2.3.3	ustawKlucz	16
5.2.3.4	ustawWartosc	16
5.2.4	Dokumentacja atrybutów składowych	16
5.2.4.1	klucz	17
5.2.4.2	wartosc	17
5.3	Dokumentacja szablonu klasy Tablica< K, W >	17
5.3.1	Opis szczegółowy	17
5.3.2	Dokumentacja konstruktora i destruktor	18
5.3.2.1	Tablica	18
5.3.2.2	~Tablica	18
5.3.3	Dokumentacja funkcji składowych	18
5.3.3.1	czypusta	18
5.3.3.2	dodaj	18
5.3.3.3	haszstring	19
5.3.3.4	operator[]	19
5.3.3.5	pobierzWartosc	19
5.3.3.6	size	20
5.3.3.7	usun	20
5.3.4	Dokumentacja atrybutów składowych	20
5.3.4.1	liczbaElementow	20

5.3.4.2	rozmiar	21
5.3.4.3	tablica	21
5.4	Dokumentacja szablonu klasy Wezel< K, W >	21
5.4.1	Opis szczegółowy	22
5.4.2	Dokumentacja konstruktora i destruktora	22
5.4.2.1	Wezel	22
5.4.2.2	Wezel	22
5.4.2.3	~Wezel	22
5.4.2.4	Wezel	22
5.4.3	Dokumentacja funkcji składowych	22
5.4.3.1	osieroc	22
5.4.4	Dokumentacja atrybutów składowych	22
5.4.4.1	lewy	22
5.4.4.2	para	22
5.4.4.3	prawy	22
<b>6</b>	<b>Dokumentacja plików</b>	<b>25</b>
6.1	Dokumentacja pliku Drzewo.h	25
6.1.1	Opis szczegółowy	26
6.2	Dokumentacja pliku Main.cpp	27
6.2.1	Dokumentacja funkcji	27
6.2.1.1	gen_random	28
6.2.1.2	main	28
6.3	Dokumentacja pliku Para.cpp	28
6.4	Dokumentacja pliku Para.h	29
6.4.1	Opis szczegółowy	29
6.5	Dokumentacja pliku Tablica.h	30
6.5.1	Opis szczegółowy	30
6.6	Dokumentacja pliku Wezel.h	31
6.6.1	Opis szczegółowy	32



# Chapter 1

## Struktura katalogów

### 1.1 Katalogi

Ta struktura katalogów jest posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

prj . . . . .	8
inc . . . . .	7
src . . . . .	9





## Chapter 2

# Indeks klas

### 2.1 Lista klas

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

<a href="#">Drzewo&lt; K, W &gt;</a>	
Modeluje pojecie drzewa binarnego. Jego atrybutem jest klasa - <a href="#">Wezel</a>	11
<a href="#">Para&lt; K, W &gt;</a>	
Modeluje pojecie pary. Jej atrybutem sa pola zawierajace klucz i wartosci	14
<a href="#">Tablica&lt; K, W &gt;</a>	
Modeluje pojecie tablicy z haszowaniem. Klasa modeluje pojecie tablicy z haszowaniem. Jej atrybutami sa pola: klucz i wartosc	17
<a href="#">Wezel&lt; K, W &gt;</a>	
Modeluje pojecie wezla. Jego atrybutem jest klasa <a href="#">Para</a>	21



## Chapter 3

# Indeks plików

### 3.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

<a href="#">Drzewo.h</a>	Definicja szablonu klasy <a href="#">Drzewo</a> Plik zawiera definicje szablonu klasy <a href="#">Drzewo</a> . . . . .	25
<a href="#">Main.cpp</a>	. . . . .	27
<a href="#">Para.cpp</a>	. . . . .	28
<a href="#">Para.h</a>	Definicja szablonu klasy <a href="#">Para</a> Plik zawiera definicje szablonu klasy <a href="#">Para</a> . . . . .	29
<a href="#">Tablica.h</a>	Definicja szablonu klasy <a href="#">Tablica</a> ( <a href="#">tablica</a> z haszowaniem) . . . . .	30
<a href="#">Wezel.h</a>	Definicja szablonu klasy <a href="#">Wezel</a> Plik zawiera definicje szablonu klasy <a href="#">Wezel</a> . . . . .	31

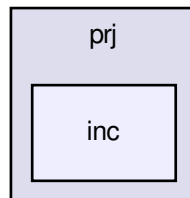


## Chapter 4

# Dokumentacja katalogów

### 4.1 Dokumentacja katalogu /home/martyna/pamsi2/lab6/prj/inc/

Directory dependency graph for /home/martyna/pamsi2/lab6/prj/inc/:

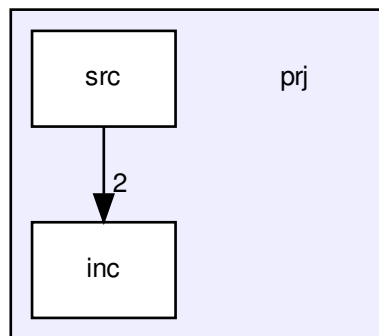


#### Pliki

- plik [Drzewo.h](#)
- plik [Para.h](#)
- plik [Tablica.h](#)
- plik [Wezel.h](#)

## 4.2 Dokumentacja katalogu /home/martyna/pamsi2/lab6/prj/

Directory dependency graph for /home/martyna/pamsi2/lab6/prj/:

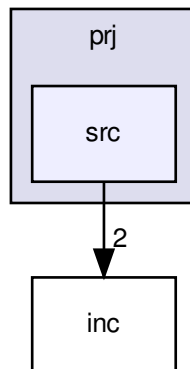


### Katalogi

- katalog [inc](#)
- katalog [src](#)

### 4.3 Dokumentacja katalogu /home/martyna/pamsi2/lab6/prj/src/

Directory dependency graph for /home/martyna/pamsi2/lab6/prj/src/:



#### Pliki

- plik [Main.cpp](#)
- plik [Para.cpp](#)





## Chapter 5

# Dokumentacja klas

### 5.1 Dokumentacja szablonu klasy Drzewo< K, W >

```
#include <Drzewo.h>
```

#### Metody publiczne

- Drzewo ()
- ~Drzewo ()
- Drzewo (const Drzewo &drzewo)
- void dodaj (K klucz, W wartosc)
- W pobierzWartosc (K klucz)
- void usun (K klucz)
- Wezel< K, W > \*& nastepnik (Wezel< K, W > \*&wezel, Wezel< K, W > \*&us)

#### Metody prywatne

- void dodaj (K klucz, W wartosc, Wezel< K, W > \*&wsk)
- Wezel< K, W > \*& usun (K klucz, Wezel< K, W > \*&wezel)

#### Atrybuty prywatne

- Wezel< K, W > \* korzen
- int liczbaElementow

#### 5.1.1 Opis szczegółowy

```
template<typename K, typename W>class Drzewo< K, W >
```

Modeluje pojecie drzewa binarnego. Jego atrybutem jest klasa Wezel.

### 5.1.2 Dokumentacja konstruktora i destruktora

5.1.2.1 `template<typename K , typename W > Drzewo< K, W >::Drzewo ( )`

Konstruktor klasy [Drzewo](#). Bezargumentowy.

5.1.2.2 `template<typename K , typename W > Drzewo< K, W >::~~Drzewo ( )`

Destruktor klasy [Drzewo](#). Czysci pamiec po obiekcie.

5.1.2.3 `template<typename K , typename W > Drzewo< K, W >::Drzewo ( const  
Drzewo< K, W > & drzewo )`

Konstruktor kopiujacy klasy [Drzewo](#). Kopiuje obiekt.

### 5.1.3 Dokumentacja funkcji składowych

5.1.3.1 `template<typename K , typename W > void Drzewo< K, W >::dodaj ( K klucz, W  
wartosc, Wezel< K, W > *& wsk ) [private]`

Funkcja dodaj. Dodaje wezel do drzewa.

Oto graf wywoływań tej funkcji:



5.1.3.2 `template<typename K , typename W > void Drzewo< K, W >::dodaj ( K klucz, W  
wartosc )`

Funkcja dodaj. Dodaje wezel do drzewa.

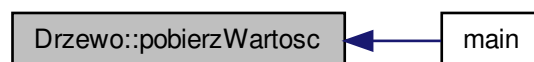
5.1.3.3 `template<typename K , typename W > Wezel< K, W > *& Drzewo< K, W  
>::nastepnik ( Wezel< K, W > *& wezel, Wezel< K, W > *& us )`

Funkcja nastepnik. Wyszukuje i wstawia w odpowiednie miejsce nastepnika.

5.1.3.4 `template<typename K , typename W > W Drzewo< K, W >::pobierzWartosc ( K  
klucz )`

Funkcja `pobierzWartosc`. Zwraca wartosc przypisana do klucza.

Oto graf wywoływań tej funkcji:



5.1.3.5 `template<typename K , typename W > Wezel< K, W > *& Drzewo< K, W >::usun  
( K klucz, Wezel< K, W > *& wezel ) [private]`

Funkcja `usun`. Usuwa wezel z drzewa.

Oto graf wywoływań tej funkcji:



5.1.3.6 `template<typename K , typename W > void Drzewo< K, W >::usun ( K klucz )`

Funkcja `usun`. Usuwa wezel z drzewa.

## 5.1.4 Dokumentacja atrybutów składowych

5.1.4.1 `template<typename K , typename W> Wezel<K,W>* Drzewo< K, W >::korzen  
[private]`

Początkowy element drzewa - korzeń.

5.1.4.2 `template<typename K, typename W> int Drzewo< K, W >::liczbaElementow`  
`[private]`

Obecna liczba elementow w tablicy.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Drzewo.h](#)

## 5.2 Dokumentacja szablonu klasy `Para< K, W >`

```
#include <Para.h>
```

### Metody publiczne

- [Para](#) (K [klucz](#), W [wartosc](#))
- [Para](#) (void)
- [~Para](#) (void)
- [Para](#) (const [Para](#) &para)
- K & [pobierzKlucz](#) (void) const
- W & [pobierzWartosc](#) (void) const
- void [ustawKlucz](#) (K [klucz](#))
- void [ustawWartosc](#) (W [wartosc](#))

### Atrybuty prywatne

- K \* [klucz](#)
- W \* [wartosc](#)

### 5.2.1 Opis szczegółowy

```
template<typename K, typename W>class Para< K, W >
```

Modeluje pojecie pary. Jej atrybutem sa pola zawierajace klucz i wartosci.

### 5.2.2 Dokumentacja konstruktora i destruktor

5.2.2.1 `template<typename K , typename W > Para< K, W >::Para ( K klucz, W wartosc )`

Konstruktor klasy [Para](#). Jego argumentami sa klucz i wartosci.

5.2.2.2 `template<typename K , typename W > Para< K, W >::Para ( void )`

Konstruktor klasy [Para](#). Bezargumentowy.

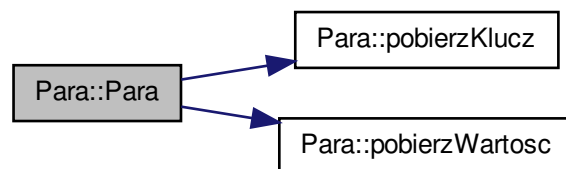
5.2.2.3 `template<typename K , typename W > Para< K, W >::~~Para ( void )`

Destruktor klasy Para. Czysci pamiec po obiekcie.

5.2.2.4 `template<typename K , typename W > Para< K, W >::~Para ( const Para< K, W > & para )`

Konstruktor kopiujacy klasy Para. Kopiuje obiekt.

Oto graf wywołań dla tej funkcji:

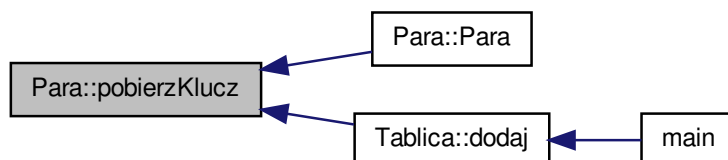


### 5.2.3 Dokumentacja funkcji składowych

5.2.3.1 `template<typename K , typename W > K & Para< K, W >::pobierzKlucz ( void ) const`

Funkcja pobierzKlucz. Zwraca klucz.

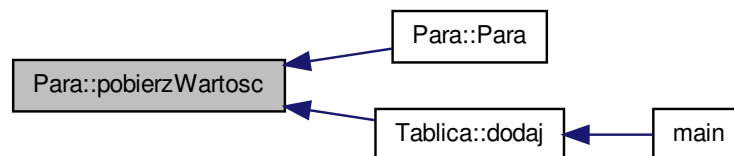
Oto graf wywoływań tej funkcji:



5.2.3.2 `template<typename K , typename W > W & Para< K, W >::pobierzWartosc ( void ) const`

Funkcja `pobierzWartosc`. Zwraca wartosc.

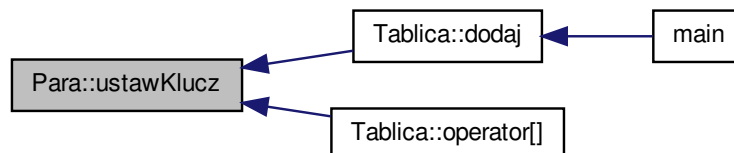
Oto graf wywołań tej funkcji:



5.2.3.3 `template<typename K , typename W > void Para< K, W >::ustawKlucz ( K klucz )`

Funkcja `ustawKlucz`. Ustawia klucz.

Oto graf wywołań tej funkcji:



5.2.3.4 `template<typename K , typename W > void Para< K, W >::ustawWartosc ( W wartosc )`

Funkcja `ustawWartosc`. Ustawia wartosc.

## 5.2.4 Dokumentacja atrybutów składowych

5.2.4.1 `template<typename K, typename W> K* Para< K, W >::klucz` `[private]`

Definicja zmiennej `klucz`.

5.2.4.2 `template<typename K, typename W> W* Para< K, W >::wartosc` `[private]`

Definicja zmiennej `wartosc`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Para.h](#)

## 5.3 Dokumentacja szablonu klasy `Tablica< K, W >`

```
#include <Tablica.h>
```

### Metody publiczne

- [Tablica](#) (int `rozmiar`)
- [~Tablica](#) ()
- void [dodaj](#) ([Para](#)< K, W > &para)
- void [usun](#) (K `klucz`)
- W [pobierzWartosc](#) (K `klucz`)
- bool [czyPusta](#) ()
- int [size](#) ()
- W & [operator\[\]](#) (K `klucz`)

### Metody prywatne

- int [haszstring](#) (K `key`)

### Atrybuty prywatne

- [Para](#)< K, W > \*\* [tablica](#)
- int [liczbaElementow](#)
- int [rozmiar](#)

### 5.3.1 Opis szczegółowy

```
template<typename K, typename W>class Tablica< K, W >
```

Modeluje pojecie tablicy z haszowaniem. Klasa modeluje pojecie tablicy z haszowaniem. Jej atrybutami sa pola: `klucz` i `wartosc`.

### 5.3.2 Dokumentacja konstruktora i destruktora

5.3.2.1 `template<typename K , typename W > Tablica< K, W >::Tablica ( int rozmiar )`

Konstruktor klasy `Tablica`. Jego argumentem jest maksymalny rozmiar tablicy.

5.3.2.2 `template<typename K , typename W > Tablica< K, W >::~~Tablica ( )`

Destruktor klasy `Tablica`. Czysci pamiec.

### 5.3.3 Dokumentacja funkcji składowych

5.3.3.1 `template<typename K, typename W> bool Tablica< K, W >::czypusta ( )`  
`[inline]`

Sprawdza zapelnienie tablicy.

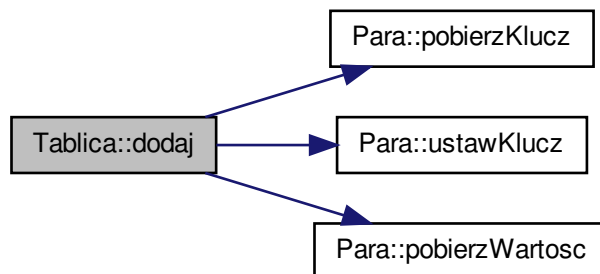
Zwraca

zwraca prawde, gdy tablica jest pusta

5.3.3.2 `template<typename K , typename W > void Tablica< K, W >::dodaj ( Para< K, W > & para )`

Funkcja `dodaj`. Dodaje krotke do tablicy.

Oto graf wywołań dla tej funkcji:





Oto graf wywołań tej funkcji:



**5.3.3.3** `template<typename K , typename W > int Tablica< K, W >::haszstring ( K key )`  
`[private]`

Funkcja haszująca dla obiektów typu string.

**5.3.3.4** `template<typename K , typename W > W & Tablica< K, W >::operator[] ( K klucz )`

Przeciążenie operatora indeksującego.

**Zwraca**

Zwraca wartość.

Oto graf wywołań dla tej funkcji:



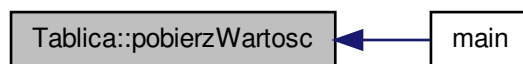
**5.3.3.5** `template<typename K , typename W > W Tablica< K, W >::pobierzWartosc ( K klucz )`

Funkcja `pobierzWartosc`.

**Zwraca**

Zwraca wartosc przypisana do danego klucza.

Oto graf wywoływań tej funkcji:



**5.3.3.6** `template<typename K, typename W> int Tablica< K, W >::size ( ) [inline]`

Podaje liczbe elementow w tablicy.

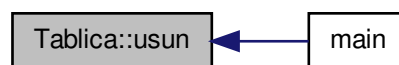
**Zwraca**

zwraca liczbe elementow w tablicy.

**5.3.3.7** `template<typename K , typename W > void Tablica< K, W >::usun ( K klucz )`

Funkcja usun. Usuwa krotke o podanym kluczu.

Oto graf wywoływań tej funkcji:

**5.3.4 Dokumentacja atrybutów składowych**

**5.3.4.1** `template<typename K, typename W> int Tablica< K, W >::liczbaElementow [private]`

Obecna liczba elementow w tablicy.

5.3.4.2 `template<typename K, typename W> int Tablica< K, W >::rozmiar` `[private]`

Maksymalna liczba elementow w tablicy.

5.3.4.3 `template<typename K, typename W> Para<K,W>** Tablica< K, W >::tablica`  
`[private]`

`Tablica` par.

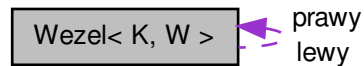
Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Tablica.h](#)

## 5.4 Dokumentacja szablonu klasy `Wezel< K, W >`

```
#include <Wezel.h>
```

Diagram współpracy dla `Wezel< K, W >`:



### Metody publiczne

- `Wezel` (`Para< K, W > para`)
- `Wezel` ()
- `~Wezel` ()
- `Wezel` (const `Wezel` &wezel)
- void `osieroc` ()

### Atrybuty publiczne

- `Para< K, W > * para`
- `Wezel * lewy`
- `Wezel * prawy`

### 5.4.1 Opis szczegółowy

```
template<typename K, typename W>class Wezel< K, W >
```

Modeluje pojecie wezla. Jego atrybutem jest klasa [Para](#).

### 5.4.2 Dokumentacja konstruktora i destruktora

```
5.4.2.1 template<typename K , typename W > Wezel< K, W >::Wezel ( Para< K, W > para )
```

Konstruktor klasy [Wezel](#). Jego argumentem jest klasa [Para](#).

```
5.4.2.2 template<typename K, typename W> Wezel< K, W >::Wezel ( ) [inline]
```

```
5.4.2.3 template<typename K , typename W > Wezel< K, W >::~~Wezel ( )
```

Destruktor klasy [Wezel](#). Czysci pamiec po obiekcie.

```
5.4.2.4 template<typename K , typename W > Wezel< K, W >::Wezel ( const Wezel< K, W > & wezel )
```

Konstruktor kopiujacy klasy [Wezel](#). Kopiuje obiekt.

### 5.4.3 Dokumentacja funkcji składowych

```
5.4.3.1 template<typename K , typename W > void Wezel< K, W >::osieroc ( )
```

Funkcja osieroc. Ustawia wskazniki na dzieci na NULL.

### 5.4.4 Dokumentacja atrybutów składowych

```
5.4.4.1 template<typename K, typename W> Wezel* Wezel< K, W >::lewy
```

Definicja zmiennej lewy.

```
5.4.4.2 template<typename K, typename W> Para<K,W>* Wezel< K, W >::para
```

Definicja zmiennej para.

```
5.4.4.3 template<typename K, typename W> Wezel* Wezel< K, W >::prawy
```

Definicja zmiennej prawy.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Wezel.h](#)



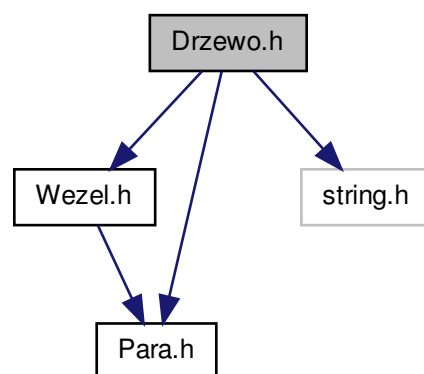
## Chapter 6

# Dokumentacja plików

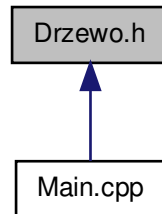
### 6.1 Dokumentacja pliku Drzewo.h

```
#include "Wezel.h" #include <string.h> #include "Para.h" ×
```

Wykres zależności załączania dla Drzewo.h:



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



## Komponenty

- class `Drzewo< K, W >`

*Modeluje pojecie drzewa binarnego. Jego atrybutem jest klasa [Wezel](#).*

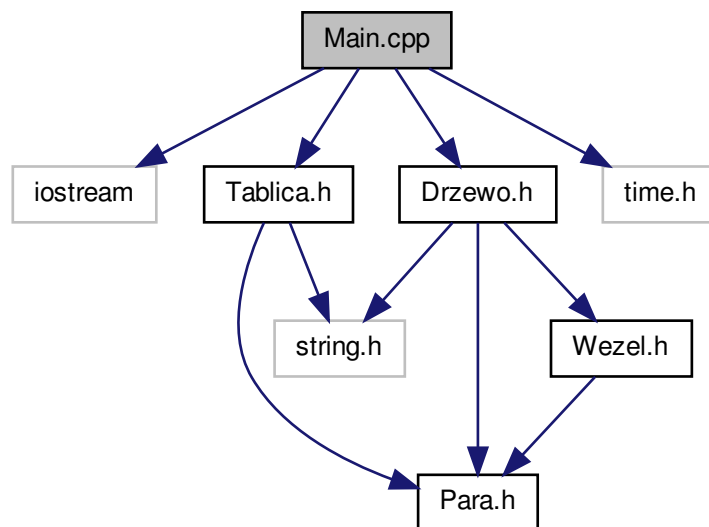
### 6.1.1 Opis szczegółowy

Definicja szablonu klasy `Drzewo` Plik zawiera definicje szablonu klasy `Drzewo`.



## 6.2 Dokumentacja pliku Main.cpp

```
#include <iostream> #include "Tablica.h" #include "Drzewo.h"
#include <time.h>
```

 Wykres zależności załączania dla Main.cpp:

### Funkcje

- string `gen_random` (const int len)
- int `main` ()

#### 6.2.1 Dokumentacja funkcji

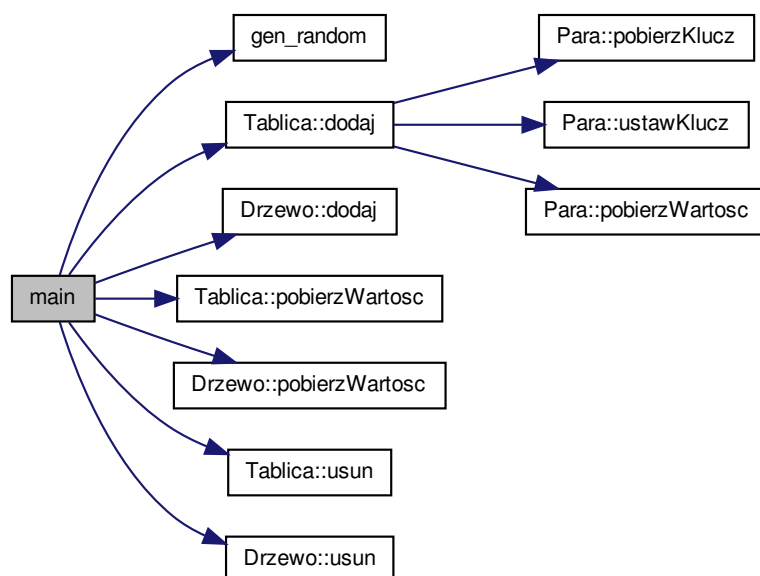
### 6.2.1.1 string gen\_random ( const int len )

Oto graf wywołań tej funkcji:



### 6.2.1.2 int main ( )

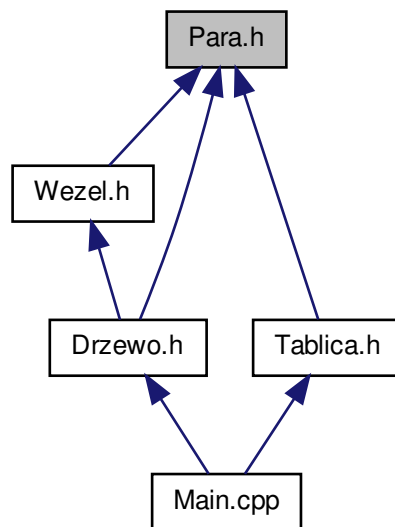
Oto graf wywołań dla tej funkcji:



## 6.3 Dokumentacja pliku Para.cpp

## 6.4 Dokumentacja pliku Para.h

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



### Komponenty

- class `Para< K, W >`

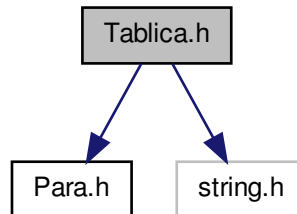
*Modeluje pojecie pary. Jej atrybutem sa pola zawierajace klucz i wartosci.*

#### 6.4.1 Opis szczegółowy

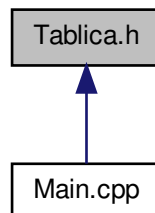
Definicja szablonu klasy `Para` Plik zawiera definicje szablonu klasy `Para`.

## 6.5 Dokumentacja pliku Tablica.h

`#include "Para.h" #include <string.h>` Wykres zależności załączania dla Tablica.h:



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



### Komponenty

- class `Tablica< K, W >`

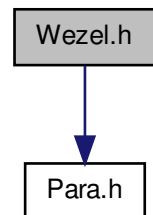
*Modeluje pojecie tablicy z haszowaniem. Klasa modeluje pojecie tablicy z haszowaniem. Jej atrybutami sa pola: klucz i wartosc.*

#### 6.5.1 Opis szczegółowy

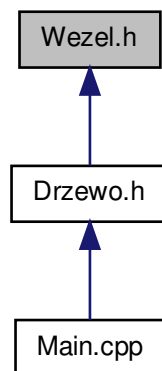
Definicja szablonu klasy `Tablica`(tablicab z haszowaniem) Plik zawiera definicje szablonu klasy `Tablica`. Jest to klasa glowna , ktora wykorzystuje klase `Para`.

## 6.6 Dokumentacja pliku Wezel.h

`#include "Para.h"` Wykres zależności załączania dla Wezel.h:



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



### Komponenty

- class `Wezel< K, W >`

*Modeluje pojecie wezla. Jego atrybutem jest klasa `Para`.*

### 6.6.1 Opis szczegółowy

Definicja szablonu klasy [Wezel](#) Plik zawiera definicje szablonu klasy [Wezel](#).