

Tablica asocjacyjna

1

Generated by Doxygen 1.8.6

Mon May 12 2014 00:27:35

Contents

| | | |
|----------|--|----------|
| 1 | Class Index | 1 |
| 1.1 | Class List | 1 |
| 2 | File Index | 3 |
| 2.1 | File List | 3 |
| 3 | Class Documentation | 5 |
| 3.1 | drzewo< Wartosc > Class Template Reference | 5 |
| 3.1.1 | Detailed Description | 5 |
| 3.1.2 | Constructor & Destructor Documentation | 6 |
| 3.1.2.1 | drzewo | 6 |
| 3.1.3 | Member Function Documentation | 6 |
| 3.1.3.1 | dodaj | 6 |
| 3.1.3.2 | Najmniejszy_Element | 6 |
| 3.1.3.3 | pobierz | 6 |
| 3.1.3.4 | Posz_Nastep | 7 |
| 3.1.3.5 | rozmiar | 8 |
| 3.1.3.6 | usun | 8 |
| 3.2 | dzialanie Class Reference | 8 |
| 3.2.1 | Detailed Description | 9 |
| 3.2.2 | Member Function Documentation | 9 |
| 3.2.2.1 | algorytm | 9 |
| 3.2.2.2 | czysc | 9 |
| 3.2.2.3 | stoper | 9 |
| 3.2.2.4 | wczytaj | 9 |
| 3.2.2.5 | wlaczStoper | 10 |
| 3.2.2.6 | wylaczStoper | 10 |
| 3.3 | hash_table< Wartosc > Class Template Reference | 10 |
| 3.3.1 | Detailed Description | 11 |
| 3.3.2 | Member Function Documentation | 11 |
| 3.3.2.1 | dodaj | 11 |
| 3.3.2.2 | hash | 11 |

| | | |
|---------|---|----|
| 3.3.2.3 | pobierz | 11 |
| 3.3.2.4 | rozmiar | 12 |
| 3.3.2.5 | usun | 12 |
| 3.3.2.6 | wypisz | 12 |
| 3.4 | Liczby< Wartosc > Class Template Reference | 12 |
| 3.4.1 | Detailed Description | 13 |
| 3.4.2 | Constructor & Destructor Documentation | 13 |
| 3.4.2.1 | Liczby | 13 |
| 3.4.2.2 | Liczby | 13 |
| 3.4.2.3 | Liczby | 13 |
| 3.4.3 | Member Function Documentation | 13 |
| 3.4.3.1 | Drugi_Elem | 13 |
| 3.4.3.2 | Drugi_Elem | 13 |
| 3.4.3.3 | Pierwszy_Elem | 14 |
| 3.4.3.4 | Pierwszy_Elem | 14 |
| 3.5 | obiekt< Wartosc > Struct Template Reference | 14 |
| 3.5.1 | Detailed Description | 14 |
| 3.5.2 | Member Data Documentation | 14 |
| 3.5.2.1 | Klucz | 14 |
| 3.5.2.2 | Wart_Klucza | 15 |
| 3.6 | Tab_Asocjacyjna< Wartosc > Class Template Reference | 15 |
| 3.6.1 | Detailed Description | 15 |
| 3.6.2 | Constructor & Destructor Documentation | 16 |
| 3.6.2.1 | Tab_Asocjacyjna | 16 |
| 3.6.3 | Member Function Documentation | 16 |
| 3.6.3.1 | Czy_Pusta | 16 |
| 3.6.3.2 | dodaj | 16 |
| 3.6.3.3 | operator[] | 16 |
| 3.6.3.4 | pobierz | 16 |
| 3.6.3.5 | rozmiar | 17 |
| 3.6.3.6 | usun | 17 |
| 3.6.3.7 | znajdz | 17 |
| 3.7 | wezel< Wartosc > Struct Template Reference | 18 |
| 3.7.1 | Detailed Description | 18 |
| 3.7.2 | Member Data Documentation | 18 |
| 3.7.2.1 | Klucz | 18 |
| 3.7.2.2 | lewy | 18 |
| 3.7.2.3 | prawy | 18 |
| 3.7.2.4 | rodzic | 18 |
| 3.7.2.5 | Wart_Klucza | 18 |

| | |
|---|-----------|
| 4 File Documentation | 19 |
| 4.1 C:/Users/Ania/Documents/GitHub/sort/inc/drzewobinarne.hh File Reference | 19 |
| 4.1.1 Detailed Description | 19 |
| 4.2 C:/Users/Ania/Documents/GitHub/sort/inc/tasocjacyjna.hh File Reference | 19 |
| 4.2.1 Detailed Description | 20 |
| 4.2.2 Function Documentation | 20 |
| 4.2.2.1 operator< | 20 |
| 4.3 C:/Users/Ania/Documents/GitHub/sort/inc/thaszujaca.hh File Reference | 20 |
| 4.3.1 Detailed Description | 21 |
| 4.4 C:/Users/Ania/Documents/GitHub/sort/inc/uruchom.hh File Reference | 21 |
| 4.4.1 Detailed Description | 21 |
| 4.5 C:/Users/Ania/Documents/GitHub/sort/src/main.cpp File Reference | 21 |
| 4.5.1 Detailed Description | 22 |
| 4.5.2 Function Documentation | 22 |
| 4.5.2.1 dzialaj | 22 |
| 4.5.2.2 main | 22 |
| 4.5.2.3 zmien_nazwy | 22 |
| 4.6 C:/Users/Ania/Documents/GitHub/sort/src/uruchom.cpp File Reference | 23 |
| Index | 24 |

Chapter 1

Class Index

1.1 Class List

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

| | |
|---|----|
| drzewo < Wartosc > | |
| W klasie drzewo wartosciami prywatnymi (dostepnymi tylko przez metody klasy) jest licznik liczba_elementow , który zlicza ile jest elementow w drzewie binarnym oraz wektor, który jest wskaźnikiem na korzeń elementu. Metodami publicznymi klasy są funkcje dodajace elemnty, usuwajace i pobierajace element z wektora | 5 |
| dzialanie | 8 |
| hash_table < Wartosc > | |
| W klasie hash_table (p. 10) wartosciami prywatnymi (dostepnymi tylko przez metody klasy) jest liczba_elementow , która definiuje stała wartosc elelmtow znajdujacych sie na liscie oraz lista, która jest wskaźnikiem na korzeń elementu. Metodami publicznymi klasy są funkcje dodajace elemnty, usuwajace i pobierajace element z wektora | 10 |
| Liczy < Wartosc > | |
| Klasa ta jest wykorzystana do stworzenia szablonu, czyli aby wartoscia moglabyc dowolna zmienna wykorzystana przez uzytkownika. Docelowo kluczem jest string czyli wartosc indeksujaca tablice. W klasie tej stworzone sa konstruktory i referencje do modyfikowania wartosci oraz odczytu | 12 |
| obiekt < Wartosc > | |
| Struktura obiekt w ktorej zdefiniowany jest element tablicy czyli klucz, który jest stringiem i jego wartosc, która jest szablonem | 14 |
| Tab_Asocjacyjna < Wartosc > | |
| Klasa Tab_Asocjacyjna (p. 15), to w niej tworzony jest wektor, w którym przechowywane sa wszystkie elementy. Indeksami tego wektora sa Klucze (string), którym odpowiadaja wartosci w zaleznosci jaka uzytkownik zdefiniuje zmienna. Znajduja sie tez rowniez rozne funkcje(metody), które mogą zostac uzyte przy wykorzystywaniu tablicy asocjacyjnej | 15 |
| wezel < Wartosc > | |
| Struktura wezel w ktorej zdefiniowany jest element tablicy czyli klucz, który jest stringiem i jego wartosc, która jest szablonem. W strukturze tej znajduje sie rowniez wskaźnik do konkretnego wezla (rodzica i jego dzieci), co ulatwia nam poruszanie sie po drzewie | 18 |

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

| | |
|---|----|
| C:/Users/Ania/Documents/GitHub/sort/inc/ drzewobinarne.hh | |
| Plik naglowkowy wykorzystywany do stworzenia drzewa binarnego do przechowywania tablicy o zadanym kluczu i wartosci. W pliku stworzona jest struktura gdzie zdefiniowany jest element wektora oraz klasa gdzie wystepuja wszystkie funkcje operujace na drzewie | 19 |
| C:/Users/Ania/Documents/GitHub/sort/inc/ tasocjacyjna.hh | |
| Plik naglowkowy tasocjacyjna, ktora sluzy do stowrzenia tablicy asocjacyjnej, tablica ta posiada klucz, ktory jest argumentem wektora, w moim przypadku oraz wartosc klucza, czyli jaka wartosc znajduje sie pod kluczem. W tym pliku stworzone sa dwie klasy. Klasa glowna Tab_Asocjacyjna (p. 15) korzysta z malej klasy Liczby (p. 12), ktora jest pomocnicza i reprezentujaca elemnty . | 19 |
| C:/Users/Ania/Documents/GitHub/sort/inc/ thaszujaca.hh | |
| Plik naglowkowy wykorzystywany do stworzenia tablicy haszujacej do przechowywania tablicy o zadanym kluczu i wartosci. W pliku stworzona jest struktura gdzie zdefiniowany jest element wektora oraz klasa gdzie wystepuja wszystkie funkcje operujace na tabilcy | 20 |
| C:/Users/Ania/Documents/GitHub/sort/inc/ uruchom.hh | |
| Klasa odpowaidajca za dziaanie i sprawdzenie zoonoci obliczeniowej kadego rodzaju tablicy asocjacyjnej. To w tej kalsie włączamy i wyłączamy stoper, wczytujemy plik z danymi i wykonujemy algorytm | 21 |
| C:/Users/Ania/Documents/GitHub/sort/src/ main.cpp | |
| Funkcja zmieniajaca nazwy dla roznym wielkosci problemu, aby za kazdym razem nie trzeba bylo wczytywac na nowo pliku | 21 |
| C:/Users/Ania/Documents/GitHub/sort/src/ uruchom.cpp | 23 |

Chapter 3

Class Documentation

3.1 drzewo< Wartosc > Class Template Reference

W klasie drzewo wartosciami prywatnymi (dostepnymi tylko przez metody klasy) jest licznik liczba_elementow, który zlicza ile jest elementow w drzewie binarnym oraz wektor, który jest wskaźnikiem na korzeń elementu. Metodami publicznymi klasy są funkcje dodajace elemnty, usuwajace i pobierajace element z wektora.

```
#include <drzewobinarne.hh>
```

Public Member Functions

- **drzewo** ()
Konstruktor inicjujacy wartosci przy wywołaniu klasy.
- void **dodaj** (string Pierwszy_Elem, Wartosc Drugi_Elem)
Funkcja dodajaca element do tablicy Funkcja dodaje kolejne elementy do drzewa binarnego badz zmienia wartosci juz istniejacych elementow, jezeli takie istnieja w niej.
- Wartosc **pobierz** (string Pierwszy_Elem)
Funkcja pobieraca element z tablicy. Funkcja pobiera wartosc, ktora znajduje sie pod wskazanym kluczem.
- void **usun** (string Pierwszy_Elem)
Usuwanie wskazanego elementu Funkcja usuwa element o wskazanym kluczu.
- int **rozmiar** ()
Sprawdzanie rozmiaru tablicy.
- struct **wezel**< Wartosc > * **Najmniejszy_Element** (struct **wezel**< Wartosc > *w)
Najmniejszy element drzewa. Funkcja wyszukuje najmniejszy element znajdujacy sie w drzewie. Funkcja wykorzystywana jest do usuwania elemntu z wektora.
- struct **wezel**< Wartosc > * **Posz_Nastep** (struct **wezel**< Wartosc > *w)
Wyszukiwanie elementu Szukanie nastepnika, ktory bedzie zastepowal miejsce usunietego elementu. Funkcja uzyteczna jest do usuwania elementu.

3.1.1 Detailed Description

```
template<typename Wartosc>class drzewo< Wartosc >
```

W klasie drzewo wartosciami prywatnymi (dostepnymi tylko przez metody klasy) jest licznik liczba_elementow, który zlicza ile jest elementow w drzewie binarnym oraz wektor, który jest wskaźnikiem na korzeń elementu. Metodami publicznymi klasy są funkcje dodajace elemnty, usuwajace i pobierajace element z wektora.

Definition at line 44 of file drzewobinarne.hh.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 `template<typename Wartosc > drzewo< Wartosc >::drzewo () [inline]`

Konstruktor inicjujący wartości przy wywołaniu klasy.

Definition at line 61 of file drzewobinarne.hh.

3.1.3 Member Function Documentation

3.1.3.1 `template<typename Wartosc > void drzewo< Wartosc >::dodaj (string Pierwszy_Elem, Wartosc Drugi_Elem)`

Funkcja dodająca element do tablicy Funkcja dodaje kolejne elementy do drzewa binarnego bądź zmienia wartości już istniejących elementów, jeżeli takie istnieją w niej.

Parameters

| | | |
|----|----------------------|--|
| in | <i>Pierwszy_Elem</i> | - jest to Klucz czyli indeks tablicy |
| in | <i>Drugi_Elem</i> | - zmienna ta jest wartością klucza, może przyjmować różne typy |

Returns

(brak)

Definition at line 147 of file drzewobinarne.hh.

3.1.3.2 `template<typename Wartosc > struct wezel< Wartosc >* drzewo< Wartosc >::Najmniejszy_Element (struct wezel< Wartosc >* w) [inline]`

Najmniejszy element drzewa. Funkcja wyszukuje najmniejszy element znajdujący się w drzewie. Funkcja wykorzystywana jest do usuwania elementu z wektora.

Parameters

| | | |
|----|----------|---|
| in | <i>w</i> | - jest to wektor, który jest drzewem binarnym |
|----|----------|---|

Returns

Funkcja zwraca wartość najmniejszego elementu znajdującego się w węźle

Definition at line 102 of file drzewobinarne.hh.

3.1.3.3 `template<typename Wartosc > Wartosc drzewo< Wartosc >::pobierz (string Pierwszy_Elem)`

Funkcja pobierająca element z tablicy. Funkcja pobiera wartość, która znajduje się pod wskazanym kluczem.

Parameters

| | | |
|----|----------------------|---|
| in | <i>Pierwszy_Elem</i> | - jest to Klucz wektora, czyli indeks tablicy |
|----|----------------------|---|

Returns

Funkcja zwraca wartość, jaka znajduje się pod wskazanym kluczem, która przyjmuje typ zmiennej w zależności od wykorzystania i zdefiniowania przez użytkownika.

Definition at line 210 of file drzewobinarne.hh.

```
3.1.3.4  template<typename Wartosc > struct wezel< Wartosc > * drzewo< Wartosc >::Posz_Nastep ( struct wezel<
        Wartosc > * w )  [inline]
```

Wyszukiwanie elementu Szukanie nastepnika, który będzie zastępował miejsce usuniętego elementu. Funkcja użyteczna jest do usuwania elementu.

Parameters

| | | |
|-----------------|----------------|--|
| <code>in</code> | <code>w</code> | - jest to wektora, który jest drzewem binarnym |
|-----------------|----------------|--|

Returns

Funkcja zwraca wartosc najmniejszego elementu znajdujacego sie w wezle

Definition at line 122 of file drzewobinarne.hh.

3.1.3.5 `template<typename Wartosc > int drzewo< Wartosc >::rozmiar ()`

Sprawdzanie rozmiaru tablicy.

Returns

Funkcja zwraca liczbe calkowita, ktora jest rozmiarem calego drzewa binarnego.

Definition at line 350 of file drzewobinarne.hh.

3.1.3.6 `template<typename Wartosc > void drzewo< Wartosc >::usun (string Pierwszy_Elem)`

Usuwanie wskazanego elementu Funkcja usuwa element o wskazanym kluczu.

Parameters

| | | |
|-----------------|----------------------------|-------------------------|
| <code>in</code> | <code>Pierwszy_Elem</code> | - jest to Klucz wektora |
|-----------------|----------------------------|-------------------------|

Returns

(brak)

Definition at line 243 of file drzewobinarne.hh.

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

- C:/Users/Ania/Documents/GitHub/sort/inc/**drzewobinarne.hh**

3.2 dzialanie Class Reference

```
#include <uruchom.hh>
```

Public Member Functions

- `LARGE_INTEGER włączStoper ()`

*Funkcja zapamietujca czas poczatkowy. Funkcja naleca do biblioteki "windows.h", stoper zostaje wczony. Funkcja naleca do funkcji `bool QueryPerformanceCounter(_out LARGE_INTEGER *lpPerformanceCount)`, funkcja ta zwraca wartosc niezerowa jeeli waczenie zako sikcesem, natomiast w przeciwnym wypadku zostanie wyrzucony bd i zwrartosc 0. Dla komputerltiprocessorowych nie ma znaczenia, ktost uywany, mog jedynie rnimalnie czasy.*

- `LARGE_INTEGER wyłączStoper ()`

*Funkcja zapamietujca czas koy. Funkcja naleca do biblioteki "windows.h", stoper zostaje wyczony, aby zosta zmierzony czas wykonania algorytmu w programie, poprzez odje czasu pocztkowego od czasu koego. Funkcja naleca do funkcji `bool QueryPerformanceCounter(_out LARGE_INTEGER *lpPerformanceCount)`, funkcja ta zwraca wartosc niezerowa jeeli wyaczenie zako sikcesem, natomiast w przeciwnym wypadku zostanie wyrzucony bd i zwrartosc 0. Dla komputerltiprocessorowych nie ma znaczenia, ktost uywany, mog jedynie rnimalnie czasy.*

- `void wczytaj (string nazwa)`

Funkcja wczytująca dane do stosu. Funkcja otwiera plik zdefiniowany w głównej funkcji przez użytkownika, sprawdza czy plik został otwarty, jeżeli został otwarty wczytana jest liczba elementów pliku, następnie wczytywane są wszystkie liczby do tablicy.

- void **algorytm** ()

Funkcja wykonująca zadany algorytm na wektorze. Funkcja wykonuje zadany algorytm na wektorze wejściowym. W naszym przypadku pobieranie elementów o zadanym kluczu z tablicy asocjacyjnej.

- int **stoper** ()

Funkcja odpowiedzialna za zmierzenie czasu działania danego algorytmu i jego zwrócenie. return Funkcja zwraca czas pobrania wartości z tablicy.

- void **czyszc** ()

Funkcja czyszczy wektor kluczy, służy do pobierania danych z tablicy asocjacyjnej. Aby nie zostały dopisywane klucze do poprzednich, gdyż wydłużyło by to czas działania.

3.2.1 Detailed Description

Definition at line 23 of file uruchom.hh.

3.2.2 Member Function Documentation

3.2.2.1 void działanie::algorytm ()

Funkcja wykonująca zadany algorytm na wektorze. Funkcja wykonuje zadany algorytm na wektorze wejściowym. W naszym przypadku pobieranie elementów o zadanym kluczu z tablicy asocjacyjnej.

Returns

(brak)

Definition at line 70 of file uruchom.cpp.

3.2.2.2 void działanie::czyszc ()

Funkcja czyszczy wektor kluczy, służy do pobierania danych z tablicy asocjacyjnej. Aby nie zostały dopisywane klucze do poprzednich, gdyż wydłużyło by to czas działania.

Definition at line 91 of file uruchom.cpp.

3.2.2.3 int działanie::stoper ()

Funkcja odpowiedzialna za zmierzenie czasu działania danego algorytmu i jego zwrócenie. return Funkcja zwraca czas pobrania wartości z tablicy.

Definition at line 81 of file uruchom.cpp.

3.2.2.4 void działanie::wczytaj (string nazwa)

Funkcja wczytująca dane do stosu. Funkcja otwiera plik zdefiniowany w głównej funkcji przez użytkownika, sprawdza czy plik został otwarty, jeżeli został otwarty wczytana jest liczba elementów pliku, następnie wczytywane są wszystkie liczby do tablicy.

Parameters

| | | |
|-----------------|--------------------|--|
| <code>in</code> | <code>nazwa</code> | - zmienna, ktostaje wprowadzona przez uytkownika do programu |
|-----------------|--------------------|--|

Returns

(brak)

Definition at line 32 of file uruchom.cpp.

3.2.2.5 LARGE_INTEGER dzialanie::włączStoper ()

Funkcja zapamietujca czas poczatkowy. Funkcja naleca do biblioteki "windows.h", stoper zostaje wczony. Funkcja naleca do funkcji `bool QueryPerformanceCounter(_out LARGE_INTEGER *lpPerformanceCount)`, funkcja ta zwraca wartosc niezerowa jeeli waczenie zako sikcesem, natomiast w przeciwnym wypadku zostanie wyrzucony bd i zwrartosc 0. Dla komputerltiprocessorowych nie ma znaczenia, ktost uywany, mog jedynie rnimalnie czasy.

Definition at line 14 of file uruchom.cpp.

3.2.2.6 LARGE_INTEGER dzialanie::wylaczStoper ()

Funkcja zapamietujca czas koy. Funkcja naleca do biblioteki "windows.h", stoper zostaje wyczony, aby zosta zmierzony czas wykonania algorytmu w programie, poprzez odje czasu pocztkowego od czasu koego. Funkcja naleca do funkcji `bool QueryPerformanceCounter(_out LARGE_INTEGER *lpPerformanceCount)`, funkcja ta zwraca wartosc niezerowa jeeli wyaczenie zako sikcesem, natomiast w przeciwnym wypadku zostanie wyrzucony bd i zwrartosc 0. Dla komputerltiprocessorowych nie ma znaczenia, ktost uywany, mog jedynie rnimalnie czasy.

Definition at line 23 of file uruchom.cpp.

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

- C:/Users/Ania/Documents/GitHub/sort/inc/**uruchom.hh**
- C:/Users/Ania/Documents/GitHub/sort/src/**uruchom.cpp**

3.3 hash_table< Wartosc > Class Template Reference

W klasie **hash_table** (p. 10) wartosciami prywatnymi (dostepnymi tylko przez metody klasy) jest liczba_elementow, ktora definiuje stala wartosc elemntow znajdujacych sie na liscie oraz lista, ktora jest wskaznikiem na korzen elementu. Metodami publicznymi klasy sa funkcje dodajace elemnty, usuwajace i pobierajace element z wektora.

```
#include <thaszuja.h>
```

Public Member Functions

- int **hash** (string Pierwszy_Elem)
Funkcja tworząca hash klucza, czyli za konkretne litery podstawiane są wartości w kodzie ASCII i dzielone jest modulo liczbę liczba_elementow możliwych dostępnych na liście.
- void **dodaj** (obiekt< Wartosc > elem)
Funkcja dodająca element do tablicy Funkcja dodaje kolejne elementy do drzewa binarnego bądź zmienia wartości już istniejących elementów, jeżeli takie istnieją w niej. Funkcja dodaje elemnty pod indeks, który zwraca funkcja hash. Pod każdym indeksem listy jest tworzona osobna lista i kilka elementów mogą się znajdować pod tym samym elementem tablicy.
- bool **usun** (string Pierwszy_Elem)
Usuwanie wskazanego elementu Funkcja usuwa element o wskazanym kluczu.
- Wartosc **pobierz** (string Pierwszy_Elem)
Funkcja pobierająca element z tablicy. Funkcja pobiera wartość, która znajduje się pod wskazanym kluczem.
- void **wypisz** ()

Funkcja pomocnicza stworzona w celu wypisania tablicy haszujacej i sprawdzenia prawidłowego działania funkcji dodania, usunięcia i pobrania elementów.

- int **rozmiar** ()

3.3.1 Detailed Description

template<typename Wartosc>class hash_table< Wartosc >

W klasie **hash_table** (p. 10) wartościami prywatnymi (dostępnymi tylko przez metody klasy) jest liczba_elementow, która definiuje stała wartość elementów znajdujących się na liście oraz lista, która jest wskaźnikiem na korzeń elementu. Metodami publicznymi klasy są funkcje dodające elementy, usuwające i pobierające element z wektora.

Definition at line 44 of file thaszujaca.hh.

3.3.2 Member Function Documentation

3.3.2.1 template<typename Wartosc> void hash_table< Wartosc >::dodaj (obiekt< Wartosc > elem)

Funkcja dodająca element do tablicy Funkcja dodaje kolejne elementy do drzewa binarnego bądź zmienia wartości już istniejących elementów, jeżeli takie istnieją w niej. Funkcja dodaje elementy pod indeks, który zwraca funkcja hash. Pod każdym indeksem listy jest tworzona osobna lista i kilka elementów mogą się znajdować pod tym samym elementem tablicy.

Parameters

| | | |
|----|------|--|
| in | elem | - jest to lista w której znajduje się klucz i jego wartość |
|----|------|--|

Returns

(brak)

Definition at line 115 of file thaszujaca.hh.

3.3.2.2 template<typename Wartosc > int hash_table< Wartosc >::hash (string Pierwszy_Elem)

Funkcja tworząca hash klucza, czyli za konkretne litery podstawiane są wartości w kodzie ASCII i dzielone jest modulo liczbę elementów możliwych dostępnych na liście.

Parameters

| | | |
|----|---------------|---------------------------------------|
| in | Pierwszy_Elem | - string z którego zliczany jest hash |
|----|---------------|---------------------------------------|

Returns

Funkcja zwraca liczbę całkowitą, liczba ta będzie indeksem listy, w której przypisany będzie obiekt.

Definition at line 173 of file thaszujaca.hh.

3.3.2.3 template<typename Wartosc > Wartosc hash_table< Wartosc >::pobierz (string Pierwszy_Elem)

Funkcja pobierająca element z tablicy. Funkcja pobiera wartość, która znajduje się pod wskazanym kluczem.

Parameters

| | | |
|-----------------|----------------------------|---|
| <code>in</code> | <code>Pierwszy_Elem</code> | - jest to Klucz, który wyszukiwana jest jego wartosc. |
|-----------------|----------------------------|---|

Returns

Funkcja zwraca wartosc jaka znajduje sie pod wskazanym kluczem, ktora przyjmuje typ zmiennej w zaleznosci od wykorzystania i zdefiniowania przez uzytkownika.

Definition at line 155 of file thaszujaca.hh.

3.3.2.4 `template<typename Wartosc> int hash_table< Wartosc >::rozmiar ()`

3.3.2.5 `template<typename Wartosc > bool hash_table< Wartosc >::usun (string Pierwszy_Elem)`

Usuwanie wskazanego elementu Funkcja usuwa element o wskazanym kluczu.

Parameters

| | | |
|-----------------|----------------------------|-------------------------|
| <code>in</code> | <code>Pierwszy_Elem</code> | - jest to Klucz wektora |
|-----------------|----------------------------|-------------------------|

Returns

(brak)

Definition at line 135 of file thaszujaca.hh.

3.3.2.6 `template<typename Wartosc> void hash_table< Wartosc >::wypisz () [inline]`

Funkcja pomocnicza stworzona w celu wypisania tablicy haszujacej i sprawdzenia prawidlowego dzialania funkcji dodania, usuniecia i pobrania elementow.

Definition at line 96 of file thaszujaca.hh.

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

- C:/Users/Ania/Documents/GitHub/sort/inc/thaszujaca.hh

3.4 Liczby< Wartosc > Class Template Reference

Klasa ta jest wykorzystana do stworzenia szablonu, czyli aby wartoscia moglabyc dowolna zmienna wykorzystana przez uzytkownika. Docelowo kluczem jest string czyli wartosc indeksujaca tablice. W klasie tej stworzone sa konstruktory i referencje do modyfikowania wartosci oraz odczytu.

```
#include <tasocjacyjna.hh>
```

Public Member Functions

- **Liczby** ()
Konstruktor inicjujacy wartosci.
- **Liczby** (const string &Klucz)
Konstruktor inicjujacy wartosci.
- **Liczby** (const string &Klucz, const Wartosc &Wart_Klucza)
Konstruktor inicjujacy wartosci.
- string & **Pierwszy_Elem** ()

Referencja do pola Klucz, dzięki niej możemy modyfikować zmienną.

- `const string & Pierwszy_Elem () const`

Funkcja umożliwiająca odczytanie wartości znajdującej się pod zmienną Klucz.

- `Wartosc & Drugi_Elem ()`

Referencja do pola Wart_Klucza, dzięki niej możemy modyfikować zmienną.

- `const Wartosc & Drugi_Elem () const`

Funkcja umożliwiająca odczytanie wartości znajdującej się pod zmienną Wart_Klucza.

3.4.1 Detailed Description

`template<typename Wartosc> class Liczby< Wartosc >`

Klasa ta jest wykorzystana do stworzenia szablonu, czyli aby wartością mogła być dowolna zmienna wykorzystana przez użytkownika. Docelowo kluczem jest string czyli wartość indeksująca tablicę. W klasie tej stworzone są konstruktory i referencje do modyfikowania wartości oraz odczytu.

Definition at line 33 of file `tasocjacyjna.hh`.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 `template<typename Wartosc> Liczby< Wartosc >::Liczby () [inline]`

Konstruktor inicjujący wartości.

Definition at line 41 of file `tasocjacyjna.hh`.

3.4.2.2 `template<typename Wartosc> Liczby< Wartosc >::Liczby (const string & Klucz) [inline]`

Konstruktor inicjujący wartości.

Definition at line 46 of file `tasocjacyjna.hh`.

3.4.2.3 `template<typename Wartosc> Liczby< Wartosc >::Liczby (const string & Klucz, const Wartosc & Wart_Klucza) [inline]`

Konstruktor inicjujący wartości.

Definition at line 51 of file `tasocjacyjna.hh`.

3.4.3 Member Function Documentation

3.4.3.1 `template<typename Wartosc> Wartosc& Liczby< Wartosc >::Drugi_Elem () [inline]`

Referencja do pola Wart_Klucza, dzięki niej możemy modyfikować zmienną.

Returns

Wart_Klucza - zmienna typu Wartosc(nasz szablon), która jest wartością Klucza.

Definition at line 71 of file `tasocjacyjna.hh`.

3.4.3.2 `template<typename Wartosc> const Wartosc& Liczby< Wartosc >::Drugi_Elem () const [inline]`

Funkcja umożliwiająca odczytanie wartości znajdującej się pod zmienną Wart_Klucza.

Returns

Wart_Klucza - zmienna typu Wartosc(nasz szablon), ktora jest wartosc Klucza.

Definition at line 78 of file tasocjacyjna.hh.

3.4.3.3 `template<typename Wartosc> string& Liczby< Wartosc >::Pierwszy_Elem () [inline]`

Referencja do pola Klucz, dzięki niej możemy modyfikować zmienną.

Returns

Klucz - string, który jest indeksem tablicy

Definition at line 57 of file tasocjacyjna.hh.

3.4.3.4 `template<typename Wartosc> const string& Liczby< Wartosc >::Pierwszy_Elem () const [inline]`

Funkcja umożliwiająca odczytanie wartości znajdującej się pod zmienną Klucz.

Returns

Klucz - string, który jest indeksem tablicy

Definition at line 64 of file tasocjacyjna.hh.

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

- C:/Users/Ania/Documents/GitHub/sort/inc/**tasocjacyjna.hh**

3.5 obiekt< Wartosc > Struct Template Reference

Struktura obiekt w której zdefiniowany jest element tablicy czyli klucz, który jest stringiem i jego wartość, która jest szablonem.

```
#include <thaszujaca.hh>
```

Public Attributes

- string **Klucz**
- Wartosc **Wart_Klucza**

3.5.1 Detailed Description

```
template<typename Wartosc> struct obiekt< Wartosc >
```

Struktura obiekt w której zdefiniowany jest element tablicy czyli klucz, który jest stringiem i jego wartość, która jest szablonem.

Definition at line 29 of file thaszujaca.hh.

3.5.2 Member Data Documentation

3.5.2.1 `template<typename Wartosc> string obiekt< Wartosc >::Klucz`

Definition at line 30 of file thaszujaca.hh.

3.5.2.2 template<typename Wartosc> Wartosc obiekt< Wartosc >::Wart_Klucza

Definition at line 31 of file thaszujaca.hh.

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

- C:/Users/Ania/Documents/GitHub/sort/inc/thaszujaca.hh

3.6 Tab_Asocjacyjna< Wartosc > Class Template Reference

Klasa **Tab_Asocjacyjna** (p. 15), to w niej tworzony jest wektor, w którym przechowywane są wszystkie elementy. Indeksami tego wektora są Klucze (string), którym odpowiadają wartości w zależności, jaka użytkownik zdefiniuje zmienna. Znajdują się też również różne funkcje (metody), które mogą zostać użyte przy wykorzystywaniu tablicy asocjacyjnej.

```
#include <tasocjacyjna.hh>
```

Public Member Functions

- **Tab_Asocjacyjna** ()
*Konstruktor **Tab_Asocjacyjna** (p. 15), w którym definiowana jest liczba elementów znajdujących się w tablicy przy inicjowaniu, a więc wartość ta wynosi 0.*
- void **dodaj** (string Pierwszy_Elem, Wartosc Drugi_Elem)
Funkcja dodająca element do tablicy Funkcja dodaje kolejne elementy do tablicy asocjacyjnej bądź zmienia wartości już istniejących elementów, jeżeli takie istnieją w niej.
- Wartosc **pobierz** (string Pierwszy_Elem)
Funkcja pobierająca element z tablicy. Funkcja pobiera wartość, która znajduje się pod wskazanym kluczem.
- void **usun** (string Pierwszy_Elem)
Usuwanie wskazanego elementu Funkcja usuwa element o wskazanym kluczu, czyli indeksie tablicy asocjacyjnej.
- string **znajdz** (Wartosc Drugi_Elem)
Znajdowanie Klucza o podanej jego wartości. Funkcja znajduje Klucz, którego elementem jest szukana wartość. Funkcja działa podobnie jak funkcja pobierz, tylko zamiast wartości znajduje klucz.
- int **rozmiar** ()
Sprawdzanie rozmiaru tablicy.
- bool **Czy_Pusta** ()
Funkcja sprawdzająca czy tablica jest pusta.
- Wartosc & **operator[]** (string Pierwszy_Elem)
Przeciążenie operatora [] Przeciążenie operatora za pomocą którego możemy odwoływać się do wskazanego elementu tablicy asocjacyjnej. Wartość wektora przy użyciu przeciążenia przyjmuje nam typ Wartosc, który użytkownik definiuje.

3.6.1 Detailed Description

```
template<typename Wartosc>class Tab_Asocjacyjna< Wartosc >
```

Klasa **Tab_Asocjacyjna** (p. 15), to w niej tworzony jest wektor, w którym przechowywane są wszystkie elementy. Indeksami tego wektora są Klucze (string), którym odpowiadają wartości w zależności, jaka użytkownik zdefiniuje zmienna. Znajdują się też również różne funkcje (metody), które mogą zostać użyte przy wykorzystywaniu tablicy asocjacyjnej.

Definition at line 107 of file tasocjacyjna.hh.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 `template<typename Wartosc > Tab_Asocjacyjna< Wartosc >::Tab_Asocjacyjna () [inline]`

Konstruktor **Tab_Asocjacyjna** (p. 15), w którym definiowana jest liczba elementów znajdujących się w tablicy przy inicjowaniu, a więc wartość ta wynosi 0.

Definition at line 124 of file `tasocjacyjna.hh`.

3.6.3 Member Function Documentation

3.6.3.1 `template<typename Wartosc > bool Tab_Asocjacyjna< Wartosc >::Czy_Pusta ()`

Funkcja sprawdzająca czy tablica jest pusta.

Returns

Jeżeli funkcja jest pusta funkcja zwraca nam true w przeciwnym wypadku false.

Definition at line 222 of file `tasocjacyjna.hh`.

3.6.3.2 `template<typename Wartosc > void Tab_Asocjacyjna< Wartosc >::dodaj (string Pierwszy_Elem, Wartosc Drugi_Elem)`

Funkcja dodająca element do tablicy Funkcja dodaje kolejne elementy do tablicy asocjacyjnej bądź zmienia wartości już istniejących elementów, jeżeli takie istnieją w niej.

Parameters

| | | |
|-----------------|----------------------------|--|
| <code>in</code> | <code>Pierwszy_Elem</code> | - jest to klucz czyli indeks tablicy |
| <code>in</code> | <code>Drugi_Elem</code> | - zmienna ta jest wartością klucza, może przyjmować różne typy |

Returns

(brak)

Definition at line 230 of file `tasocjacyjna.hh`.

3.6.3.3 `template<typename Wartosc > Wartosc & Tab_Asocjacyjna< Wartosc >::operator[] (string Pierwszy_Elem)`

Przeciążenie operatora [] Przeciążenie operatora za pomocą którego możemy odwoływać się do wskazanego elementu tablicy asocjacyjnej. Wartość wektora przy użyciu przeciążenia przyjmuje nam typ `Wartosc`, który użytkownik definiuje.

Parameters

| | | |
|-----------------|----------------------------|---|
| <code>in</code> | <code>Pierwszy_Elem</code> | - jest to klucz naszej tablicy, czyli string. To dzięki niemu mamy dostęp do wartości klucza. |
|-----------------|----------------------------|---|

Returns

Przeciążenie zwraca wartość do elementu znajdującego się o podanym kluczu.

Definition at line 183 of file `tasocjacyjna.hh`.

3.6.3.4 `template<typename Wartosc > Wartosc Tab_Asocjacyjna< Wartosc >::pobierz (string Pierwszy_Elem)`

Funkcja pobierająca element z tablicy. Funkcja pobiera wartość, która znajduje się pod wskazanym kluczem.

Parameters

| | | |
|-----------|----------------------|---|
| <i>in</i> | <i>Pierwszy_Elem</i> | - jest to Klucz wektora, czyli indeks tablicy |
|-----------|----------------------|---|

Returns

Funkcja zwraca wartosc jaka znajduje sie pod wskazanym kluczem, która przyjmuje typ zmiennej w zależności od wykorzystania i zdefiniowania przez użytkownika.

Definition at line 267 of file `tasocjacyjna.hh`.

3.6.3.5 `template<typename Wartosc > int Tab_Asocjacyjna< Wartosc >::rozmiar ()`

Sprawdzanie rozmiaru tablicy.

Returns

Funkcja zwraca liczbe całkowita, która jest rozmiarem całej tablicy asocjacyjnej

Definition at line 216 of file `tasocjacyjna.hh`.

3.6.3.6 `template<typename Wartosc > void Tab_Asocjacyjna< Wartosc >::usun (string Pierwszy_Elem)`

Usuwanie wskazanego elementu Funkcja usuwa element o wskazanym kluczu, czyli indeksie tablicy asocjacyjnej.

Parameters

| | | |
|-----------|----------------------|---|
| <i>in</i> | <i>Pierwszy_Elem</i> | - jest to Klucz wektora, czyli indeks tablicy |
|-----------|----------------------|---|

Returns

(brak)

Definition at line 300 of file `tasocjacyjna.hh`.

3.6.3.7 `template<typename Wartosc > string Tab_Asocjacyjna< Wartosc >::znajdz (Wartosc Drugi_Elem)`

Znajdowanie Klucza o podanej jego wartosci. Funkcja znajduje Klucz, którego elementem jest szukana wartosc. Funkcja działa podobnie jak funkcja `pobierz`, tylko zamiast wartosci znajduje klucz.

Parameters

| | | |
|-----------|-------------------|--|
| <i>in</i> | <i>Drugi_Elem</i> | - do tej wartosci, która przyjmuje dolowny typ poszukujemy Klucz |
|-----------|-------------------|--|

Returns

Funkcja zwraca nazwe klucza, czyli indeks wektora.

Definition at line 318 of file `tasocjacyjna.hh`.

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

- `C:/Users/Ania/Documents/GitHub/sort/inc/tasocjacyjna.hh`

3.7 wezel< Wartosc > Struct Template Reference

Struktura wezel w ktorej zdefiniowany jest element tablicy czyli klucz, ktory jest stringiem i jego wartosc, ktora jest szablonem. W strukturze tej znajduje sie rowniez wskaznik do konkretnego wezla (rodzica i jego dzieci), co ulatwia nam poruszanie sie po drzewie.

```
#include <drzewobinarne.hh>
```

Public Attributes

- **wezel** * **lewy**
- **wezel** * **prawy**
- **wezel** * **rodzic**
- string **Klucz**
- Wartosc **Wart_Klucza**

3.7.1 Detailed Description

```
template<typename Wartosc> struct wezel< Wartosc >
```

Struktura wezel w ktorej zdefiniowany jest element tablicy czyli klucz, ktory jest stringiem i jego wartosc, ktora jest szablonem. W strukturze tej znajduje sie rowniez wskaznik do konkretnego wezla (rodzica i jego dzieci), co ulatwia nam poruszanie sie po drzewie.

Definition at line 29 of file drzewobinarne.hh.

3.7.2 Member Data Documentation

3.7.2.1 `template<typename Wartosc> string wezel< Wartosc >::Klucz`

Definition at line 31 of file drzewobinarne.hh.

3.7.2.2 `template<typename Wartosc> wezel* wezel< Wartosc >::lewy`

Definition at line 30 of file drzewobinarne.hh.

3.7.2.3 `template<typename Wartosc> wezel* wezel< Wartosc >::prawy`

Definition at line 30 of file drzewobinarne.hh.

3.7.2.4 `template<typename Wartosc> wezel* wezel< Wartosc >::rodzic`

Definition at line 30 of file drzewobinarne.hh.

3.7.2.5 `template<typename Wartosc> Wartosc wezel< Wartosc >::Wart_Klucza`

Definition at line 32 of file drzewobinarne.hh.

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

- C:/Users/Ania/Documents/GitHub/sort/inc/**drzewobinarne.hh**

Chapter 4

File Documentation

4.1 C:/Users/Ania/Documents/GitHub/sort/inc/drzewobinarne.hh File Reference

Plik naglowkowy wykorzystywany do stworzenia drzewa binarnego do przechowywania tablicy o zadanym kluczu i wartosci. W pliku stworzona jest struktura gdzie zdefiniowany jest element wektora oraz klasa gdzie wystepuja wszystkie funkcje operujace na drzewie.

```
#include <iostream>
#include <string>
```

Classes

- struct **wezel**< **Wartosc** >

Struktura wezel w ktorej zdefiniowany jest element tablicy czyli klucz, ktory jest stringiem i jego wartosc, ktora jest szablonem. W strukturze tej znajduje sie rowniez wskaznik do konkretnego wezla (rodzica i jego dzieci), co ulatwia nam poruszanie sie po drzewie.

- class **drzewo**< **Wartosc** >

W klasie drzewo wartosciami prywatnymi (dostepnymi tylko przez metody klasy) jest licznik liczba_elementow, ktory zlicza ile jest elementow w drzewie binarnym oraz wektor, ktory jest wskaznikiem na korzen elementu. Metodami publicznymi klasy sa funkcje dodajace elemnty, usuwajace i pobierajace element z wektora.

4.1.1 Detailed Description

Plik naglowkowy wykorzystywany do stworzenia drzewa binarnego do przechowywania tablicy o zadanym kluczu i wartosci. W pliku stworzona jest struktura gdzie zdefiniowany jest element wektora oraz klasa gdzie wystepuja wszystkie funkcje operujace na drzewie.

Definition in file **drzewobinarne.hh**.

4.2 C:/Users/Ania/Documents/GitHub/sort/inc/tasocjacyjna.hh File Reference

Plik naglowkowy tasocjacyjna, ktora sluzi do stowrzenia tablicy asocjacyjnej, tablica ta posiada klucz, ktory jest argumentem wektora, w moim przypadku oraz wartosc klucza, czyli jaka wartosc znajduje sie pod kluczem. W tym pliku stworzone sa dwie klasy. Klasa glowna **Tab_Asocjacyjna** (p. 15) korzysta z malej klasy **Liczby** (p. 12), ktora jest pomocnicza i reprezentujaca elemnty.

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
```

Classes

- class **Liczby**< **Wartosc** >

Klasa ta jest wykorzystana do stworzenia szablonu, czyli aby wartoscia moglabyc dowolna zmienna wykorzystana przez uzytkownika. Docelowo kluczem jest string czyli wartosc indeksujaca tablice. W klasie tej stworzone sa konstruktory i referencje do modyfikowania wartosci oraz odczytu.

- class **Tab_Asocjacyjna**< **Wartosc** >

*Klasa **Tab_Asocjacyjna** (p. 15), to w niej tworzony jest wektor, w którym przechowywane sa wszystkie elementy. Indeksami tego wektora sa Klucze (string), ktorym odpowiadaja wartosci w zaleznosci jaka uzytkownik zdefiniuje zmienna. Znajduja sie tez rowniez rozne funkcje/metody, ktore moga zostac uzyte przy wykorzystywaniu tablicy asocjacyjnej.*

Functions

- template<typename **Wartosc** >

bool **operator**< (**Liczby**< **Wartosc** > pierwszy, **Liczby**< **Wartosc** > drugi)

Przeciazanie operatora < Przeciazanie porownujace klucze, czyli stringi. Przeciazanie to wykorzystane jest przy sortowaniu uzytym przy kazdym dodawaniu elementow do tablicy.

4.2.1 Detailed Description

Plik naglowkowy **tasocjacyjna**, ktora sluzy do stowrzenia tablicy asocjacyjnej, tablica ta posiada klucz, ktory jest argumentem wektora, w moim przypadku oraz wartosc klucza, czyli jaka wartosc znajduje sie pod kluczem. W tym pliku stworzone sa dwie klasy. Klasa glowna **Tab_Asocjacyjna** (p. 15) korzysta z malej klasy **Liczby** (p. 12), ktora jest pomocnicza i reprezentujaca elemnty.

Definition in file **tasocjacyjna.hh**.

4.2.2 Function Documentation

4.2.2.1 template<typename **Wartosc** > bool operator< (**Liczby**< **Wartosc** > pierwszy, **Liczby**< **Wartosc** > drugi)

Przeciazanie operatora < Przeciazanie porownujace klucze, czyli stringi. Przeciazanie to wykorzystane jest przy sortowaniu uzytym przy kazdym dodawaniu elementow do tablicy.

Parameters

| | | |
|----|-----------------|---|
| in | <i>pierwszy</i> | - element porownywany nalzezacy do klasy Liczby (p. 12) o szablonie Wartosc |
| in | <i>drugi</i> | - element porownywany nalzezacy do klasy Liczby (p. 12) o szablonie Wartosc |

Returns

Funkcja zwraca true jezeli pierwszy porownywany element jest mniejszy od drugiego w przeciwnym wypadku false.

Definition at line 93 of file **tasocjacyjna.hh**.

4.3 C:/Users/Ania/Documents/GitHub/sort/inc/thaszujaca.hh File Reference

Plik naglowkowy wykorzystywany do stworzenia tablicy haszujacej do przechowywania tablicy o zadanym kluczu i wartosci. W pliku stworzona jest struktura gdzie zdefiniowany jest element wektora oraz klasa gdzie wystepuja wszystkie funkcje operujace na tabilcy.

```
#include <iostream>
#include <string>
#include <list>
```

Classes

- struct **obiekt**< **Wartosc** >

Struktura obiekt w ktorej zdefiniowany jest element tablicy czyli klucz, który jest stringiem i jego wartosc, która jest szablonem.

- class **hash_table**< **Wartosc** >

*W klasie **hash_table** (p. 10) wartosciami prywatnymi (dostepnymi tylko przez metody klasy) jest liczba_elementow, która definiuje stała wartosc elementow znajdujacych sie na liscie oraz lista, która jest wskaźnikiem na korzen elementu. Metodami publicznymi klasy sa funkcje dodajace elemnty, usuwajace i pobierajace element z wektora.*

4.3.1 Detailed Description

Plik naglowkowy wykorzystywany do stworzenia tablicy haszujacej do przechowywania tablicy o zadanym kluczu i wartosci. W pliku stworzona jest struktura gdzie zdefiniowany jest element wektora oraz klasa gdzie wystepuja wszystkie funkcje operujace na tablicy.

Definition in file **thaszujaca.hh**.

4.4 C:/Users/Ania/Documents/GitHub/sort/inc/uruchom.hh File Reference

Klasa odpowiadajca za dziaanie i sprawdzenie zoonoci obliczeniowej kadego rodzaju tablicy asocjacyjnej. To w tej kalsie włączamy i wyłączamy stoper, wczytujemy plik z danymi i wykonujemy algorytm.

```
#include <windows.h>
#include "thaszujaca.hh"
#include <vector>
```

Classes

- class **dzialanie**

4.4.1 Detailed Description

Klasa odpowiadajca za dziaanie i sprawdzenie zoonoci obliczeniowej kadego rodzaju tablicy asocjacyjnej. To w tej kalsie włączamy i wyłączamy stoper, wczytujemy plik z danymi i wykonujemy algorytm.

Definition in file **uruchom.hh**.

4.5 C:/Users/Ania/Documents/GitHub/sort/src/main.cpp File Reference

Funkcja zmieniajaca nazwy dla roznnych wielkosci problemu, aby za kazdym razem nie trzeba bylo wczytywac na nowo pliku.

```
#include "uruchom.hh"
#include <sstream>
#include <fstream>
#include <time.h>
#include <string>
#include <iostream>
```

Functions

- string **zmien_nazwy** (int i)
- void **dzialaj** ()

Funkcja otwiera plik do zapisania wnioskow i danych z obliczonym czasem pobierania wartosci z tablicy asocjacyjnej. W funkcji tej wykorzystane jest rowniez powtorzenie danego problemu, aby miec usredniony czas.

- int **main** ()

Przyklad uzycia i poprawnego dzialania tablicy asocjacyjnej.

4.5.1 Detailed Description

Funkcja zmieniajaca nazwy dla roznnych wielkosci problemu, aby za kazdym razem nie trzeba bylo wczytywac na nowo pliku.

Parameters

| | | |
|-----------------|----------------|--|
| <code>in</code> | <code>i</code> | - zmienna wykorzystana aby zdefiniowac jaka wielkosc problemu ma byc sprawdzona w programie. |
|-----------------|----------------|--|

Returns

Funkcja zwraca nazwy plikow do wczytania.

Definition in file **main.cpp**.

4.5.2 Function Documentation

4.5.2.1 void dzialaj ()

Funkcja otwiera plik do zapisania wnioskow i danych z obliczonym czasem pobierania wartosci z tablicy asocjacyjnej. W funkcji tej wykorzystane jest rowniez powtorzenie danego problemu, aby miec usredniony czas.

Definition at line 45 of file main.cpp.

4.5.2.2 int main ()

Przyklad uzycia i poprawnego dzialania tablicy asocjacyjnej.

Definition at line 76 of file main.cpp.

4.5.2.3 string zmien_nazwy (int i)

Definition at line 21 of file main.cpp.

4.6 C:/Users/Ania/Documents/GitHub/sort/src/uruchom.cpp File Reference

```
#include "uruchom.hh"  
#include <fstream>  
#include <iostream>  
#include <cstdlib>
```

Index

algorytm
 działanie, 9

C:/Users/Ania/Documents/GitHub/sort/inc/drzewobinarne.-hash_table< Wartosc >, 10
 hh, 19

C:/Users/Ania/Documents/GitHub/sort/inc/tasocjacyjna.-
 hh, 19

C:/Users/Ania/Documents/GitHub/sort/inc/thaszujaca.-
 hh, 20

C:/Users/Ania/Documents/GitHub/sort/inc/uruchom.hh,
 21

C:/Users/Ania/Documents/GitHub/sort/src/main.cpp, 21

C:/Users/Ania/Documents/GitHub/sort/src/uruchom.-
 cpp, 23

Czy_Pusta
 Tab_Asocjacyjna, 16

czysc
 działanie, 9

dodaj
 drzewo, 6
 hash_table, 11
 Tab_Asocjacyjna, 16

Drugi_Elem
 Liczby, 13

drzewo
 dodaj, 6
 drzewo, 6
 Najmniejszy_Element, 6
 pobierz, 6
 Posz_Nastep, 6
 rozmiar, 8
 usun, 8

drzewo< Wartosc >, 5

dzialaj
 main.cpp, 22

działanie, 8
 algorytm, 9
 czysc, 9
 stoper, 9
 wczytaj, 9
 włączStoper, 10
 wylączStoper, 10

hash
 hash_table, 11

hash_table
 dodaj, 11
 hash, 11
 pobierz, 11

rozmiar, 12

usun, 12

wypisz, 12

Klucz
 obiekt, 14
 wezel, 18

lewy
 wezel, 18

Liczby
 Drugi_Elem, 13
 Liczby, 13
 Pierwszy_Elem, 14

Liczby< Wartosc >, 12

main
 main.cpp, 22

main.cpp
 działaj, 22
 main, 22
 zmien_nazwy, 22

Najmniejszy_Element
 drzewo, 6

obiekt
 Klucz, 14
 Wart_Klucza, 14

obiekt< Wartosc >, 14

operator<
 tasocjacyjna.hh, 20

operator[]
 Tab_Asocjacyjna, 16

Pierwszy_Elem
 Liczby, 14

pobierz
 drzewo, 6
 hash_table, 11
 Tab_Asocjacyjna, 16

Posz_Nastep
 drzewo, 6

prawy
 wezel, 18

rodzic
 wezel, 18

rozmiar
 drzewo, 8

- hash_table, 12
- Tab_Asocjacyjna, 17
- stoper
 - dzialanie, 9
- Tab_Asocjacyjna
 - Czy_Pusta, 16
 - dodaj, 16
 - operator[], 16
 - pobierz, 16
 - rozmiar, 17
 - Tab_Asocjacyjna, 16
 - Tab_Asocjacyjna, 16
 - usun, 17
 - znajdz, 17
- Tab_Asocjacyjna< Wartosc >, 15
- tasocjacyjna.hh
 - operator<, 20
- usun
 - drzewo, 8
 - hash_table, 12
 - Tab_Asocjacyjna, 17
- Wart_Klucza
 - obiekt, 14
 - wezel, 18
- wczytaj
 - dzialanie, 9
- wezel
 - Klucz, 18
 - lewy, 18
 - prawy, 18
 - rodzic, 18
 - Wart_Klucza, 18
- wezel< Wartosc >, 18
- wlaczStoper
 - dzialanie, 10
- wylaczStoper
 - dzialanie, 10
- wypisz
 - hash_table, 12
- zmien_nazwy
 - main.cpp, 22
- znajdz
 - Tab_Asocjacyjna, 17