Tablica asocjacyjna

Generated by Doxygen 1.8.6

Mon Apr 7 2014 01:21:44

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 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** ()
  Konstrukor inicjujacy wartosci.
- **Liczby** (const string &Klucz)
  Konstrukor inicjujacy wartosci.
- **Liczby** (const string &Klucz, const Wartosc &Wart_Klucza)
  Konstrukor inicjujacy wartosci.
- string & **Pierwszy_Elem** ()
  Referencja do pola Klucz, dzieki niej mozemy modyfikowac zmienna.
- const string & **Pierwszy_Elem** () const
  Funkcja umozliwiajaca odczytanie wartosci znajdujacej sie pod zmienna Klucz.
- Wartosc & **Drugi_Elem** ()
  Referencja do pola Wart_Klucza, dzieki niej mozemy modyfikowac zmienna.
- const Wartosc & **Drugi_Elem** () const
  Funkcja umozliwiajaca odczytanie wartosci znajdujacej sie pod zmienna Wart_Klucza.

3.1.1 Detailed Description

template<typename Wartosc>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.

3.1.2 Member Function Documentation

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

Referencja do pola Wart_Klucza, dzieki niej mozemy modyfikowac zmienna.

Returns

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

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

Funkcja umozliwiajaca odczytanie wartosci znajdujacej sie pod zmienna Wart_Klucza.

Returns

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

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

Referencja do pola Klucz, dzieki niej mozemy modyfikowac zmienna.

Returns

Klucz - string, ktory jest indeksem tablicy

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

Funkcja umozliwiajaca odczytanie wartosci znajdujacej sie pod zmienna Klucz.

Returns

Klucz - string, ktory jest indeksem tablicy

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

- inc/**tasocjacyjna.hh**

3.2 Tab_Asocjacyjna< Wartosc > Class Template Reference

Klasa **Tab_Asocjacyjna** (p. 6), to w niej tworzony jest wektor, w ktorym 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.

`#include <tasocjacyjna.hh>`

Public Member Functions

- **Tab_Asocjacyjna** ()

  *Konstruktor **Tab_Asocjacyjna** (p. 6), w kotrym definiowana jest liczba elementow znjadujacych sie w tablicy przy inicjacjowaniu, a wiec wartosc ta wynosi 0.*
- void **dodaj** (string Pierwszy_Elem, Wartosc Drugi_Elem)

  Funkcja dodajaca element do tablicy Funkcja dodaje kolejne elementy do tablicy asocjacyjnej badz zmienia wartosci juz istniejcych elementow, jezeli takie istnieja w niej.
- Wartosc **pobierz** (string Pierwszy_Elem)

  Funkcja pobieraca elemnt 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, czyli indeksie tablicy asocjacyjnej.

- string **znajdz** (Wartosc Drugi_Elem)

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

- int **rozmiar** ()

  Sprawdzanie rozmiaru tablicy.

- bool **Czy_Pusta** ()

  Funkcja sprawdzajaca czy tablica jest pusta.

- Wartosc & **operator[]** (string Pierwszy_Elem)

  Przeciazenie operatora [] Przeciazenie operatora za pomoca ktorego mozemy odwolywac sie do wskazanego elementu tablicy asocjacyjnej. Wartosc wektora przy uzyciu przeciazenia przyjmuje nam typ Wartosc, ktory uzytkownik definiuje.

3.2.1 Detailed Description

template<typename Wartosc>class Tab_Asocjacyjna< Wartosc >

Klasa **Tab_Asocjacyjna** (p. 6), to w niej tworzony jest wektor, w ktorym 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.

3.2.2 Member Function Documentation

3.2.2.1 template<typename Wartosc > bool Tab_Asocjacyjna< Wartosc >::Czy_Pusta ( )

Funkcja sprawdzajaca czy tablica jest pusta.

Returns

Jezeli funkcja jest pusta funkcja zwraca nam true w przciwnym wypadku flase.

3.2.2.2 template<typename Wartosc > void Tab_Asocjacyjna< Wartosc >::dodaj (string *Pierwszy_Elem,* Wartosc *Drugi_Elem*)

Funkcja dodajaca element do tablicy Funkcja dodaje kolejne elementy do tablicy asocjacyjnej badz zmienia wartosci juz istniejcych elementow, jezeli takie istnieja w niej.

Parameters

in	*Pierwszy_Elem*	- jest to Klucz czyli indeks tablicy
in	*Drugi_Elem*	- zmienna ta jest wartosc klucza, moze przyjmowac rozne typy

Returns

(brak)

3.2.2.3 template<typename Wartosc > Wartosc & Tab_Asocjacyjna< Wartosc >::operator[] (string *Pierwszy_Elem*)

Przeciazenie operatora [] Przeciazenie operatora za pomoca ktorego mozemy odwolywac sie do wskazanego elementu tablicy asocjacyjnej. Wartosc wektora przy uzyciu przeciazenia przyjmuje nam typ Wartosc, ktory uzytkownik definiuje.

Parameters

in	*Pierwszy_Elem*	- jest to klucz naszej tablicy, czyli string. To dzieki niemu mamy dostep do wartosci klucza.

Returns

Przeciazenie zwraca wartosc do elemntu znajdujacego sie o poadanym kluczu.

3.2.2.4 template<typename Wartosc> Wartosc Tab_Asocjacyjna< Wartosc >::pobierz (string *Pierwszy_Elem*)

Funkcja pobieraca elemnt z tablicy. Funkcja pobiera wartosc, ktora znajduje sie pod wskazanym kluczem.

Parameters

in	*Pierwszy_Elem*	- jest to Klucz wektora, czyli indeks tablicy

Returns

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

3.2.2.5 template<typename Wartosc> int Tab_Asocjacyjna< Wartosc >::rozmiar ( )

Sprawdzanie rozmiaru tablicy.

Returns

Funkcja zwraca liczbe calkowita, ktora jest rozmiarem calej tablicy asocjacyjnej

3.2.2.6 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

in	*Pierwszy_Elem*	- jest to Klucz wektora, czyli indeks tablicy

Returns

(brak)

3.2.2.7 template<typename Wartosc> string Tab_Asocjacyjna< Wartosc >::znajdz (Wartosc *Drugi_Elem*)

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

Parameters

in	*Drugi_Elem*	- do tej wartosci, ktora przyjmuje dolowny typ poszukujemy Klucz

Returns

Funkcja zwraca nazwe klucza, czyli indeks wektora.

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

- inc/**tasocjacyjna.hh**

Chapter 4

File Documentation

4.1 inc/tasocjacyjna.hh File Reference

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. 6) korzysta z malej klasy **Liczby** (p. 5), ktora jest pomocnicza i rezprezentujaca 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. 6), to w niej tworzony jest wektor, w ktorym 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)

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

4.1.1 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. 6) korzysta z malej klasy **Liczby** (p. 5), ktora jest pomocnicza i rezprezentujaca elemnty.

4.1.2 Function Documentation

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

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

Parameters

in	*pierwszy*	- element porownywany nalzezacy do klasy **Liczby** (p. 5) o szablonie Wartosc
in	*drugi*	- element porownywany nalzezacy do klasy **Liczby** (p. 5) o szablonie Wartosc

Returns

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

4.2 src/main.cpp File Reference

Funkcja glowna ktorej glownym zalozeniem jest wczytanie plikow z rozna wielkoscia elementow znajdujacych siliku, obliczenie sredniej wartosci czasu, w jakim zostaje wykonany algorytm (w naszym przypadku pomnozenie przez 2),nastepnie program porownuje poprawnosc wykoannia mnozenia z plikiem sprawdzajacym. Uzytkownik musi w programie zdefiniowac: liczbe powtorzen (zmienna j), ilosc plikow - do ilu wykonywane jest mnozenie (zmienna i), nazwy plikow (string czesc_1, i, czesc_2 - wszystko opcjonalnie).

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

Functions

- int **main** ()

4.2.1 Detailed Description

Funkcja glowna ktorej glownym zalozeniem jest wczytanie plikow z rozna wielkoscia elementow znajdujacych siliku, obliczenie sredniej wartosci czasu, w jakim zostaje wykonany algorytm (w naszym przypadku pomnozenie przez 2),nastepnie program porownuje poprawnosc wykoannia mnozenia z plikiem sprawdzajacym. Uzytkownik musi w programie zdefiniowac: liczbe powtorzen (zmienna j), ilosc plikow - do ilu wykonywane jest mnozenie (zmienna i), nazwy plikow (string czesc_1, i, czesc_2 - wszystko opcjonalnie). Przyklad uzycia i poprawnego dzialania tablicy asocjacyjnej.

Returns

(brak)

Index