

Tablica asocjacyjna

Generated by Doxygen 1.8.1.2

Wed May 21 2014 12:25:18

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	Dzialanie Class Reference	3
2.1.1	Detailed Description	3
2.1.2	Member Function Documentation	3
2.1.2.1	sprawdz	3
2.1.2.2	wczytajDaneWejsciowe	4
2.1.2.3	wykonajAlgorytm	4
2.2	elem Struct Reference	4
2.2.1	Detailed Description	4
2.3	Para< Wartosc > Class Template Reference	4
2.3.1	Detailed Description	5
2.3.2	Member Function Documentation	5
2.3.2.1	klucz	5
2.3.2.2	klucz	5
2.3.2.3	wartosc	5
2.3.2.4	wartosc	6
2.4	Tab_asocjacyjna< Wartosc > Class Template Reference	6
2.4.1	Constructor & Destructor Documentation	6
2.4.1.1	Tab_asocjacyjna	6
2.4.2	Member Function Documentation	6
2.4.2.1	czy_pusta	6
2.4.2.2	dodaj	7
2.4.2.3	pobierz	7
2.4.2.4	rozmiar	7
2.4.2.5	usun	7
2.4.2.6	zmien	7
2.5	tab_hash Class Reference	8
2.5.1	Detailed Description	8

2.5.2	Member Function Documentation	8
2.5.2.1	dodaj	8
2.5.2.2	pobierz	8
2.5.2.3	usun	8
2.5.2.4	wypisz	9
2.6	Tablica_asocjacyjna< Wartosc > Class Template Reference	9
2.6.1	Detailed Description	10
2.6.2	Constructor & Destructor Documentation	10
2.6.2.1	Tablica_asocjacyjna	10
2.6.3	Member Function Documentation	10
2.6.3.1	czy_znalazlo	10
2.6.3.2	dodaj	10
2.6.3.3	usun	10
2.6.3.4	zmien	11
2.6.3.5	znajdz	11
2.7	Tester Class Reference	11
2.7.1	Detailed Description	12
2.7.2	Member Function Documentation	12
2.7.2.1	zamienNazwy	12
2.8	wezel< Wartosc > Struct Template Reference	12
2.8.1	Detailed Description	12

Chapter 1

Class Index

1.1 Class List

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

Dzialanie	Klasa modelujaca gowna czesc programu	3
elem	Struktura modelujca pare klucz-wartosc	4
Para< Wartosc >	Klasa modeluje pare klucz-wartosc, ktora jest podstawowym elementem tablicy asocjacyjnej Zar typ Klucz jak i Wartosc musz miec konstruktory parametryczne, ktore inicjuja ich wartosci . . .	4
Tab_asocjacyjna< Wartosc >		6
tab_hash	Klasa modelujaca tablice haszujaca z funkcja mieszajaca	8
Tablica_asocjacyjna< Wartosc >	Klasa modeluje pojecie tablicy asocjacyjnej Tablica asocjacyjna jest to tablica, w ktorej do wartosci mozemy sie odwolywac poprzez klucze. Odwolanie poprez indeks rowniez jest mo- zliwe	9
Tester	Klasa Tester Klasa modeluje narzedzie do benchmarkowania algorytmu. Poprzez wielokrotne wynonywanie algorytmu mozemy wyciagnac sredni czas dzialania, co jest dokladniejszym po- miarem	11
wezel< Wartosc >	Klasa modeluje pare klucz-wartosc, ktora jest podstawowym elementem tablicy asocjacyjnej Zar typ Klucz jak i Wartosc musz miec konstruktory parametryczne, ktore inicjuja ich wartosci . . .	12

Chapter 2

Class Documentation

2.1 Działanie Class Reference

Klasa modelująca gówna czesc programu.

```
#include <dzialanie.hh>
```

Public Member Functions

- void `wczytajDaneWejscowe` (string nazwa)
metoda wczytuje dane do tablicy znajdujacej sie w zmiennej wejscie.
- void `wykonajAlgorytm` ()
Metoda wykonuje algorytm na danych wejscowych (tablicy)
- bool `sprawdz` ()
Metoda sprawdza poprawnosc algorytmu.
- int `uruchom` (int j)
Metoda wykonuje jednorazowy test algorytmu Metoda: -wlacza zegar -wykonuje algorytm -wylacza zegar -sprawdza poprawnosc algorytmu return czas wykonywania algorytmu w milisekundach.
- int `wez_rozmiar` ()

2.1.1 Detailed Description

Klasa modelująca gówna czesc programu.

Klasa modeluje gówna czesc porgramu, ktorego zadaniem jest: -wczytanie danych -zmierzenie czasu dzialania algorytmu -sprawdzenie poprawnosc tego algorytmu, majac oczekiwany wynik

2.1.2 Member Function Documentation

2.1.2.1 bool Działanie::sprawdz ()

Metoda sprawdza poprawnosc algorytmu.

Wczytywane sa poprawne dane wynikowe, a nastepnie sa one porownywane z tymi otrzymanymi przez wykonanie algorytmu

Returns

0 - gdy algorytm jest poprawny, -1 - gdy nie.

2.1.2.2 void Dzialanie::wczytajDaneWejsciove (string nazwa)

metoda wczytuje dane do tablicy znajdujacej sie w zmiennej wejscie.

Format danych w pliku jest nastepujacy: pierwszy wiersz - ilosc elementow, a nastepnie w kolumnie kolejne wartosci tablicy.

Parameters

<i>nazwaPliku</i>	- nazwa pliku do otwarcia
-------------------	---------------------------

Returns

void

2.1.2.3 void Dzialanie::wykonajAlgorytm ()

Metoda wykonuje algorytm na danych wejsciowych (tablicy)

Algorytm do wykonania : pomnoz kazdy element razy 2.

Returns

void

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

- inc/dzialanie.hh
- src/dzialanie.cpp

2.2 elem Struct Reference

Struktura modelujca pare klucz-wartosc.

```
#include <hash.hh>
```

Public Attributes

- std::string **klucz**
- int **wartosc**

2.2.1 Detailed Description

Struktura modelujca pare klucz-wartosc.

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

- inc/hash.hh

2.3 Para< Wartosc > Class Template Reference

Klasa modeluje pare klucz-wartosc, ktora jest podstawowym elementem tablicy asocjacyjnej Zar typ Klucz jak i Wartosc musz miec konstruktory parametryczne, ktore inicjuja ich wartosci.

```
#include <slownik.hh>
```


Public Member Functions

- [Para](#) ()
Konstruktor inicjujacy wartosci key i val.
- [Para](#) (const std::string &key)
Konstruktor inicjujacy wartosci key i val.
- [Para](#) (const std::string &key, const Wartosc &val)
Konstruktor inicjujacy wartosci key i val.
- std::string & [klucz](#) ()
Metoda zwraca referencje do pola key, dzieki niej mozemy modyfikowac jego wartosc.
- Wartosc & [wartosc](#) ()
Metoda zwraca referencje do pola val, dzieki niej mozemy modyfikowac jego wartosc.
- const std::string & [klucz](#) () const
Metoda zwraca wartosc pola key, sluzzy ona do odczytu tej wartosci.
- const Wartosc & [wartosc](#) () const
Metoda zwraca wartosc pola val, sluzzy ona do odczytu tej wartosci.

2.3.1 Detailed Description

```
template<typename Wartosc>class Para< Wartosc >
```

Klasa modeluje pare klucz-wartosc, ktora jest podstawowym elementem tablicy asocjacyjnej Zar typ Klucz jak i Wartosc musz miec konstruktory parametryczne, ktore inicjuja ich wartosci.

2.3.2 Member Function Documentation

2.3.2.1 `template<typename Wartosc> std::string& Para< Wartosc >::klucz () [inline]`

Metoda zwraca referencje do pola key, dzieki niej mozemy modyfikowac jego wartosc.

Returns

referencja do pola key

2.3.2.2 `template<typename Wartosc> const std::string& Para< Wartosc >::klucz () const [inline]`

Metoda zwraca wartosc pola key, sluzzy ona do odczytu tej wartosci.

Returns

wartosc pola key

2.3.2.3 `template<typename Wartosc> Wartosc& Para< Wartosc >::wartosc () [inline]`

Metoda zwraca referencje do pola val, dzieki niej mozemy modyfikowac jego wartosc.

Returns

referencja do pola val

2.3.2.4 `template<typename Wartosc> const Wartosc& Para< Wartosc >::wartosc () const [inline]`

Metoda zwraca wartosc pola val, sluzy ona do odczytu tej wartosci.

Returns

wartosc pola val

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

- inc/slownik.hh

2.4 Tab_asocjacyjna< Wartosc > Class Template Reference

Public Member Functions

- [Tab_asocjacyjna](#) ()
Metoda dodaje pare klucz-wartosc do tablicy.
- bool [dodaj](#) (std::string klucz, Wartosc wartosc)
Metoda dodaje pare klucz-wartosc do drzewa W przypadku gdy dany klucz juz istnieje to wartosc, ktora jest z nim skojarzona zostaje zastapiona ta podana w argumencie "wartosc".
- Wartosc [pobierz](#) (std::string klucz) const
Metoda pobiera wartosc przypisana pod zadany klucz.
- void [zmien](#) (string klucz, Wartosc wartosc)
Metoda zmienia wartosc pod danym kluczem na ta zadana.
- bool [usun](#) (string klucz)
Metoda usuwa pozycje zawierajaca podany klucz.
- const bool [czy_pusta](#) ()
Metoda zwraca informacje czy drzewo jest puste.
- const int [rozmiar](#) ()
Metoda zwraca ilosc elementow na drzewie.
- void [wypisz](#) (string klucz)
Metoda pomocnicza, ktora wypisuje zawartosc drzewa Powstala w celach testowych.

2.4.1 Constructor & Destructor Documentation

2.4.1.1 `template<typename Wartosc > Tab_asocjacyjna< Wartosc >::Tab_asocjacyjna () [inline]`

Metoda dodaje pare klucz-wartosc do tablicy.

Parameters

<i>klucz</i>	
<i>wartosc</i>	

2.4.2 Member Function Documentation

2.4.2.1 `template<typename Wartosc > const bool Tab_asocjacyjna< Wartosc >::czy_pusta () [inline]`

Metoda zwraca informacje czy drzewo jest puste.

Returns

true gdy puste
false gdy nie puste

2.4.2.2 `template<typename Wartosc > bool Tab_asocjacyjna< Wartosc >::dodaj (std::string klucz, Wartosc wartosc)`

Metoda dodaje pare klucz-wartosc do drzewa W przypadku gdy dany klucz juz istnieje to wartosc, ktora jest z nim skojarzona zostaje zastapiona ta podana w argumencie "wartosc".

Parameters

in	<i>klucz-nazwa</i>	klucza
in	<i>wartosc-wartosc</i>	do zapisania

2.4.2.3 `template<typename Wartosc > Wartosc Tab_asocjacyjna< Wartosc >::pobierz (std::string klucz) const`

Metoda pobiera wartosc przypisana pod zadany klucz.

Parameters

in	<i>klucz-</i>	nazwa klucza, ktorego szukamy
----	---------------	-------------------------------

Returns

wartosc spod zadanego klucza

2.4.2.4 `template<typename Wartosc > const int Tab_asocjacyjna< Wartosc >::rozmiar () [inline]`

Metoda zwraca ilosc elementow na drzewie.

Returns

ilosc elementow

2.4.2.5 `template<typename Wartosc > bool Tab_asocjacyjna< Wartosc >::usun (string klucz)`

Metoda usuwa pozycje zawierajaca podany klucz.

Parameters

in	<i>klucz-klucz</i>	do usuniecia
----	--------------------	--------------

2.4.2.6 `template<typename Wartosc > void Tab_asocjacyjna< Wartosc >::zmien (string klucz, Wartosc wartosc) [inline]`

Metoda zmienia wartosc pod danym kluczem na ta zadana.

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

- inc/drzewo.hh

2.5 tab_hash Class Reference

Klasa modelująca tablice haszująca z funkcją mieszająca.

```
#include <hash.hh>
```

Public Member Functions

- void [dodaj](#) (std::string klucz, int wartosc)
Metoda dodająca parę klucz-wartosc do tablicy Dla klucza zostaje wygenerowany indeks tablicy, następnie para zostaje dodana pod ten indeks.
- int [pobierz](#) (std::string klucz)
Metoda pobiera wartosc spod zadanego klucza Generowany jest indeks tablicy i w liscie spod tego indeksu zostaje wyszukana zadana para.
- bool [usun](#) (std::string klucz)
Metoda usuwa z tablicy pozycje o zadanym kluczu.
- void [wypisz](#) ()
Metoda pomocnicza, sluzi do wypisania zawartosci tablicy.

2.5.1 Detailed Description

Klasa modelująca tablice haszująca z funkcją mieszająca.

2.5.2 Member Function Documentation

2.5.2.1 void tab_hash::dodaj (std::string klucz, int wartosc)

Metoda dodająca parę klucz-wartosc do tablicy Dla klucza zostaje wygenerowany indeks tablicy, następnie para zostaje dodana pod ten indeks.

Parameters

<i>klucz-klucz</i>	do dodania
<i>wartosc-wartosc</i>	do dodania

2.5.2.2 int tab_hash::pobierz (std::string klucz)

Metoda pobiera wartosc spod zadanego klucza Generowany jest indeks tablicy i w liscie spod tego indeksu zostaje wyszukana zadana para.

Parameters

<i>klucz</i>	- klucz pary ktorej szukamy
--------------	-----------------------------

Returns

wartosc spod klucza

2.5.2.3 bool tab_hash::usun (std::string klucz)

Metoda usuwa z tablicy pozycje o zadanym kluczu.

Parameters

<i>klucz</i>	- klucz ktory zostanie usuniety
--------------	---------------------------------

Returns

true - gdy klucz zostal usuniety
false - gdy klucza nie znaleziono

2.5.2.4 void tab_hash::wypisz ()

Metoda pomocnicza, sluzi do wypisania zawartosci tablicy.

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

- inc/hash.hh
- src/hash.cpp

2.6 Tablica_asocjacyjna< Wartosc > Class Template Reference

Klasa modeluje pojecie tablicy asocjacyjnej Tablica asocjacyjna jest to tablica, w ktorej do wartosci mozemy sie odwoływac poprzez klucze. Odwołanie poprzez indeks rowniez jest mozliwe.

```
#include <slownik.hh>
```

Public Member Functions

- [Tablica_asocjacyjna](#) ()
Metoda dodaje pare klucz-wartosc do tablicy.
- void [dodaj](#) (std::string klucz, Wartosc wartosc)
Metoda dodaje pare klucz-wartosc do tablicy W przypadku gdy dany klucz juz istnieje to wartosc, ktora jest z nim skojarzona zostaje zastapiona ta podana w argumentie "wartosc" Po kazdorazowym dodaniu nowego klucza, tablica jest sortowana alfabetycznie wzgledem nazw kluczy. Takie rozwiazanie daje mozliwosc przeszukiwania binarnego co powoduje ze dostep do elementu jest w czasie logn.
- void [zmien](#) (string klucz, Wartosc wartosc)
Metoda pobiera wartosc przypisana pod zadany klucz.
- Wartosc [pobierz](#) (std::string klucz)
- void [usun](#) (string klucz)
Metoda usuwa pozycje zawierajaca podany klucz. Gdy klucz istnieje, to jest usuwany, a flaga czy_istnieje ustawiana na wartosc true, przeciwnie flaga ma wartosc false.
- string [znajdz](#) (Wartosc wartosc)
Metoda znajduje pozycje(nazwe klucza), pod ktora znajduje sie podana wartosc.
- const bool [czy_znalazlo](#) () const
Metoda zwraca stan flagi czy_istnieje. Flaga czy istnieje jest ustawiana za kazdym razem gdy wykonujemy funkcje, ktora przeszukuje tablice po kluczach. Gdy zadany klucz istnieje to flaga jest ustawiana. Natomiast gdy danego klucza nie ma w zbiorze to flaga jest zerowana.
- const bool [czy_pusta](#) ()
- const int [rozmiar](#) ()
- Wartosc [operator\[\]](#) (string klucz) const
- Wartosc & [operator\[\]](#) (string klucz)

2.6.1 Detailed Description

```
template<typename Wartosc>class Tablica_asocjacyjna< Wartosc >
```

Klasa modeluje pojecie tablicy asocjacyjnej Tablica asocjacyjna jest to tablica, w ktorej do wartosci mozemy sie odwozywac poprzez klucze. Odwołanie poprzez indeks rowniez jest mozliwe.

2.6.2 Constructor & Destructor Documentation

2.6.2.1 `template<typename Wartosc > Tablica_asocjacyjna< Wartosc >::Tablica_asocjacyjna () [inline]`

Metoda dodaje pare klucz-wartosc do tablicy.

zmienna pomocnicza, ktora sluzy do okreslania czy wartosc o podanym kluczu

znajduje sie w zbiorze

Parameters

<i>klucz</i>	
<i>wartosc</i>	

2.6.3 Member Function Documentation

2.6.3.1 `template<typename Wartosc > const bool Tablica_asocjacyjna< Wartosc >::czy_znalazlo () const [inline]`

Metoda zwraca stan flagi czy_istnieje. Flaga czy istnieje jest ustawiana za kazdym razem gdy wykonujemy funkcje, ktora przeszukuje tablice po kluczach. Gdy zadany klucz istnieje to flaga jest ustawiana. Natomiast gdy danego klucza nie ma w zbiorze to flaga jest zerowana.

Returns

true - zadany klucz istnieje w zbiorze
false - klucz nie istnieje.

2.6.3.2 `template<typename Wartosc > void Tablica_asocjacyjna< Wartosc >::dodaj (std::string klucz, Wartosc wartosc)`

Metoda dodaje pare klucz-wartosc do tablicy W przypadku gdy dany klucz juz istnieje to wartosc, ktora jest z nim skojarzona zostaje zastapiona ta

podana w argumencie "wartosc" Po kazdorazowym dodaniu nowego klucza, tablica jest sortowana alfabetycznie wzgledem nazw kluczy. Takie rozwiazanie daje mozliwosc przeszukiwania binarnego co powoduje ze dostep do elementu jest w czasie logn.

Parameters

<i>in</i>	<i>klucz-nazwa</i>	klucza
<i>in</i>	<i>wartosc-wartosc</i>	do zapisania

2.6.3.3 `template<typename Wartosc > void Tablica_asocjacyjna< Wartosc >::usun (string klucz)`

Metoda usuwa pozycje zawierajaca podany klucz. Gdy klucz istnieje, to jest usuwany, a flaga czy_istnieje ustawiana na wartosc true, przeciwnie flaga ma wartosc false.

Parameters

in	klucz-klucz	do usuniecia
----	-------------	--------------

2.6.3.4 `template<typename Wartosc > void Tablica_asocjacyjna< Wartosc >::zmien (string klucz, Wartosc wartosc)`
`[inline]`

Metoda pobiera wartosc przypisana pod zadany klucz.

Jesli klucz nie istnieje to flaga czy_istnieje jest ustawiana na false, w przeciwnym wypadku ma ona wartosc true. Po wywołaniu metody "pobierz" mozna sprawdzic stan flagi metoda [czy_znalazlo\(\)](#) i jego podstawie okreslic czy operacja zostala wykonana poprawnie.

Parameters

in	klucz-	nazwa klucza, ktorego szukamy
----	--------	-------------------------------

Returns

wartosc spod zadanego klucza

2.6.3.5 `template<typename Wartosc > string Tablica_asocjacyjna< Wartosc >::znajdz (Wartosc wartosc)`

Metoda znajduje pozycje(nazwe klucza), pod ktora znajduje sie podana wartosc.

Returns

gdy wartosc istnieje: nazwa klucza,
 gdy nie istnieje: pusta zmienna string

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

- inc/slownik.hh

2.7 Tester Class Reference

Klasa [Tester](#) Klasa modeluje narzedzie do benchmarkowania algorytmu. Poprzez wielokrotne wynonywanie algorytmu mozemy wyciagnac sredni czas dzialania, co jest dokladniejszym pomiarem.

```
#include <tester.hh>
```

Public Member Functions

- [Tester](#) ()
konstruktor klasy Konstruktor inicjuje wartosci: -powtorzenia (ile razy wykonywac algorytm) -ilosc (ile mamy zestawow danych) -wejscie (nazwa pliku z danymi wejscowymi) -wynik (nazwa pliku z poprawnym wynikiem algorytmu)
- void [otworzPlik](#) (string nazwa)
Metoda otwierajaca referencje do pliku CSV.
- void [zamknijPlik](#) ()
Metoda otwierajaca referencje do pliku CSV.
- void [symulacja](#) ()
Metoda symulujaca badanie algorytmu Metoda wykonuje symulacje dzialania algorytmu. Wykonuje algorytm zadana liczbe razy dla zadanych zestawow danych. Wyniki zapisuje do pliku CSV o nazwie rezultat.csv Format zapisu: rozmiar_problemu,ilosc_powtorzen,sredni_czas.

- void [zamienNazwy](#) (int numer)

Metoda aktualizuje nazwe plikow wejscowych Format plikow wejscowych jest scisle okreslony. Dane wejscowe: wejscie%.txt, gdzie % to numer zestawu Dane wynikowe: wynik%.txt, gdzie % to numer zestawu Zestawy nalazy numerowac od 0.

2.7.1 Detailed Description

Klasa [Tester](#) Klasa modeluje narzedzie do benchmarkowania algorytmu. Poprzez wielokrotne wynonywanie algorytmu mozemy wyciagnac sredni czas dzialania, co jest dokladniejszym pomiarem.

2.7.2 Member Function Documentation

2.7.2.1 void Tester::zamienNazwy (int numer)

Metoda aktualizuje nazwe plikow wejscowych Format plikow wejscowych jest scisle okreslony. Dane wejscowe: wejscie%.txt, gdzie % to numer zestawu Dane wynikowe: wynik%.txt, gdzie % to numer zestawu Zestawy nalazy numerowac od 0.

Parameters

<i>numer</i>	- % zostaje zamieniony na wartosc 'numer'
--------------	---

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

- inc/tester.hh
- src/tester.cpp

2.8 wezel< Wartosc > Struct Template Reference

Klasa modeluje pare klucz-wartosc, ktora jest podstawowym elementem tablicy asocjacyjnej Zar typ Klucz jak i Wartosc musz miec konstruktory parametryczne, ktore inicjuja ich wartosci.

```
#include <drzewo.hh>
```

Public Attributes

- jakie_dziecko **dziecko**
- string **klucz**
- Wartosc **dane**
- [wezel](#) * **rodzic**
- [wezel](#) * **lewy**
- [wezel](#) * **prawy**

2.8.1 Detailed Description

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

Klasa modeluje pare klucz-wartosc, ktora jest podstawowym elementem tablicy asocjacyjnej Zar typ Klucz jak i Wartosc musz miec konstruktory parametryczne, ktore inicjuja ich wartosci.

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

- inc/drzewo.hh

Index

czy_pusta
 Tab_asocjacyjna, 6
czy_znalazlo
 Tablica_asocjacyjna, 10

dodaj
 Tab_asocjacyjna, 7
 tab_hash, 8
 Tablica_asocjacyjna, 10

Dzialanie, 3
 sprawdz, 3
 wczytajDaneWejscowe, 3
 wykonajAlgorytm, 4

elem, 4

klucz
 Para, 5

Para
 klucz, 5
 wartosc, 5

Para< Wartosc >, 4
pobierz

 Tab_asocjacyjna, 7
 tab_hash, 8

rozmiar
 Tab_asocjacyjna, 7

sprawdz
 Dzialanie, 3

Tab_asocjacyjna
 czy_pusta, 6
 dodaj, 7
 pobierz, 7
 rozmiar, 7
 Tab_asocjacyjna, 6
 Tab_asocjacyjna, 6
 usun, 7
 zmien, 7

Tab_asocjacyjna< Wartosc >, 6

tab_hash, 8
 dodaj, 8
 pobierz, 8
 usun, 8
 wypisz, 9

Tablica_asocjacyjna
 czy_znalazlo, 10
 dodaj, 10

 Tablica_asocjacyjna, 10
 Tablica_asocjacyjna, 10
 usun, 10
 zmien, 11
 znajdz, 11
Tablica_asocjacyjna< Wartosc >, 9
Tester, 11
 zamienNazwy, 12

usun
 Tab_asocjacyjna, 7
 tab_hash, 8
 Tablica_asocjacyjna, 10

wartosc
 Para, 5
wczytajDaneWejscowe
 Dzialanie, 3
wezel< Wartosc >, 12
wykonajAlgorytm
 Dzialanie, 4
wypisz
 tab_hash, 9

zamienNazwy
 Tester, 12

zmien
 Tab_asocjacyjna, 7
 Tablica_asocjacyjna, 11
znajdz
 Tablica_asocjacyjna, 11