PAMSI 6,7

Wygenerowano przez Doxygen 1.8.6

Cz, 29 maj 2014 03:04:49

Spis treści

1 Strona główna **1**

2 Indeks klas **1**

2.1 Lista klas . . . 1

3 Indeks plików **1**

3.1 Lista plików . . . 1

4 Dokumentacja klas **2**

4.1 Dokumentacja klasy DrzewoAsocjacyjne . . . 2

4.1.1 Opis szczegółowy . . . 2

4.1.2 Dokumentacja funkcji składowych . . . 2

4.2 Dokumentacja struktury ParaKluczWartosc . . . 4

4.2.1 Opis szczegółowy . . . 4

4.3 Dokumentacja klasy StrukturaDanych . . . 4

4.3.1 Opis szczegółowy . . . 5

4.3.2 Dokumentacja konstruktora i destruktora . . . 5

4.3.3 Dokumentacja funkcji składowych . . . 5

4.3.4 Dokumentacja przyjaciół i funkcji związanych . . . 8

4.4 Dokumentacja klasy TablicaAsocjacyjna . . . 9

4.4.1 Opis szczegółowy . . . 10

4.4.2 Dokumentacja konstruktora i destruktora . . . 10

4.4.3 Dokumentacja funkcji składowych . . . 10

4.5 Dokumentacja klasy TablicaHaszujaca . . . 11

4.5.1 Opis szczegółowy . . . 11

4.5.2 Dokumentacja konstruktora i destruktora . . . 12

4.5.3 Dokumentacja funkcji składowych . . . 12

4.6 Dokumentacja klasy Wierzcholek . . . 13

4.6.1 Opis szczegółowy . . . 13

5 Dokumentacja plików **13**

5.1 Dokumentacja pliku definicje.h . . . 13

5.1.1 Opis szczegółowy . . . 14

5.2 Dokumentacja pliku DrzewoAsocjacyjne.cpp . . . 14

5.3 Dokumentacja pliku DrzewoAsocjacyjne.h . . . 14

5.4 Dokumentacja pliku ParaKluczWartosc.h . . . 14

5.5 Dokumentacja pliku StrukturaDanych.cpp . . . 15

5.5.1 Dokumentacja funkcji . . . 15

5.6 Dokumentacja pliku StrukturaDanych.h . . . 15

5.7 Dokumentacja pliku TablicaAsocjacyjna.cpp . . . 15

5.8 Dokumentacja pliku TablicaAsocjacyjna.h 16
5.9 Dokumentacja pliku TablicaHaszujaca.h 16
5.10 Dokumentacja pliku TestTablicAsocjacyjnych.h 16
5.10.1 Dokumentacja funkcji 17

6 Sprawozdanie z wykonania programu labolatorium 6 i 7. **18**

Indeks **22**

1 Strona główna

Laboratorium 6 i 7.

Implementacja tablic asocjacyjnych na roznych strukturach danych. Wykorzystano: tablice, drzewo binarne, tablice haszujaca.

Autor

Jakub Chmiel 200314

2 Indeks klas

2.1 Lista klas

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

DrzewoAsocjacyjne
Tablica asocjacyjna na drzewie binarnym **2**

ParaKluczWartosc
Struktura pary klucz i wartosc **4**

StrukturaDanych
Struktura danych o funkcjonalnosci tablicy **4**

TablicaAsocjacyjna
Tablica asocjacyjna **9**

TablicaHaszujaca
Tablica haszujaca **11**

Wierzcholek
Tablica asocjacyjna na drzewie binarnym **13**

3 Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

definicje.h
Plik zawiera ogolne instrukcje preprocesora wspolne dla wszystkich plikow zrodlowych **13**

DrzewoAsocjacyjne.cpp 14

DrzewoAsocjacyjne.h 14

ParaKluczWartosc.h 14

StrukturaDanych.cpp 15

StrukturaDanych.h 15

TablicaAsocjacyjna.cpp 15

TablicaAsocjacyjna.h 16

TablicaHaszujaca.h 16

TestTablicAsocjacyjnych.h 16

4 Dokumentacja klas

4.1 Dokumentacja klasy DrzewoAsocjacyjne

Tablica asocjacyjna na drzewie binarnym.

`#include <DrzewoAsocjacyjne.h>`

Metody publiczne

- void dodaj (TYP_KLUCZ &klucz, TYP_WARTOSC &wartosc)

  Dodaje pare o okreslonym kluczu i wartosci.
- void usun (TYP_KLUCZ klucz)

  Zmienia wartosc na o okreslonym kluczu na wartosc zerowa.
- void zmien (TYP_KLUCZ &klucz, TYP_WARTOSC &wartosc)

  Zmienia pare o okreslonym kluczu i wartosci, jesli istnieje.
- const TYP_WARTOSC pobierz (const TYP_KLUCZ &klucz)

  Pobiera wartosc na okreslonym kluczu.
- const int ilosc_elementow ()

  Podaje ilosc elementow znajdujacych sie na drzewie.
- const bool czy_pusta ()

  Podaje czy drzewo jest puste.
- void wypisz_wszystko ()

  Wypisuje wszystkie elementy na drzewie na standardowe wyjscie.
- void przepisz_strukture (StrukturaDanych &dane, int ile)

  Przepisuje wybrana ilosc elementow ze struktury do tablicy.

4.1.1 Opis szczegółowy

Tablica asocjacyjna na drzewie binarnym.

4.1.2 Dokumentacja funkcji składowych

4.1.2.1 const bool DrzewoAsocjacyjne::czy_pusta ( )

Podaje czy drzewo jest puste.

Zwraca

true - brak elementow. false - istnieja elementy.

4.1.2.2 void DrzewoAsocjacyjne::dodaj (TYP_KLUCZ & *klucz,* TYP_WARTOSC & *wartosc*)

Dodaje pare o okreslonym kluczu i wartosci.

Parametry

klucz	Nowy lub juz istniejacy klucz.
wartosc	Nowa wartosc.

4.1.2.3 const int DrzewoAsocjacyjne::ilosc_elementow ( )

Podaje ilosc elementow znajdujacych sie na drzewie.

Zwraca

Ilosc elementow.

4.1.2.4 const TYP_WARTOSC DrzewoAsocjacyjne::pobierz (const TYP_KLUCZ & *klucz*)

Pobiera wartosc na okreslonym kluczu.

Parametry

klucz	Juz istniejacy klucz.

Zwraca

Odnaleziona wartosc.

4.1.2.5 void DrzewoAsocjacyjne::przepisz_strukture (StrukturaDanych & *dane,* int *ile*)

Przepisuje wybrana ilosc elementow ze struktury do tablicy.

Parametry

dane	zrodlowa struktura danych.
ile	ilosc elementow do przepisania.

4.1.2.6 void DrzewoAsocjacyjne::usun (TYP_KLUCZ *klucz*)

Zmienia wartosc na o okreslonym kluczu na wartosc zerowa.

Parametry

klucz	Klucz do usuniecia wartosci.

4.1.2.7 void DrzewoAsocjacyjne::zmien (TYP_KLUCZ & *klucz,* TYP_WARTOSC & *wartosc*)

Zmienia pare o okreslonym kluczu i wartosci, jesli istnieje.

Parametry

klucz	Juz istniejacy klucz.

wartosc	Nowa wartosc.

Dokumentacja dla tej klasy została wygenerowana z plików:

- DrzewoAsocjacyjne.h
- DrzewoAsocjacyjne.cpp

4.2 Dokumentacja struktury ParaKluczWartosc

Struktura pary klucz i wartosc.

`#include <ParaKluczWartosc.h>`

Metody publiczne

- ParaKluczWartosc (TYP_KLUCZ klucz, TYP_WARTOSC wartosc)
  Kontruktor z podanymi wartosciami pol.
- ParaKluczWartosc ()
  Kontruktor ustawia pola na oznaczenia braku pary.
- bool czy_istnieje ()
  Sprawdza czy dana para istnieje.
- bool czy_zajety ()
  Sprawdza czy klucz oznacza ze para istenieje i ma sens.

Atrybuty publiczne

- TYP_KLUCZ **klucz**
- TYP_WARTOSC **wartosc**

4.2.1 Opis szczegółowy

Struktura pary klucz i wartosc.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- ParaKluczWartosc.h

4.3 Dokumentacja klasy StrukturaDanych

Struktura danych o funkcjonalnosci tablicy.

`#include <StrukturaDanych.h>`

Metody publiczne

- StrukturaDanych (StrukturaDanych &inna)
  Konstruktor kopiujacy.
- const int ilosc_elementow ()
  zwraca ilosc elementow w strukturze.
- const TYP element_na (int i)
  zwraca wartosc elementu na danym indeksie
- StrukturaDanych & operator= (StrukturaDanych &T)
  Operator przypisania.
- StrukturaDanych & operator+= (StrukturaDanych &T)
  Operator dodawania z przypisaniem.

Statyczne metody publiczne

- static bool zmien_element (StrukturaDanych &T, int index, TYP e)

  Nadpisuje element na danym indeksie.
- static bool kopiuj_wycinek (StrukturaDanych &zrodlo, StrukturaDanych &cel, int i_od, int i_do)

  Dodaje elementy ze struktury zrodlowej do docelowej z indeksow od i_od do i_do.
- static bool zamien_elementy (StrukturaDanych &T, int i, int j)

  Zamienia kolejnosc dwoch dowolnych elementow.
- static bool odwroc_kolejnosc (StrukturaDanych &T)

  Odwraca kolejnosc wszystkich elementow struktury.
- static bool dodaj_element (StrukturaDanych &T, TYP e)

  Dodaje element na koniec struktury.
- static StrukturaDanych dodaj_elementy (StrukturaDanych &T1, StrukturaDanych &T2)

  Laczy 2 struktury ze soba.
- static void wypisz_wszystko (StrukturaDanych &T)

  Wypisuje dane ze struktury na standardowe wyjscie.

Przyjaciele

- StrukturaDanych operator+ (StrukturaDanych T1, StrukturaDanych &T2)

  Laczy 2 struktury ze soba.
- bool operator== (StrukturaDanych &T1, StrukturaDanych &T2)

  Operator porownania.

4.3.1 Opis szczegółowy

Struktura danych o funkcjonalnosci tablicy.

4.3.2 Dokumentacja konstruktora i destruktora

4.3.2.1 StrukturaDanych::StrukturaDanych (StrukturaDanych & *inna*)

Konstruktor kopiujacy.

Parametry

inna	kopiowana StrukturaDanych

4.3.3 Dokumentacja funkcji składowych

4.3.3.1 bool StrukturaDanych::dodaj_element (StrukturaDanych & *T*, TYP *e*) [static]

Dodaje element na koniec struktury.

Parametry

T	docelowa StrukturaDanych
e	element do dodania

Zwraca

- true sukces
- false porazka

4.3.3.2 StrukturaDanych StrukturaDanych::dodaj_elementy (StrukturaDanych & *T1,* StrukturaDanych & *T2*) [static]

Laczy 2 struktury ze soba.

Parametry

T1	pierwsza StrukturaDanych
T2	druga StrukturaDanych

Zwraca

Struktura bedaca polaczeniem dwoch wejsciowych struktur.

4.3.3.3 const TYP StrukturaDanych::element_na (int *i*)

zwraca wartosc elementu na danym indeksie

Parametry

i	indeks

Zwraca

wartosc na i-tym indeksie

4.3.3.4 const int StrukturaDanych::ilosc_elementow ( )

zwraca ilosc elementow w strukturze.

Zwraca

ilosc elementow.

4.3.3.5 bool StrukturaDanych::kopiuj_wycinek (StrukturaDanych & *zrodlo,* StrukturaDanych & *cel,* int *i_od,* int *i_do*) [static]

Dodaje elementy ze struktury zrodlowej do docelowej z indeksow od i_od do i_do.

Parametry

&zrodlo	Struktura zrodlowa.
&cel	Struktura docelowa
i_od	Indeks pierwszego elementu ktory chcemy kopiowac.
i_do	Indeks ostatniego elementu ktory chcemy kopiowac.

Zwraca

Stan powodzenia funkcji.

4.3.3.6 bool StrukturaDanych::odwroc_kolejnosc (StrukturaDanych & *T*) [static]

Odwraca kolejnosc wszystkich elementow struktury.

Parametry

T	docelowa StrukturaDanych

Zwraca

- true sukces
- false porazka

4.3.3.7 StrukturaDanych & StrukturaDanych::operator+= (StrukturaDanych & *T*)

Operator dodawania z przypisaniem.

Parametry

T	StrukturaDanych

Zwraca

Struktura z dodanymi na koniec elementami struktury T

4.3.3.8 StrukturaDanych & StrukturaDanych::operator= (StrukturaDanych & *T*)

Operator przypisania.

Parametry

T	StrukturaDanych

Zwraca

taka sama struktura jak parametr T

4.3.3.9 void StrukturaDanych::wypisz_wszystko (StrukturaDanych & *T*) [static]

Wypisuje dane ze struktury na standardowe wyjscie.

Parametry

T	docelowa StrukturaDanych

4.3.3.10 bool StrukturaDanych::zamien_elementy (StrukturaDanych & *T*, int *i*, int *j*) [static]

Zamienia kolejnosc dwoch dowolnych elementow.

Parametry

T	docelowa StrukturaDanych
i	indeks 1
j	indeks 2

Zwraca

- true sukces
- false porazka

4.3.3.11 bool StrukturaDanych::zmien_element (StrukturaDanych & *T*, int *index*, TYP *e*) [static]

Nadpisuje element na danym indeksie.

Parametry

&T	Docelowa struktura.
index	Indeks elementu do nadpisu.
e	nowy element.

Zwraca

Stan powodzenia funkcji.

4.3.4 Dokumentacja przyjaciół i funkcji związanych

4.3.4.1 StrukturaDanych operator+ (StrukturaDanych *T1*, StrukturaDanych & *T2*) [friend]

Laczy 2 struktury ze soba.

Parametry

T1	pierwsza StrukturaDanych
T2	druga StrukturaDanych

Zwraca

Struktura bedaca polaczeniem dwoch wejsciowych struktur.

4.3.4.2 bool operator== (StrukturaDanych & *T1*, StrukturaDanych & *T2*) [friend]

Operator porownania.

Parametry

T1	pierwsza StrukturaDanych
T2	druga StrukturaDanych

Zwraca

- true struktury maja identyczne elementy
- false elementy nie sa identyczne

Dokumentacja dla tej klasy została wygenerowana z plików:

- StrukturaDanych.h
- StrukturaDanych.cpp

4.4 Dokumentacja klasy TablicaAsocjacyjna

Tablica asocjacyjna.

`#include <TablicaAsocjacyjna.h>`

Metody publiczne

- TablicaAsocjacyjna (TablicaAsocjacyjna &inna)

  Konstruktor kopiujacy.
- void dodaj (TYP_KLUCZ &klucz, TYP_WARTOSC &wartosc)

  Dodaje nowa pare klucz i wartosc lub zmienia juz istniejaca.
- void usun (TYP_KLUCZ &klucz)

  Usuwa dany element, przesuwa wszystkie elementy za nim o jedno miejsce do tylu.
- void zmien (TYP_KLUCZ &klucz, TYP_WARTOSC &wartosc)

  Zmienia juz istniejaca pare klucz i wartosc.
- const TYP_WARTOSC pobierz (const TYP_KLUCZ &klucz)

  Pobiera wartosc spod klucza lub wartosc zero jesli nie istenieje.
- const int ilosc_elementow ()

  zwraca ilosc elementow w tablicy.
- const bool czy_pusta ()

  Sprawdza czy tablica nie zawiera zadnej pary klucz i wartosc.
- void **wypisz_wszystko** ()
- void przepisz_strukture (StrukturaDanych &dane, int ile)

  Przepisuje wybrana ilosc elementow ze struktury do tablicy.

4.4.1 Opis szczegółowy

Tablica asocjacyjna.

4.4.2 Dokumentacja konstruktora i destruktora

4.4.2.1 TablicaAsocjacyjna::TablicaAsocjacyjna (TablicaAsocjacyjna & *inna*)

Konstruktor kopiujacy.

Parametry

inna	kopiowana TablicaAsocjacyjna

4.4.3 Dokumentacja funkcji składowych

4.4.3.1 const bool TablicaAsocjacyjna::czy_pusta ( )

Sprawdza czy tablica nie zawiera zadnej pary klucz i wartosc.

Zwraca

-true - tablica pusta -false - tablica zawiera co najmniej jedna pare

4.4.3.2 void TablicaAsocjacyjna::dodaj (TYP_KLUCZ & *klucz,* TYP_WARTOSC & *wartosc*)

Dodaje nowa pare klucz i wartosc lub zmienia juz istniejaca.

Parametry

klucz	klucz pod jakim bedzie znajdzowac sie nowa wartosc.
wartosc	wartosc jaka bedzie sie znajdowac pod kluczem.

4.4.3.3 const int TablicaAsocjacyjna::ilosc_elementow ( )

zwraca ilosc elementow w tablicy.

Zwraca

ilosc elementow.

4.4.3.4 const TYP_WARTOSC TablicaAsocjacyjna::pobierz (const TYP_KLUCZ & *klucz*)

Pobiera wartosc spod klucza lub wartosc zero jesli nie istenieje.

Parametry

klucz	klucz pod jakim bedzie znajdowac sie wartosc.

4.4.3.5 void TablicaAsocjacyjna::przepisz_strukture (StrukturaDanych & *dane,* int *ile*)

Przepisuje wybrana ilosc elementow ze struktury do tablicy.

Parametry

dane	zrodlowa struktura danych.
ile	ilosc elementow do przepisania.

4.4.3.6 void TablicaAsocjacyjna::usun (TYP_KLUCZ & *klucz*)

Usuwa dany element, przesuwa wszystkie elementy za nim o jedno miejsce do tylu.

Parametry

klucz	klucz pod jakim znajduje sie para.

4.4.3.7 void TablicaAsocjacyjna::zmien (TYP_KLUCZ & *klucz,* TYP_WARTOSC & *wartosc*)

Zmienia juz istniejaca pare klucz i wartosc.

Parametry

klucz	klucz pod jakim bedzie znajdzowac sie nowa wartosc.
wartosc	wartosc jaka bedzie sie znajdowac pod kluczem.

Dokumentacja dla tej klasy została wygenerowana z plików:

- TablicaAsocjacyjna.h
- TablicaAsocjacyjna.cpp

4.5 Dokumentacja klasy TablicaHaszujaca

Tablica haszujaca.

`#include <TablicaHaszujaca.h>`

Metody publiczne

- TablicaHaszujaca (TablicaHaszujaca &inna)

  Konstruktor kopiujacy.
- void dodaj (TYP_KLUCZ &klucz, TYP_WARTOSC &wartosc)

  Dodaje nowa pare klucz i wartosc lub zmienia istniejaca.
- void usun (TYP_KLUCZ &klucz)

  Usuwa pare o danym kluczu jesli istenieje.
- void zmien (TYP_KLUCZ &klucz, TYP_WARTOSC &wartosc)

  Zmienia pare klucz i wartosc jesli istnieje.
- const TYP_WARTOSC pobierz (const TYP_KLUCZ &klucz)

  Zwraca wartosc danej pary.
- const int ilosc_elementow ()

  Zwraca ilosc elementow na tablicy haszujacej.
- const bool **czy_pusta** ()
- void wypisz_wszystko ()

  Wypisuje wszystkie elementy tablicy na standardowe wyjscie.
- void przepisz_strukture (StrukturaDanych &dane, int ile)

  Przepisuje wybrana ilosc elementow ze struktury do tablicy.

4.5.1 Opis szczegółowy

Tablica haszujaca.

4.5.2 Dokumentacja konstruktora i destruktora

4.5.2.1 TablicaHaszujaca::TablicaHaszujaca (TablicaHaszujaca & *inna*)

Konstruktor kopiujacy.

Parametry

inna	kopiowana TablicaAsocjacyjna

4.5.3 Dokumentacja funkcji składowych

4.5.3.1 void TablicaHaszujaca::dodaj (TYP_KLUCZ & *klucz,* TYP_WARTOSC & *wartosc*)

Dodaje nowa pare klucz i wartosc lub zmienia istniejaca.

Parametry

klucz	klucz pary.
wartosc	nowa wartosc.

4.5.3.2 const TYP_WARTOSC TablicaHaszujaca::pobierz (const TYP_KLUCZ & *klucz*)

Zwraca wartosc danej pary.

Parametry

klucz	klucz pary.

Zwraca

wartosc danej pary.

4.5.3.3 void TablicaHaszujaca::przepisz_strukture (StrukturaDanych & *dane,* int *ile*)

Przepisuje wybrana ilosc elementow ze struktury do tablicy.

Parametry

dane	zrodlowa struktura danych.
ile	ilosc elementow do przepisania.

4.5.3.4 void TablicaHaszujaca::usun (TYP_KLUCZ & *klucz*)

Usuwa pare o danym kluczu jesli istenieje.

Parametry

klucz	klucz pary do usuniecia.

4.5.3.5 void TablicaHaszujaca::zmien (TYP_KLUCZ & *klucz,* TYP_WARTOSC & *wartosc*)

Zmienia pare klucz i wartosc jesli istnieje.

Parametry

klucz	klucz pary do zmiany.

wartosc	nowa wartosc.

Dokumentacja dla tej klasy została wygenerowana z plików:

- TablicaHaszujaca.h
- TablicaHaszujaca.cpp

4.6 Dokumentacja klasy Wierzcholek

Tablica asocjacyjna na drzewie binarnym.

`#include <DrzewoAsocjacyjne.h>`

Metody publiczne

- **Wierzcholek** (Wierzcholek *p_ojciec)
- void **usun** ()
- Wierzcholek & lewy ()

  Pobiera nastepny wierzcholek z lewej strony. Jesli nie istenieje to tworzy.
- Wierzcholek & prawy ()

  Pobiera nastepny wierzcholek z prawej strony. Jesli nie istenieje to tworzy.
- bool **czy_lewy** ()
- bool **czy_prawy** ()
- void wypisz_rekurencyjnie ()

  Wypisuje elementy w wierzcholku i potomkach na podany strumien.

Atrybuty publiczne

- ParaKluczWartosc para

  Para klucza i wartosci na danym wierzcholku.

4.6.1 Opis szczegółowy

Tablica asocjacyjna na drzewie binarnym.

Dokumentacja dla tej klasy została wygenerowana z plików:

- DrzewoAsocjacyjne.h
- DrzewoAsocjacyjne.cpp

5 Dokumentacja plików

5.1 Dokumentacja pliku definicje.h

Plik zawiera ogolne instrukcje preprocesora wspolne dla wszystkich plikow zrodlowych.

`#include <string>`

Definicje

- #define TYP string

  Typ uzyty w strukturze, zawiera klucze do tablic.
- #define TYP_KLUCZ string

  Typ klucza w tablicach asocjacyjnych.
- #define TYP_WARTOSC int

  Typ wartosci w tablicach asocjacyjnych.
- #define KLUCZ_ZERO ""

  Wartosc klucza znaczaca brak klucza.
- #define WARTOSC_ZERO 0

  Wartosc domyslna dla typu wartosci.
- #define LICZBA_LIST 100000

  Liczba list w tablicy haszujacej.

5.1.1 Opis szczegółowy

Plik zawiera ogolne instrukcje preprocesora wspolne dla wszystkich plikow zrodlowych.

5.2 Dokumentacja pliku DrzewoAsocjacyjne.cpp

```
#include "DrzewoAsocjacyjne.h"
```

5.3 Dokumentacja pliku DrzewoAsocjacyjne.h

```
#include <iostream>
#include "definicje.h"
#include "ParaKluczWartosc.h"
#include "StrukturaDanych.h"
```

Komponenty

- class Wierzcholek

  Tablica asocjacyjna na drzewie binarnym.
- class DrzewoAsocjacyjne

  Tablica asocjacyjna na drzewie binarnym.

5.4 Dokumentacja pliku ParaKluczWartosc.h

```
#include "definicje.h"
```

Komponenty

- struct ParaKluczWartosc

  Struktura pary klucz i wartosc.

5.5 Dokumentacja pliku StrukturaDanych.cpp

```
#include "StrukturaDanych.h"
```

Funkcje

- StrukturaDanych operator+ (StrukturaDanych T1, StrukturaDanych &T2)

  Laczy 2 struktury ze soba.
- bool operator== (StrukturaDanych &T1, StrukturaDanych &T2)

  Operator porownania.

5.5.1 Dokumentacja funkcji

5.5.1.1 StrukturaDanych operator+ (StrukturaDanych *T1,* StrukturaDanych & *T2*)

Laczy 2 struktury ze soba.

Parametry

T1	pierwsza StrukturaDanych
T2	druga StrukturaDanych

Zwraca

Struktura bedaca polaczeniem dwoch wejsciowych struktur.

5.5.1.2 bool operator== (StrukturaDanych & *T1,* StrukturaDanych & *T2*)

Operator porownania.

Parametry

T1	pierwsza StrukturaDanych
T2	druga StrukturaDanych

Zwraca

- true struktury maja identyczne elementy
- false elementy nie sa identyczne

5.6 Dokumentacja pliku StrukturaDanych.h

```
#include <iostream>
#include "definicje.h"
```

Komponenty

- class StrukturaDanych

  Struktura danych o funkcjonalnosci tablicy.

5.7 Dokumentacja pliku TablicaAsocjacyjna.cpp

```
#include "TablicaAsocjacyjna.h"
```

5.8 Dokumentacja pliku TablicaAsocjacyjna.h

```
#include <iostream>
#include "definicje.h"
#include "ParaKluczWartosc.h"
#include "StrukturaDanych.h"
```

Komponenty

- class TablicaAsocjacyjna

  Tablica asocjacyjna.

5.9 Dokumentacja pliku TablicaHaszujaca.h

```
#include <iostream>
#include <vector>
#include "definicje.h"
#include "ParaKluczWartosc.h"
#include "StrukturaDanych.h"
```

Komponenty

- class TablicaHaszujaca

  Tablica haszujaca.

5.10 Dokumentacja pliku TestTablicAsocjacyjnych.h

```
#include <iostream>
#include <fstream>
#include <time.h>
#include <stdlib.h>
#include "definicje.h"
#include "StrukturaDanych.h"
#include "TablicaAsocjacyjna.h"
#include "DrzewoAsocjacyjne.h"
#include "TablicaHaszujaca.h"
```

Definicje

- #define PLIK_DANYCH "dane.txt"

  nazwa pliku z danymi wejsciowymi, takze dla wyjscia generowanych liczb.
- #define LICZBA_POWTORZEN 1000

  ilosc powtorzen pomiaru czasu dla kazdego rozmiaru problemu.
- #define LICZBA_WIELKOSCI 4

  ilosc roznych rozmiarow problemu.
- #define WIELKOSCI_PROBLEMU {100,1000,10000,100000}

  tablica zawierajaca wszystkie mierzone rozmiary problemu.
- #define WIELKOSC_GENEROWANYCH_DANYCH 100000

  ilosc danych do wygenerowania.

- #define DLUGOSC_KLUCZA 6

  Liczba liter w generowanym kluczu.

Funkcje

- void **algorytm1** (TYP_KLUCZ &klucz)
- void **algorytm2** (TYP_KLUCZ &klucz)
- void **algorytm3** (TYP_KLUCZ &klucz)
- void **zmien_ilosc_elementow1** (StrukturaDanych &dane, int ile)
- void **zmien_ilosc_elementow2** (StrukturaDanych &dane, int ile)
- void **zmien_ilosc_elementow3** (StrukturaDanych &dane, int ile)
- bool wczytaj_dane (const char ∗nazwa_pliku, StrukturaDanych &tablica)

  wczytuje dane z pliku
- bool zapisz_dane (const char ∗nazwa_pliku, int ∗col_rozmiar_problemu, double ∗col_czas, int rozmiar)

  zapisuje dane do pliku .csv dane zawieraja: 1 kolumna: wielkosc problemu 2 kolumna: czas potrzebny do zrealizowanego danego problemu
- void generuj_dane (char ∗nazwa_pliku)

  Generuje nowe dane do pliku.
- void testuj_algorytm (void(∗algorytm)(TYP_KLUCZ &klucz), void(∗zmien_ilosc_elementow)(StrukturaDanych &dane, int ile), StrukturaDanych &dane, const char ∗plik_wyjsciowy)

  wykonuje testy czasu algorytmu dla przygotowanych parametrow zmierzone czasu zapisuje do pliku
- int main ()

  funkcja sprawdza poprawnosc dzialania struktur danych.

Zmienne

- TablicaAsocjacyjna **tab1**
- DrzewoAsocjacyjne **tab2**
- TablicaHaszujaca **tab3**

5.10.1 Dokumentacja funkcji

5.10.1.1 int main ( )

funkcja sprawdza poprawnosc dzialania struktur danych.

Parametry

void(∗algorytm)(StrukturaDanych&	dane) Wskaznik na wybrany algorytm.
dane	Dane do obrobki, niezmieniane.
ile_liczb	Wielkosc problemu do sprawdzenia, niewielka aby dalo sie zobaczyc w konsoli.

Zwraca

Zwraca Strukture po wykonaniu algorytmu. funkcja main

5.10.1.2 void testuj_algorytm (void(∗)(TYP_KLUCZ &klucz) *algorytm,* void(∗)(StrukturaDanych &dane, int ile) *zmien_ilosc_elementow,* StrukturaDanych & *dane,* const char ∗ *plik_wyjsciowy*)

wykonuje testy czasu algorytmu dla przygotowanych parametrow zmierzone czasu zapisuje do pliku

Parametry

double(∗algorytm)	funkcja z algorytmem do testowania
∗tablica	dane dla algorytmu
rozmiar	rozmiar tablicy
∗plik_wyjsciowy	nazwa pliku do zapisu zmierzonych czasow

5.10.1.3 bool wczytaj_dane (const char ∗ *nazwa_pliku,* StrukturaDanych & *tablica*)

wczytuje dane z pliku

Format:

liczba_danych

dana1

dana2

.

Parametry

∗nazwa_pliku	nazwa pliku z danymi
∗&tablica	tablica docelowa (usuwana w przypadku !=NULL)
&rozmiar	rozmiar tablicy docelowej

5.10.1.4 bool zapisz_dane (const char ∗ *nazwa_pliku,* int ∗ *col_rozmiar_problemu,* double ∗ *col_czas,* int *rozmiar*)

zapisuje dane do pliku .csv dane zawieraja: 1 kolumna: wielkosc problemu 2 kolumna: czas potrzebny do zrealizowanego danego problemu

Parametry

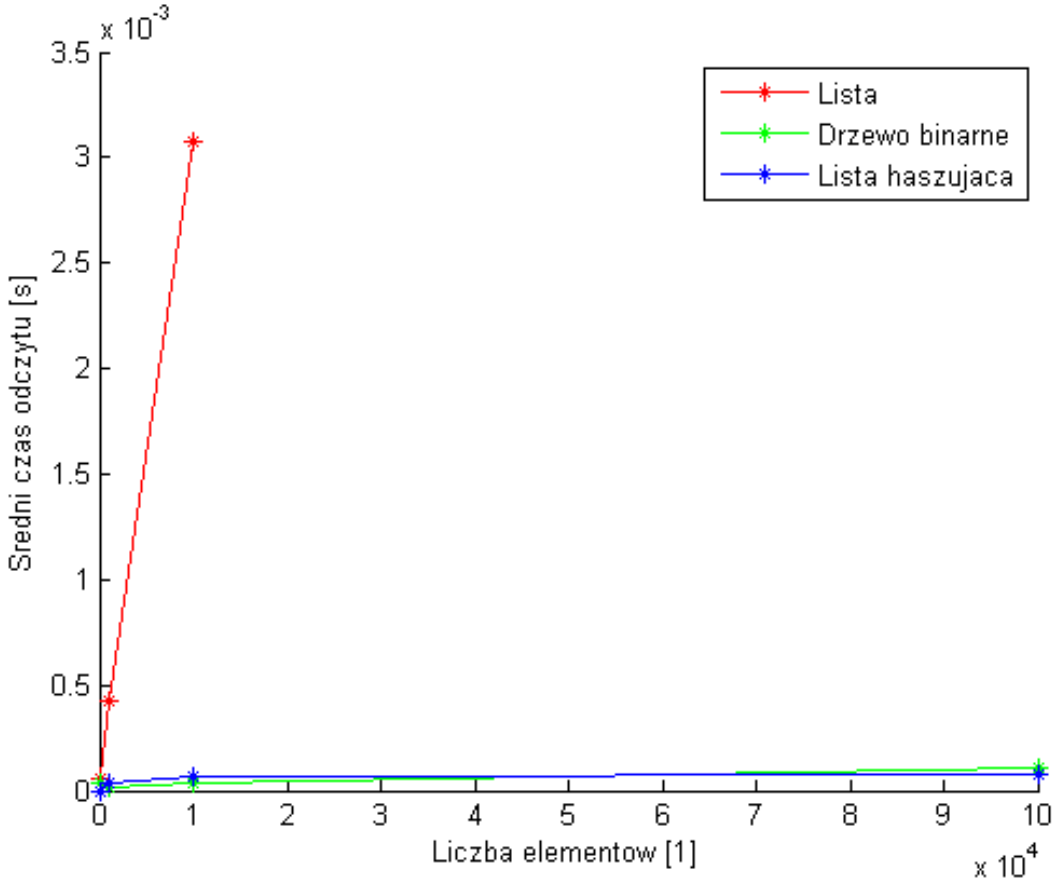
∗nazwa_pliku	nazwa pliku do zapisu
∗col_rozmiar_problemu	tablica z 1 kolumna
∗col_czas	druga kolumna
rozmiar	rozmiar obu tablic

Zwraca

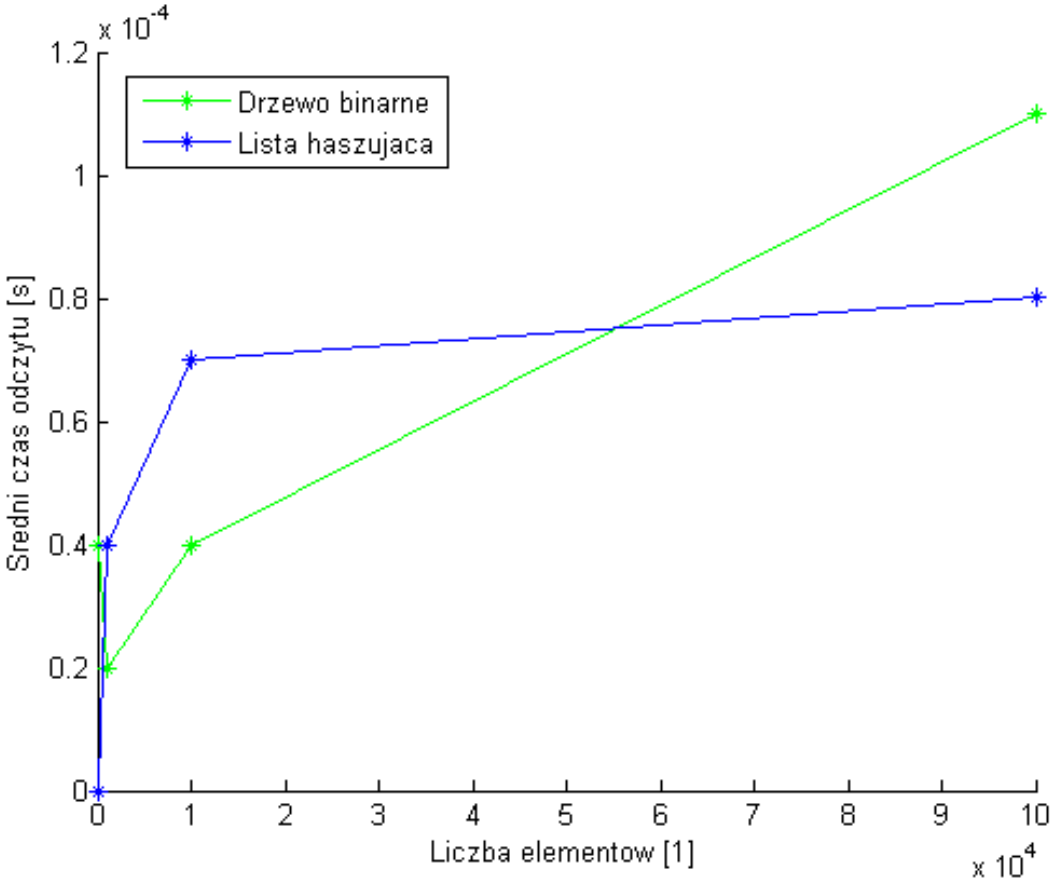
- true sukces
- false blad

6 Sprawozdanie z wykonania programu labolatorium 6 i 7.

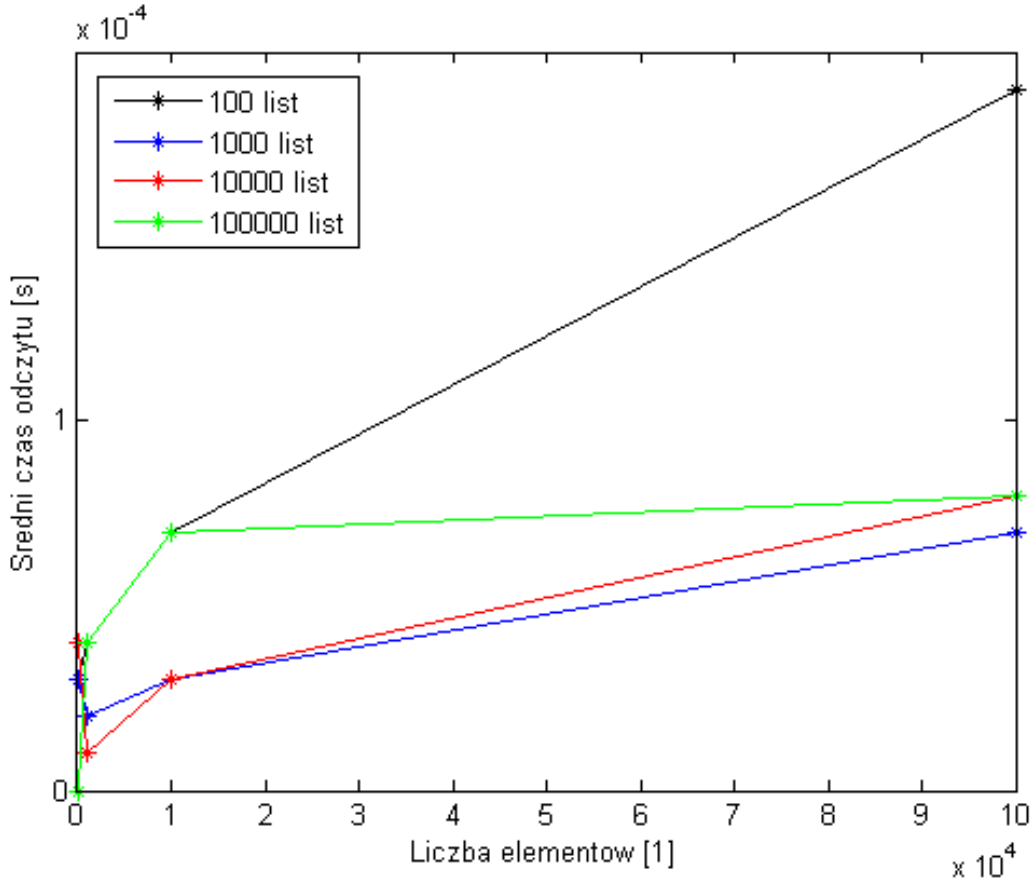
Pomiar czasu dla wyszukiwania elementu na strukturach asocjacyjnych zaimplenentowanych na tablicy, drzewie binarnym i tablicy haszujacej. Porownanie wszystkich implementacji znajduje sie na wykresie 1.
Z wykresu wynika, że implementacja o najwiekszej zlozonosci obliczeniowej to implementacja na nieposortowanej tablicy (zlozonosc co najmniej liniowa). Dokladniejsze porownanie dwoch nastepnych implementacji znajduje sie na wykresie 2.
Implementacja na drzewie binarnym ma zlozonosc logarytmiczna co widac na wykresie. Zlozonosc implementacji na tablicy haszujacej zalezy od ilosc list jakie zawiera. W szczegolnym przypadku moze byc szybsza od implementacji na drzewie i o mniejszej zlozonosci (stalej). Szczegolowy pomiar czasu dla roznych wartosci liczby list znajduje sie na wykresie 3.
Z wykresu wynika, ze jesli ilosc list jest nie mniejsza niz ilosc elementow ktore zawiera cala tablica to czas odczytu jest staly (wykres zielony). Ze wzrostem elementow ponad liczba list, czas zaczyna rosnac liniowo (wykresy pozostale).



Rysunek 1: Wykres pomiaru czasu wyszukiwania na wszystkich implementacjach struktur asocjacyjnych.



Rysunek 2: Wykres pomiaru czasu wyszukiwania na drzewie binarnym i tablicy haszujacej.



Rysunek 3: Wykres pomiaru czasu wyszukiwania na roznych rozmiarow tablicy haszujacej.

Skorowidz

czy_pusta
    DrzewoAsocjacyjne, 2
    TablicaAsocjacyjna, 10

definicje.h, 13
dodaj
    DrzewoAsocjacyjne, 3
    TablicaAsocjacyjna, 10
    TablicaHaszujaca, 12
dodaj_element
    StrukturaDanych, 5
dodaj_elementy
    StrukturaDanych, 5
DrzewoAsocjacyjne, 2
    czy_pusta, 2
    dodaj, 3
    ilosc_elementow, 3
    pobierz, 3
    przepisz_strukture, 3
    usun, 3
    zmien, 3
DrzewoAsocjacyjne.cpp, 14
DrzewoAsocjacyjne.h, 14

element_na
    StrukturaDanych, 7

ilosc_elementow
    DrzewoAsocjacyjne, 3
    StrukturaDanych, 7
    TablicaAsocjacyjna, 10

kopiuj_wycinek
    StrukturaDanych, 7

main
    TestTablicAsocjacyjnych.h, 17

odwroc_kolejnosc
    StrukturaDanych, 7
operator+
    StrukturaDanych, 8
    StrukturaDanych.cpp, 15
operator+=
    StrukturaDanych, 7
operator=
    StrukturaDanych, 8
operator==
    StrukturaDanych, 9
    StrukturaDanych.cpp, 15

ParaKluczWartosc, 4
ParaKluczWartosc.h, 14
pobierz
    DrzewoAsocjacyjne, 3
    TablicaAsocjacyjna, 10
    TablicaHaszujaca, 12

przepisz_strukture
    DrzewoAsocjacyjne, 3
    TablicaAsocjacyjna, 10
    TablicaHaszujaca, 12

StrukturaDanych, 4
    dodaj_element, 5
    dodaj_elementy, 5
    element_na, 7
    ilosc_elementow, 7
    kopiuj_wycinek, 7
    odwroc_kolejnosc, 7
    operator+, 8
    operator+=, 7
    operator=, 8
    operator==, 9
    StrukturaDanych, 5
    StrukturaDanych, 5
    wypisz_wszystko, 8
    zamien_elementy, 8
    zmien_element, 8
StrukturaDanych.cpp, 15
    operator+, 15
    operator==, 15
StrukturaDanych.h, 15

TablicaAsocjacyjna, 9
    czy_pusta, 10
    dodaj, 10
    ilosc_elementow, 10
    pobierz, 10
    przepisz_strukture, 10
    TablicaAsocjacyjna, 10
    TablicaAsocjacyjna, 10
    usun, 11
    zmien, 11
TablicaAsocjacyjna.cpp, 15
TablicaAsocjacyjna.h, 16
TablicaHaszujaca, 11
    dodaj, 12
    pobierz, 12
    przepisz_strukture, 12
    TablicaHaszujaca, 12
    TablicaHaszujaca, 12
    usun, 12
    zmien, 12
TablicaHaszujaca.h, 16
TestTablicAsocjacyjnych.h, 16
    main, 17
    testuj_algorytm, 17
    wczytaj_dane, 18
    zapisz_dane, 18
testuj_algorytm
    TestTablicAsocjacyjnych.h, 17

usun

DrzewoAsocjacyjne, 3
TablicaAsocjacyjna, 11
TablicaHaszujaca, 12

wczytaj_dane
TestTablicAsocjacyjnych.h, 18
Wierzcholek, 13
wypisz_wszystko
StrukturaDanych, 8

zamien_elementy
StrukturaDanych, 8
zapisz_dane
TestTablicAsocjacyjnych.h, 18
zmien
DrzewoAsocjacyjne, 3
TablicaAsocjacyjna, 11
TablicaHaszujaca, 12
zmien_element
StrukturaDanych, 8