

PAMSI LAB V

0.3

Wygenerowano przez Doxygen 1.8.6

Cz, 16 kwi 2015 11:52:17

Spis treści

1	Strona główna	1
1.1	Programu	1
2	Indeks hierarchiczny	3
2.1	Hierarchia klas	3
3	Indeks klas	5
3.1	Lista klas	5
4	Indeks plików	7
4.1	Lista plików	7
5	Dokumentacja klas	9
5.1	Dokumentacja klasy BenchmarkInterfejs	9
5.1.1	Opis szczegółowy	9
5.1.2	Dokumentacja funkcji składowych	9
5.1.2.1	_Generator	9
5.1.2.2	_IleKolizja	10
5.1.2.3	_Test	10
5.1.2.4	_Wczytaj	10
5.1.2.5	_WykonajTest	10
5.1.2.6	_Zaladuj	10
5.2	Dokumentacja struktury Element	11
5.2.1	Opis szczegółowy	11
5.2.2	Dokumentacja funkcji składowych	11
5.2.2.1	operator=	11
5.2.3	Dokumentacja atrybutów składowych	11
5.2.3.1	_Klucz	11
5.2.3.2	_Wartosc	11
5.3	Dokumentacja szablonu klasy StosTab< Typ >	11
5.3.1	Opis szczegółowy	12
5.3.2	Dokumentacja konstruktora i destruktoru	12
5.3.2.1	StosTab	12

5.3.2.2	StosTab	12
5.3.2.3	~StosTab	13
5.3.3	Dokumentacja funkcji składowych	13
5.3.3.1	_Adres	13
5.3.3.2	_Pokaz	13
5.3.3.3	_Pop	13
5.3.3.4	_Push	13
5.3.3.5	_Rozmiar	14
5.3.3.6	_Wartosc	14
5.3.3.7	_Zwolnij	14
5.3.4	Dokumentacja atrybutów składowych	14
5.3.4.1	_L	14
5.3.4.2	_RozmiarL	14
5.3.4.3	_RozmiarT	14
5.4	Dokumentacja szablonu klasy Struktury< Typ >	15
5.4.1	Opis szczegółowy	15
5.4.2	Dokumentacja funkcji składowych	15
5.4.2.1	_Pokaz	15
5.4.2.2	_Pop	15
5.4.2.3	_Push	16
5.4.2.4	_Rozmiar	16
5.4.2.5	_Zwolnij	16
5.5	Dokumentacja klasy StrukturyBenchmark	16
5.5.1	Opis szczegółowy	17
5.5.2	Dokumentacja funkcji składowych	17
5.5.2.1	_IleKolizja	17
5.5.2.2	_Test	17
5.5.2.3	_Ustaw	17
5.5.2.4	_Wczytaj	18
5.5.2.5	_Zaladuj	18
5.5.2.6	Przydziel	18
5.5.3	Dokumentacja atrybutów składowych	18
5.5.3.1	_Klucze	18
5.5.3.2	_Wartosci	18
5.5.3.3	W	18
5.6	Dokumentacja klasy TablicaAso	19
5.6.1	Opis szczegółowy	19
5.6.2	Dokumentacja funkcji składowych	19
5.6.2.1	operator()	19
5.6.2.2	operator[]	19

5.6.2.3	Usun_wszystkie	20
5.7	Dokumentacja klasy TablicaHash	20
5.7.1	Opis szczegółowy	21
5.7.2	Dokumentacja konstruktora i destruktor	21
5.7.2.1	TablicaHash	21
5.7.2.2	~TablicaHash	21
5.7.3	Dokumentacja funkcji składowych	21
5.7.3.1	_Dodaj	21
5.7.3.2	_Haszuj	21
5.7.3.3	_Kolizja	22
5.7.3.4	_Stworz_Rekord	22
5.7.3.5	_Usun	22
5.7.3.6	_Wyczysc	22
5.7.3.7	_Zwroc	22
5.7.3.8	IloscKolizji	23
5.7.4	Dokumentacja atrybutów składowych	23
5.7.4.1	_Tab	23
5.7.4.2	K	23
6	Dokumentacja plików	25
6.1	Dokumentacja pliku BenchmarkInterfejs.cpp	25
6.1.1	Opis szczegółowy	25
6.2	Dokumentacja pliku BenchmarkInterfejs.hh	25
6.2.1	Dokumentacja definicji	26
6.2.1.1	DLUGOSC_SLOWA	26
6.2.1.2	ILOSC	26
6.3	Dokumentacja pliku Element.hh	26
6.4	Dokumentacja pliku Main.cpp	26
6.4.1	Opis szczegółowy	26
6.4.2	Dokumentacja definicji	26
6.4.2.1	ILOSC_POW	26
6.4.3	Dokumentacja funkcji	27
6.4.3.1	main	27
6.5	Dokumentacja pliku StosTab.hh	27
6.6	Dokumentacja pliku strona-glowna.dox	27
6.7	Dokumentacja pliku Struktury.hh	27
6.8	Dokumentacja pliku StrukturyBenchmark.cpp	27
6.9	Dokumentacja pliku StrukturyBenchmark.hh	27
6.10	Dokumentacja pliku TablicaAso.cpp	28
6.10.1	Opis szczegółowy	28

6.11 Dokumentacja pliku TablicaAso.hh	28
6.12 Dokumentacja pliku TablicaHash.cpp	28
6.12.1 Opis szczegółowy	28
6.13 Dokumentacja pliku TablicaHash.hh	28
6.13.1 Dokumentacja definicji	29
6.13.1.1 ROZMIAR	29
Indeks	30

Rozdział 1

Strona główna

Autor

Bartłomiej Ankowski

Data

16.04.2015

Wersja

0.3

1.1 Programu

Rozdział 2

Indeks hierarchiczny

2.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

BenchmarkInterfejs	9
StrukturyBenchmark	16
Element	11
Struktury< Typ >	15
StosTab< Typ >	11
Struktury< Element >	15
StosTab< Element >	11
TablicaHash	20
TablicaAso	19

Rozdział 3

Indeks klas

3.1 Lista klas

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

BenchmarkInterfejs	
Modeluje pojecie Interfejsu Benchmark'u	9
Element	
Element Element jest struktura lacząca Klucz i wartosc, razem tworzy rekord tablicy haszujacej	11
StosTab< Typ >	11
Struktury< Typ >	
Modeluje pojecie Struktury danych, klasa bazowa dla Stosu,Kolejki i Listy,zarowno w implemen-	
tacji wskaznikowej jak i tablicowej	15
StrukturyBenchmark	16
TablicaAso	19
TablicaHash	20

Rozdział 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

BenchmarkInterfejs.cpp	
Definicje Metod klasy BenchmarkInterfejs	25
BenchmarkInterfejs.hh	25
Element.hh	26
Main.cpp	
Funkcja glowna programu	26
StosTab.hh	27
Struktury.hh	27
StrukturyBenchmark.cpp	27
StrukturyBenchmark.hh	27
TablicaAso.cpp	
Definicje Metod klasy TablicaAso	28
TablicaAso.hh	28
TablicaHash.cpp	
Definicje Metod klasy TablicaHash	28
TablicaHash.hh	28

Rozdział 5

Dokumentacja klas

5.1 Dokumentacja klasy BenchmarkInterfejs

Modeluje pojecie Interfejsu Benchmark'u.

```
#include <BenchmarkInterfejs.hh>
```

Diagram dziedziczenia dla BenchmarkInterfejs

Metody publiczne

- void [_WykonajTest](#) (const unsigned int Ilosc_Pow)
Metoda wykonujaca test odpowiedniej struktury.
- void [_Generator](#) () const

Metody chronione

- virtual void [_Test](#) (const unsigned int Ilosc) const =0
Metoda Wykonujaca pojedyncza operacje.
- virtual void [_Wczytaj](#) (string PlikWart, string PlikKlucz)=0
Metoda wczytujaca dane z pliku Metoda ma za zadanie wczytac dane z pliku wejsciowego.
- virtual void [_Zaladuj](#) (const unsigned int n) const =0
- virtual void [_IleKolizja](#) ()=0

5.1.1 Opis szczegółowy

Modeluje pojecie Interfejsu Benchmark'u.

Klasa bazowa dla implementowania benchmarku dla kolejnych struktur danych

Definicja w linii 31 pliku BenchmarkInterfejs.hh.

5.1.2 Dokumentacja funkcji składowych

5.1.2.1 void BenchmarkInterfejs::_Generator () const

Metoda generujaca dane

Metoda ma za zadanie wygenerowac plik z danymi z przedzialu (1-99), jest wywoływana gdy uzytkownik nie poda w argumencie wywołania programu nazwy pliku wejsciowego z danymi

Parametry

in	<i>PlikOut</i>	- Nazwa plik w ktorym zapisywane sa wygenerowane dane
in	<i>n</i>	- Ilosc wygenerowanych danych

Definicja w linii 39 pliku BenchmarkInterfejs.cpp.

5.1.2.2 `virtual void BenchmarkInterfejs::_IleKolizja () [protected], [pure virtual]`

Implementowany w [StrukturyBenchmark](#).

5.1.2.3 `virtual void BenchmarkInterfejs::_Test (const unsigned int llosc) const [protected], [pure virtual]`

Metoda Wykonujaca pojedyncza operacje.

Metoda ma za zadanie wykonan pojedyncza operacja, ktorej czas jest rejestrowany

Parametry

in	<i>llosc</i>	- Liczba danych poddana testowi
----	--------------	---------------------------------

Implementowany w [StrukturyBenchmark](#).

5.1.2.4 `virtual void BenchmarkInterfejs::_Wczytaj (string PlikWart, string PlikKlucz) [protected], [pure virtual]`

Metoda wczytujaca dane z pliku Metoda ma za zadanie wczytac dane z pliku wejscowego.

Parametry

in	<i>PlikIn</i>	- Nazwa pliku wejscowego
in	<i>n</i>	- liczba wczytywanych danych

Implementowany w [StrukturyBenchmark](#).

5.1.2.5 `void BenchmarkInterfejs::_WykonajTest (const unsigned int llosc_Pow)`

Metoda wykonujaca test odpowiedniej struktury.

Metoda ma za zadanie wykonac Benchmark dla struktury, dla ustawionej ilosci danych i okreslona przez argument metody ilosc powtorzen.

Parametry

in	<i>llosc_Pow</i>	- okresla ile razy ma sie wykonac test
----	------------------	--

Definicja w linii 15 pliku BenchmarkInterfejs.cpp.

Oto graf wywoływań tej funkcji:

5.1.2.6 `virtual void BenchmarkInterfejs::_Zaladuj (const unsigned int n) const [protected], [pure virtual]`

Implementowany w [StrukturyBenchmark](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

- [BenchmarkInterfejs.hh](#)
- [BenchmarkInterfejs.cpp](#)

5.2 Dokumentacja struktury Element

`Element` `Element` jest struktura łącząca Klucz i wartość, razem tworzy rekord tablicy haszującej.

```
#include <Element.hh>
```

Metody publiczne

- `Element & operator=` (const `Element` &E)

Atrybuty publiczne

- `std::string _Klucz`
Pole Klucz Pole przechowuje klucz odpowiedniego rekordu.
- `int _Wartosc`
Pole Wartosc Pole przechowuje Wartosc skojarzona z danym kluczem.

5.2.1 Opis szczegółowy

`Element` `Element` jest struktura łącząca Klucz i wartość, razem tworzy rekord tablicy haszującej.

Definicja w linii 23 pliku `Element.hh`.

5.2.2 Dokumentacja funkcji składowych

5.2.2.1 `Element& Element::operator= (const Element & E) [inline]`

Definicja w linii 40 pliku `Element.hh`.

5.2.3 Dokumentacja atrybutów składowych

5.2.3.1 `std::string Element::_Klucz`

Pole Klucz Pole przechowuje klucz odpowiedniego rekordu.

Definicja w linii 31 pliku `Element.hh`.

5.2.3.2 `int Element::_Wartosc`

Pole Wartosc Pole przechowuje Wartosc skojarzona z danym kluczem.

Definicja w linii 38 pliku `Element.hh`.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- `Element.hh`

5.3 Dokumentacja szablonu klasy `StosTab< Typ >`

```
#include <StosTab.hh>
```

Diagram dziedziczenia dla `StosTab< Typ >`

Diagram współpracy dla `StosTab< Typ >`:

Metody publiczne

- [StosTab](#) ()
Konstruktor Podczas tworzenia obiektu tej klasy automatycznie alokowana jest tablica o rozmiarze 1 oraz ustawienie obecnej liczby elementów listy na 0.
- [StosTab](#) (const [StosTab](#) &K)
Konstruktor Kopiujący.
- virtual [~StosTab](#) ()
Destruktor obiektu.
- void [_Pokaz](#) () const
Metoda wypisująca elementy Stosu.
- Typ [_Pop](#) (unsigned int Pozycja=0)
Metoda ściągająca element ze stosu.
- void [_Push](#) (Typ k, unsigned int Pozycja=0)
Metoda dodająca element do tablicy.
- unsigned int [_Rozmiar](#) () const
Metoda zwracająca rozmiar listy.
- Typ [_Wartosc](#) (unsigned int Index) const
- Typ & [_Adres](#) (unsigned int Index)

Metody prywatne

- void [_Zwolnij](#) ()

Atrybuty prywatne

- Typ * [_L](#)
Pole klasy [StosTab](#).
- unsigned int [_RozmiarL](#)
Pole Klasy [StosTab](#).
- unsigned int [_RozmiarT](#)
Pole Klasy [StosTab](#).

5.3.1 Opis szczegółowy

```
template<class Typ>class StosTab< Typ >
```

Definicja w linii 16 pliku StosTab.hh.

5.3.2 Dokumentacja konstruktora i destruktora

5.3.2.1 `template<class Typ> StosTab< Typ >::StosTab () [inline]`

Konstruktor Podczas tworzenia obiektu tej klasy automatycznie alokowana jest tablica o rozmiarze 1 oraz ustawienie obecnej liczby elementów listy na 0.

Definicja w linii 66 pliku StosTab.hh.

5.3.2.2 `template<class Typ> StosTab< Typ >::StosTab (const StosTab< Typ > & K) [inline]`

Konstruktor Kopiujący.

Definicja w linii 77 pliku StosTab.hh.

5.3.2.3 `template<class Typ> virtual StosTab< Typ >::~~StosTab () [inline]`

Destruktor obiektu.

Definicja w linii 88 pliku StosTab.hh.

5.3.3 Dokumentacja funkcji składowych

5.3.3.1 `template<class Typ> Typ& StosTab< Typ >::_Adres (unsigned int Index) [inline]`

Definicja w linii 193 pliku StosTab.hh.

5.3.3.2 `template<class Typ> void StosTab< Typ >::_Pokaz () const [inline],[virtual]`

Metoda wypisująca elementy Stosu.

Metoda ma za zadanie wypisać wszystkie elementy znajdujące się obecnie na liście danych

Implementuje [Struktury< Typ >](#).

Definicja w linii 97 pliku StosTab.hh.

5.3.3.3 `template<class Typ> Typ StosTab< Typ >::_Pop (unsigned int Pozycja = 0) [inline],[virtual]`

Metoda ściągająca element ze stosu.

Metoda ma za zadanie ściągnąć ostatni element stosu, w przypadku gdy tablica jest do połowy pusta następuje utworzenie nowej tablicy o dwa razy mniejszym rozmiarze

Parametry

in	<i>Pozycja</i>	- numer elementu który zostanie usunięty z listy i zostanie zwrócona jego wartość
----	----------------	---

Zwraca

Zwraca wybrany przez użytkownika element

Implementuje [Struktury< Typ >](#).

Definicja w linii 119 pliku StosTab.hh.

Oto graf wywołań tej funkcji:

5.3.3.4 `template<class Typ> void StosTab< Typ >::_Push (Typ k, unsigned int Pozycja = 0) [inline],[virtual]`

Metoda dodająca element do tablicy.

Metoda ma za zadanie dodać nowy element na końcu stosu, w przypadku zapelnienia tablicy następuje utworzenie nowej tablicy i przepisanie elementów

Parametry

in	<i>k</i>	- wartość jaką chcemy dodać do listy
in	<i>Pozycja</i>	- Pozycja na której chcemy dodać wartość

Implementuje [Struktury< Typ >](#).

Definicja w linii 158 pliku StosTab.hh.

5.3.3.5 `template<class Typ> unsigned int StosTab< Typ >::_Rozmiar () const [inline],[virtual]`

Metoda zwracająca rozmiar listy.

Metoda zwraca informacje o obecnej ilości danych w strukturze

Zwraca

Zwraca ilość elementów listy

Implementuje [Struktury< Typ >](#).

Definicja w linii 189 pliku StosTab.hh.

Oto graf wywołań tej funkcji:

5.3.3.6 `template<class Typ> Typ StosTab< Typ >::_Wartosc (unsigned int Index) const [inline]`

Definicja w linii 192 pliku StosTab.hh.

5.3.3.7 `template<class Typ> void StosTab< Typ >::_Zwolnij () [inline],[private],[virtual]`

Metoda zwalniania pamięci

Metoda ma za zadanie zwolnić pamięć zajmowaną przez dane, dopóki ilość elementów listy nie wynosi 0 wykonywana jest metoda `_Pop`, aby opróżnić stos i zwolnić pamięć

Implementuje [Struktury< Typ >](#).

Definicja w linii 51 pliku StosTab.hh.

5.3.4 Dokumentacja atrybutów składowych

5.3.4.1 `template<class Typ> Typ* StosTab< Typ >::_L [private]`

Pole klasy [StosTab](#).

Pole zawiera wskaźnik na typ całkowity, służy do alokacji pamięci na dynamiczną tablicę

Definicja w linii 26 pliku StosTab.hh.

5.3.4.2 `template<class Typ> unsigned int StosTab< Typ >::_RozmiarL [private]`

Pole klasy [StosTab](#).

Pole przechowuje informacje o ilości obecnie znajdujących się elementów na liście danych

Definicja w linii 34 pliku StosTab.hh.

5.3.4.3 `template<class Typ> unsigned int StosTab< Typ >::_RozmiarT [private]`

Pole klasy [StosTab](#).

Pole przechowuje informacje o obecnym rozmiarze tablicy danych

Definicja w linii 42 pliku StosTab.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [StosTab.hh](#)

5.4 Dokumentacja szablonu klasy Struktury< Typ >

Modeluje pojecie [Struktury](#) danych, klasa bazowa dla Stosu,Kolejki i Listy,zarowno w implemenetacji wskaznikowej jak i tablicowej.

```
#include <Struktury.hh>
```

Diagram dziedziczenia dla Struktury< Typ >

Metody publiczne

- virtual void [_Push](#) (Typ k, unsigned int Pozycja)=0
Metoda dodajaca kolejny element struktury.
- virtual Typ [_Pop](#) (unsigned int Pozycja)=0
Metoda usuwajaca element.
- virtual unsigned int [_Rozmiar](#) () const =0
Metoda zwracajaca rozmiar [Struktury](#).
- virtual void [_Pokaz](#) () const =0
Metoda wyswietlajaca dane.
- virtual void [_Zwolnij](#) ()=0
Metoda zwalnijaca pamiec.

5.4.1 Opis szczegółowy

```
template<class Typ>class Struktury< Typ >
```

Modeluje pojecie [Struktury](#) danych, klasa bazowa dla Stosu,Kolejki i Listy,zarowno w implemenetacji wskaznikowej jak i tablicowej.

Definicja w linii 26 pliku Struktury.hh.

5.4.2 Dokumentacja funkcji składowych

5.4.2.1 `template<class Typ> virtual void Struktury< Typ >::_Pokaz () const [pure virtual]`

Metoda wyswietlajaca dane.

Metoda ma za zadanie wyswietlic wszystkie dane nalezace do struktury

Implementowany w [StosTab< Typ >](#) i [StosTab< Element >](#).

5.4.2.2 `template<class Typ> virtual Typ Struktury< Typ >::_Pop (unsigned int Pozycja) [pure virtual]`

Metoda usuwajaca element.

Metoda ma za zadanie usunac element i w zaleznosci od implementowanej struktury bedzie to usuwany element usuwany z poczatku,końca lub w przypadku listy z dowolnego jej miejsca

Parametry

in	Pozycja	- Numer elementu ,ktory zostanie dodany. Argument ma znaczenie tylko w przypadku listy i domyślnie jest ustawiony, tak aby element był dodawany zawsze na poczatku listy
----	---------	--

Zwraca

Zwraca wartosc elementu z odpowiedniego dla wybranej struktury miejsca

Implementowany w [StosTab< Typ >](#) i [StosTab< Element >](#).

5.4.2.3 `template<class Typ> virtual void Struktury< Typ >::_Push (Typ k, unsigned int Pozycja) [pure virtual]`

Metoda dodajaca kolejny element struktury.

Metoda ma za zadanie dodac kolejny element do naszej struktury oraz zapisac w nim odpowiednia wartosc. W zaleznosci od implementowanej struktury element bedzie dodawany na poczatku lub na koncu struktury danych.

Parametry

<code>in</code>	<code>k</code>	- wartosc typu calkowitnego, ktora bedzie umieszczona w strukturze
-----------------	----------------	--

Implementowany w [StosTab< Typ >](#) i [StosTab< Element >](#).

5.4.2.4 `template<class Typ> virtual unsigned int Struktury< Typ >::_Rozmiar () const [pure virtual]`

Metoda zwracajaca rozmiar [Struktury](#).

Metoda ma zadanie zwrocic bierzaca liczbe elementow nalezacych do danej struktury

Zwraca

- Bierzaca liczba elementow [Struktury](#) danych

Implementowany w [StosTab< Typ >](#) i [StosTab< Element >](#).

5.4.2.5 `template<class Typ> virtual void Struktury< Typ >::_Zwolnij () [pure virtual]`

Metoda zwalnijajaca pamiec.

Metoda ma za zadanie zwolnic pamiec uzywana przy zapelnienianiu danej struktury danymi

Implementowany w [StosTab< Typ >](#) i [StosTab< Element >](#).

Dokumentacja dla tej klasy zostala wygenerowana z pliku:

- [Struktury.hh](#)

5.5 Dokumentacja klasy StrukturyBenchmark

```
#include <StrukturyBenchmark.hh>
```

Diagram dziedziczenia dla StrukturyBenchmark

Diagram wspolpracy dla StrukturyBenchmark:

Metody publiczne

- void [_Ustaw](#) ([TablicaAso](#) &A)
- void [_Wczytaj](#) (string PlikWart, string PlikKlucz)
Metoda Wczytujaca dane.
- void [_IleKolizja](#) ()

Metody prywatne

- void `_Test` (const unsigned int n) const
Metoda wykonująca test dla odpowiedniej struktury.
- void `_Zaladuj` (const unsigned int n) const
- void `Przydziel` ()

Atrybuty prywatne

- `TablicaAso` * `W`
Pole StrukturyBenchmark Pole zawiera wskaźnik na [Struktury](#), za pomocą niego i metod wirtualnych będą wywoływane odpowiednie dla danej struktury metody.
- int * `_Wartosci`
Pole StrukturyBenchmark Pole zawiera wskaźnik na typ całkowity, służy on do alokowania pamięci dla wczytanych z pliku danych.
- std::string * `_Klucze`
Pole StrukturyBenchmark Pole zawiera wskaźnik na string, służy on do alokowania pamięci dla wczytanych z pliku danych.

Dodatkowe Dziedziczone Składowe

5.5.1 Opis szczegółowy

Klasa modeluje pojęcie Benchmarku przeznaczonego dla struktur danych przechowujące dane
Definicja w linii 23 pliku StrukturyBenchmark.hh.

5.5.2 Dokumentacja funkcji składowych

5.5.2.1 void StrukturyBenchmark::_IleKolizja () [inline], [virtual]

Implementuje [BenchmarkInterfejs](#).

Definicja w linii 83 pliku StrukturyBenchmark.hh.

5.5.2.2 void StrukturyBenchmark::_Test (const unsigned int n) const [private], [virtual]

Metoda wykonująca test dla odpowiedniej struktury.

Metoda ma za zadanie wykonać zapelnienie struktury danymi o zadanej w argumencie ilości

Parametry

in	n	- ilość danych którą zapelniona struktura
----	---	---

Implementuje [BenchmarkInterfejs](#).

Definicja w linii 4 pliku StrukturyBenchmark.cpp.

5.5.2.3 void StrukturyBenchmark::_Ustaw (TablicaAso & A) [inline]

Definicja w linii 70 pliku StrukturyBenchmark.hh.

Oto graf wywołań tej funkcji:

5.5.2.4 void StrukturyBenchmark::_Wczytaj (string PlikWart, string PlikKlucz) [virtual]

Metoda Wczytujaca dane.

Metoda ma za zadanie wczytac dane wejsciowe o podanej przez argument nazwie oraz przypisac wskaznik do zaalokowanych w pamieci danych

Parametry

in	PlikIn	- nazwa pliku wejsciowego z danymi
in	Ilosc	- Ilosc danych jaka bedzie wczytywana

Implementuje [BenchmarkInterfejs](#).

Definicja w linii 27 pliku StrukturyBenchmark.cpp.

Oto graf wywoływań tej funkcji:

5.5.2.5 void StrukturyBenchmark::_Zaladuj (const unsigned int n) const [private], [virtual]

Implementuje [BenchmarkInterfejs](#).

Definicja w linii 10 pliku StrukturyBenchmark.cpp.

5.5.2.6 void StrukturyBenchmark::_Przydziel () [private]

Definicja w linii 16 pliku StrukturyBenchmark.cpp.

5.5.3 Dokumentacja atrybutów składowych

5.5.3.1 std::string* StrukturyBenchmark::_Klucze [private]

Pole StrkturyBenchmark Pole zawiera wskaznik na string, sluzy on do alokowania pamieci dla wczytanych z pliku danych.

Definicja w linii 51 pliku StrukturyBenchmark.hh.

5.5.3.2 int* StrukturyBenchmark::_Wartosci [private]

Pole StrkturyBenchmark Pole zawiera wskaznik na typ calkowity, sluzy on do alokowania pamieci dla wczytanych z pliku danych.

Definicja w linii 43 pliku StrukturyBenchmark.hh.

5.5.3.3 TablicaAso* StrukturyBenchmark::W [private]

Pole StrukturyBenchmark Pole zawiera wskaźnik na [Struktury](#), za pomoca niego i metod wirtualnych beda wywoływane odpowiednie dla danej strktury metody.

Definicja w linii 35 pliku StrukturyBenchmark.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

- [StrukturyBenchmark.hh](#)
- [StrukturyBenchmark.cpp](#)

5.6 Dokumentacja klasy TablicaAso

```
#include <TablicaAso.hh>
```

Diagram dziedziczenia dla TablicaAso

Diagram współpracy dla TablicaAso:

Metody publiczne

- const int `operator()` (const std::string &Klucz) const
operator Indeksowania(odczyt)
- int & `operator[]` (const std::string &Klucz)
operator indeksowania(zapis)
- void `Usun_wszystkie` ()
Metoda Usuwejaca.

Dodatkowe Dziedziczone Składowe

5.6.1 Opis szczegółowy

Definicja w linii 15 pliku TablicaAso.hh.

5.6.2 Dokumentacja funkcji składowych

5.6.2.1 const int TablicaAso::operator() (const std::string & Klucz) const

operator Indeksowania(odczyt)

Przeiazienie operatora indeksowania pozwala nam w latwy sposob odczytac wartosc pasujaca do zadanego poprzez argument Klucza

Parametry

in	Klucz	- Klucz dla ktorego zostanie odczytana wartosc
----	-------	--

Zwraca

Zwraca wartosc pasujaca do danego Klucza

Definicja w linii 18 pliku TablicaAso.cpp.

5.6.2.2 int & TablicaAso::operator[] (const std::string & Klucz)

operator indeksowania(zapis)

Przciazeniei operatora indeksowania pozwala nam w latwy sposob dokonac zapisu wartosci w polu skojarzonym z kluczem podanym poprzez argument. Zapis dokonuje sie poprzez zwrocenie referencji pola pasuajcego do danego klucza

Parametry

in	Klucz	- Klucz dla ktorego dodana zostanie wartosc
----	-------	---

Zwraca

Zwraca referencje do pola zkojarzonego z podanym poprzez argument Kluczem

Definicja w linii 13 pliku TablicaAso.cpp.

5.6.2.3 void TablicaAso::Usun_wszystkie ()

Metoda Usuwajaca.

Metoda ma za zadanie usunac cala bierzaca zawartosc tablicy

Definicja w linii 23 pliku TablicaAso.cpp.

Dokumentacja dla tej klasy została wygenerowana z plików:

- [TablicaAso.hh](#)
- [TablicaAso.cpp](#)

5.7 Dokumentacja klasy TablicaHash

```
#include <TablicaHash.hh>
```

Diagram dziedziczenia dla TablicaHash

Diagram współpracy dla TablicaHash:

Metody publiczne

- void [IloscKolizji](#) ()
Wyswietla ilosc kolizji.

Metody chronione

- void [_Kolizja](#) ()
- unsigned int [_Haszuj](#) (const std::string Klucz) const
Funkcja Haszujaca.
- void [_Wyczyszc](#) ()
Metoda kasujaca.
- void [_Usun](#) (const std::string Klucz)
Metoda Usuwajaca.
- [Element](#) [_Stworz_Rekord](#) (const int W, const std::string K)
Metoda Tworzaca element.
- [TablicaHash](#) ()
Konstruktor.
- [~TablicaHash](#) ()
Destruktor.
- const int [_Zwroc](#) (const std::string &Klucz) const
Metoda odczytujaca Wartosc.
- int & [_Dodaj](#) (const std::string &Klucz)
Metoda umozliwiajaca dodanie wartosci.

Atrybuty prywatne

- [StosTab](#) < [Element](#) > * [_Tab](#) [[ROZMIAR](#)]
Pole klasy [TablicaHash](#).
- int [K](#)
Pole Klasy [TablicaHash](#).

5.7.1 Opis szczegółowy

Definicja w linii 23 pliku TablicaHash.hh.

5.7.2 Dokumentacja konstruktora i destruktor

5.7.2.1 TablicaHash::TablicaHash () [protected]

Konstruktor.

Konstruktor ma za zadanie przypisać do każdego wskaźnika tablicy NULL

Definicja w linii 13 pliku TablicaHash.cpp.

5.7.2.2 TablicaHash::~~TablicaHash () [protected]

Destruktor.

Definicja w linii 20 pliku TablicaHash.cpp.

5.7.3 Dokumentacja funkcji składowych

5.7.3.1 int & TablicaHash::_Dodaj (const std::string & *Klucz*) [protected]

Metoda umożliwiająca dodanie wartości.

Metoda ma za zadanie umożliwić przypisanie odpowiedniej wartości w polu skojarzonym z podanym poprzez argument kluczem, przypisanie odbywa się za pomocą zwracania przez tą metodę referencji do odpowiedniego pola

Parametry

in	<i>Klucz</i>	- Klucz dla którego zostanie dodana wartość
----	--------------	---

Zwraca

Zwraca referencje w celu umożliwienia przypisania wartości w odpowiednie pole

Definicja w linii 52 pliku TablicaHash.cpp.

5.7.3.2 unsigned int TablicaHash::_Haszuj (const std::string *Klucz*) const [protected]

Funkcja Haszująca.

Funkcja ta ma za zadanie określić indeks na którym zostanie dodany nowy [Element](#) do stosu. Metoda zlicza sumę znaków Klucza i zwraca resztę z dzielenia wyznaczonej sumy przez rozmiar tablicy haszującej

Parametry

in	<i>Klucz</i>	- Klucz dla którego zostanie wyznaczony indeks tablicy
----	--------------	--

Zwraca

Indeks tablicy, odpowiedni dla danego klucza

Definicja w linii 26 pliku TablicaHash.cpp.

5.7.3.3 void TablicaHash::_Kolizja () [protected]

Metoda Kolizja

Metoda ma za zadanie zwiększyć wartość przechowywaną w polu K, Wywoływana jest w przypadku wystąpienia kolizji

Definicja w linii 96 pliku TablicaHash.cpp.

5.7.3.4 Element TablicaHash::_Stworz_Rekord (const int W, const std::string K) [protected]

Metoda Tworząca element.

Metoda ma za zadanie stworzyć [Element](#)

Parametry

in	W	- Wartość składająca się na Element
in	K	- Klucz składający się na dany Element

Zwraca

Zwraca Scalny [Element](#)

Definicja w linii 34 pliku TablicaHash.cpp.

5.7.3.5 void TablicaHash::_Usun (const std::string Klucz) [protected]

Metoda Usuująca.

Metoda ma za zadanie usunąć element zdefiniowany przez podany przez argument Klucz

Parametry

in	Klucz	- Klucz dla którego zostanie usunięty pasujący Element
----	-------	--

5.7.3.6 void TablicaHash::_Wyczyszc () [protected]

Metoda kasująca.

Metoda ma za zadanie usunąć całą aktualną zawartość tablic haszujących

Definicja w linii 42 pliku TablicaHash.cpp.

5.7.3.7 const int TablicaHash::_Zwroc (const std::string & Klucz) const [protected]

Metoda odczytująca Wartość.

Metoda ma za zadanie zwrócić wartość skojarzoną z podanym przez argument Kluczem

Parametry

in	Klucz	- klucz dla którego zostanie zwrócona odpowiednia wartość, w przypadku braku pasującego w bazie klucza zostanie wyświetlony stosowny komunikat
----	-------	--

Zwraca

Zwraca wartość skojarzoną z kluczem

Definicja w linii 75 pliku TablicaHash.cpp.

5.7.3.8 void TablicaHash::IloscKolizji ()

Wyswietla ilosc kolizji.

Metoda ma za zadanie wyswietlic ilosc wystapien kolizji

Definicja w linii 98 pliku TablicaHash.cpp.

5.7.4 Dokumentacja atrybutów składowych

5.7.4.1 StosTab<Element>* TablicaHash::_Tab[ROZMIAR] [private]

Pole klasy [TablicaHash](#).

Pole to jest tablica wskaźników o rozmiarze będącym liczbą pierwszą. Wskaźniki wskazują na abstrakcyjny typ danych, zamodelowany na wcześniejszych zajęciach. Zadeklarowany stos pracuje na typie danych [Element](#), łączącym klucz z przypisaną do niego wartością

Definicja w linii 36 pliku TablicaHash.hh.

5.7.4.2 int TablicaHash::K [private]

Pole Klasy [TablicaHash](#).

Pole to przechowuje informacje o ilości kolizji jakie wystąpiły podczas dodawania rekordów do tablicy

Definicja w linii 45 pliku TablicaHash.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

- [TablicaHash.hh](#)
- [TablicaHash.cpp](#)

Rozdział 6

Dokumentacja plików

6.1 Dokumentacja pliku BenchmarkInterfejs.cpp

Definicje Metod klasy [BenchmarkInterfejs](#).

```
#include "BenchmarkInterfejs.hh"
#include <sys/time.h>
#include <iostream>
#include <fstream>
```

Wykres zależności załączania dla BenchmarkInterfejs.cpp:

6.1.1 Opis szczegółowy

Definicje Metod klasy [BenchmarkInterfejs](#).

Definicja w pliku [BenchmarkInterfejs.cpp](#).

6.2 Dokumentacja pliku BenchmarkInterfejs.hh

```
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <cstring>
#include <vector>
```

Wykres zależności załączania dla BenchmarkInterfejs.hh: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [BenchmarkInterfejs](#)

Modeluje pojecie Interfejsu Benchmark'u.

Definicje

- #define [DLUGOSC_SLOWA](#) 10
- #define [ILOSC](#) 1000000

6.2.1 Dokumentacja definicji

6.2.1.1 `#define DLUGOSC_SLOWA 10`

Definicja w linii 3 pliku `BenchmarkInterfejs.hh`.

6.2.1.2 `#define ILOSC 1000000`

Definicja w linii 4 pliku `BenchmarkInterfejs.hh`.

6.3 Dokumentacja pliku `Element.hh`

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

Komponenty

- struct `Element`

`Element Element` jest struktura łącząca Klucz i wartość, razem tworzy rekord tablicy haszującej.

6.4 Dokumentacja pliku `Main.cpp`

funkcja główna programu

```
#include "StosTab.hh"
#include "TablicaAso.hh"
#include "StrukturyBenchmark.hh"
```

Wykres zależności załączania dla `Main.cpp`:

Definicje

- `#define ILOSC_POW 10`

Funkcje

- int `main` (int argc, char *argv[])

6.4.1 Opis szczegółowy

funkcja główna programu

Definicja w pliku `Main.cpp`.

6.4.2 Dokumentacja definicji

6.4.2.1 `#define ILOSC_POW 10`

Definicja w linii 4 pliku `Main.cpp`.

6.4.3 Dokumentacja funkcji

6.4.3.1 `int main (int argc, char * argv[])`

Definicja w linii 10 pliku Main.cpp.

Oto graf wywołań dla tej funkcji:

6.5 Dokumentacja pliku StosTab.hh

```
#include <cstdlib>
#include <iostream>
#include "Struktury.hh"
```

Wykres zależności załączania dla StosTab.hh: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [StosTab](#)< [Typ](#) >

6.6 Dokumentacja pliku strona-glowna.dox

6.7 Dokumentacja pliku Struktury.hh

```
#include <iostream>
#include <cstring>
#include <sstream>
#include <fstream>
```

Wykres zależności załączania dla Struktury.hh: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [Struktury](#)< [Typ](#) >

Modeluje pojecie [Struktury](#) danych, klasa bazowa dla Stosu, Kolejki i Listy, zarowno w implemenetacji wskaznikowej jak i tablicowej.

6.8 Dokumentacja pliku StrukturyBenchmark.cpp

```
#include "StrukturyBenchmark.hh"
```

Wykres zależności załączania dla StrukturyBenchmark.cpp:

6.9 Dokumentacja pliku StrukturyBenchmark.hh

```
#include "BenchmarkInterfejs.hh"
#include "TablicaAso.hh"
```

Wykres zależności załączania dla StrukturyBenchmark.hh: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [StrukturyBenchmark](#)

6.10 Dokumentacja pliku TablicaAso.cpp

Definicje Metod klasy [TablicaAso](#).

```
#include "TablicaAso.hh"
```

Wykres zależności załączania dla TablicaAso.cpp:

6.10.1 Opis szczegółowy

Definicje Metod klasy [TablicaAso](#).

Definicja w pliku [TablicaAso.cpp](#).

6.11 Dokumentacja pliku TablicaAso.hh

```
#include "TablicaHash.hh"
```

Wykres zależności załączania dla TablicaAso.hh: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [TablicaAso](#)

6.12 Dokumentacja pliku TablicaHash.cpp

Definicje Metod klasy [TablicaHash](#).

```
#include "TablicaHash.hh"
```

Wykres zależności załączania dla TablicaHash.cpp:

6.12.1 Opis szczegółowy

Definicje Metod klasy [TablicaHash](#).

Definicja w pliku [TablicaHash.cpp](#).

6.13 Dokumentacja pliku TablicaHash.hh

```
#include <iostream>
#include <cstring>
#include <sstream>
#include <fstream>
#include "Element.hh"
#include "StosTab.hh"
```

Wykres zależności załączania dla TablicaHash.hh: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class `TablicaHash`

Definicje

- `#define ROZMIAR 224729`

6.13.1 Dokumentacja definicji

6.13.1.1 `#define ROZMIAR 224729`

Definicja w linii 3 pliku TablicaHash.hh.

Skorowidz

- ~StosTab
 - StosTab, [12](#)
- ~TablicaHash
 - TablicaHash, [21](#)
- _Adres
 - StosTab, [13](#)
- _Dodaj
 - TablicaHash, [21](#)
- _Generator
 - BenchmarkInterfejs, [9](#)
- _Haszuj
 - TablicaHash, [21](#)
- _IleKolizja
 - BenchmarkInterfejs, [10](#)
 - StrukturyBenchmark, [17](#)
- _Klucz
 - Element, [11](#)
- _Klucze
 - StrukturyBenchmark, [18](#)
- _Kolizja
 - TablicaHash, [21](#)
- _L
 - StosTab, [14](#)
- _Pokaz
 - StosTab, [13](#)
 - Struktury, [15](#)
- _Pop
 - StosTab, [13](#)
 - Struktury, [15](#)
- _Push
 - StosTab, [13](#)
 - Struktury, [16](#)
- _Rozmiar
 - StosTab, [13](#)
 - Struktury, [16](#)
- _RozmiarL
 - StosTab, [14](#)
- _RozmiarT
 - StosTab, [14](#)
- _Stworz_Rekord
 - TablicaHash, [22](#)
- _Tab
 - TablicaHash, [23](#)
- _Test
 - BenchmarkInterfejs, [10](#)
 - StrukturyBenchmark, [17](#)
- _Ustaw
 - StrukturyBenchmark, [17](#)
- _Usun
 - TablicaHash, [22](#)
- _Wartosc
 - Element, [11](#)
 - StosTab, [14](#)
- _Wartosci
 - StrukturyBenchmark, [18](#)
- _Wczytaj
 - BenchmarkInterfejs, [10](#)
 - StrukturyBenchmark, [17](#)
- _Wyczysc
 - TablicaHash, [22](#)
- _WykonajTest
 - BenchmarkInterfejs, [10](#)
- _Zaladuj
 - BenchmarkInterfejs, [10](#)
 - StrukturyBenchmark, [18](#)
- _Zwolnij
 - StosTab, [14](#)
 - Struktury, [16](#)
- _Zwroc
 - TablicaHash, [22](#)
- BenchmarkInterfejs, [9](#)
 - _Generator, [9](#)
 - _IleKolizja, [10](#)
 - _Test, [10](#)
 - _Wczytaj, [10](#)
 - _WykonajTest, [10](#)
 - _Zaladuj, [10](#)
- BenchmarkInterfejs.cpp, [25](#)
- BenchmarkInterfejs.hh, [25](#)
 - DLUGOSC_SLOWA, [26](#)
 - ILOSC, [26](#)
- DLUGOSC_SLOWA
 - BenchmarkInterfejs.hh, [26](#)
- Element, [11](#)
 - _Klucz, [11](#)
 - _Wartosc, [11](#)
 - operator=, [11](#)
- Element.hh, [26](#)
- ILOSC
 - BenchmarkInterfejs.hh, [26](#)
- ILOSC_POW
 - Main.cpp, [26](#)
- IloscKolizji
 - TablicaHash, [22](#)
- K

TablicaHash, [23](#)

main

 Main.cpp, [27](#)

Main.cpp, [26](#)

 ILOSC_POW, [26](#)

 main, [27](#)

operator()

 TablicaAso, [19](#)

operator=

 Element, [11](#)

Przydziel

 StrukturyBenchmark, [18](#)

ROZMIAR

 TablicaHash.hh, [29](#)

StosTab

 ~StosTab, [12](#)

 _Adres, [13](#)

 _L, [14](#)

 _Pokaz, [13](#)

 _Pop, [13](#)

 _Push, [13](#)

 _Rozmiar, [13](#)

 _RozmiarL, [14](#)

 _RozmiarT, [14](#)

 _Wartosc, [14](#)

 _Zwolnij, [14](#)

 StosTab, [12](#)

 StosTab, [12](#)

StosTab< Typ >, [11](#)

StosTab.hh, [27](#)

strona-glowna.dox, [27](#)

Struktury

 _Pokaz, [15](#)

 _Pop, [15](#)

 _Push, [16](#)

 _Rozmiar, [16](#)

 _Zwolnij, [16](#)

Struktury< Typ >, [15](#)

Struktury.hh, [27](#)

StrukturyBenchmark, [16](#)

 _IleKolizja, [17](#)

 _Klucze, [18](#)

 _Test, [17](#)

 _Ustaw, [17](#)

 _Wartosci, [18](#)

 _Wczytaj, [17](#)

 _Zaladuj, [18](#)

 Przydziel, [18](#)

 W, [18](#)

StrukturyBenchmark.cpp, [27](#)

StrukturyBenchmark.hh, [27](#)

TablicaAso, [19](#)

 operator(), [19](#)

 Usun_wszystkie, [19](#)

TablicaAso.cpp, [28](#)

TablicaAso.hh, [28](#)

TablicaHash, [20](#)

 ~TablicaHash, [21](#)

 _Dodaj, [21](#)

 _Haszuj, [21](#)

 _Kolizja, [21](#)

 _Stworz_Rekord, [22](#)

 _Tab, [23](#)

 _Usun, [22](#)

 _Wyczysc, [22](#)

 _Zwroc, [22](#)

 IloscKolizji, [22](#)

 K, [23](#)

 TablicaHash, [21](#)

 TablicaHash, [21](#)

TablicaHash.cpp, [28](#)

TablicaHash.hh, [28](#)

 ROZMIAR, [29](#)

Usun_wszystkie

 TablicaAso, [19](#)

W

 StrukturyBenchmark, [18](#)