LAB3

0.1

Wygenerowano przez Doxygen 1.8.6

Śr, 22 kwi 2015 20:18:38

Spis treści

| 1 | Clas | s Index | | | 1 |
|---|------|----------|------------|--|---|
| | 1.1 | Class I | _ist | | 1 |
| 2 | File | Index | | | 3 |
| | 2.1 | File Lis | st | | 3 |
| 3 | Clas | s Docu | mentatior | 1 | 5 |
| | 3.1 | Bench | mark< T | > Class Template Reference | 5 |
| | | 3.1.1 | Detailed | Description | 5 |
| | | 3.1.2 | Construc | ctor & Destructor Documentation | 5 |
| | | | 3.1.2.1 | Benchmark | 5 |
| | | | 3.1.2.2 | Benchmark | 6 |
| | | | 3.1.2.3 | \sim Benchmark | 6 |
| | | 3.1.3 | Member | Function Documentation | 6 |
| | | | 3.1.3.1 | GenerujLiczbyCalkowiteLosowe | 6 |
| | | | 3.1.3.2 | GenerujLiczbyZmiennoprzecinkowe | 6 |
| | | | 3.1.3.3 | StworzLiczbyOdniesienia | 7 |
| | | | 3.1.3.4 | Testuj | 7 |
| | | | 3.1.3.5 | TransformacjaBoxa_Mullera | 8 |
| | | | 3.1.3.6 | UstalRozmiarTablicyZlozonosciObliczeniowej | 8 |
| | | | 3.1.3.7 | ZapiszWynikiZlozonosciObliczeniowej | 8 |
| | | 3.1.4 | Member | Data Documentation | 8 |
| | | | 3.1.4.1 | IloscDanych | 8 |
| | | | 3.1.4.2 | LiczbyGaussowe | 9 |
| | | | 3.1.4.3 | rozmiar | 9 |
| | | | 3.1.4.4 | ZlozonoscObliczeniowa | 9 |
| | 3.2 | Kolejka | a < T > CI | ass Template Reference | 9 |
| | | 3.2.1 | Detailed | Description | 9 |
| | | 3.2.2 | Construc | ctor & Destructor Documentation | 9 |
| | | | 3.2.2.1 | Kolejka | 9 |
| | | | 3.2.2.2 | ~Kolejka | 0 |
| | | 3.2.3 | Member | Function Documentation | 0 |

iv SPIS TREŚCI

| | | 3.2.3.1 pop_back | 10 |
|-----|---------|---|----|
| | | 3.2.3.2 push_front | 10 |
| | | 3.2.3.3 show | 10 |
| | | 3.2.3.4 size | 10 |
| 3.3 | List< 7 | > Class Template Reference | 11 |
| | 3.3.1 | Detailed Description | 11 |
| | 3.3.2 | Constructor & Destructor Documentation | 11 |
| | | 3.3.2.1 List | 11 |
| | | 3.3.2.2 ~List | 12 |
| | 3.3.3 | Member Function Documentation | 12 |
| | | 3.3.3.1 operator[] | 12 |
| | | 3.3.3.2 pop_back | 12 |
| | | 3.3.3.3 pop_front | 12 |
| | | 3.3.3.4 push | 12 |
| | | 3.3.3.5 push_back | 13 |
| | | 3.3.3.6 push_front | 14 |
| | | 3.3.3.7 show | 14 |
| | | 3.3.3.8 showOdKonca | 14 |
| | | 3.3.3.9 size | 15 |
| | 3.3.4 | Member Data Documentation | 15 |
| | | 3.3.4.1 _size | 15 |
| | | 3.3.4.2 head | 15 |
| | | 3.3.4.3 tail | 15 |
| 3.4 | Node< | T $>$ Struct Template Reference | 15 |
| | 3.4.1 | Detailed Description | 15 |
| | 3.4.2 | Member Data Documentation | 15 |
| | | 3.4.2.1 next | 15 |
| | | 3.4.2.2 val | 16 |
| 3.5 | NodeL | $<$ T $>$ Struct Template Reference \dots 1 | 16 |
| | 3.5.1 | Detailed Description | 16 |
| | 3.5.2 | Member Data Documentation | 16 |
| | | 3.5.2.1 next | 16 |
| | | 3.5.2.2 prev | 16 |
| | | 3.5.2.3 val | 16 |
| 3.6 | Stack< | T > Class Template Reference | 16 |
| | 3.6.1 | Detailed Description | 17 |
| | 3.6.2 | Constructor & Destructor Documentation | 17 |
| | | 3.6.2.1 Stack | 17 |
| | | 3.6.2.2 ~Stack | 17 |
| | 3.6.3 | Member Function Documentation | 17 |

SPIS TREŚCI

| | | 3.6.3.1 | operator[] | 17 |
|------|--|---|---|--|
| | | 3.6.3.2 | peek | 17 |
| | | 3.6.3.3 | pop | 18 |
| | | 3.6.3.4 | push | 18 |
| | | 3.6.3.5 | size | 18 |
| | 3.6.4 | Member | Data Documentation | 18 |
| | | 3.6.4.1 | capacity | 18 |
| | | 3.6.4.2 | storage | 18 |
| | | 3.6.4.3 | top | 19 |
| File | Docume | entation | | 21 |
| | | | h File Reference | 21 |
| | | | | 22 |
| | | | · | 22 |
| | | | | 22 |
| | | 4.1.2.2 | · | 22 |
| 4.2 | inc/Kol | ejka.hh Fil | · | 23 |
| | 4.2.1 | • | | 23 |
| 4.3 | inc/List | | | 23 |
| | 4.3.1 | Detailed | Description | 24 |
| 4.4 | inc/Sta | ck.hh File | Reference | 24 |
| | 4.4.1 | Detailed | Description | 25 |
| | 4.4.2 | Function | Documentation | 26 |
| | | 4.4.2.1 | operator<< | 26 |
| 4.5 | inc/Str | uktury.hh F | File Reference | 26 |
| 4.6 | inc/Zap | oiszStosKo | olejkaLista.hh File Reference | 27 |
| | 4.6.1 | Function | Documentation | 28 |
| | | 4.6.1.1 | WczytajListe | 28 |
| | | 4.6.1.2 | ZapiszKolejnoLiczbyKolejki | 28 |
| | | 4.6.1.3 | ZapiszKolejnoLiczbyListy | 29 |
| | | 4.6.1.4 | ZapiszKolejnoLiczbyStosu | 30 |
| 4.7 | src/ma | in.cpp File | Reference | 30 |
| | 4.7.1 | Function | Documentation | 30 |
| | | 4.7.1.1 | main | 31 |
| 4.8 | src/Zap | oiszStosKo | olejkaLista.cpp File Reference | 31 |
| | 4.8.1 | Function | Documentation | 32 |
| | | 4.8.1.1 | WczytajListe | 32 |
| | | 4.8.1.2 | ZapiszKolejnoLiczbyKolejki | 32 |
| | | 4.8.1.3 | ZapiszKolejnoLiczbyListy | 32 |
| | | 4.8.1.4 | ZapiszKolejnoLiczbyStosu | 33 |
| | 4.1 4.2 4.3 4.4 4.5 4.6 | File Docume 4.1 inc/Ber 4.1.1 4.1.2 4.2 inc/Kol 4.2.1 4.3 inc/List 4.3.1 4.4 inc/Sta 4.4.1 4.4.2 4.5 inc/Str 4.6 inc/Zap 4.6.1 4.7 src/ma 4.7.1 4.8 src/Zap | 3.6.3.2 3.6.3.3 3.6.3.4 3.6.3.5 3.6.4.1 3.6.4.2 3.6.4.3 File Documentation 4.1 inc/Benchmark.h 4.1.1 Detailed 4.1.2 Function 4.1.2.1 4.1.2.2 4.2 inc/Kolejka.hh File 4.2.1 Detailed 4.3 inc/Lista.hh File file 4.4.1 Detailed 4.4.2 Function 4.4.2.1 4.5 inc/Stack.hh File 4.4.1 Detailed 4.4.2 Function 4.4.2.1 4.5 inc/Struktury.hh File 4.6.1.1 4.6.1.2 4.6.1.3 4.6.1.4 4.7 src/main.cpp File 4.7.1 Function 4.7.1.1 4.8 src/ZapiszStosko 4.8.1 Function 4.8.1.1 4.8.1.2 4.8.1.3 | 3.6.3.2 peek 3.6.3.3 pop. 3.6.3.4 push 3.6.3.5 size 3.6.4 Member Data Documentation. 3.6.4.1 capacity 3.6.4.2 storage 3.6.4.3 top. File Documentation 4.1 inc/Benchmark.hh File Reference 4.1.1 Detailed Description 4.1.2 Function Documentation 4.1.2.1 operator<< 4.1.2 operator>> 4.2 inc/Kolejka.hh File Reference 4.3.1 Detailed Description 4.3 inc/List.hh File Reference 4.3.1 Detailed Description 4.4 inc/Stack.hh File Reference 4.4.1 Detailed Description 4.5 inc/List.hh File Reference 4.6 inc/ZapiszStoskolejka.list.hh File Reference 4.6 inc/ZapiszStoskolejka.list.hh File Reference 4.6.1 Function Documentation 4.6.1.1 Wczytaji.liste 4.6.1.2 ZapiszKolejnoLiczbyKolejki 4.6.1.3 ZapiszKolejnoLiczbyStosu 4.7 src/main.cpp File Reference 4.7.1 Function Documentation 4.7.1 Function Documentation 4.8.1.1 Wczytaji.liste 4.8.1.2 ZapiszKolejnoLiczbyStosu 4.8 src/ZapiszStoskolejka.lista.cpp File Reference 4.8.1 Function Documentation 4.8.1.1 Wczytaji.liste 4.8.1.2 Tunction Documentation 4.8.1.1 Wczytaji.liste 4.8.1.2 ZapiszKolejnoLiczbyKolejki 4.8.1.3 ZapiszKolejnoLiczbyKolejki 4.8.1.3 ZapiszKolejnoLiczbyKolejki 4.8.1.3 ZapiszKolejnoLiczbyKolejki 4.8.1.3 ZapiszKolejnoLiczbyKolejki 4.8.1.3 ZapiszKolejnoLiczbyKolejki 4.8.1.3 ZapiszKolejnoLiczbyKolejki |

Rozdział 1

Indeks klas

1.1 Lista klas

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

| enchmark < T > |
|--|
| olejka< T > |
| Klasa Kolejka (str. 9) sluzy do wykonywania podstawowych operacji na Kolejce: dodaj,odejmij element. Przechowuje informacje o ilosci wszysktich elementow |
| st <t></t> |
| Klasa List (str. 11) sluzy do wykonywania podstawowych operacji na Liscie: dodaj,odejmij element. Przechowuje informacje o ilosci wszysktich elementow |
| ode <t></t> |
| odeL <t></t> |
| ack< T > |

2 Indeks klas

Rozdział 2

Indeks plików

2.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

| Benchmark.hh | |
|---|----|
| Klasa Benchmark (str. 5) sluzy do przechowywania wynikow zlozonosci obliczeniowej i danych wejsciowych,generowania liczb rozkladu Gaussowego | 21 |
| Kolejka.hh | |
| Struktura przechowujaca wartosc wezla i wskaznik na nastepny element typu Node (str. 15) | 23 |
| Lista.hh | |
| Struktura przechowujaca wartosc wezla i wskaznik na nastepny element typu Node (str. 15) | 23 |
| main.cpp | 30 |
| Stack.hh | |
| Klasa Stack (str. 16) sluzy do przechowywania, dodawania, zdejmowania kolejnych elementow | |
| stosu | 24 |
| Struktury.hh | 26 |
| ZapiszStosKolejkaLista.cpp | 31 |
| | 27 |

Indeks plików

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja szablonu klasy Benchmark< T >

#include <Benchmark.hh>

Metody publiczne

- · Benchmark ()
- Benchmark (int roz, unsigned long int max, T *Struktura=nullptr)
- ∼Benchmark ()
- void UstalRozmiarTablicyZlozonosciObliczeniowej (int roz)
- std::ostream & GenerujLiczbyZmiennoprzecinkowe (long int rozmiar, std::ostream &strum)
- std::ostream & GenerujLiczbyCalkowiteLosowe (long int rozmiar, std::ostream &strum)
- void TransformacjaBoxa_Mullera (float *a)
- void StworzLiczbyOdniesienia (int p[], int ile)
- void Testuj (int MaxIloscDanych, void(*wsk_fun)(T *, int))
- std::ostream & ZapiszWynikiZlozonosciObliczeniowej (std::ostream &Strm)

Atrybuty publiczne

- unsigned long int ** ZlozonoscObliczeniowa
- T * LiczbyGaussowe
- · unsigned long int IloscDanych
- int rozmiar

3.1.1 Opis szczegółowy

template<typename T>class Benchmark< T>

Definicja w linii 16 pliku Benchmark.hh.

3.1.2 Dokumentacja konstruktora i destruktora

3.1.2.1 template<typename T > Benchmark< T >::Benchmark()

brief Konstruktor bezparametryczny

Konstruktor bezparametryczny klasy **Benchmark** (str. 5) Ustawia wszystkie zmienne i wskazniki na wartosc 0 Definicja w linii 97 pliku Benchmark.hh.

6 Dokumentacja klas

3.1.2.2 template<typename T > Benchmark < T >::Benchmark (int roz, unsigned long int max, T * Struktura = nullptr)

brief Konstruktor z 2 argumentami int, unsigned long int

Konstruktor parametryczny klasy Benchmark (str. 5)

Parametry

| in | roz | - typ int, rozmiar tablicy ZlozonoscObliczeniowa, |
|----|------|--|
| in | max- | unsigned long int, ilosc liczb zmiennoprzecinkowych tablicy LiczbyGaussowe |

Definicja w linii 82 pliku Benchmark.hh.

3.1.2.3 template<typename T > Benchmark<T>::~Benchmark()

brief destruktor klasy Benchmark (str. 5)

Definicja w linii 106 pliku Benchmark.hh.

- 3.1.3 Dokumentacja funkcji składowych
- 3.1.3.1 template<typename T > std::ostream & Benchmark< T >::GenerujLiczbyCalkowiteLosowe (long int *rozmiar*, std::ostream & *strum*)

brief Funkcja generuje liczby typu double z rozkladu Gaussa

Funkcja generuje liczby typu int o rozkladzie Gaussa i zapisuje do strumienia

Parametry

| in | rozmiar | - long int, ilosc wygenerowanych elementow |
|----|---------|--|
| in | &strum | - referencja do strumienia wyjsciowego |

Zwraca

Zwraca referencje do strumienia wyjsciowego

Warunek wstępny

Prawidlowe dzialanie funkcji TransformacjaBoxa_Mullera

Definicja w linii 204 pliku Benchmark.hh.

3.1.3.2 template<typename T > std::ostream & Benchmark< T >::GenerujLiczbyZmiennoprzecinkowe (long int *rozmiar*, std::ostream & *strum*)

brief Funkcja generuje liczby typu double z rozkladu Gaussa

Funkcja generuje liczby typu float o rozkladzie Gaussa i zapisuje do strumienia

Parametry

| in | rozmiar | - long int, ilosc wygenerowanych elementow |
|----|---------|--|
| in | &strum | - referencja do strumienia wyjsciowego |

Zwraca

Zwraca referencje do strumienia wyjsciowego

Warunek wstępny

Prawidlowe dzialanie funkcji TransformacjaBoxa Mullera

Definicja w linii 175 pliku Benchmark.hh.

3.1.3.3 template < typename T > void Benchmark < T >::StworzLiczbyOdniesienia (int p[], int ile)

brief Funkcja tworzy wygodne liczby odniesienia dla danej funkcji testowej o zadanym maximum

Funkcja tworzy liczby odniesienia, by lepiej dobrac ilosci danych do testu

Parametry

| in | p | - int, w tablicy ustalane sa argumenty liczb odniesienia |
|----|-------------|--|
| in | ile-int,jak | duzo ma zostac stworzonych liczb odniesienia |

Warunek wstępny

Ilosc liczb odniesienia musi byc podzielna przez 4

Definicja w linii 293 pliku Benchmark.hh.

3.1.3.4 template<typename T > void Benchmark< T >::Testuj (int MaxlloscDanych, void(*)(T *, int) wsk_fun)

brief Funkcja mierzy czas trwania funkcji dla okreslonej ilosci danych

Funkcja sluzy do badania zlozonosci obliczeniowej danej funkcji

Parametry

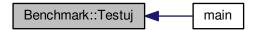
| in | MaxlloscDanych | - int, maksymalna ilosc danych do testu |
|----|----------------|---|
| in | wsk_fun= | wskaznik na funkcje testowana o arg:double*,int |

Warunek wstępny

Funkcja w argumencie musi być typu(TypSzablonuBenchmark,maxLiczbaElementow) llosc cykli ustawia sie w konstruktorze Benchmarku,maxElem=75 000 000

Definicja w linii 247 pliku Benchmark.hh.

Oto graf wywoływań tej funkcji:



3.1.3.5 template < typename T > void Benchmark < T >::TransformacjaBoxa_Mullera (float * a)

brief Funkcja zamienia liczby float z rozkladu rownomiernego na Gaussowy

Funkcja zamienia liczby z rozkladu rownomiernego na rozklad Gaussa

8 Dokumentacja klas

Parametry

| in | a[] | - typ float,wskaznik na tablice 2-elementowa |
|----|-----|--|
|----|-----|--|

Warunek wstępny

2 liczby z rozkladu rownomiernego

Definicja w linii 231 pliku Benchmark.hh.

3.1.3.6 template<typename T > void Benchmark< T >::UstalRozmiarTablicyZlozonosciObliczeniowej (int roz)

brief Ustalenie rozmiaru tablicy zlozonosci obliczeniowej

Funkcja ustala na nowo rozmiar tablicy ZlozonoscObliczeniowa

Parametry

| in | roz | - typ int, rozmiar tablicy ZlozonoscObliczeniowa |
|----|-----|--|

Warunek wstępny

zmienna roz musi byc dodatnia

Definicja w linii 119 pliku Benchmark.hh.

3.1.3.7 template<typename T > std::ostream & Benchmark< T >::ZapiszWynikiZlozonosciObliczeniowej (std::ostream & Strm)

Funkcja sluzy do zapisania wynikow z tablicy ZlozonoscObliczeniowa

Parametry

| in | &Strum | - referencja do strumienia wyjsciowego |
|----|--------|--|
| in | roz | - ilosc elementow w tablicy ZlozonoscObliczniowa przez 2 |

Zwraca

Zwraca referencje do strumienia wyjsciowego

Warunek wstępny

Poprawnie wczytane wartości tablicy ZlozonoścObliczeniowa

Definicja w linii 313 pliku Benchmark.hh.

3.1.4 Dokumentacja atrybutów składowych

3.1.4.1 template<typename T> unsigned long int Benchmark< T>::lloscDanych

brief ilosc liczb z rozkladu Gaussa

Definicja w linii 30 pliku Benchmark.hh.

3.1.4.2 template < typename T> T* Benchmark < T>::LiczbyGaussowe

brief wskaznik na tablice przechowujaca elementy z rozkladu gaussa

Definicja w linii 26 pliku Benchmark.hh.

3.1.4.3 template<typename T> int Benchmark< T>::rozmiar

brief ilosc liczb zlozonosci obliczeniowej

Definicja w linii 34 pliku Benchmark.hh.

3.1.4.4 template<typename T> unsigned long int** Benchmark<T>::ZlozonoscObliczeniowa

brief wskaznik na tablice przechowująca wartości zlozoności obliczeniowej(ilość danych,czas)

Definicja w linii 22 pliku Benchmark.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

· Benchmark.hh

3.2 Dokumentacja szablonu klasy Kolejka < T >

klasa **Kolejka** (str. 9) sluzy do wykonywania podstawowych operacji na Kolejce: dodaj,odejmij element. Przechowuje informacje o ilosci wszysktich elementow.

```
#include <Kolejka.hh>
```

Metody publiczne

- · Kolejka ()
- ∼Kolejka ()
- int size ()
- void push_front (T value)
- void pop_back ()
- void show ()

Atrybuty prywatne

- Node< T > * head
- int _size

3.2.1 Opis szczegółowy

template<typename T>class Kolejka<T>

Definicja w linii 21 pliku Kolejka.hh.

3.2.2 Dokumentacja konstruktora i destruktora

3.2.2.1 template<typename T > Kolejka < T >::Kolejka ()

brief Konstruktor bezparametryczny

Konstruktor bezparametryczny, ustawia parametry na 0

Definicja w linii 59 pliku Kolejka.hh.

10 Dokumentacja klas

3.2.2.2 template<typename T > Kolejka< T >:: \sim Kolejka ()

Destruktor, usuwa kolejne elementy kolejki zaczynajac od poczatku

Definicja w linii 69 pliku Kolejka.hh.

3.2.3 Dokumentacja funkcji składowych

3.2.3.1 template<typename T > void Kolejka< T >::pop_back()

brief Funkcja zdejmuje element z konca kolejki

Funkcja usuwa element z konca kolejki

Warunek wstępny

Kolejka (str. 9) nie moze byc pusta

Definicja w linii 106 pliku Kolejka.hh.

3.2.3.2 template<typename T > void Kolejka< T >::push_front (T value)

brief Funkcja dodaje element na poczatek kolejki

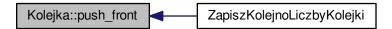
Funkcja sluzy do dodania elementu na poczatek kolejki

Parametry

| in | value-typ | int, wartosc elementu zmiennej dodanej do kolejki |
|----|-----------|---|
|----|-----------|---|

Definicja w linii 92 pliku Kolejka.hh.

Oto graf wywoływań tej funkcji:



3.2.3.3 template<typename T > void Kolejka< T >::show ()

brief Funkcja wyswietla wszystkie elementy na standardowe wyjscie

Funkcja wyswietla elementy kolejki

Definicja w linii 134 pliku Kolejka.hh.

3.2.3.4 template<typename T > int Kolejka< T >::size ()

brief Funkcja zwraca rozmiar kolejki

Zwraca

Funkcja zwraca wartosc rozmiaru kolejki

Definicja w linii 82 pliku Kolejka.hh.

3.2.4 Dokumentacja atrybutów składowych

```
3.2.4.1 template<typename T> int Kolejka<T>::_size [private]
```

brief Informacja o rozmiarze kolejki

Definicja w linii 30 pliku Kolejka.hh.

```
3.2.4.2 template<typename T> Node<T>* Kolejka<T>::head [private]
```

brief wskaznik do ktorego doczepione sa kolejne elementy

Definicja w linii 26 pliku Kolejka.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

· Kolejka.hh

3.3 Dokumentacja szablonu klasy List< T >

klasa **List** (str. 11) sluzy do wykonywania podstawowych operacji na Liscie: dodaj,odejmij element. Przechowuje informacje o ilosci wszysktich elementow.

```
#include <Lista.hh>
```

Metody publiczne

- · List ()
- \sim List ()
- int size ()
- void push_front (T value)
- void pop_front ()
- void push_back (T value)
- void pop_back ()
- void show ()
- void showOdKonca ()
- void **push** (T value, int nr=0)
- T & operator[] (int a)

Atrybuty publiczne

- NodeL< T>* head
- NodeL< T > * tail
- int size

3.3.1 Opis szczegółowy

template<typename T>class List< T>

Definicja w linii 25 pliku Lista.hh.

12 Dokumentacja klas

3.3.2 Dokumentacja konstruktora i destruktora

3.3.2.1 template<typename T > List< T >::List ()

brief Konstruktor bezparametryczny

Konstruktor bezparametryczny, ustawia parametry na 0

Definicja w linii 101 pliku Lista.hh.

3.3.2.2 template < typename T > List < T >:: \sim List ()

brief Destruktor

Destruktor, usuwa kolejne elementy listy zaczynajac od poczatku

Definicja w linii 112 pliku Lista.hh.

3.3.3 Dokumentacja funkcji składowych

3.3.3.1 template<typename T> T& List< T>::operator[](int a) [inline]

Przeciazony operator indeksowania zwraca referencje do elementu o indeksie a

Definicja w linii 87 pliku Lista.hh.

3.3.3.2 template<typename T > void List< T >::pop_back()

brief Funkcja zdejmuje element z konca listy

Funkcja usuwa element z konca listy

Warunek wstępny

Lista nie moze byc pusta

Definicja w linii 229 pliku Lista.hh.

3.3.3.3 template<typename T > void List< T >::pop_front ()

brief Funkcja zdejmuje element z poczatku listy

Funkcja usuwa element z poczatku listy

Warunek wstępny

Lista nie moze byc pusta

Definicja w linii 152 pliku Lista.hh.

3.3.3.4 template < typename T > void List < T >::push (T value, int nr = 0)

brief Funkcja dodaje element przed elementem o indeksie nr

Funkcja dodaje element przed elementem o indeksie nr

Parametry

| in | value-wybrany | typ, wartosc elementu dodanego do listy |
|----|---------------|--|
| in | nr- | indeks elementu przed ktorym ma byc dodany element |

Warunek wstępny

indeksowanie od 0

Definicja w linii 196 pliku Lista.hh.

3.3.3.5 template<typename T > void List< T >::push_back (T value)

brief Funkcja dodaje element na koniec listy

Funkcja dodaje element na koniec listy

Parametry

| in | valuo | - typ int. wartosc elementu dodanego na koniec listy |
|-----|-------|--|
| T11 | vaiue | - typ int, wartosc elementu dodanego na koniec listy |

Definicja w linii 171 pliku Lista.hh.

Oto graf wywoływań tej funkcji:



3.3.3.6 template<typename T > void List< T >::push_front (T value)

brief Funkcja dodaje element na poczatek listy

Funkcja sluzy do dodania elementu na poczatek listy

Parametry

| | | in | value-typ | int, wartosc elementu zmiennej dodanej do listy |
|--|--|----|-----------|---|
|--|--|----|-----------|---|

Definicja w linii 135 pliku Lista.hh.

Oto graf wywoływań tej funkcji:



14 Dokumentacja klas

3.3.3.7 template<typename T > void List< T >::show ()

brief Funkcja wyswietla wszystkie elementy na standardowe wyjscie

brief Funkcja pokazujaca na strumieniu std::cout zawartosc listy

Funkcja wyswietla elementy listy

Definicja w linii 259 pliku Lista.hh.

3.3.3.8 template<typename T > void List< T >::showOdKonca ()

brief Funkcja pokazujaca na strumieniu std::cout zawartosc listy od konca

Funkcja wyswietla elementy listy od konca

Definicja w linii 273 pliku Lista.hh.

3.3.3.9 template<typename T > int List< T >::size ()

brief Funkcja zwraca rozmiar listy

Zwraca

Funkcja zwraca wartosc rozmiaru listy

Definicja w linii 125 pliku Lista.hh.

3.3.4 Dokumentacja atrybutów składowych

3.3.4.1 template<typename T> int List< T>::_size

brief Informacja o rozmiarze listy

Definicja w linii 39 pliku Lista.hh.

3.3.4.2 template<typename T> NodeL<T>* List< T>::head

brief wskaznik do ktorego doczepione sa kolejne elementy listy

Definicja w linii 31 pliku Lista.hh.

3.3.4.3 template<typename T> NodeL<T>* List< T>::tail

brief wskaznik pokazujacy na koniec listy

Definicja w linii 35 pliku Lista.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

· Lista.hh

3.4 Dokumentacja szablonu struktury Node < T >

#include <Kolejka.hh>

Atrybuty publiczne

- T val
- Node< T > * next

3.4.1 Opis szczegółowy

template<typename T>struct Node< T>

Definicja w linii 10 pliku Kolejka.hh.

3.4.2 Dokumentacja atrybutów składowych

3.4.2.1 template<typename T> Node<T>* Node< T>::next

Definicja w linii 13 pliku Kolejka.hh.

3.4.2.2 template < typename T > T Node < T >::val

Definicja w linii 12 pliku Kolejka.hh.

Dokumentacja dla tej struktury została wygenerowana z pliku:

· Kolejka.hh

3.5 Dokumentacja szablonu struktury NodeL< T>

#include <Lista.hh>

Atrybuty publiczne

- T val
- NodeL< T> * next
- NodeL< T > * prev

3.5.1 Opis szczegółowy

template < typename T > struct NodeL < T >

Definicja w linii 12 pliku Lista.hh.

3.5.2 Dokumentacja atrybutów składowych

3.5.2.1 template<typename T> NodeL<T>* NodeL< T>::next

Definicja w linii 15 pliku Lista.hh.

3.5.2.2 template<typename T> NodeL<T>* NodeL< T>::prev

Definicja w linii 16 pliku Lista.hh.

16 Dokumentacja klas

3.5.2.3 template<typename T> T NodeL< T>::val

Definicja w linii 14 pliku Lista.hh.

Dokumentacja dla tej struktury została wygenerowana z pliku:

· Lista.hh

3.6 Dokumentacja szablonu klasy Stack< T >

```
#include <Stack.hh>
```

Metody publiczne

- Stack (int capacity=10)
- void push (T value)
- T peek ()
- int size ()
- \sim Stack ()
- void pop ()
- T & operator[] (int a)

Atrybuty publiczne

- T * top
- · int capacity
- T * storage

3.6.1 Opis szczegółowy

template<typename T>class Stack< T>

Definicja w linii 11 pliku Stack.hh.

3.6.2 Dokumentacja konstruktora i destruktora

```
3.6.2.1 template < typename T > Stack < T >::Stack ( int capacity = 10 )
```

Konstruktor parametryczny klasy Stack (str. 16)

Parametry

| | in | capacity | - typ int, rozmiar stosu |
|--|----|----------|--------------------------|
|--|----|----------|--------------------------|

Definicja w linii 66 pliku Stack.hh.

3.6.2.2 template<typename T > Stack< T >:: \sim Stack ()

Destruktor klasy Stack (str. 16)

Definicja w linii 125 pliku Stack.hh.

3.6.3 Dokumentacja funkcji składowych

3.6.3.1 template<typename T> T& Stack< T>::operator[](int a) [inline]

brief Przeciazony operator indeksowania, umozliwia traktowanie stosu jak tablicy Definicja w linii 54 pliku Stack.hh.

3.6.3.2 template<typename T > T Stack< T >::peek ()

Funkcja pokazuje element znajdujacy sie na szczycie stosu

Warunek wstępny

Stos nie moze byc pusty

Definicja w linii 104 pliku Stack.hh.

3.6.3.3 template<typename T > void Stack< T >::pop ()

Funkcja pop zdejmuje ostatni element ze stosu

Warunek wstępny

Stos nie moze byc pusty

Definicja w linii 136 pliku Stack.hh.

3.6.3.4 template<typename T > void Stack< T >::push (T value)

Funkcja dodaje element na koniec tablicy stosu

Parametry

| in | value | - typ int, wartosc dodana do stosu |
|----|-------|------------------------------------|

Warunek końcowy

wykorzystana metoda podwajania do powiekszania stosu

Definicja w linii 83 pliku Stack.hh.

Oto graf wywoływań tej funkcji:



18 Dokumentacja klas

3.6.3.5 template<typename T > int Stack< T >::size ()

Funkcja pokazuje ilosc elementow stosu

Zwraca

zwraca ilosc elementow stosu

Definicja w linii 116 pliku Stack.hh.

3.6.4 Dokumentacja atrybutów składowych

3.6.4.1 template < typename T> int Stack < T>::capacity

Definicja w linii 21 pliku Stack.hh.

3.6.4.2 template<typename T> T* Stack< T>::storage

Definicja w linii 25 pliku Stack.hh.

3.6.4.3 template<typename T> T* Stack< T>::top

Definicja w linii 17 pliku Stack.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

· Stack.hh

Rozdział 4

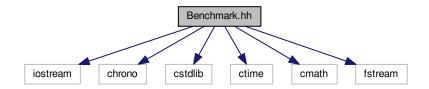
Dokumentacja plików

4.1 Dokumentacja pliku Benchmark.hh

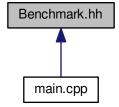
Klasa **Benchmark** (str. 5) sluzy do przechowywania wynikow zlozoności obliczeniowej i danych wejsciowych, generowania liczb rozkladu Gaussowego.

```
#include <iostream>
#include <chrono>
#include <cstdlib>
#include <ctime>
#include <cmath>
#include <fstream>
```

Wykres zależności załączania dla Benchmark.hh:



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



20 Dokumentacja plików

Komponenty

class Benchmark< T >

Funkcje

template<typename T >
 std::ostream & operator<< (std::ostream &Strm, const Benchmark< T > &ben)

template<typename T >
 std::istream & operator>> (std::istream &Strm, Benchmark< T > &ben)

4.1.1 Dokumentacja funkcji

4.1.1.1 template < typename T > std::ostream & operator << (std::ostream & Strm, const Benchmark < T > & ben)

Funkcja operatorowa pozwala na wypisanie wszystkich liczb tablicy LiczbyGaussowe

Parametry

| in,out | &Strm | - referencja do strumienia wyjsciowego |
|--------|-------|--|
| in,out | &ben | refencja do klasy typu Benchmark (str. 5) |

Zwraca

Zwraca referencje do strumienia wyjsciowego

Warunek wstępny

Poprawne wczytanie liczb tablicy LiczbyGaussowe

Definicja w linii 140 pliku Benchmark.hh.

4.1.1.2 template < typename T > std::istream & operator >> (std::istream & Strm, Benchmark < T > & ben)

Funkcja operatorowa pozwala na wczytanie liczby typu double do tablicy LiczbyGaussowe

Parametry

| in,out | &Strm | - referencja do strumienia wejsciowego |
|--------|-------|--|
| in,out | &ben | referencja do klasy typu Benchmark (str. 5) |

Zwraca

Zwraca referencje do strumienia wejsciowego

Warunek wstępny

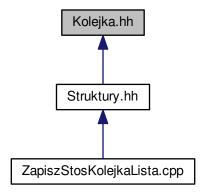
Liczba tylko typu double

Definicja w linii 158 pliku Benchmark.hh.

4.2 Dokumentacja pliku Kolejka.hh

Struktura przechowujaca wartosc wezla i wskaznik na nastepny element typu Node (str. 15).

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



Komponenty

- struct Node< T >
- class Kolejka< T >

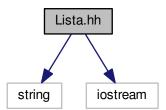
klasa **Kolejka** (str. 9) sluzy do wykonywania podstawowych operacji na Kolejce: dodaj, odejmij element. Przechowuje informacje o ilosci wszysktich elementow.

4.3 Dokumentacja pliku Lista.hh

Struktura przechowujaca wartosc wezla i wskaznik na nastepny element typu **Node** (str. 15).

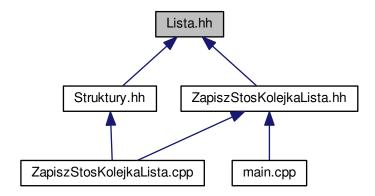
#include <string>
#include <iostream>

Wykres zależności załączania dla Lista.hh:



22 Dokumentacja plików

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



Komponenty

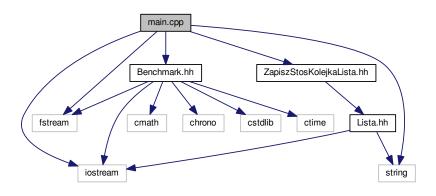
- struct NodeL< T >
- class List< T >

klasa **List** (str. 11) sluzy do wykonywania podstawowych operacji na Liscie: dodaj,odejmij element. Przechowuje informacje o ilosci wszysktich elementow.

4.4 Dokumentacja pliku main.cpp

```
#include <iostream>
#include <fstream>
#include <string>
#include "Benchmark.hh"
#include "ZapiszStosKolejkaLista.hh"
```

Wykres zależności załączania dla main.cpp:



Funkcje

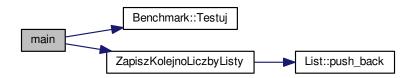
• int main ()

4.4.1 Dokumentacja funkcji

4.4.1.1 int main ()

Definicja w linii 9 pliku main.cpp.

Oto graf wywołań dla tej funkcji:

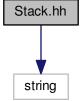


4.5 Dokumentacja pliku Stack.hh

Klasa Stack (str. 16) sluzy do przechowywania, dodawania, zdejmowania kolejnych elementow stosu.

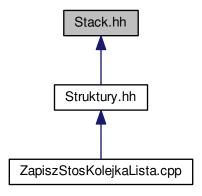
#include <string>

Wykres zależności załączania dla Stack.hh:



24 Dokumentacja plików

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



Komponenty

class Stack< T >

Funkcje

template<typename T >
 std::ostream & operator<< (std::ostream &out, const Stack< T > &stack)

4.5.1 Dokumentacja funkcji

4.5.1.1 template < typename T > std::ostream & operator << (std::ostream & out, const Stack < T > & stack)

Funkcja operatorowa sluzy do wyswietlania stosu,zbedna

Parametry

| in | &out | - referencja do strumienia wyjsciowego |
|----|---------|--|
| in | &stack- | referencja do stosu |

Zwraca

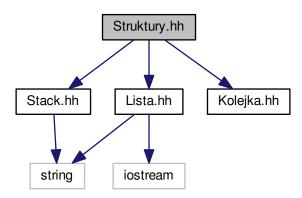
zwraca referencje do strumienia wyjsciowego

Definicja w linii 151 pliku Stack.hh.

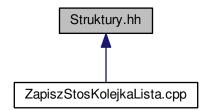
4.6 Dokumentacja pliku Struktury.hh

```
#include "Stack.hh"
#include "Lista.hh"
#include "Kolejka.hh"
```

Wykres zależności załączania dla Struktury.hh:



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

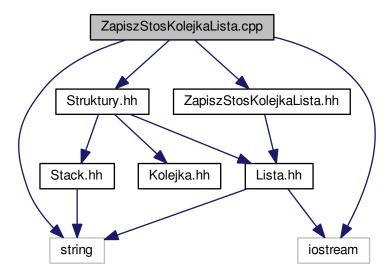


4.7 Dokumentacja pliku ZapiszStosKolejkaLista.cpp

```
#include <string>
#include <iostream>
#include "Struktury.hh"
#include "ZapiszStosKolejkaLista.hh"
```

26 Dokumentacja plików

Wykres zależności załączania dla ZapiszStosKolejkaLista.cpp:



Funkcje

- void ZapiszKolejnoLiczbyStosu (double *Gausowe, int size)
- void ZapiszKolejnoLiczbyListy (double *Gausowe, int size)
- void ZapiszKolejnoLiczbyKolejki (double *Gausowe, int size)
- void **WczytajListe** (std::istream &Strm, unsigned long int lloscDanych, **List**< int > *dane)

4.7.1 Dokumentacja funkcji

4.7.1.1 void WczytajListe (std::istream & Strm, unsigned long int IloscDanych, List < int > * dane)

Funkcja sluzy do Wczytania Listy o elementach typu int

Parametry

|] | Strm - referencja do strumienia wejsciowego |
|---|---|
|] | IloscDanych - typ int, oznacza jak wiele danych ma byc wczytane |
|] | dane - typ List <int>*, lista gdzie beda wczytane dane</int> |

Definicja w linii 50 pliku ZapiszStosKolejkaLista.cpp.

Oto graf wywołań dla tej funkcji:



4.7.1.2 void ZapiszKolejnoLiczbyKolejki (double * Gausowe, int size)

Funkcja sluzy do zapisywania danych na Kolejce utworzonej w tej funkcji

Parametry

|] | Gausowe - typ double *, wskaznik na tablice typu double |
|---|---|
|] | size - rozmiar, jak wiele liczb ma byc zapisane |

Definicja w linii 37 pliku ZapiszStosKolejkaLista.cpp.

Oto graf wywołań dla tej funkcji:



4.7.1.3 void ZapiszKolejnoLiczbyListy (double * Gausowe, int size)

Funkcja sluzy do zapisywania danych na liscie utworzonej w tej funkcji

Parametry

| J | Gausowe - typ double *, wskaznik na tablice typu double |
|---|---|
|] | size - rozmiar, jak wiele liczb ma byc zapisane |

Definicja w linii 25 pliku ZapiszStosKolejkaLista.cpp.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



28 Dokumentacja plików

4.7.1.4 void ZapiszKolejnoLiczbyStosu (double * Gausowe, int size)

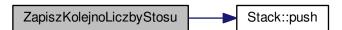
Funkcja sluzy do zapisywania danych na stosie utworzonym w tej funkcji

Parametry

|] | Gausowe - typ double *, wskaznik na tablice typu double |
|---|---|
| J | size - rozmiar, jak wiele liczb ma byc zapisane |

Definicja w linii 13 pliku ZapiszStosKolejkaLista.cpp.

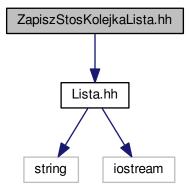
Oto graf wywołań dla tej funkcji:



4.8 Dokumentacja pliku ZapiszStosKolejkaLista.hh

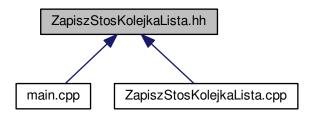
#include "Lista.hh"

Wykres zależności załączania dla ZapiszStosKolejkaLista.hh:



30 Dokumentacja plików

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



Funkcje

- void ZapiszKolejnoLiczbyStosu (double *Gausowe, int size)
- void ZapiszKolejnoLiczbyListy (double *Gausowe, int size)
- void ZapiszKolejnoLiczbyKolejki (double *Gausowe, int size)
- void WczytajListe (std::istream &Strm, unsigned long int lloscDanych, List< int > *dane)

4.8.1 Dokumentacja funkcji

4.8.1.1 void WczytajListe (std::istream & Strm, unsigned long int IloscDanych, List < int > * dane)

brief Funkcja sluzy do Wczytania Listy o elementach typu int

Funkcja sluzy do Wczytania Listy o elementach typu int

Parametry

|] | Strm - referencja do strumienia wejsciowego |
|---|---|
| J | IloscDanych - typ int, oznacza jak wiele danych ma byc wczytane |
| J | dane - typ List <int>*, lista gdzie beda wczytane dane</int> |

Definicja w linii 50 pliku ZapiszStosKolejkaLista.cpp.

Oto graf wywołań dla tej funkcji:



4.8.1.2 void ZapiszKolejnoLiczbyKolejki (double * Gausowe, int size)

brief Funkcja sluzy do zapisywania danych na Kolejce utworzonej w tej funkcji Funkcja sluzy do zapisywania danych na Kolejce utworzonej w tej funkcji

Parametry

|] | Gausowe - typ double *, wskaznik na tablice typu double |
|---|---|
| J | size - rozmiar, jak wiele liczb ma byc zapisane |

Definicja w linii 37 pliku ZapiszStosKolejkaLista.cpp.

Oto graf wywołań dla tej funkcji:



4.8.1.3 void ZapiszKolejnoLiczbyListy (double * Gausowe, int size)

brief Funkcja sluzy do zapisywania danych na liscie utworzonej w tej funkcji

Funkcja sluzy do zapisywania danych na liscie utworzonej w tej funkcji

Parametry

| J | Gausowe - typ double *, wskaznik na tablice typu double |
|---|---|
| J | size - rozmiar, jak wiele liczb ma byc zapisane |

Definicja w linii 25 pliku ZapiszStosKolejkaLista.cpp.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



32 Dokumentacja plików

4.8.1.4 void ZapiszKolejnoLiczbyStosu (double * Gausowe, int size)

brief Funkcja sluzy do zapisywania danych na stosie utworzonym w tej funkcji

Funkcja sluzy do zapisywania danych na stosie utworzonym w tej funkcji

Parametry

|] | Gausowe - typ double *, wskaznik na tablice typu double |
|---|---|
|] | size - rozmiar, jak wiele liczb ma byc zapisane |

Definicja w linii 13 pliku ZapiszStosKolejkaLista.cpp.

Oto graf wywołań dla tej funkcji:

