

Struktury_Tablice_Dynamiczne

1

Wygenerowano przez Doxygen 1.7.1

Thu Mar 26 2015 06:32:47

Spis treści

1	Program tworzący struktury danych.	1
2	Indeks klas	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 Benchmark	9
5.1.1	Opis szczegółowy	10
5.1.2	Dokumentacja konstruktora i destruktora	10
5.1.2.1	Benchmark	10
5.1.2.2	~Benchmark	11
5.1.3	Dokumentacja funkcji składowych	11
5.1.3.1	Czas_Start	11
5.1.3.2	Czas_Stop	11
5.1.3.3	Licz_Srednia	12
5.1.3.4	Losuj	12
5.1.3.5	Tablica	13
5.1.3.6	Wczytaj_Dane	13
5.1.3.7	Zapisz_Wyniki	13
5.1.4	Dokumentacja atrybutów składowych	14
5.1.4.1	_tablica_liczb	14

5.1.4.2	Ilosc_Danych	14
5.1.4.3	iterator	14
5.1.4.4	iterator_sredniej	14
5.1.4.5	rozmiar_tablic	14
5.1.4.6	srednia_jednego_problemu	14
5.1.4.7	stoper_start	14
5.1.4.8	stoper_stop	14
5.1.4.9	wielkosc_problemu	14
5.2	Dokumentacja szablonu klasy Kolejka< TYP >	15
5.2.1	Opis szczegółowy	16
5.2.2	Dokumentacja funkcji składowych	16
5.2.2.1	POP	16
5.2.2.2	PUSH	16
5.2.2.3	SHOW	17
5.2.2.4	SIZE	17
5.3	Dokumentacja szablonu klasy Lista< TYP >	17
5.3.1	Opis szczegółowy	18
5.3.2	Dokumentacja konstruktora i destruktor	18
5.3.2.1	Lista	18
5.3.2.2	~Lista	18
5.3.3	Dokumentacja funkcji składowych	18
5.3.3.1	POP	18
5.3.3.2	Powieszenie_Pamieci	19
5.3.3.3	PUSH	19
5.3.3.4	Rozmiar	19
5.3.3.5	SHOW	20
5.3.3.6	SIZE	20
5.3.3.7	Zmniejszenie_Pamieci	21
5.3.4	Dokumentacja atrybutów składowych	21
5.3.4.1	_rozmiar_listy	21
5.3.4.2	koniec	21
5.3.4.3	poczatek	21
5.3.4.4	tab	21
5.4	Dokumentacja szablonu klasy Stos< TYP >	22

5.4.1	Opis szczegółowy	23
5.4.2	Dokumentacja funkcji składowych	23
5.4.2.1	POP	23
5.4.2.2	PUSH	23
5.4.2.3	SHOW	24
5.4.2.4	SIZE	24
6	Dokumentacja plików	27
6.1	Dokumentacja pliku Benchmark.cpp	27
6.1.1	Opis szczegółowy	28
6.2	Dokumentacja pliku Benchmark.hh	28
6.2.1	Opis szczegółowy	29
6.3	Dokumentacja pliku Kolejka.hh	29
6.3.1	Opis szczegółowy	30
6.4	Dokumentacja pliku Lista.cpp	30
6.4.1	Dokumentacja funkcji	31
6.4.1.1	main	31
6.5	Dokumentacja pliku Lista.hh	32
6.5.1	Opis szczegółowy	33
6.6	Dokumentacja pliku Stos.hh	33
6.6.1	Opis szczegółowy	35
6.7	Dokumentacja pliku strona.dox	35
6.8	Dokumentacja pliku Struktury.cpp	35
6.8.1	Dokumentacja funkcji	36
6.8.1.1	main	36
6.9	Dokumentacja pliku Test.cpp	36
6.9.1	Dokumentacja definicji	37
6.9.1.1	STALA	37
6.9.2	Dokumentacja funkcji	37
6.9.2.1	funkcja	37
6.9.2.2	main	38

Rozdział 1

Program tworzący struktury danych.

Autor

Lukasz Sak

Wersja

1

Program posiada definicje struktur danych: [Stos](#), [Lista](#), [Kolejka](#). Struktury posiadają większość takich samych metod. PUSH() - wrzucająca dane do struktury, POP() - usuwający odpowiednią dane ze struktury, SIZE() - zwracający ilość elementów w strukturze, SHOW() - wyświetlający elementy struktury. Program posiada szablon dzięki któremu możemy używać różnych typów danych.

Rozdział 2

Indeks klas

2.1 Hierarchia klas

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

Benchmark	9
Lista< TYP >	17
Kolejka< TYP >	15
Stos< TYP >	22

Rozdział 3

Indeks klas

3.1 Lista klas

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

Benchmark (Klasa Benchmark)	9
Kolejka< TYP > (Klasa Kolejka)	15
Lista< TYP > (Klasa Lista)	17
Stos< TYP > (Klasa Stos)	22

Rozdział 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

Benchmark.cpp (Metody klasy Benchmark)	27
Benchmark.hh (Definicja klasy Benchmark)	28
Kolejka.hh (Definicja klasy Kolejka)	29
Lista.cpp	30
Lista.hh (Definicja klasy Lista)	32
Stos.hh (Definicja klasy Stos)	33
Struktury.cpp	35
Test.cpp	36

Rozdział 5

Dokumentacja klas

5.1 Dokumentacja klasy Benchmark

Klasa [Benchmark](#).

```
#include <Benchmark.hh>
```

Metody publiczne

- [Benchmark](#) (unsigned int rozmiar_problemu, double stala)
Inicjalizator klasy [Benchmark](#).
- double [Tablica](#) (int i)
- float [Licz_Srednia](#) ()
- float [Czas_Start](#) ()
- float [Czas_Stop](#) ()
- void [Zapisz_Wyniki](#) ()
- [~Benchmark](#) ()

Atrybuty publiczne

- unsigned int * [wielkosc_problemu](#)
- unsigned int [Ilosc_Danych](#)

Metody prywatne

- void [Losuj](#) (int *tablica_liczb, unsigned int rozmiar)
- unsigned int [Wczytaj_Dane](#) ()

Atrybuty prywatne

- int * `_tablica_liczb`
- float `stoper_start`
- unsigned int `iterator`
- unsigned int `iterator_sredniej`
- float * `stoper_stop`
- float * `srednia_jednego_problemu`
- unsigned int `rozmiar_tablic`

5.1.1 Opis szczegółowy

Klasa ta modeluje nam test dla funkcji Składa się z pól:

Parametry

- [in] `_tablica_liczb` - która przechowuje nasze dane ktorymi bedziemy testowali funkcje
- [in] `stoper_start` - przechowuje poczatek mierzenia czasu
- [in] `iterator` - sluzy nam do iterowania od 0 do 9 (10 prob) zatrzymania czasu
- [in] `iterator_sredniej` - sluzy nam do iterowania kolejnego pomiaru sredniej w zaleznosci od ilosci prob
- [in] `stoper_stop` - przechowuje nam 10 wynikow pomiaru czasu (obliczony wynik jednego pomiaru)
- [in] `srednia_jednego_problemu` - przechowuje tablice sredniego czasu wykonania pomiarow dla poszczegolnych prob
- [in] `ilosc_problemu` - przechowuje nam jak duzo prob bedzie wykonywanych
- [in] `Ilosc_Danych` - ilosc danych na ktorych bedziemy pracowali
- [in] `wielkosc_problemu` - ilosc pojedynczego problemu(ilosci danych na 1 probe)

Definicja w linii 30 pliku Benchmark.hh.

5.1.2 Dokumentacja konstruktora i destruktor

5.1.2.1 `Benchmark::Benchmark (unsigned int rozmiar_problemu, double stala)`

Inicjalizator ten służy do określania początkowych wartości pól klasy oraz wyboru na jakich danych bedziemy pracowali (losowe/wczytane)

Opis argumentów:

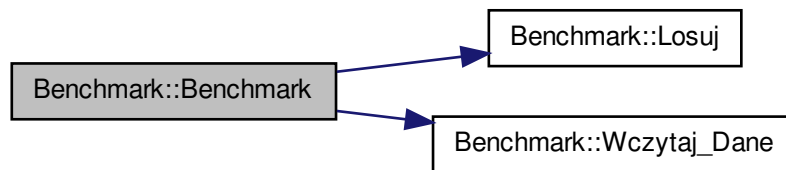
Parametry

- [in] `rozmiar_problemu` - ilosc maksymalnej liczby wprowadzanych danych

[in] *stala* - stała przez którą będziemy mnożyli, aby np. uzyskać więcej wyników najlepszy przedział (1.1-10)

Definicja w linii 15 pliku Benchmark.cpp.

Oto graf wywołań dla tej funkcji:



5.1.2.2 `Benchmark::~~Benchmark () [inline]`

Definicja w linii 125 pliku Benchmark.hh.

5.1.3 Dokumentacja funkcji składowych

5.1.3.1 `float Benchmark::Czas_Start ()`

Definicja w linii 66 pliku Benchmark.cpp.

Oto graf wywołań tej funkcji:



5.1.3.2 `float Benchmark::Czas_Stop ()`

Definicja w linii 73 pliku Benchmark.cpp.

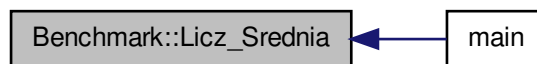
Oto graf wywołań tej funkcji:



5.1.3.3 `float Benchmark::Licz_Srednia ()`

Definicja w linii 83 pliku `Benchmark.cpp`.

Oto graf wywołań tej funkcji:



5.1.3.4 `void Benchmark::Losuj (int * tablica_liczb, unsigned int rozmiar)` [private]

Definicja w linii 56 pliku `Benchmark.cpp`.

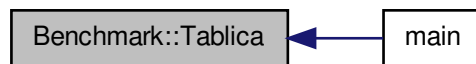
Oto graf wywołań tej funkcji:



5.1.3.5 double Benchmark::Tablica (int i) [inline]

Definicja w linii 83 pliku Benchmark.hh.

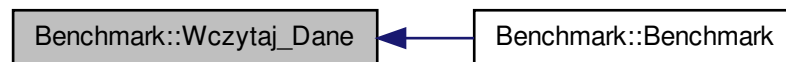
Oto graf wywołań tej funkcji:



5.1.3.6 unsigned int Benchmark::Wczytaj_Dane () [private]

Definicja w linii 114 pliku Benchmark.cpp.

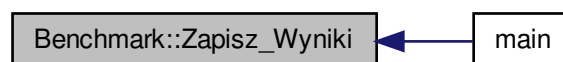
Oto graf wywołań tej funkcji:



5.1.3.7 void Benchmark::Zapisz_Wyniki ()

Definicja w linii 95 pliku Benchmark.cpp.

Oto graf wywołań tej funkcji:



5.1.4 Dokumentacja atrybutów składowych

5.1.4.1 `int* Benchmark::_tablica_liczb` `[private]`

Definicja w linii 32 pliku `Benchmark.hh`.

5.1.4.2 `unsigned int Benchmark::Ilosc_Danych`

Definicja w linii 74 pliku `Benchmark.hh`.

5.1.4.3 `unsigned int Benchmark::iterator` `[private]`

Definicja w linii 34 pliku `Benchmark.hh`.

5.1.4.4 `unsigned int Benchmark::iterator_sredniej` `[private]`

Definicja w linii 35 pliku `Benchmark.hh`.

5.1.4.5 `unsigned int Benchmark::rozmiar_tablic` `[private]`

Definicja w linii 38 pliku `Benchmark.hh`.

5.1.4.6 `float* Benchmark::srednia_jednego_problemu` `[private]`

Definicja w linii 37 pliku `Benchmark.hh`.

5.1.4.7 `float Benchmark::stoper_start` `[private]`

Definicja w linii 33 pliku `Benchmark.hh`.

5.1.4.8 `float* Benchmark::stoper_stop` `[private]`

Definicja w linii 36 pliku `Benchmark.hh`.

5.1.4.9 `unsigned int* Benchmark::wielkosc_problemu`

Definicja w linii 73 pliku `Benchmark.hh`.

Dokumentacja dla tej klasy została wygenerowana z plików:

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

5.2 Dokumentacja szablonu klasy Kolejka< TYP >

Klasa [Kolejka](#).

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

Diagram dziedziczenia dla Kolejka< TYP >

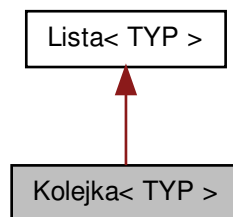
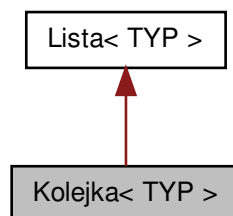


Diagram współpracy dla Kolejka< TYP >:



Metody publiczne

- void [PUSH](#) (TYP liczba)
- int [POP](#) ()
- void [SHOW](#) ()
- unsigned int [SIZE](#) ()

5.2.1 Opis szczegółowy

```
template<typename TYP> class Kolejka< TYP >
```

Klasa ta modeluje nam Kolejke Składa się z pól klasy [Lista](#) oraz metod PUSH, POP, SIZE, SHOW Klasa w calosci wykorzystuje implementacje listy

Definicja w linii 24 pliku Kolejka.hh.

5.2.2 Dokumentacja funkcji składowych

5.2.2.1 `template<typename TYP> int Kolejka< TYP >::POP ()`
`[inline]`

Definicja w linii 40 pliku Kolejka.hh.

Oto graf wywoływań tej funkcji:



5.2.2.2 `template<typename TYP> void Kolejka< TYP >::PUSH (TYP liczba)`
`[inline]`

Reimplementowana z [Lista< TYP >](#).

Definicja w linii 33 pliku Kolejka.hh.

Oto graf wywoływań tej funkcji:



5.2.2.3 `template<typename TYP> void Kolejka< TYP >::SHOW ()`
`[inline]`

Reimplementowana z [Lista< TYP >](#).

Definicja w linii 48 pliku Kolejka.hh.

5.2.2.4 `template<typename TYP> unsigned int Kolejka< TYP >::SIZE ()`
`[inline]`

Reimplementowana z [Lista< TYP >](#).

Definicja w linii 57 pliku Kolejka.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

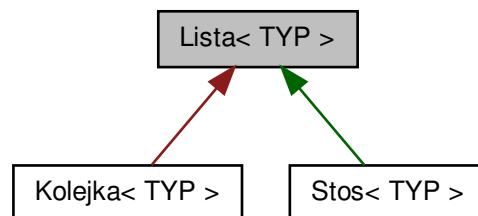
- [Kolejka.hh](#)

5.3 Dokumentacja szablonu klasy Lista< TYP >

Klasa [Lista](#).

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

Diagram dziedziczenia dla Lista< TYP >



Metody publiczne

- void [Rozmiar](#) ()
- [Lista](#) ()
- [~Lista](#) ()
- void [PUSH](#) (TYP liczba)
- void [Powiekszenie_Pamieci](#) ()

- void [Zmniejszenie_Pamieci](#) ()
- int [POP](#) (int *liczba*)
- unsigned int [SIZE](#) ()
- void [SHOW](#) ()

Atrybuty prywatne

- TYP * [tab](#)
- unsigned int [_rozmiar_listy](#)
- unsigned int [poczatek](#)
- unsigned int [koniec](#)

5.3.1 Opis szczegółowy

template<typename TYP> class Lista< TYP >

Klasa ta modeluje nam Listę wartości typu TYP Składa się z pól:

Parametry

- [in] **tab* - tablica naszych liczb;
- [in] *poczatek* - pierwsza liczba w naszej tablicy
- [in] *koniec* - ostatnia liczba w naszej tablicy
- [in] *_rozmiar_listy* - rozmiar stworzonej tablicy dynamicznej

Definicja w linii 27 pliku Lista.hh.

5.3.2 Dokumentacja konstruktora i destruktor

5.3.2.1 template<typename TYP> Lista< TYP >::Lista () [inline]

Definicja w linii 37 pliku Lista.hh.

5.3.2.2 template<typename TYP> Lista< TYP >::~~Lista () [inline]

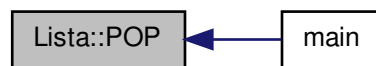
Definicja w linii 38 pliku Lista.hh.

5.3.3 Dokumentacja funkcji składowych

5.3.3.1 template<typename TYP> int Lista< TYP >::POP (int *liczba*) [inline]

Definicja w linii 97 pliku Lista.hh.

Oto graf wywołań tej funkcji:



5.3.3.2 `template<typename TYP> void Lista< TYP >::Powiekszenie_Pamieci () [inline]`

Definicja w linii 63 pliku Lista.hh.

5.3.3.3 `template<typename TYP> void Lista< TYP >::PUSH (TYP liczba) [inline]`

Reimplementowana w [Kolejka< TYP >](#) i [Stos< TYP >](#).

Definicja w linii 47 pliku Lista.hh.

Oto graf wywołań tej funkcji:



5.3.3.4 `template<typename TYP> void Lista< TYP >::Rozmiar () [inline]`

Definicja w linii 36 pliku Lista.hh.

Oto graf wywołań tej funkcji:



5.3.3.5 `template<typename TYP> void Lista< TYP >::SHOW ()` `[inline]`

Reimplementowana w [Kolejka< TYP >](#) i [Stos< TYP >](#).

Definicja w linii 123 pliku `Lista.hh`.

Oto graf wywołań tej funkcji:

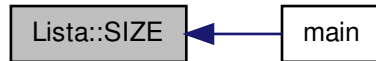


5.3.3.6 `template<typename TYP> unsigned int Lista< TYP >::SIZE ()` `[inline]`

Reimplementowana w [Kolejka< TYP >](#) i [Stos< TYP >](#).

Definicja w linii 112 pliku `Lista.hh`.

Oto graf wywołań tej funkcji:



5.3.3.7 `template<typename TYP> void Lista< TYP >::Zmniejszenie_Pamieci
() [inline]`

Definicja w linii 71 pliku Lista.hh.

5.3.4 Dokumentacja atrybutów składowych

5.3.4.1 `template<typename TYP> unsigned int Lista< TYP >::_rozmiar_listy
[private]`

Definicja w linii 31 pliku Lista.hh.

5.3.4.2 `template<typename TYP> unsigned int Lista< TYP >::koniec
[private]`

Definicja w linii 33 pliku Lista.hh.

5.3.4.3 `template<typename TYP> unsigned int Lista< TYP >::poczatek
[private]`

Definicja w linii 32 pliku Lista.hh.

5.3.4.4 `template<typename TYP> TYP* Lista< TYP >::tab [private]`

Definicja w linii 30 pliku Lista.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Lista.hh](#)

5.4 Dokumentacja szablonu klasy Stos< TYP >

Klasa [Stos](#).

```
#include <Stos.hh>
```

Diagram dziedziczenia dla Stos< TYP >

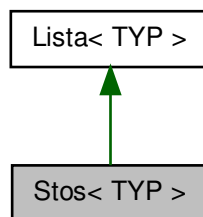
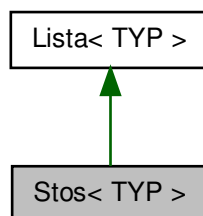


Diagram współpracy dla Stos< TYP >:



Metody publiczne

- void [PUSH](#) (TYP liczba)
- int [POP](#) ()
- void [SHOW](#) ()
- unsigned int [SIZE](#) ()

5.4.1 Opis szczegółowy

```
template<typename TYP> class Stos< TYP >
```

Klasa ta modeluje nam [Stos](#) Składa się z pól klasy [Lista](#) ktore zostania uzyte oraz metod PUSH, POP, SIZE, SHOW Klasa w calosci wykorzystuje implementacje listy

Definicja w linii 24 pliku Stos.hh.

5.4.2 Dokumentacja funkcji składowych

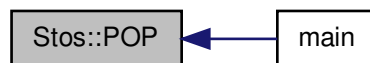
5.4.2.1 `template<typename TYP> int Stos< TYP >::POP () [inline]`

Definicja w linii 41 pliku Stos.hh.

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



5.4.2.2 `template<typename TYP> void Stos< TYP >::PUSH (TYP liczba) [inline]`

Reimplementowana z [Lista< TYP >](#).

Definicja w linii 34 pliku Stos.hh.

Oto graf wywołań tej funkcji:



5.4.2.3 `template<typename TYP> void Stos< TYP >::SHOW ()` `[inline]`

Reimplementowana z [Lista< TYP >](#).

Definicja w linii 49 pliku `Stos.hh`.

Oto graf wywołań tej funkcji:

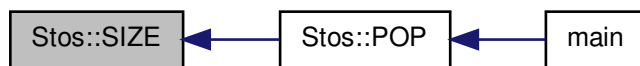


5.4.2.4 `template<typename TYP> unsigned int Stos< TYP >::SIZE ()` `[inline]`

Reimplementowana z [Lista< TYP >](#).

Definicja w linii 58 pliku `Stos.hh`.

Oto graf wywołań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Stos.hh](#)

Rozdział 6

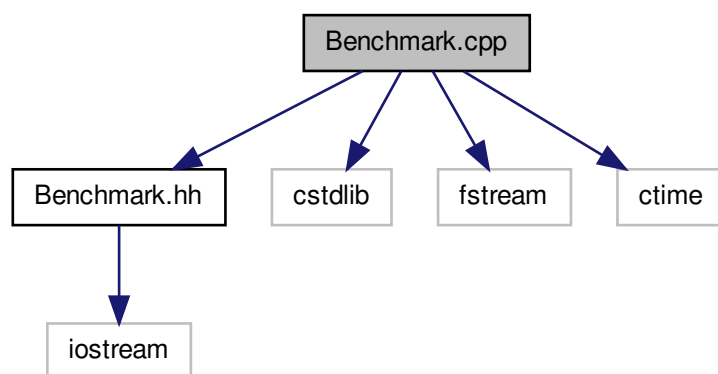
Dokumentacja plików

6.1 Dokumentacja pliku Benchmark.cpp

Metody klasy [Benchmark](#).

```
#include "Benchmark.hh"  
#include <cstdlib>  
#include <fstream>  
#include <ctime>
```

Wykres zależności załączania dla Benchmark.cpp:



6.1.1 Opis szczegółowy

Plik zawiera definicje metod klasy [Benchmark](#)

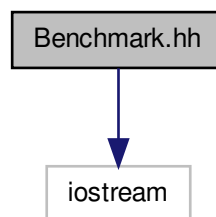
Definicja w pliku [Benchmark.cpp](#).

6.2 Dokumentacja pliku Benchmark.hh

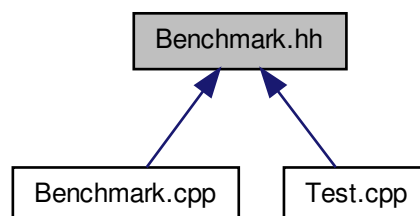
Definicja klasy [Benchmark](#).

```
#include <iostream>
```

Wykres zależności załączania dla Benchmark.hh:



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



Komponenty

- class [Benchmark](#)

Klasa [Benchmark](#).

6.2.1 Opis szczegółowy

Plik zawiera definicje klasy [Benchmark](#) która będzie wyznaczała nam punkty do wyznaczenia złożoności obliczeniowej.

Definicja w pliku [Benchmark.hh](#).

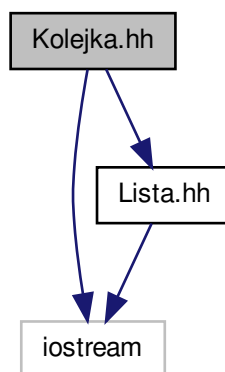
6.3 Dokumentacja pliku Kolejka.hh

Definicja klasy [Kolejka](#).

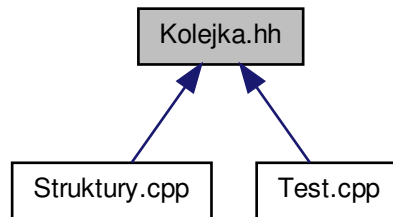
```
#include <iostream>
```

```
#include "Lista.hh"
```

Wykres zależności załączania dla Kolejka.hh:



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



Komponenty

- class `Kolejka< TYP >`

Klasa `Kolejka`.

6.3.1 Opis szczegółowy

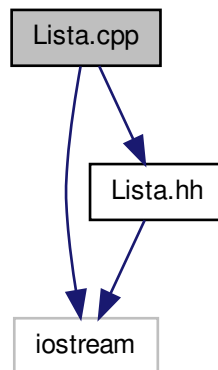
Plik zawiera definicje klasy `Kolejka`, która będzie strukturą naszych danych. Klasa ta posiada szablon, dzięki czemu możemy pracować na różnych typach danych

Definicja w pliku `Kolejka.hh`.

6.4 Dokumentacja pliku `Lista.cpp`

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

Wykres zależności załączania dla Lista.cpp:



Funkcje

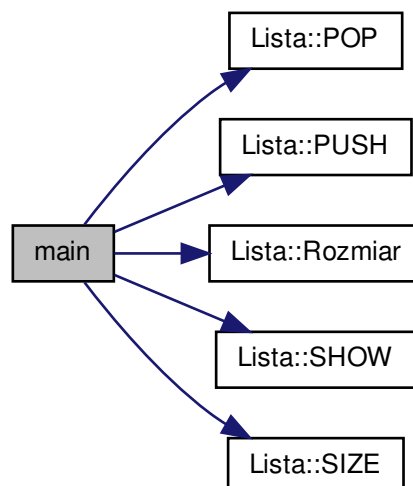
- int `main` ()

6.4.1 Dokumentacja funkcji

6.4.1.1 int `main` ()

Definicja w linii 6 pliku Lista.cpp.

Oto graf wywołań dla tej funkcji:

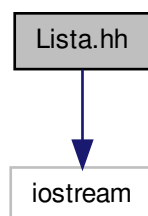


6.5 Dokumentacja pliku Lista.hh

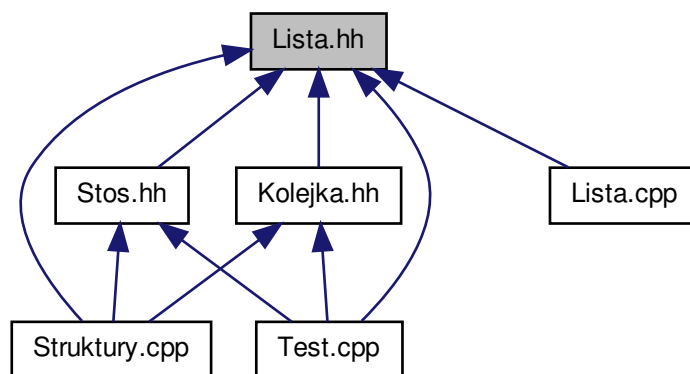
Definicja klasy [Lista](#).

```
#include <iostream>
```

Wykres zależności załączania dla Lista.hh:



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



Komponenty

- class `Lista< TYP >`

Klasa `Lista`.

6.5.1 Opis szczegółowy

Plik zawiera definicje klasy `Lista` która będzie struktura danych oparta na tablicy dynamicznej

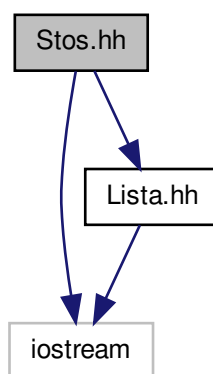
Definicja w pliku `Lista.hh`.

6.6 Dokumentacja pliku Stos.hh

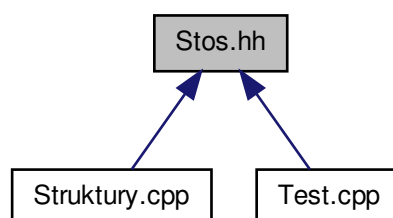
Definicja klasy `Stos`.

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

Wykres zależności załączania dla Stos.hh:



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



Komponenty

- class `Stos< TYP >`

Klasa Stos.

6.6.1 Opis szczegółowy

Plik zawiera definicje klasy [Stos](#), która będzie struktura naszych danych. Klasa ta posiada szablon, dzięki czemu możemy pracować na różnych typach danych

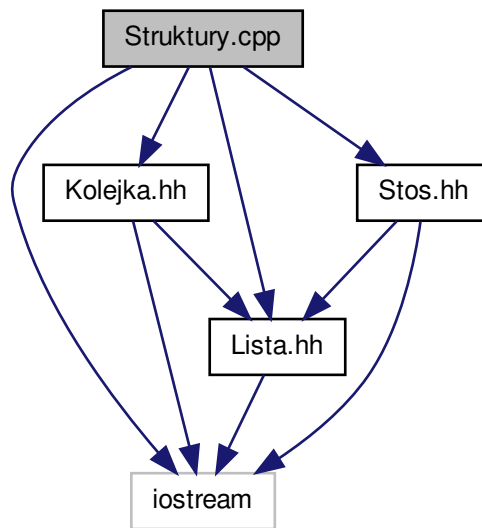
Definicja w pliku [Stos.hh](#).

6.7 Dokumentacja pliku strona.dox

6.8 Dokumentacja pliku Struktury.cpp

```
#include <iostream>
#include "Lista.hh"
#include "Kolejka.hh"
#include "Stos.hh"
```

Wykres zależności załączania dla Struktury.cpp:



Funkcje

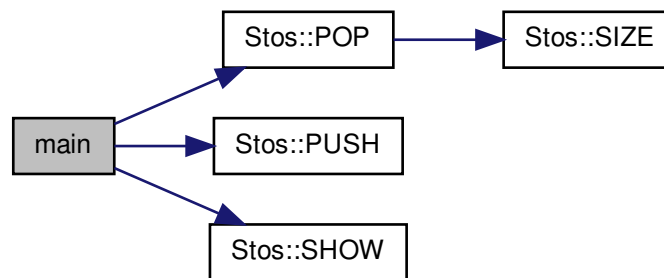
- int [main](#) ()

6.8.1 Dokumentacja funkcji

6.8.1.1 `int main ()`

Definicja w linii 8 pliku `Struktury.cpp`.

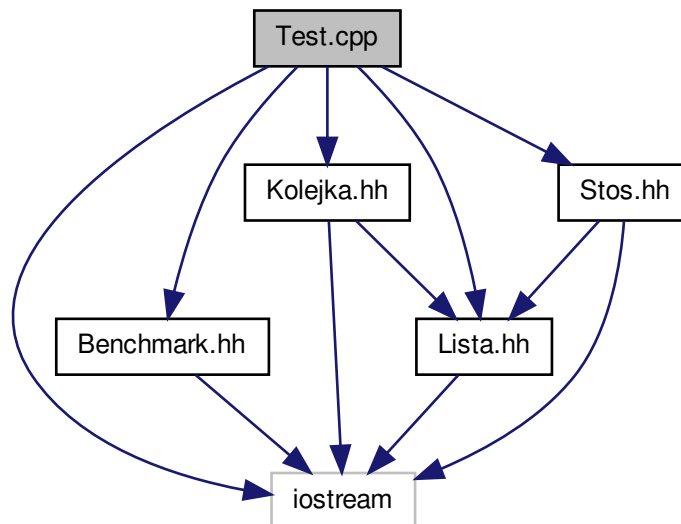
Oto graf wywołań dla tej funkcji:



6.9 Dokumentacja pliku `Test.cpp`

```
#include <iostream>
#include "Benchmark.hh"
#include "Lista.hh"
#include "Kolejka.hh"
#include "Stos.hh"
```

Wykres zależności załączania dla Test.cpp:



Definicje

- `#define STALA 10`

Funkcje

- `double funkcja (double x)`
- `int main ()`

6.9.1 Dokumentacja definicji

6.9.1.1 `#define STALA 10`

Definicja w linii 7 pliku Test.cpp.

6.9.2 Dokumentacja funkcji

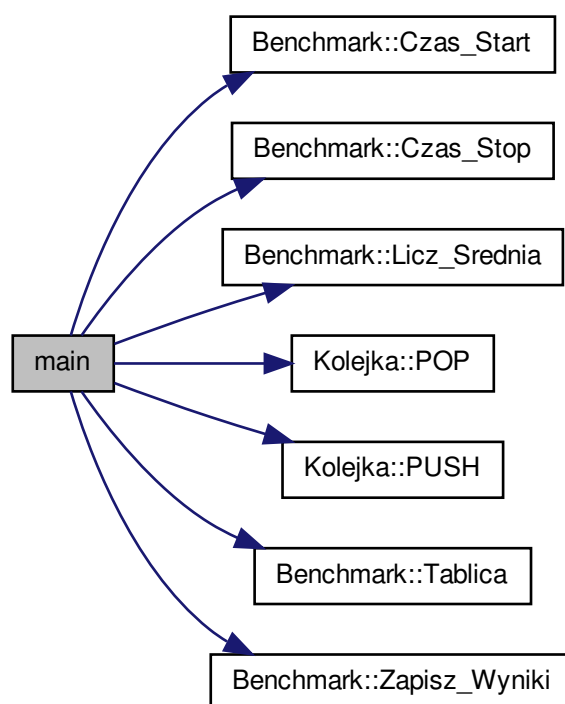
6.9.2.1 `double funkcja (double x)`

Definicja w linii 10 pliku Test.cpp.

6.9.2.2 int main ()

Definicja w linii 14 pliku Test.cpp.

Oto graf wywołań dla tej funkcji:



Skorowidz

- ~Benchmark
 - Benchmark, [11](#)
- ~Lista
 - Lista, [18](#)
- _rozmiar_listy
 - Lista, [21](#)
- _tablica_liczb
 - Benchmark, [14](#)
- Benchmark, [9](#)
 - ~Benchmark, [11](#)
 - _tablica_liczb, [14](#)
 - Benchmark, [10](#)
 - Czas_Start, [11](#)
 - Czas_Stop, [11](#)
 - Ilosc_Danych, [14](#)
 - iterator, [14](#)
 - iterator_sredniej, [14](#)
 - Licz_Srednia, [12](#)
 - Losuj, [12](#)
 - rozmiar_tablic, [14](#)
 - srednia_jednego_problemu, [14](#)
 - stoper_start, [14](#)
 - stoper_stop, [14](#)
 - Tablica, [12](#)
 - Wczytaj_Dane, [13](#)
 - wielkosc_problemu, [14](#)
 - Zapisz_Wyniki, [13](#)
- Benchmark.cpp, [27](#)
- Benchmark.hh, [28](#)
- Czas_Start
 - Benchmark, [11](#)
- Czas_Stop
 - Benchmark, [11](#)
- funkcja
 - Test.cpp, [37](#)
- Ilosc_Danych
 - Benchmark, [14](#)
- iterator
 - Benchmark, [14](#)
- iterator_sredniej
 - Benchmark, [14](#)
- Kolejka, [15](#)
 - POP, [16](#)
 - PUSH, [16](#)
 - SHOW, [16](#)
 - SIZE, [17](#)
- Kolejka.hh, [29](#)
- koniec
 - Lista, [21](#)
- Licz_Srednia
 - Benchmark, [12](#)
- Lista, [17](#)
 - ~Lista, [18](#)
 - _rozmiar_listy, [21](#)
 - koniec, [21](#)
 - Lista, [18](#)
 - poczatek, [21](#)
 - POP, [18](#)
 - Powiekszenie_Pamieci, [19](#)
 - PUSH, [19](#)
 - Rozmiar, [19](#)
 - SHOW, [20](#)
 - SIZE, [20](#)
 - tab, [21](#)
 - Zmniejszenie_Pamieci, [21](#)
- Lista.cpp, [30](#)
 - main, [31](#)
- Lista.hh, [32](#)
- Losuj
 - Benchmark, [12](#)
- main
 - Lista.cpp, [31](#)
 - Struktury.cpp, [36](#)
 - Test.cpp, [37](#)
- poczatek
 - Lista, [21](#)

POP
 Kolejka, [16](#)
 Lista, [18](#)
 Stos, [23](#)
Powiekszenie_Pamieci
 Lista, [19](#)
PUSH
 Kolejka, [16](#)
 Lista, [19](#)
 Stos, [23](#)

Rozmiar
 Lista, [19](#)
rozmiar_tablic
 Benchmark, [14](#)

SHOW
 Kolejka, [16](#)
 Lista, [20](#)
 Stos, [24](#)
SIZE
 Kolejka, [17](#)
 Lista, [20](#)
 Stos, [24](#)
srednia_jednego_problemu
 Benchmark, [14](#)
STALA
 Test.cpp, [37](#)
stoper_start
 Benchmark, [14](#)
stoper_stop
 Benchmark, [14](#)
Stos, [22](#)
 POP, [23](#)
 PUSH, [23](#)
 SHOW, [24](#)
 SIZE, [24](#)
Stos.hh, [33](#)
strona.dox, [35](#)
Struktury.cpp, [35](#)
 main, [36](#)

tab
 Lista, [21](#)
Tablica
 Benchmark, [12](#)
Test.cpp, [36](#)
 funkcja, [37](#)
 main, [37](#)
 STALA, [37](#)

Wczytaj_Dane
 Benchmark, [13](#)
wielkosc_problemu
 Benchmark, [14](#)

Zapisz_Wyniki
 Benchmark, [13](#)
Zmniejszenie_Pamieci
 Lista, [21](#)