

Struktury

1

Wygenerowano przez Doxygen 1.7.1

Thu Mar 26 2015 06:26:58

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 Element< TYP >	15
5.2.1	Opis szczegółowy	15
5.2.2	Dokumentacja konstruktora i destruktoru	15
5.2.2.1	Element	15
5.2.2.2	~Element	16
5.2.3	Dokumentacja atrybutów składowych	16
5.2.3.1	nastepny	16
5.2.3.2	wartosc	16
5.3	Dokumentacja szablonu klasy Kolejka< TYP >	16
5.3.1	Opis szczegółowy	17
5.3.2	Dokumentacja funkcji składowych	17
5.3.2.1	POP	17
5.3.2.2	PUSH	17
5.3.2.3	SHOW	18
5.3.2.4	SIZE	18
5.4	Dokumentacja szablonu klasy Lista< TYP >	18
5.4.1	Opis szczegółowy	19
5.4.2	Dokumentacja konstruktora i destruktoru	19
5.4.2.1	Lista	19
5.4.2.2	~Lista	19
5.4.3	Dokumentacja funkcji składowych	19
5.4.3.1	POP	19
5.4.3.2	PUSH	20
5.4.3.3	SHOW	20
5.4.3.4	SIZE	20
5.4.4	Dokumentacja atrybutów składowych	20

5.4.4.1	ostatni	20
5.4.4.2	pierwszy	20
5.5	Dokumentacja szablonu klasy Stos< TYP >	21
5.5.1	Opis szczegółowy	22
5.5.2	Dokumentacja funkcji składowych	22
5.5.2.1	POP	22
5.5.2.2	PUSH	22
5.5.2.3	SHOW	23
5.5.2.4	SIZE	23
6	Dokumentacja plików	25
6.1	Dokumentacja pliku Benchmark.cpp	25
6.1.1	Opis szczegółowy	26
6.2	Dokumentacja pliku Benchmark.hh	26
6.2.1	Opis szczegółowy	27
6.3	Dokumentacja pliku Element.hh	27
6.3.1	Opis szczegółowy	28
6.4	Dokumentacja pliku Kolejka.hh	28
6.4.1	Opis szczegółowy	29
6.5	Dokumentacja pliku Lista.hh	29
6.5.1	Opis szczegółowy	31
6.6	Dokumentacja pliku Stos.hh	31
6.6.1	Opis szczegółowy	32
6.7	Dokumentacja pliku strona.dox	32
6.8	Dokumentacja pliku Struktury.cpp	32
6.8.1	Dokumentacja funkcji	33
6.8.1.1	main	33
6.9	Dokumentacja pliku Test.cpp	34
6.9.1	Dokumentacja definicji	35
6.9.1.1	STALA	35
6.9.2	Dokumentacja funkcji	35
6.9.2.1	funkcja	35
6.9.2.2	main	36

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 odpowiedni dane ze struktury, SIZE() - zwracający ilość elementów w strukturze, SHOW() - wyświetlający elementy struktury. Struktury są zrobione na szablonach, zatem można używać kilku typów danych.

Rozdział 2

Indeks klas

2.1 Hierarchia klas

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

Benchmark	9
Element< TYP >	15
Lista< TYP >	18
Kolejka< TYP >	16
Stos< TYP >	21

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
Element< TYP > (Klasa Element)	15
Kolejka< TYP > (Klasa Kolejka)	16
Lista< TYP > (Klasa Lista)	18
Stos< TYP > (Klasa Stos)	21

Rozdział 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

Benchmark.cpp (Metody klasy Benchmark)	25
Benchmark.hh (Definicja klasy Benchmark)	26
Element.hh (Definicja klasy Element)	27
Kolejka.hh (Definicja klasy Kolejka)	28
Lista.hh (Definicja klasy Lista)	29
Stos.hh (Definicja klasy Stos)	31
Struktury.cpp	32
Test.cpp	34

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:

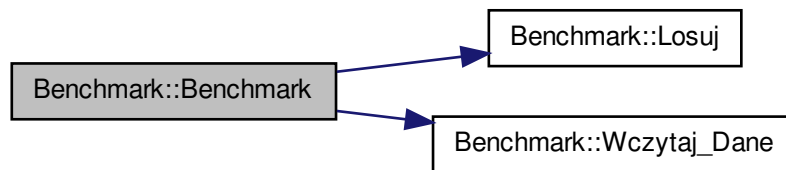
Parametry

- [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.

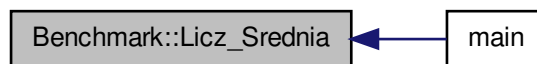
Oto graf wywołań tej funkcji:



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`.

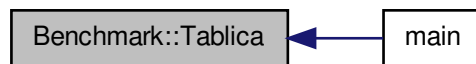
Oto graf wywołań tej funkcji:



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

Definicja w linii 83 pliku Benchmark.hh.

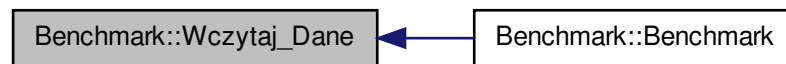
Oto graf wywołań tej funkcji:



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

Definicja w linii 114 pliku Benchmark.cpp.

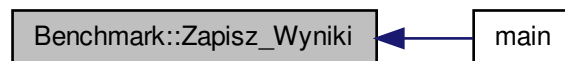
Oto graf wywołań tej funkcji:



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:

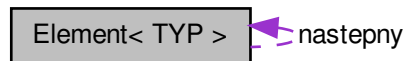
- [Benchmark.hh](#)
- [Benchmark.cpp](#)

5.2 Dokumentacja szablonu klasy `Element< TYP >`

Klasa `Element`.

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

Diagram współpracy dla `Element< TYP >`:



Metody publiczne

- `Element()`
- `~Element()`

Atrybuty publiczne

- `TYP wartosc`
- `Element * nastepny`

5.2.1 Opis szczegółowy

```
template<class TYP> class Element< TYP >
```

Klasa ta modeluje nam pojedyncza dana oraz wskaźnik na kolejna dana

Parametry

[in] `*nastepny` - jako wskaźnik na kolejny element

[in] `wartosc` - wartosc naszego elementu zdefiniowana przez nas przy uzyciu szablonu

Definicja w linii 22 pliku `Element.hh`.

5.2.2 Dokumentacja konstruktora i destruktora

5.2.2.1 `template<class TYP> Element< TYP >::Element() [inline]`

Definicja w linii 29 pliku `Element.hh`.

5.2.2.2 `template<class TYP> Element< TYP >::~~Element () [inline]`

Definicja w linii 34 pliku Element.hh.

5.2.3 Dokumentacja atrybutów składowych

5.2.3.1 `template<class TYP> Element* Element< TYP >::nastepny`

Definicja w linii 27 pliku Element.hh.

5.2.3.2 `template<class TYP> TYP Element< TYP >::wartosc`

Definicja w linii 26 pliku Element.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Element.hh](#)

5.3 Dokumentacja szablonu klasy Kolejka< TYP >

Klasa [Kolejka](#).

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

Diagram dziedziczenia dla Kolejka< TYP >

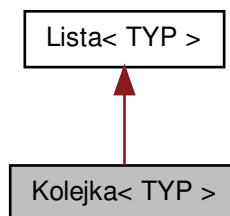
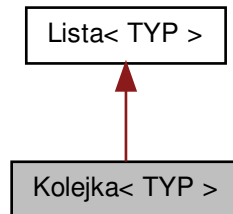


Diagram współpracy dla Kolejka< TYP >:



Metody publiczne

- void **PUSH** (TYP liczba)
- int **POP** ()
- void **SHOW** ()
- unsigned int **SIZE** ()

5.3.1 Opis szczegółowy

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

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

Definicja w linii 25 pliku Kolejka.hh.

5.3.2 Dokumentacja funkcji składowych

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

Definicja w linii 41 pliku Kolejka.hh.

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

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

Definicja w linii 34 pliku Kolejka.hh.

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

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

Definicja w linii 49 pliku Kolejka.hh.

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

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

Definicja w linii 58 pliku Kolejka.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

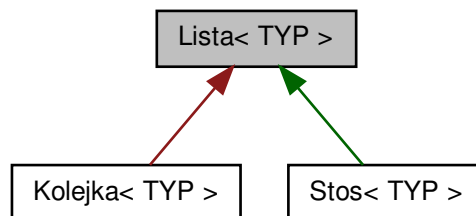
- [Kolejka.hh](#)

5.4 Dokumentacja szablonu klasy Lista< TYP >

Klasa [Lista](#).

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

Diagram dziedziczenia dla Lista< TYP >



Metody publiczne

- [Lista \(\)](#)
- [~Lista \(\)](#)
- void [PUSH](#) (TYP liczba)
- int [POP](#) (unsigned int Numer_Elementu)
- unsigned int [SIZE](#) ()
- void [SHOW](#) ()

Atrybuty chronione

- `Element< TYP > * pierwszy`
- `Element< TYP > * ostatni`

5.4.1 Opis szczegółowy

```
template<typename TYP> class Lista< TYP >
```

Klasa ta modeluje nam Listę Składa się z metod PUSH, POP, SIZE, SHOW oraz pól:

Parametry

[in] **pierwszy* - jako wskaźnik na pierwszy element

Definicja w linii 27 pliku Lista.hh.

5.4.2 Dokumentacja konstruktora i destruktor

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

Definicja w linii 33 pliku Lista.hh.

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

Definicja w linii 34 pliku Lista.hh.

5.4.3 Dokumentacja funkcji składowych

```
5.4.3.1 template<typename TYP> int Lista< TYP >::POP ( unsigned int  
Numer_Elementu ) [inline]
```

Definicja w linii 59 pliku Lista.hh.

Oto graf wywołań tej funkcji:



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

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

Definicja w linii 44 pliku Lista.hh.

Oto graf wywołań tej funkcji:



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

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

Definicja w linii 108 pliku Lista.hh.

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

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

Definicja w linii 88 pliku Lista.hh.

5.4.4 Dokumentacja atrybutów składowych

5.4.4.1 `template<typename TYP> Element<TYP>* Lista< TYP >::ostatni` `[protected]`

Definicja w linii 31 pliku Lista.hh.

5.4.4.2 `template<typename TYP> Element<TYP>* Lista< TYP >::pierwszy` `[protected]`

Definicja w linii 30 pliku Lista.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Lista.hh](#)

5.5 Dokumentacja szablonu klasy Stos< TYP >

Klasa [Stos](#).

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

Diagram dziedziczenia dla Stos< TYP >

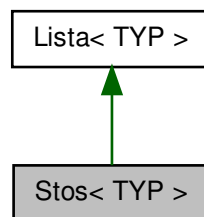
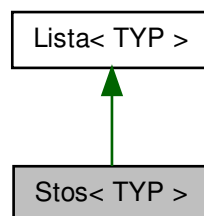


Diagram współpracy dla Stos< TYP >:



Metody publiczne

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

5.5.1 Opis szczegółowy

template<typename TYP> class Stos< TYP >

Klasa ta modeluje nam [Stos](#) Składa się z pól klasy [Lista](#) (początek) które zostanie użyta aby pokazywało na ostatni element stosu oraz metod PUSH, POP, SIZE, SHOW

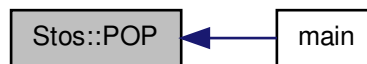
Definicja w linii 25 pliku Stos.hh.

5.5.2 Dokumentacja funkcji składowych

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

Definicja w linii 50 pliku Stos.hh.

Oto graf wywołań tej funkcji:



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

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

Definicja w linii 37 pliku Stos.hh.

Oto graf wywołań tej funkcji:



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

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

Definicja w linii 58 pliku Stos.hh.

Oto graf wywołań tej funkcji:

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

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

Definicja w linii 67 pliku Stos.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Stos.hh](#)

Rozdział 6

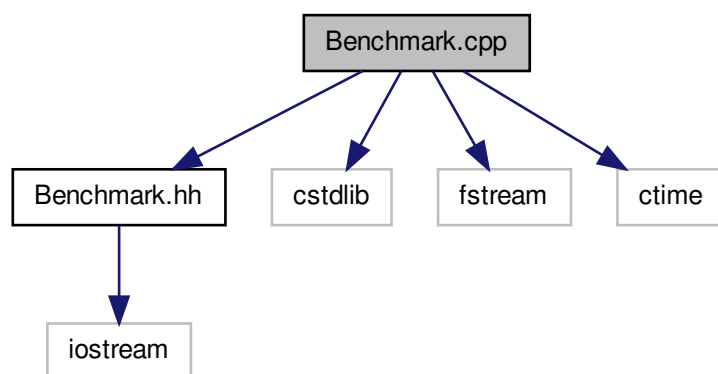
Dokumentacja plików

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](#)

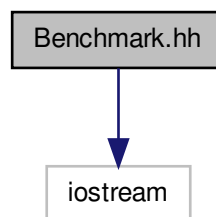
Definicja w pliku [Benchmark.cpp](#).

6.2 Dokumentacja pliku Benchmark.hh

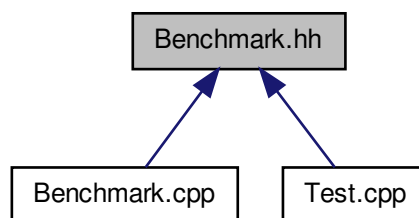
Definicja klasy [Benchmark](#).

```
#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

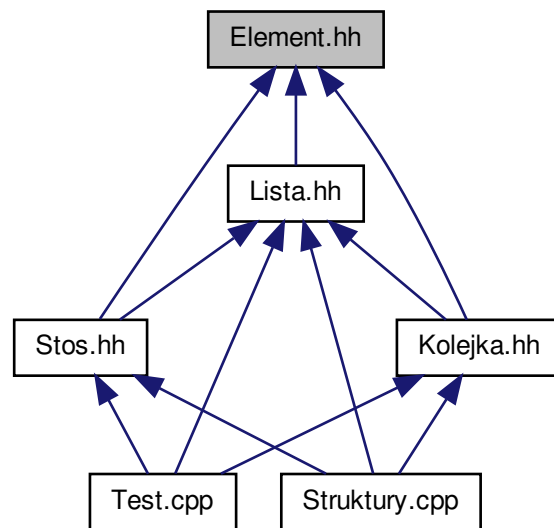
Plik zawiera definicje klasy `Benchmark` która będzie wyznaczała nam punkty do wyznaczenia złożoności obliczeniowej.

Definicja w pliku `Benchmark.hh`.

6.3 Dokumentacja pliku Element.hh

Definicja klasy `Element`.

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



Komponenty

- class `Element< TYP >`

Klasa [Element](#).

6.3.1 Opis szczegółowy

Plik zawiera definicje klasy [Element](#), która będzie pojedynczym elementem naszej struktury. Klasa ta posiada szablon, dzięki czemu możemy pracować na różnych typach danych

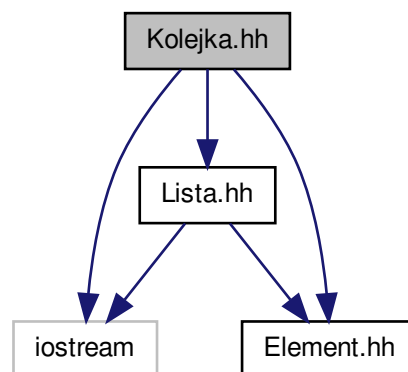
Definicja w pliku [Element.hh](#).

6.4 Dokumentacja pliku Kolejka.hh

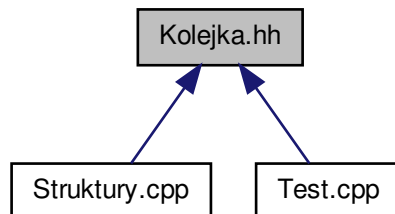
Definicja klasy [Kolejka](#).

```
#include <iostream>
#include "Element.hh"
#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](#).

6.4.1 Opis szczegółowy

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

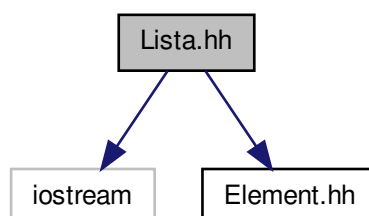
Definicja w pliku [Kolejka.hh](#).

6.5 Dokumentacja pliku Lista.hh

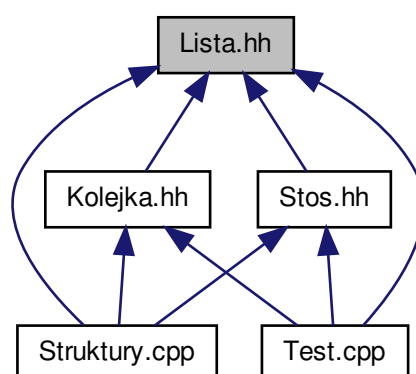
Definicja klasy [Lista](#).

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

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 strukturą naszych danych. Klasa ta posiada szablon, dzięki czemu możemy pracować na różnych typach danych

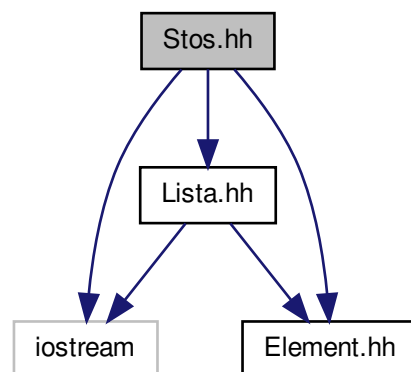
Definicja w pliku [Lista.hh](#).

6.6 Dokumentacja pliku Stos.hh

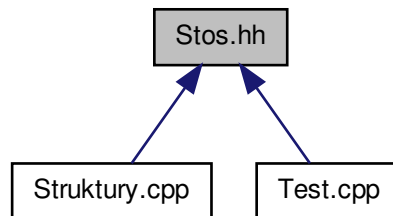
Definicja klasy [Stos](#).

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

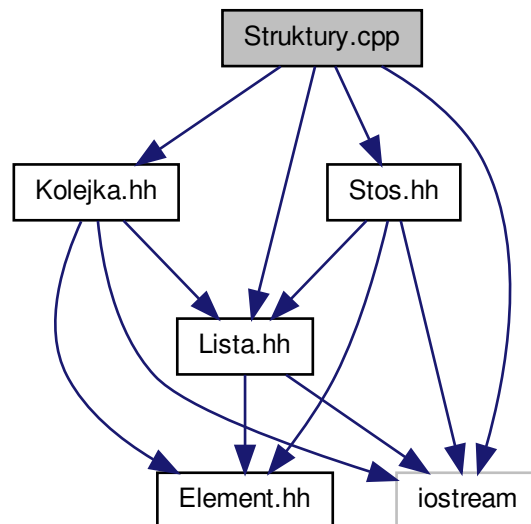
Definicja w pliku `Stos.hh`.

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

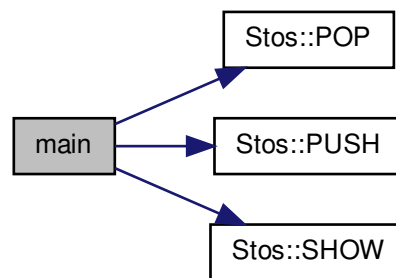
- int `main` ()

6.8.1 Dokumentacja funkcji

6.8.1.1 int `main` ()

Definicja w linii 8 pliku Struktury.cpp.

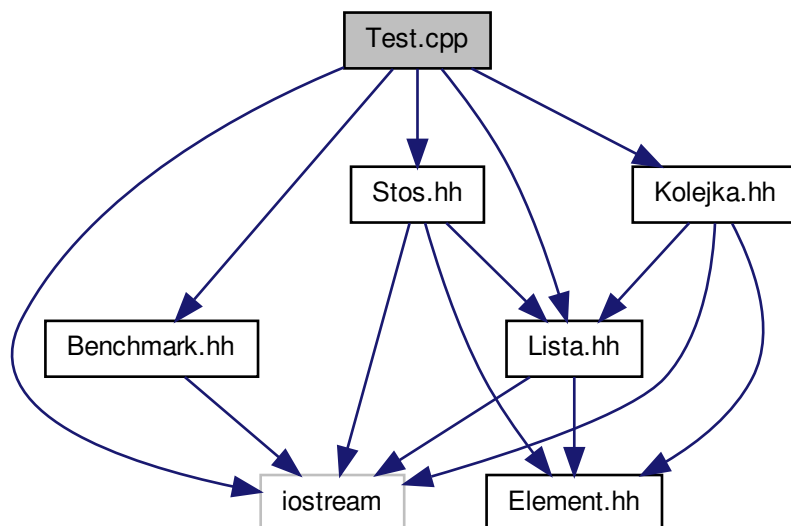
Oto graf wywołań dla tej funkcji:



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

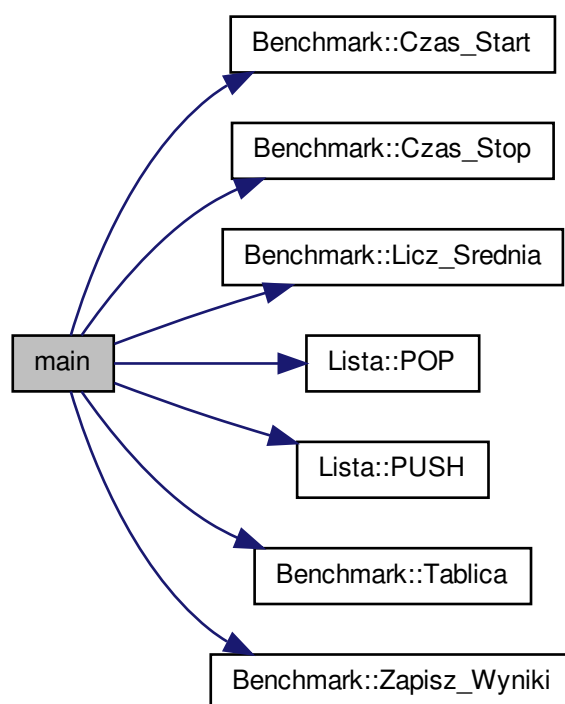
6.9.2.1 `double funkcja (double x)`

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](#)
- ~Element
 - Element, [15](#)
- ~Lista
 - Lista, [19](#)
- _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, [25](#)
- Benchmark.hh, [26](#)
- Czas_Start
 - Benchmark, [11](#)
- Czas_Stop
 - Benchmark, [11](#)
- Element, [15](#)
 - ~Element, [15](#)
 - Element, [15](#)
 - nastepny, [16](#)
 - wartosc, [16](#)
- Element.hh, [27](#)
- funkcja
 - Test.cpp, [35](#)
- Ilosc_Danych
 - Benchmark, [14](#)
- iterator
 - Benchmark, [14](#)
- iterator_sredniej
 - Benchmark, [14](#)
- Kolejka, [16](#)
 - POP, [17](#)
 - PUSH, [17](#)
 - SHOW, [17](#)
 - SIZE, [18](#)
- Kolejka.hh, [28](#)
- Licz_Srednia
 - Benchmark, [12](#)
- Lista, [18](#)
 - ~Lista, [19](#)
 - Lista, [19](#)
 - ostatni, [20](#)
 - pierwszy, [20](#)
 - POP, [19](#)
 - PUSH, [19](#)
 - SHOW, [20](#)
 - SIZE, [20](#)
- Lista.hh, [29](#)
- Losuj
 - Benchmark, [12](#)
- main
 - Struktury.cpp, [33](#)
 - Test.cpp, [35](#)
- nastepny
 - Element, [16](#)
- ostatni

Lista, [20](#)

pierwszy
Lista, [20](#)

POP
Kolejka, [17](#)
Lista, [19](#)
Stos, [22](#)

PUSH
Kolejka, [17](#)
Lista, [19](#)
Stos, [22](#)

rozmiar_tablic
Benchmark, [14](#)

SHOW
Kolejka, [17](#)
Lista, [20](#)
Stos, [22](#)

SIZE
Kolejka, [18](#)
Lista, [20](#)
Stos, [23](#)

srednia_jednego_problemu
Benchmark, [14](#)

STALA
Test.cpp, [35](#)

stoper_start
Benchmark, [14](#)

stoper_stop
Benchmark, [14](#)

Stos, [21](#)
POP, [22](#)
PUSH, [22](#)
SHOW, [22](#)
SIZE, [23](#)

Stos.hh, [31](#)

strona.dox, [32](#)

Struktury.cpp, [32](#)
main, [33](#)

Tablica
Benchmark, [12](#)

Test.cpp, [34](#)
funkcja, [35](#)
main, [35](#)
STALA, [35](#)

wartosc
Element, [16](#)

Wczytaj_Dane
Benchmark, [13](#)

wielkosc_problemu
Benchmark, [14](#)

Zapisz_Wyniki
Benchmark, [13](#)