

Zadanie 5

Generated by Doxygen 1.8.9.1

Tue Apr 21 2015 21:46:18

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	ArrayLista Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	5
3.1.2.1	ArrayLista	6
3.1.2.2	~ArrayLista	6
3.1.3	Member Function Documentation	6
3.1.3.1	ile_elementow	6
3.1.3.2	pop	6
3.1.3.3	push	6
3.1.3.4	push	6
3.1.3.5	size	6
3.2	Benchmarker Class Reference	7
3.2.1	Detailed Description	7
3.2.2	Member Function Documentation	7
3.2.2.1	generujdane	7
3.2.2.2	generujdane_string	7
3.2.2.3	testuj	8
3.3	HaszTab Class Reference	8
3.3.1	Detailed Description	9
3.3.2	Constructor & Destructor Documentation	9
3.3.2.1	HaszTab	9
3.3.2.2	~HaszTab	9
3.3.3	Member Function Documentation	9
3.3.3.1	mieszaj	9
3.3.3.2	odczytaj	9

3.3.3.3	push	10
3.3.3.4	size_k1	10
3.3.3.5	size_k2	10
3.4	Kolejka Class Reference	10
3.4.1	Detailed Description	11
3.4.2	Constructor & Destructor Documentation	11
3.4.2.1	Kolejka	11
3.4.2.2	~Kolejka	11
3.4.3	Member Function Documentation	11
3.4.3.1	pop	11
3.4.3.2	push	11
3.4.3.3	size	11
3.5	Lista Class Reference	12
3.5.1	Detailed Description	12
3.5.2	Constructor & Destructor Documentation	12
3.5.2.1	Lista	12
3.5.2.2	~Lista	12
3.5.3	Member Function Documentation	12
3.5.3.1	pop	12
3.5.3.2	pop	13
3.5.3.3	push	13
3.5.3.4	push	13
3.5.3.5	size	13
3.6	Stos Class Reference	13
3.6.1	Detailed Description	14
3.6.2	Constructor & Destructor Documentation	14
3.6.2.1	Stos	14
3.6.2.2	~Stos	14
3.6.3	Member Function Documentation	14
3.6.3.1	pop	14
3.6.3.2	push	14
3.6.3.3	size	14
4	File Documentation	15
4.1	ArrayLista.cpp File Reference	15
4.1.1	Detailed Description	15
4.2	ArrayLista.hh File Reference	15
4.2.1	Detailed Description	16
4.3	Benchmark.cpp File Reference	16
4.3.1	Detailed Description	17

4.3.2	Macro Definition Documentation	17
4.3.2.1	BENCHMARK_CPP	17
4.4	Benchmark.hh File Reference	17
4.4.1	Detailed Description	18
4.5	HaszTab.cpp File Reference	18
4.5.1	Detailed Description	19
4.6	HaszTab.hh File Reference	19
4.6.1	Detailed Description	20
4.7	Kolejka.cpp File Reference	20
4.7.1	Detailed Description	20
4.8	Kolejka.hh File Reference	21
4.8.1	Detailed Description	21
4.9	Lista.cpp File Reference	21
4.9.1	Detailed Description	22
4.10	Lista.hh File Reference	22
4.10.1	Detailed Description	22
4.11	main.cpp File Reference	23
4.11.1	Detailed Description	23
4.11.2	Function Documentation	23
4.11.2.1	main	23
4.12	Stos.cpp File Reference	24
4.12.1	Detailed Description	24
4.13	Stos.hh File Reference	24
4.13.1	Detailed Description	25
Index		27

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

ArrayLista	Klasa ArrayLista	5
Benchmark	Klasa Benchmark	7
HaszTab	Klasa HaszTab	8
Kolejka	Klasa Kolejka	10
Lista	Klasa Lista	12
Stos	Klasa Stos	13

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

ArrayLista.cpp		
Metody klasy ArrayLista	15
ArrayLista.hh		
Definicja klasy ArrayLista	15
Benchmark.cpp		
Metody klasy Benchmark	16
Benchmark.hh		
Definicja klasy Benchmark	17
HaszTab.cpp		
Metody klasy HaszTab	18
HaszTab.hh		
Definicja klasy HaszTab	19
Kolejka.cpp		
Metody klasy Kolejka	20
Kolejka.hh		
Definicja klasy Kolejka	21
Lista.cpp		
Metody klasy Lista	21
Lista.hh		
Definicja klasy Lista	22
main.cpp		
Modul glowny	23
Stos.cpp		
Metody klasy Stos	24
Stos.hh		
Definicja klasy Stos	24

Chapter 3

Class Documentation

3.1 ArrayLista Class Reference

Klasa [ArrayLista](#).

```
#include <ArrayLista.hh>
```

Public Member Functions

- [ArrayLista](#) ()

Konstruktor bezparametryczny. Konstruktor inicjalizujący tablice listy. rozmiar - rozmiar tablicy dla listy, inicjalizowane wartoscia 1. ilosc_elementow - ilosc elementow listy, inicjalizowane wartoscia 0. Tworzona jest tablica dynamiczna o jednym elemencie.

- [~ArrayLista](#) ()

Destruktor bezparametryczny stosu.

- void [push](#) (int, int)

Metoda umieszczająca element na liście. Metoda inkrementuje rozmiar podczas umieszczania elementu na liście.

- void [push](#) (int wartosc)

Przeciążenie operacji push. Powiększa listę podczas dodawania elementu do 200%. Następuje inkrementacja rozmiar listy.

- int [pop](#) ()

Metoda zdejmująca element z listy. Metoda dekrementuje ilosc_elementow przy zdejmowaniu z listy. Tablica listy jest zmniejszana podczas zdejmowania elementu o połowę gdy ilosc elementow znajdujących się na niej jest równa połowie jej rozmiaru.

- int [size](#) ()

Metoda zwracająca rozmiar tablicy na której oparta jest lista.

- int [ile_elementow](#) ()

Metoda zwracająca ilosc elementow jaka znajduje się na liście.

3.1.1 Detailed Description

Klasa [ArrayLista](#).

3.1.2 Constructor & Destructor Documentation

3.1.2.1 ArrayLista::ArrayLista ()

Konstruktor bezparametryczny. Konstruktor inicjalizujący tablice listy. rozmiar - rozmiar tablicy dla listy, inicjalizowane wartością 1. ilość_elementów - ilość elementów listy, inicjalizowane wartością 0. Tworzona jest tablica dynamiczna o jednym elemencie.

3.1.2.2 ArrayLista::~~ArrayLista ()

Destruktor bezparametryczny stosu.

3.1.3 Member Function Documentation

3.1.3.1 int ArrayLista::ile_elementow ()

Metoda zwracająca ilość elementów jaka znajduje się na liście.

Returns

ileść_elementów - typu int, ilość elementów jaka znajduje się na liście.

3.1.3.2 int ArrayLista::pop ()

Metoda zdejmująca element z listy. Metoda dekrementuje ilość_elementów przy zdejmowaniu z listy. Tablica listy jest zmniejszana podczas zdejmowania elementu o połowę gdy ilość elementów znajdujących się na niej jest równa połowie jej rozmiaru.

Returns

wartość - typu int, wartość zdejmowana ze stosu.

3.1.3.3 void ArrayLista::push (int wartość, int zwiększanie)

Metoda umieszczająca element na liście Metoda inkrementuje rozmiar podczas umieszczania elementu na liście.

Parameters

<i>zwiększanie</i>	- typu int, mnożnik rozszerzania tablicy podczas dodawania elementów listy .
<i>wartość</i>	- typu int, wartość umieszczana na stosie.

3.1.3.4 void ArrayLista::push (int wartość) [inline]

Przeciążenie operacji push. Powiększa listę podczas dodawania elementu do 200%. Następuje inkrementacja rozmiaru listy.

Parameters

<i>wartość</i>	- typu int, wartość umieszczana na liście.
----------------	--

3.1.3.5 int ArrayLista::size ()

Metoda zwracająca rozmiar tablicy na której oparta jest lista.

Returns

rozmiar - typu int, rozmiar tablicy listy.

The documentation for this class was generated from the following files:

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

3.2 Benchmarker Class Reference

Klasa [Benchmarker](#).

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

Public Member Functions

- `template<class T, typename T2>`
`long int testuj (T *, T2 *, int, int)`
Szablon metody przeprowadzająca sprawdzenie czasu działania funkcji. Typy: [Lista](#), [Stos](#), [Kolejka](#), Tablica haszująca.
- `int * generuj dane (int l_danych)`
Metoda generująca wartości losowe z przedziału 0 10000.
- `string * generuj dane_string (int l_danych)`
Metoda generująca wartości losowe z przedziału A-Z + a-z.

3.2.1 Detailed Description

Klasa [Benchmarker](#).

3.2.2 Member Function Documentation

3.2.2.1 `int * Benchmarker::generuj dane (int l_danych)`

Metoda generująca wartości losowe z przedziału 0 10000.

Parameters

<code>l_danych</code>	- typu int, liczba generowanych danych.
-----------------------	---

Returns

`dane` - typu `long int`, tablica z wygenerowanymi danymi.

3.2.2.2 `string * Benchmarker::generuj dane_string (int l_danych)`

Metoda generująca wartości losowe z przedziału A-Z + a-z.

Parameters

<code>l_danych</code>	- typu int, liczba generowanych danych.
-----------------------	---

Returns

`dane` - typu *string*, tablica z wygenerowanymi danymi(np.: Axb CtQ).

3.2.2.3 `template<class T, typename T2 > template long int Benchmarker::testuj< HaszTab, string > (T *, T2 *, int, int)`

Szablon metody przeprowadzająca sprawdzenie czasu działania funkcji. Typy: [Lista](#), [Stos](#), [Kolejka](#), Tablica haszująca.

Template Parameters

<code>Tab</code>	- typu T*, wskaznik na zaimplementowany stos/liste/kolejke/tablice haszującą.
<code>dane</code>	- typu int*, wskaznik na tablice z danymi generowanymi.
<code>liczba_przejsc</code>	- typu int, liczba przejsc przez dane.
<code>liczba_danych</code>	- typu int, liczba danych w tablicy.

Returns

`czas_calkowity_usredniony` - typu long int, czas sredni działania funkcji.

The documentation for this class was generated from the following files:

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

3.3 HaszTab Class Reference

Klasa [HaszTab](#).

```
#include <HaszTab.hh>
```

Public Member Functions

- [HaszTab](#) ()
Konstruktor bezparametryczny. Konstruktor inicjalizujący tablice haszującą. rozmiar_k1 - wielkosc tablicy pierwszej. rozmiar_k2 - wielkosc tablicy drugiej. Tworzona jest macierz elementow (string wartosc,string klucz) o zadanej wielkości.
- [~HaszTab](#) ()
Destruktor bezparametryczny tablicy haszującej.
- void [push](#) (string wartosc, string klucz)
Przeciążenie operacji push. Metoda umieszczająca element o zadanej wartości i kluczu na tablicy haszującej.
- int [mieszaj](#) (string klucz_umieszczany, int modulacja)
Metoda mieszająca klucz tablicy haszującej. Metoda zmienia zadany klucz na indeksy tablic.
- string [odczytaj](#) (string klucz_szukany)
Metoda czytująca element z tablicy haszującej. Metoda czytuje element o określonym kluczu. Wartości oraz klucze nie są usuwane podczas czytowania.
- int [size_k1](#) ()
Metoda zwracająca rozmiar tablicy pierwszej na ktorej oparta jest tablica haszująca.
- int [size_k2](#) ()
Metoda zwracająca rozmiar tablicy drugiej na ktorej oparta jest tablica haszująca.

3.3.1 Detailed Description

Klasa [HaszTab](#).

3.3.2 Constructor & Destructor Documentation

3.3.2.1 HaszTab::HaszTab ()

Konstruktor bezparametryczny. Konstruktor inicjalizujący tablice haszującą. rozmiar_k1 - wielkość tablicy pierwszej. rozmiar_k2 - wielkość tablicy drugiej. Tworzona jest macierz elementów (string wartość,string klucz) o zadanej wielkości.

3.3.2.2 HaszTab::~HaszTab ()

Destruktor bezparametryczny tablicy haszującej.

3.3.3 Member Function Documentation

3.3.3.1 int HaszTab::mieszaj (string *klucz_umieszczany*, int *modulacja*)

Metoda mieszająca klucz tablicy haszującej. Metoda zmienia zadany klucz na indeksy tablic.

Parameters

<i>klucz_↔ umieszczany</i>	- typu string, zadany klucz.
<i>modulacja</i>	- typu int, wielkość tablicy która moduluje klucz.

Returns

indeks - typu int, zmodulowany indeks elementu.

3.3.3.2 string HaszTab::odczytaj (string *klucz_szukany*)

Metoda zczytująca element z tablicy haszującej. Metoda zczytuje element o określonym kluczu. Wartości oraz klucze nie są usuwane podczas zczytywania.

Parameters

<i>klucz_szukany</i>	- typu string, szukany klucz.
----------------------	-------------------------------

Returns

wartosc - typu int, wartosc zdejmowana ze stosu.

Here is the call graph for this function:



3.3.3.3 void HaszTab::push (string wartosc, string klucz)

Przeciążenie operacji push. Metoda umieszczająca element o zadanej wartości i kluczu na tablicy haszującej.

Parameters

<i>wartosc</i>	- typu string, zadana wartość.
<i>klucz</i>	- typu string, zadany klucz.

Here is the call graph for this function:



3.3.3.4 int HaszTab::size_k1 ()

Metoda zwracająca rozmiar tablicy pierwszej na której oparta jest tablica haszująca.

Returns

rozmiar - typu int, rozmiar tablicy pierwszej.

3.3.3.5 int HaszTab::size_k2 ()

Metoda zwracająca rozmiar tablicy drugiej na której oparta jest tablica haszująca.

Returns

rozmiar - typu int, rozmiar tablicy drugiej.

The documentation for this class was generated from the following files:

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

3.4 Kolejka Class Reference

Klasa [Kolejka](#).

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

Public Member Functions

- [Kolejka](#) ()
Konstruktor bezparametryczny. Konstruktor inicjalizujący strażnika_początek i strażnika_koniec kolejki wartościami NULL, oraz rozmiar kolejki wartością 0.
- [~Kolejka](#) ()

Destruktor bezparametryczny kolejki.

- void [push](#) (int)
Metoda umieszczajaca element na koncu kolejki. Metoda inkrementuje rozmiar podczas umieszczania elementu w kolejce.
- int [pop](#) ()
Metoda zdejmujaca element z poczatku kolejki. Metoda dekrementuje rozmiar przy zdejmowaniu elementu.
- int [size](#) ()
Metoda zwracajaca wielkosc kolejki.

3.4.1 Detailed Description

Klasa [Kolejka](#).

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Kolejka::Kolejka ()

Konstruktor bezparametryczny. Konstruktor inicjalizujacy straznika_poczatek i straznika_koniec kolejki wartosciami NULL , oraz rozmiar kolejki wartoscia 0.

3.4.2.2 Kolejka::~~Kolejka ()

Destruktor bezparametryczny kolejki.

3.4.3 Member Function Documentation

3.4.3.1 int Kolejka::pop ()

Metoda zdejmujaca element z poczatku kolejki. Metoda dekrementuje rozmiar przy zdejmowaniu elementu.

Returns

wartosc - typu int, wartosc zdejmowana z kolejki.

3.4.3.2 void Kolejka::push (int wartosc)

Metoda umieszczajaca element na koncu kolejki. Metoda inkrementuje rozmiar podczas umieszczania elementu w kolejce.

Parameters

<i>wartosc</i>	- typu int, wartosc umieszczana na koncu kolejki.
----------------	---

3.4.3.3 int Kolejka::size ()

Metoda zwracajaca wielkosc kolejki.

Returns

rozmiar - typu int,rozmiar kolejki.

The documentation for this class was generated from the following files:

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

3.5 Lista Class Reference

Klasa [Lista](#).

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

Public Member Functions

- [Lista](#) ()
Konstruktor bezparametryczny. Konstruktor inicjalizujący strażnika listy wartością NULL oraz rozmiar wartością 0.
- [~Lista](#) ()
Destruktor bezparametryczny listy.
- void [push](#) (int, int)
Metoda umieszczająca element określonej pozycji na liście <0,rozmiar>. Metoda inkrementuje rozmiar podczas umieszczania elementu na liście.
- int [pop](#) (int)
Metoda zdejmująca element z określonej pozycji listy <0,rozmiar>. Metoda dekrementuje rozmiar przy zdejmowaniu elementu.
- void [push](#) (int wartosc)
Przeciążenie operacji push. Umieszcza element domyślnie na pozycji 1. Następuje inkrementacja rozmiaru listy.
- int [pop](#) ()
Przeciążenie operacji pop dla listy. Pobiera domyślnie element listy z pozycji 1. Następuje dekrementacja rozmiaru listy.
- int [size](#) ()
Metoda zwracająca wielkość listy.

3.5.1 Detailed Description

Klasa [Lista](#).

3.5.2 Constructor & Destructor Documentation

3.5.2.1 [Lista::Lista](#) ()

Konstruktor bezparametryczny. Konstruktor inicjalizujący strażnika listy wartością NULL oraz rozmiar wartością 0.

3.5.2.2 [Lista::~~Lista](#) ()

Destruktor bezparametryczny listy.

3.5.3 Member Function Documentation

3.5.3.1 [int Lista::pop](#) (int *pozycja*)

Metoda zdejmująca element z określonej pozycji listy <0,rozmiar>. Metoda dekrementuje rozmiar przy zdejmowaniu elementu.

Parameters

<i>pozycja</i>	- typu int, numer elementu który ma być zdjęty z listy.
----------------	---

Returns

wartosc - typu int, wartosc zdejmowana z listy.

3.5.3.2 `int Lista::pop () [inline]`

Przeciazenie operacji pop dla listy. Pobiera domyslnie element listy z pozycji 1. Nastepuje dekrementacja rozmiar listy.

Returns

wartosc - typu int, wartosc zdejmowana z listy.

3.5.3.3 `void Lista::push (int wartosc, int pozycja)`

Metoda umieszczajaca element okreslonej pozycji na liscie <0,rozmiar>. Metoda inkrementuje rozmiar podczas umieszczania elementu na liscie.

Parameters

<i>wartosc</i>	- typu int, wartosc umieszczana na liscie.
<i>pozycja</i>	- typu int, pozycja na ktorej jest umieszczana wartosc.

3.5.3.4 `void Lista::push (int wartosc) [inline]`

Przeciazenie operacji push. Umieszcza element domyslnie na pozycji 1. Nastepuje inkrementacja rozmiar listy.

Parameters

<i>wartosc</i>	- typu int, wartosc umieszczana na liscie.
----------------	--

3.5.3.5 `int Lista::size ()`

Metoda zwracajaca wielkosc listy.

Returns

rozmiar - typu int, rozmiar listy.

The documentation for this class was generated from the following files:

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

3.6 Stos Class Reference

Klasa [Stos](#).

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

Public Member Functions

- [Stos](#) ()
Konstruktor bezparametryczny. Konstruktor inicjalizujacy straznika stosu wartoscia NULL ,oraz rozmiar wartoscia 0.
- [~Stos](#) ()
Destruktor bezparametryczny stosu.
- void [push](#) (int)

Metoda umieszczajaca element na stosie Metoda inkrementuje rozmiar podczas umieszczania elementu na stosie.

- `int pop()`

Metoda zdejmujaca element ze stosu. Metoda dekrementuje rozmiar przy zdejmowaniu ze stosu.

- `int size()`

Metoda zwracajaca wielkosc stosu.

3.6.1 Detailed Description

Klasa [Stos](#).

3.6.2 Constructor & Destructor Documentation

3.6.2.1 `Stos::Stos()`

Konstruktor bezparametryczny. Konstruktor inicjalizujący strażnika stosu wartością NULL, oraz rozmiar wartością 0.

3.6.2.2 `Stos::~Stos()`

Destruktor bezparametryczny stosu.

3.6.3 Member Function Documentation

3.6.3.1 `int Stos::pop()`

Metoda zdejmująca element ze stosu. Metoda dekrementuje rozmiar przy zdejmowaniu ze stosu.

Returns

wartosc - typu int, wartosc zdejmowana ze stosu.

3.6.3.2 `void Stos::push(int wartosc)`

Metoda umieszczająca element na stosie Metoda inkrementuje rozmiar podczas umieszczania elementu na stosie.

Parameters

<i>wartosc</i>	- typu int, wartosc umieszczana na stosie.
----------------	--

3.6.3.3 `int Stos::size()`

Metoda zwracająca wielkość stosu.

Returns

rozmiar - typu int, rozmiar stosu.

The documentation for this class was generated from the following files:

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

Chapter 4

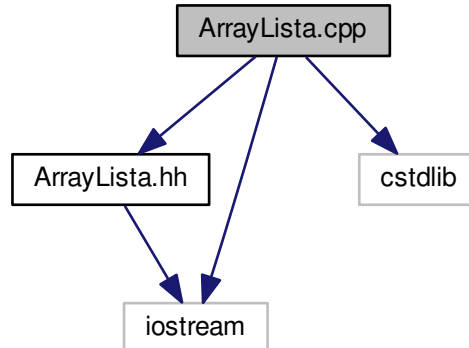
File Documentation

4.1 ArrayList.cpp File Reference

Metody klasy [ArrayList](#).

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

Include dependency graph for ArrayList.cpp:



4.1.1 Detailed Description

Metody klasy [ArrayList](#).

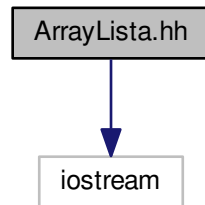
Plik zawiera metody klasy [ArrayList](#).

4.2 ArrayList.hh File Reference

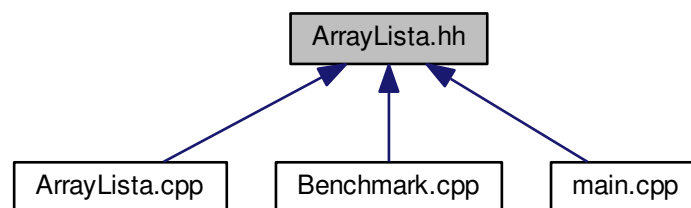
Definicja klasy [ArrayList](#).

```
#include <iostream>
```

Include dependency graph for ArrayLista.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class [ArrayLista](#)

Klasa [ArrayLista](#).

4.2.1 Detailed Description

Definicja klasy [ArrayLista](#).

Plik zawiera definicje klasy modulujacej pojecie listy jednokierunkowej opartej na tablicy dynamicznej.

4.3 Benchmark.cpp File Reference

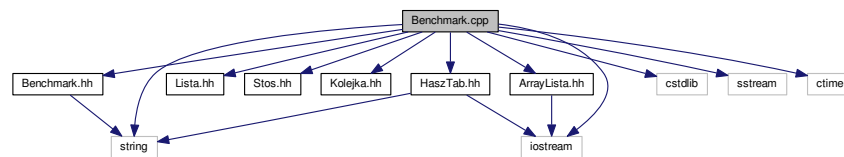
Metody klasy [Benchmark](#).

```

#include "Benchmark.hh"
#include "Lista.hh"
#include "Stos.hh"
#include "Kolejka.hh"
#include "ArrayLista.hh"
#include "HaszTab.hh"
#include <cstdlib>
#include <iostream>
#include <sstream>
#include <string>
#include <ctime>

```

Include dependency graph for Benchmark.cpp:



Macros

- `#define BENCHMARK_CPP`

4.3.1 Detailed Description

Metody klasy [Benchmark](#).

Plik zawiera metody klasy [Benchmark](#).

4.3.2 Macro Definition Documentation

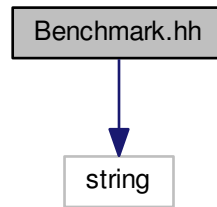
4.3.2.1 #define BENCHMARK_CPP

4.4 Benchmark.hh File Reference

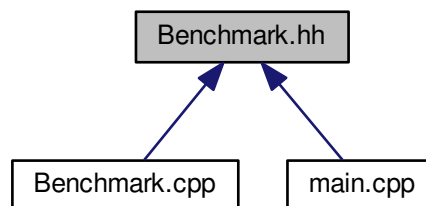
Definicja klasy [Benchmark](#).

```
#include <string>
```

Include dependency graph for Benchmark.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class [Benchmark](#)

Klasa [Benchmark](#).

4.4.1 Detailed Description

Definicja klasy [Benchmark](#).

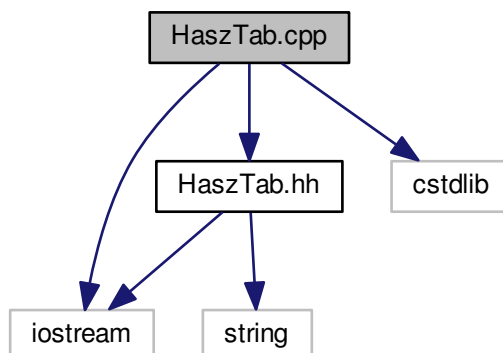
Plik zawiera definicje klasy [Benchmark](#).

4.5 HaszTab.cpp File Reference

Metody klasy [HaszTab](#).

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


Include dependency graph for HaszTab.cpp:



4.5.1 Detailed Description

Metody klasy [HaszTab](#).

Plik zawiera metody klasy [HaszTab](#).

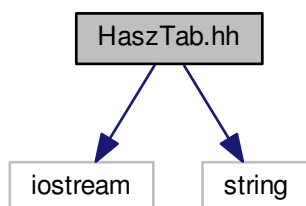
4.6 HaszTab.hh File Reference

Definicja klasy [HaszTab](#).

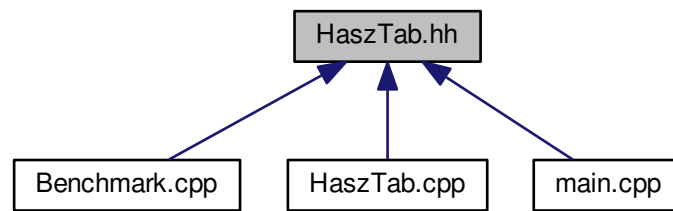
```
#include <iostream>
```

```
#include <string>
```

Include dependency graph for HaszTab.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class [HaszTab](#)
Klasa [HaszTab](#).

4.6.1 Detailed Description

Definicja klasy [HaszTab](#).

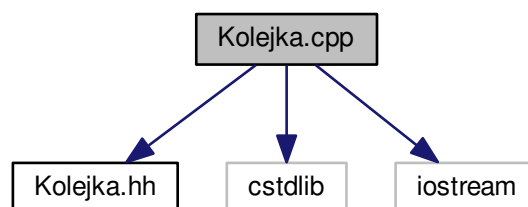
Plik zawiera definicje klasy modulujacej pojecie tablicy haszujacej.

4.7 Kolejka.cpp File Reference

Metody klasy [Kolejka](#).

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

Include dependency graph for `Kolejka.cpp`:



4.7.1 Detailed Description

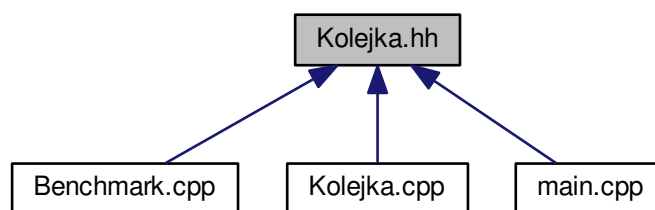
Metody klasy [Kolejka](#).

Plik zawiera metody klasy [Kolejka](#).

4.8 Kolejka.hh File Reference

Definicja klasy [Kolejka](#).

This graph shows which files directly or indirectly include this file:



Classes

- class [Kolejka](#)

Klasa [Kolejka](#).

4.8.1 Detailed Description

Definicja klasy [Kolejka](#).

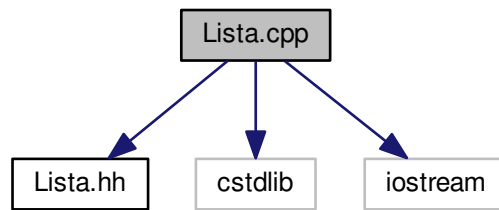
Plik zawiera definicje klasy [Kolejka](#).

4.9 Lista.cpp File Reference

Metody klasy [Lista](#).

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

Include dependency graph for Lista.cpp:



4.9.1 Detailed Description

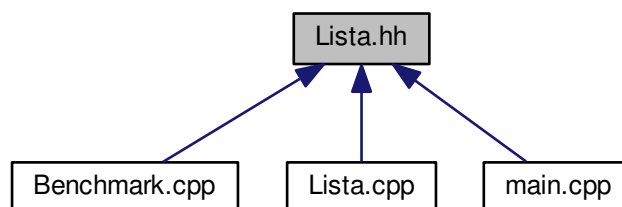
Metody klasy [Lista](#).

Plik zawiera metody klasy [Lista](#).

4.10 Lista.hh File Reference

Definicja klasy [Lista](#).

This graph shows which files directly or indirectly include this file:



Classes

- class [Lista](#)
Klasa [Lista](#).

4.10.1 Detailed Description

Definicja klasy [Lista](#).

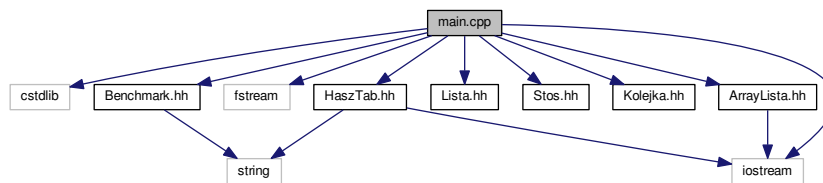
Plik zawiera definicje klasy modulujacej pojecie listy jednokierunkowej.

4.11 main.cpp File Reference

Modul glowny.

```
#include <cstdlib>
#include <iostream>
#include <fstream>
#include "Benchmark.hh"
#include "Lista.hh"
#include "Stos.hh"
#include "Kolejka.hh"
#include "ArrayLista.hh"
#include "HaszTab.hh"
```

Include dependency graph for main.cpp:



Functions

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

Funkcja glowna programu.

4.11.1 Detailed Description

Modul glowny.

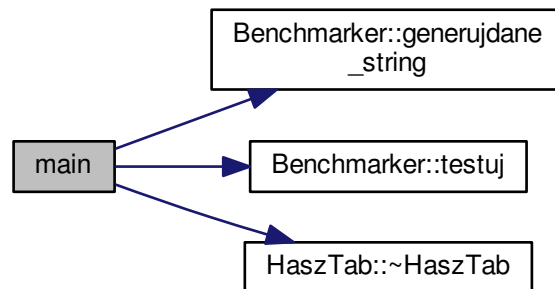
Plik zawiera funkcje main.

4.11.2 Function Documentation

4.11.2.1 int main (int argc, char * argv[])

Funkcja glowna programu.

Here is the call graph for this function:

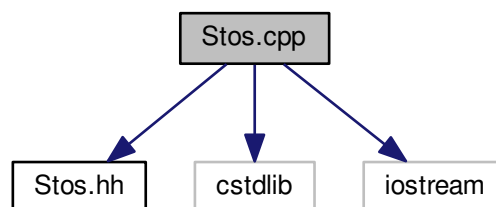


4.12 Stos.cpp File Reference

Metody klasy [Stos](#).

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

Include dependency graph for `Stos.cpp`:



4.12.1 Detailed Description

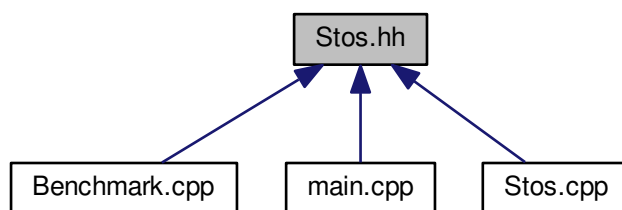
Metody klasy [Stos](#).

Plik zawiera metody klasy [Stos](#).

4.13 Stos.hh File Reference

Definicja klasy [Stos](#).

This graph shows which files directly or indirectly include this file:



Classes

- class [Stos](#)
Klasa [Stos](#).

4.13.1 Detailed Description

Definicja klasy [Stos](#).

Plik zawiera definicje klasy [Stos](#).

Index

- ~ArrayList
 - ArrayList, [6](#)
- ~HaszTab
 - HaszTab, [9](#)
- ~Kolejka
 - Kolejka, [11](#)
- ~Lista
 - Lista, [12](#)
- ~Stos
 - Stos, [14](#)
- ArrayList, [5](#)
 - ~ArrayList, [6](#)
 - ArrayList, [5](#)
 - ile_elementow, [6](#)
 - pop, [6](#)
 - push, [6](#)
 - size, [6](#)
- ArrayList.cpp, [15](#)
- ArrayList.hh, [15](#)
- BENCHMARK_CPP
 - Benchmark.cpp, [17](#)
- Benchmark.cpp, [16](#)
 - BENCHMARK_CPP, [17](#)
- Benchmark.hh, [17](#)
- Benchmarker, [7](#)
 - generujdane, [7](#)
 - generujdane_string, [7](#)
 - testuj, [8](#)
- generujdane
 - Benchmarker, [7](#)
- generujdane_string
 - Benchmarker, [7](#)
- HaszTab, [8](#)
 - ~HaszTab, [9](#)
 - HaszTab, [9](#)
 - mieszaj, [9](#)
 - odczytaj, [9](#)
 - push, [9](#)
 - size_k1, [10](#)
 - size_k2, [10](#)
- HaszTab.cpp, [18](#)
- HaszTab.hh, [19](#)
- ile_elementow
 - ArrayList, [6](#)
- Kolejka, [10](#)
 - ~Kolejka, [11](#)
 - Kolejka, [11](#)
 - pop, [11](#)
 - push, [11](#)
 - size, [11](#)
- Kolejka.cpp, [20](#)
- Kolejka.hh, [21](#)
- Lista, [12](#)
 - ~Lista, [12](#)
 - Lista, [12](#)
 - pop, [12](#)
 - push, [13](#)
 - size, [13](#)
- Lista.cpp, [21](#)
- Lista.hh, [22](#)
- main
 - main.cpp, [23](#)
- main.cpp, [23](#)
 - main, [23](#)
- mieszaj
 - HaszTab, [9](#)
- odczytaj
 - HaszTab, [9](#)
- pop
 - ArrayList, [6](#)
 - Kolejka, [11](#)
 - Lista, [12](#)
 - Stos, [14](#)
- push
 - ArrayList, [6](#)
 - HaszTab, [9](#)
 - Kolejka, [11](#)
 - Lista, [13](#)
 - Stos, [14](#)
- size
 - ArrayList, [6](#)
 - Kolejka, [11](#)
 - Lista, [13](#)
 - Stos, [14](#)
- size_k1
 - HaszTab, [10](#)
- size_k2
 - HaszTab, [10](#)
- Stos, [13](#)
 - ~Stos, [14](#)
 - pop, [14](#)

- push, [14](#)
 - size, [14](#)
 - Stos, [14](#)
- Stos.cpp, [24](#)
- Stos.hh, [24](#)
- testuj
 - Benchmark, [8](#)