

## Zadanie 6

Generated by Doxygen 1.8.9.1

Sun May 3 2015 20:11:20



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	ArrayLista Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	6
3.1.2.1	ArrayLista	6
3.1.2.2	ArrayLista	6
3.1.2.3	~ArrayLista	6
3.1.3	Member Function Documentation	6
3.1.3.1	ile_elementow	6
3.1.3.2	mergesort	6
3.1.3.3	pop	6
3.1.3.4	push	7
3.1.3.5	push	7
3.1.3.6	scal	7
3.1.3.7	size	7
3.2	Benchmarker Class Reference	7
3.2.1	Detailed Description	8
3.2.2	Member Function Documentation	8
3.2.2.1	generujdane	8
3.2.2.2	generujdane_string	8
3.2.2.3	testuj	8
3.3	HaszTab Class Reference	9
3.3.1	Detailed Description	9
3.3.2	Constructor & Destructor Documentation	9
3.3.2.1	HaszTab	9
3.3.2.2	~HaszTab	10

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

4.2.1	Detailed Description	18
4.3	Benchmark.cpp File Reference	18
4.3.1	Detailed Description	19
4.3.2	Macro Definition Documentation	19
4.3.2.1	BENCHMARK_CPP	19
4.4	Benchmark.hh File Reference	19
4.4.1	Detailed Description	20
4.5	HaszTab.cpp File Reference	20
4.5.1	Detailed Description	21
4.6	HaszTab.hh File Reference	21
4.6.1	Detailed Description	22
4.7	Kolejka.cpp File Reference	22
4.7.1	Detailed Description	22
4.8	Kolejka.hh File Reference	23
4.8.1	Detailed Description	23
4.9	Lista.cpp File Reference	23
4.9.1	Detailed Description	24
4.10	Lista.hh File Reference	24
4.10.1	Detailed Description	24
4.11	main.cpp File Reference	25
4.11.1	Detailed Description	25
4.11.2	Function Documentation	25
4.11.2.1	main	25
4.12	Stos.cpp File Reference	26
4.12.1	Detailed Description	26
4.13	Stos.hh File Reference	26
4.13.1	Detailed Description	27
<b>Index</b>		<b>29</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">ArrayLista</a>	Klasa <a href="#">ArrayLista</a> . . . . .	5
<a href="#">Benchmarker</a>	Klasa <a href="#">Benchmarker</a> . . . . .	7
<a href="#">HaszTab</a>	Klasa <a href="#">HaszTab</a> . . . . .	9
<a href="#">Kolejka</a>	Klasa <a href="#">Kolejka</a> . . . . .	11
<a href="#">Lista</a>	Klasa <a href="#">Lista</a> . . . . .	12
<a href="#">Stos</a>	Klasa <a href="#">Stos</a> . . . . .	14





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">ArrayLista.cpp</a>		
Metody klasy <a href="#">ArrayLista</a>	. . . . .	17
<a href="#">ArrayLista.hh</a>		
Definicja klasy <a href="#">ArrayLista</a>	. . . . .	17
<a href="#">Benchmark.cpp</a>		
Metody klasy <a href="#">Benchmark</a>	. . . . .	18
<a href="#">Benchmark.hh</a>		
Definicja klasy <a href="#">Benchmark</a>	. . . . .	19
<a href="#">HaszTab.cpp</a>		
Metody klasy <a href="#">HaszTab</a>	. . . . .	20
<a href="#">HaszTab.hh</a>		
Definicja klasy <a href="#">HaszTab</a>	. . . . .	21
<a href="#">Kolejka.cpp</a>		
Metody klasy <a href="#">Kolejka</a>	. . . . .	22
<a href="#">Kolejka.hh</a>		
Definicja klasy <a href="#">Kolejka</a>	. . . . .	23
<a href="#">Lista.cpp</a>		
Metody klasy <a href="#">Lista</a>	. . . . .	23
<a href="#">Lista.hh</a>		
Definicja klasy <a href="#">Lista</a>	. . . . .	24
<a href="#">main.cpp</a>		
Modul glowny	. . . . .	25
<a href="#">Stos.cpp</a>		
Metody klasy <a href="#">Stos</a>	. . . . .	26
<a href="#">Stos.hh</a>		
Definicja klasy <a href="#">Stos</a>	. . . . .	26



## 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](#) (int)
- [~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 rozmiaru 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.*
- void [mergesort](#) ()  
*Metoda sortująca elementy tablicy algorytmem "mergesort". Metoda dzieli główną tablicę. Jest wywoływana w rekurencji. Wywołuje scalanie podtablicy lewej i prawej.*
- void [scal](#) ([ArrayLista](#) \*tab\_l, [ArrayLista](#) \*tab\_p)  
*Metoda scalająca i sortująca, od najmniejszej do największej, liczby w scalanych tablicach. Sortuje te liczby do tablicy głównej.*

#### 3.1.1 Detailed Description

Klasa [ArrayLista](#).

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 ArrayList::ArrayList ( )

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

#### 3.1.2.2 ArrayList::ArrayList ( int wielkosc )

#### 3.1.2.3 ArrayList::~~ArrayList ( )

Destruktor bezparametryczny stosu.

### 3.1.3 Member Function Documentation

#### 3.1.3.1 int ArrayList::ile\_elementow ( )

Metoda zwracająca ilosc elementow jaka znajduje sie na liscie.

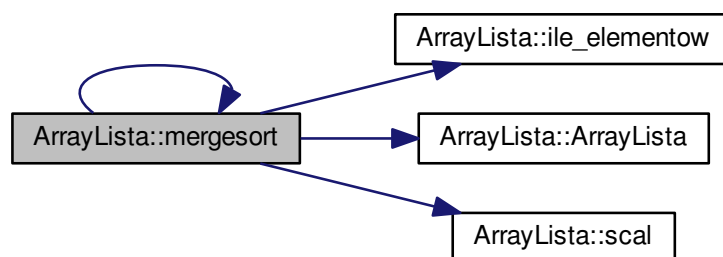
##### Returns

ilosc\_elementow - typu int, ilosc elementow jaka znajduje sie na liscie.

#### 3.1.3.2 void ArrayList::mergesort ( )

Metoda sortująca elementy tablicy algorytmem "mergesort". Metoda dzieli główną tablicę. Jest wywoływana w rekurencji. Wywołuje scalanie podtablicy lewej i prawej.

Here is the call graph for this function:



#### 3.1.3.3 int ArrayList::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.

### Returns

wartosc - typu int, wartosc zdejmowana ze stosu.

#### 3.1.3.4 void ArrayLista::push ( int wartosc, int zwiekszenie )

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

### Parameters

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

#### 3.1.3.5 void ArrayLista::push ( int wartosc ) [inline]

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

### Parameters

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

#### 3.1.3.6 void ArrayLista::scal ( ArrayLista \* tab\_l, ArrayLista \* tab\_p )

Metoda scalająca i sortująca, od najmniejszej do największej, liczby w scalanych tablicach. Sortuje te liczby do tablicy głównej.

### Parameters

<i>tab_l</i>	- typu <a href="#">ArrayLista</a> , lewa podtablica.
<i>tab_p</i>	- typu <a href="#">ArrayLista</a> , prawa podtablica.

#### 3.1.3.7 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< ArrayLista, int > ( 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.

<i>liczba_przejsc</i>	- typu int, liczba przejsc przez dane.
<i>liczba_danych</i>	- typu int, liczba danych w tablicy.

#### Returns

czas\_calkowity\_usredniony - typu long int, czas sredni dzialania 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)  
*Przeciazenie 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 - wielkosc tablicy pierwszej. rozmiar\_k2 - wielkosc tablicy drugiej. Tworzona jest macierz elementow (string wartosc,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 czytująca element z tablicy haszującej. Metoda czytuje element o określonym kluczu. Wartości oraz klucze nie są usuwane podczas czytania.

#### 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* )

Przeciazenie 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 ktorej 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 ktorej 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 umieszczająca element na końcu kolejki. Metoda inkrementuje rozmiar podczas umieszczania elementu w kolejce.*
- int [pop](#) ()  
*Metoda zdejmująca element z początku kolejki. Metoda dekrementuje rozmiar przy zdejmowaniu elementu.*
- int [size](#) ()  
*Metoda zwracająca wielkość kolejki.*

### 3.4.1 Detailed Description

Klasa [Kolejka](#).

### 3.4.2 Constructor & Destructor Documentation

#### 3.4.2.1 Kolejka::Kolejka ( )

Konstruktor bezparametryczny. Konstruktor inicjalizujący straznika\_początek i straznika\_koniec kolejki wartościami NULL , oraz rozmiar kolejki wartością 0.

#### 3.4.2.2 Kolejka::~~Kolejka ( )

Destruktor bezparametryczny kolejki.

### 3.4.3 Member Function Documentation

#### 3.4.3.1 int Kolejka::pop ( )

Metoda zdejmująca element z początku 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 umieszczająca element na końcu 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 zwracająca wielkość 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]

Przeciążenie operacji pop dla listy. Pobiera domyślnie element listy z pozycji 1. Następuje dekrementacja rozmiar listy.

#### Returns

wartosc - typu int, wartosc zdejmowana z listy.

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

Metoda umieszczająca element określonej pozycji na liście <0,rozmiar>. Metoda inkrementuje rozmiar podczas umieszczania elementu na liście.

#### Parameters

<i>wartosc</i>	- typu int, wartosc umieszczana na liście.
<i>pozycja</i>	- typu int, pozycja na której jest umieszczana wartosc.

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

Przeciążenie operacji push. Umieszcza element domyślnie na pozycji 1. Następuje inkrementacja rozmiar listy.

#### Parameters

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

### 3.5.3.5 int Lista::size ( )

Metoda zwracająca wielkość 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 inicjalizujący strażnika stosu wartością NULL, oraz rozmiar wartości 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

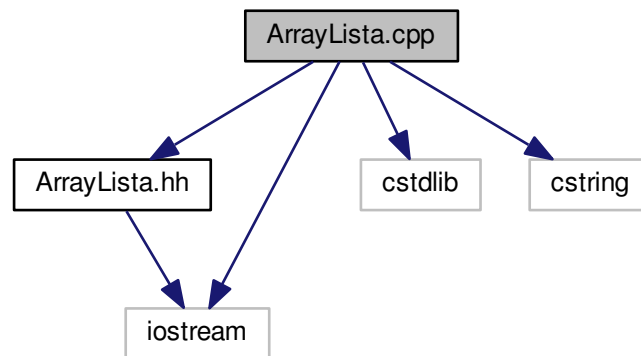
# File Documentation

### 4.1 ArrayList.cpp File Reference

Metody klasy [ArrayList](#).

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

Include dependency graph for ArrayList.cpp:



#### 4.1.1 Detailed Description

Metody klasy [ArrayList](#).

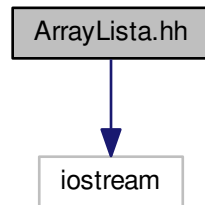
Plik zawiera metody klasy [ArrayList](#).

### 4.2 ArrayList.hh File Reference

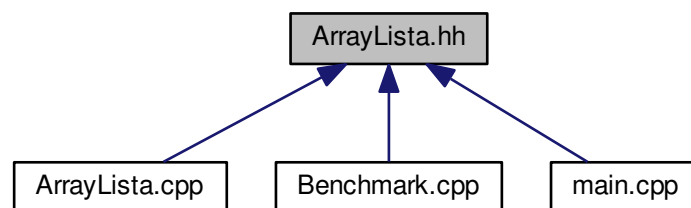
Definicja klasy [ArrayList](#).

```
#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](#).

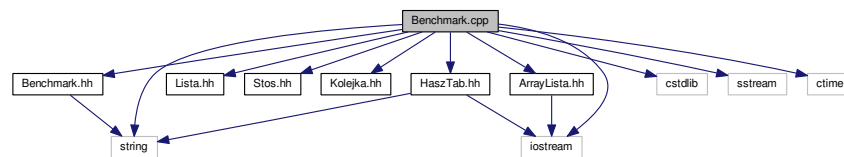


```

#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

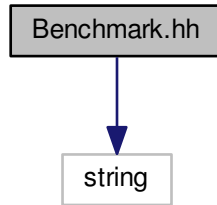
#### 4.3.2.1 #define BENCHMARK\_CPP

## 4.4 Benchmark.hh File Reference

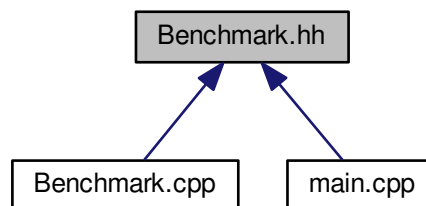
Definicja klasy [Benchmark](#).

```
#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](#).

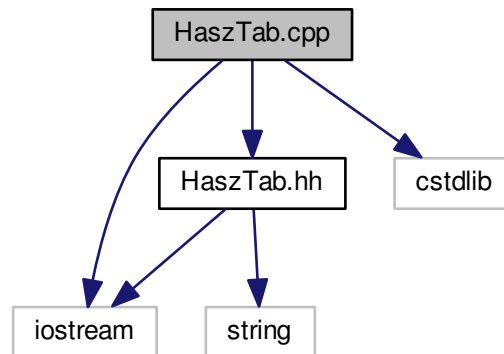
Plik zawiera definicje klasy [Benchmark](#).

## 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](#).

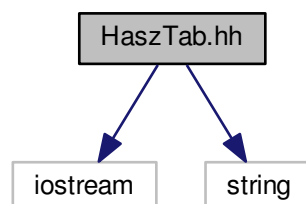
## 4.6 HaszTab.hh File Reference

Definicja klasy [HaszTab](#).

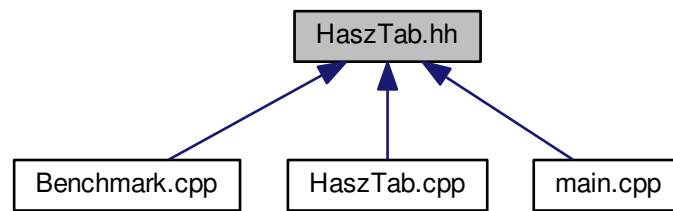
```
#include <iostream>
```

```
#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](#).

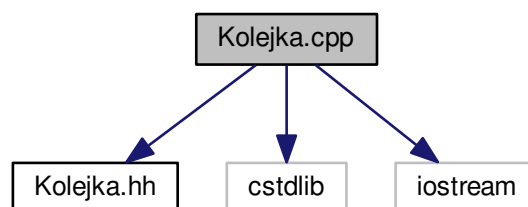
Plik zawiera definicje klasy modulujacej pojecie tablicy haszujacej.

## 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

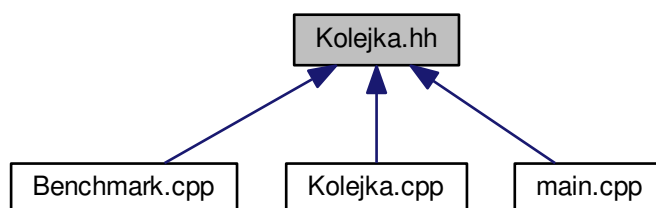
Metody klasy [Kolejka](#).

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](#).

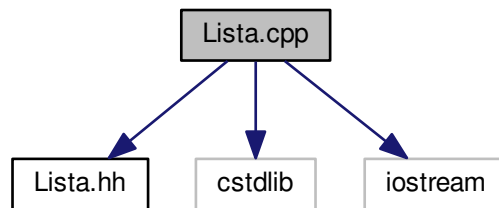
Plik zawiera definicje klasy [Kolejka](#).

## 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

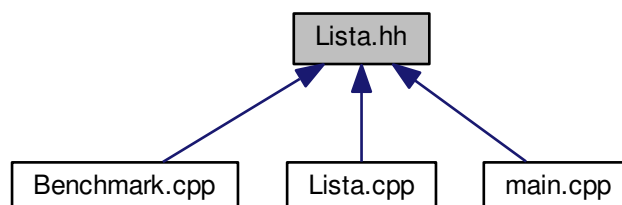
Metody klasy [Lista](#).

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](#).

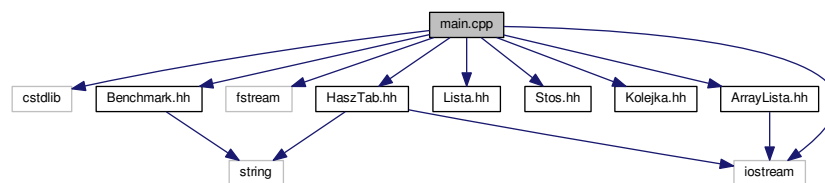
Plik zawiera definicje klasy modulujacej pojecie listy jednokierunkowej.

## 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.

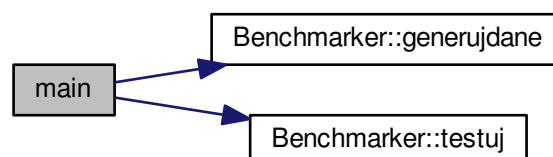
Plik zawiera funkcje main.

#### 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:

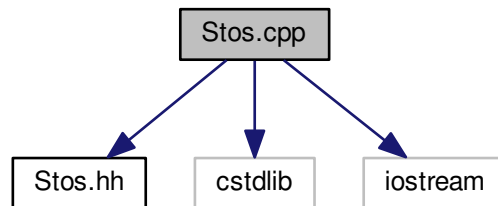


## 4.12 Stos.cpp File Reference

Metody klasy [Stos](#).

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

Include dependency graph for Stos.cpp:



### 4.12.1 Detailed Description

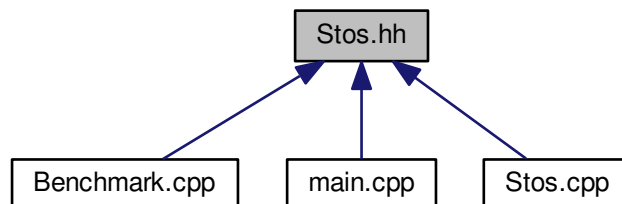
Metody klasy [Stos](#).

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

- ~ArrayLista
  - ArrayLista, 6
- ~HaszTab
  - HaszTab, 9
- ~Kolejka
  - Kolejka, 12
- ~Lista
  - Lista, 13
- ~Stos
  - Stos, 15
- ArrayLista, 5
  - ~ArrayLista, 6
  - ArrayLista, 6
  - ile\_elementow, 6
  - mergesort, 6
  - pop, 6
  - push, 7
  - scal, 7
  - size, 7
- ArrayLista.cpp, 17
- ArrayLista.hh, 17
- BENCHMARK\_CPP
  - Benchmark.cpp, 19
- Benchmark.cpp, 18
  - BENCHMARK\_CPP, 19
- Benchmark.hh, 19
- Benchmarker, 7
  - generujdane, 8
  - generujdane\_string, 8
  - testuj, 8
- generujdane
  - Benchmarker, 8
- generujdane\_string
  - Benchmarker, 8
- HaszTab, 9
  - ~HaszTab, 9
  - HaszTab, 9
  - mieszaj, 10
  - odczytaj, 10
  - push, 10
  - size\_k1, 11
  - size\_k2, 11
- HaszTab.cpp, 20
- HaszTab.hh, 21
- ile\_elementow
  - ArrayLista, 6
- Kolejka, 11
  - ~Kolejka, 12
  - Kolejka, 12
  - pop, 12
  - push, 12
  - size, 12
- Kolejka.cpp, 22
- Kolejka.hh, 23
- Lista, 12
  - ~Lista, 13
  - Lista, 13
  - pop, 13
  - push, 14
  - size, 14
- Lista.cpp, 23
- Lista.hh, 24
- main
  - main.cpp, 25
- main.cpp, 25
  - main, 25
- mergesort
  - ArrayLista, 6
- mieszaj
  - HaszTab, 10
- odczytaj
  - HaszTab, 10
- pop
  - ArrayLista, 6
  - Kolejka, 12
  - Lista, 13
  - Stos, 15
- push
  - ArrayLista, 7
  - HaszTab, 10
  - Kolejka, 12
  - Lista, 14
  - Stos, 15
- scal
  - ArrayLista, 7
- size
  - ArrayLista, 7
  - Kolejka, 12
  - Lista, 14
  - Stos, 15
- size\_k1
  - HaszTab, 11

size\_k2  
    HaszTab, [11](#)  
Stos, [14](#)  
    ~Stos, [15](#)  
    pop, [15](#)  
    push, [15](#)  
    size, [15](#)  
    Stos, [15](#)  
Stos.cpp, [26](#)  
Stos.hh, [26](#)  
  
testuj  
    Benchmark, [8](#)