

## Zadanie 3

Generated by Doxygen 1.8.9.1

Wed Apr 15 2015 12:56:37



# 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 . . . . .	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	testuj . . . . .	7
3.3	Kolejka Class Reference . . . . .	8
3.3.1	Detailed Description . . . . .	8
3.3.2	Constructor & Destructor Documentation . . . . .	8
3.3.2.1	Kolejka . . . . .	8
3.3.2.2	~Kolejka . . . . .	8
3.3.3	Member Function Documentation . . . . .	9
3.3.3.1	pop . . . . .	9
3.3.3.2	push . . . . .	9
3.3.3.3	size . . . . .	9

3.4	Lista Class Reference . . . . .	9
3.4.1	Detailed Description . . . . .	10
3.4.2	Constructor & Destructor Documentation . . . . .	10
3.4.2.1	Lista . . . . .	10
3.4.2.2	~Lista . . . . .	10
3.4.3	Member Function Documentation . . . . .	10
3.4.3.1	pop . . . . .	10
3.4.3.2	pop . . . . .	10
3.4.3.3	push . . . . .	10
3.4.3.4	push . . . . .	11
3.4.3.5	size . . . . .	11
3.5	Stos Class Reference . . . . .	11
3.5.1	Detailed Description . . . . .	11
3.5.2	Constructor & Destructor Documentation . . . . .	11
3.5.2.1	Stos . . . . .	11
3.5.2.2	~Stos . . . . .	12
3.5.3	Member Function Documentation . . . . .	12
3.5.3.1	pop . . . . .	12
3.5.3.2	push . . . . .	12
3.5.3.3	size . . . . .	12
<b>4</b>	<b>File Documentation</b>	<b>13</b>
4.1	ArrayLista.cpp File Reference . . . . .	13
4.1.1	Detailed Description . . . . .	13
4.2	ArrayLista.hh File Reference . . . . .	13
4.2.1	Detailed Description . . . . .	14
4.3	Benchmark.cpp File Reference . . . . .	14
4.3.1	Detailed Description . . . . .	15
4.3.2	Macro Definition Documentation . . . . .	15
4.3.2.1	BENCHMARK_CPP . . . . .	15
4.4	Benchmark.hh File Reference . . . . .	15
4.4.1	Detailed Description . . . . .	16
4.5	Kolejka.cpp File Reference . . . . .	16
4.5.1	Detailed Description . . . . .	16
4.6	Kolejka.hh File Reference . . . . .	16
4.6.1	Detailed Description . . . . .	17
4.7	Lista.cpp File Reference . . . . .	17
4.7.1	Detailed Description . . . . .	17
4.8	Lista.hh File Reference . . . . .	18
4.8.1	Detailed Description . . . . .	18

---

4.9	main.cpp File Reference	18
4.9.1	Detailed Description	19
4.9.2	Function Documentation	19
4.9.2.1	main	19
4.10	Stos.cpp File Reference	19
4.10.1	Detailed Description	20
4.11	Stos.hh File Reference	20
4.11.1	Detailed Description	20
	<b>Index</b>	<b>21</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="#">Benchmark</a>	Klasa <a href="#">Benchmark</a> . . . . .	7
<a href="#">Kolejka</a>	Klasa <a href="#">Kolejka</a> . . . . .	8
<a href="#">Lista</a>	Klasa <a href="#">Lista</a> . . . . .	9
<a href="#">Stos</a>	Klasa <a href="#">Stos</a> . . . . .	11





## 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>	. . . . .	13
<a href="#">ArrayLista.hh</a>		
Definicja klasy <a href="#">ArrayLista</a>	. . . . .	13
<a href="#">Benchmark.cpp</a>		
Metody klasy <a href="#">Benchmark</a>	. . . . .	14
<a href="#">Benchmark.hh</a>		
Definicja klasy <a href="#">Benchmark</a>	. . . . .	15
<a href="#">Kolejka.cpp</a>		
Metody klasy <a href="#">Kolejka</a>	. . . . .	16
<a href="#">Kolejka.hh</a>		
Definicja klasy <a href="#">Kolejka</a>	. . . . .	16
<a href="#">Lista.cpp</a>		
Metody klasy <a href="#">Lista</a>	. . . . .	17
<a href="#">Lista.hh</a>		
Definicja klasy <a href="#">Lista</a>	. . . . .	18
<a href="#">main.cpp</a>		
Modul glowny	. . . . .	18
<a href="#">Stos.cpp</a>		
Metody klasy <a href="#">Stos</a>	. . . . .	19
<a href="#">Stos.hh</a>		
Definicja klasy <a href="#">Stos</a>	. . . . .	20



## 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 listy.*

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

*Metoda umieszczająca element na liscie Metoda inkrementuje rozmiar podczas umieszczania elementu na liscie.*

- 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 polowe gdy ilosc elementow znajdujących się na niej jest równa polowie jej rozmiaru.*

- int [size](#) ()

*Metoda zwracająca rozmiar tablicy na ktorej oparta jest lista.*

- int [ile\\_elementow](#) ()

*Metoda zwracająca ilosc elementow jaka znajduje się na liscie.*

#### 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. *ilosc\_elementow* - ilość elementów listy, inicjalizowane wartością 0. Tworzona jest tablica dynamiczna o jednym elemencie.

### 3.1.2.2 ArrayLista::~~ArrayLista ( )

Destruktor bezparametryczny listy.

## 3.1.3 Member Function Documentation

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

Metoda zwracająca ilość elementów jaka znajduje się na liście.

#### Returns

*ilosc\_elementow* - typu int, ilość elementów jaka znajduje się na liście.

### 3.1.3.2 int ArrayLista::pop ( )

Metoda zdejmująca element z listy. Metoda dekrementuje *ilosc\_elementow* 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

*wartosc* - typu int, wartość zdejmowana ze stosu.

### 3.1.3.3 void ArrayLista::push ( int *wartosc*, 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>wartosc</i>	- typu int, wartość umieszczana na stosie.

### 3.1.3.4 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, 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 >`  
`long int testuj (T *, int *, int, int)`  
*Szablon metody przeprowadzająca sprawdzenie czasu działania funkcji. Typy: [Lista](#), [Stos](#), [Kolejka](#).*
- `int * generuj dane (int)`  
*Metoda generująca wartości losowe z przedziału 0 10000.*

### 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 `template<class T > template long int Benchmarker::testuj< ArrayLista > ( T *, int *, int, int )`

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

##### Template Parameters

<code>Tab</code>	- typu <code>T*</code> , wskaźnik na zaimplementowany stos/listę/kolejkę.
<code>dane</code>	- typu <code>int*</code> , wskaźnik na tablicę z danymi generowanymi.
<code>liczba_przejsc</code>	- typu int, liczba przejść przez dane.

<code>liczba_danych</code>	- 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 Kolejka Class Reference

Klasa [Kolejka](#).

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

### Public Member Functions

- [Kolejka](#) ()  
*Konstruktor bezparametryczny. Konstruktor inicjalizujacy straznika\_poczek i straznika\_koniec kolejki wartosciami NULL , oraz rozmiar kolejki wartoscia 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.*

### 3.3.1 Detailed Description

Klasa [Kolejka](#).

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 Kolejka::Kolejka ( )

Konstruktor bezparametryczny. Konstruktor inicjalizujacy straznika\_poczek i straznika\_koniec kolejki wartosciami NULL , oraz rozmiar kolejki wartoscia 0.

#### 3.3.2.2 Kolejka::~~Kolejka ( )

Destruktor bezparametryczny kolejki.

### 3.3.3 Member Function Documentation

#### 3.3.3.1 `int Kolejka::pop ( )`

Metoda zdejmująca element z początku kolejki. Metoda dekrementuje rozmiar przy zdejmowaniu elementu.

##### Returns

wartosc - typu int, wartosc zdejmowana z kolejki.

#### 3.3.3.2 `void Kolejka::push ( int wartosc )`

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

##### Parameters

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

#### 3.3.3.3 `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](#)

## 3.4 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](#) ()

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

- `int size ()`

*Metoda zwracajaca wielkosc listy.*

### 3.4.1 Detailed Description

Klasa [Lista](#).

### 3.4.2 Constructor & Destructor Documentation

#### 3.4.2.1 `Lista::Lista ( )`

Konstruktor bezparametryczny. Konstruktor inicjalizujacy straznika listy wartoscia NULL oraz rozmiar wartoscia 0.

#### 3.4.2.2 `Lista::~~Lista ( )`

Destruktor bezparametryczny listy.

### 3.4.3 Member Function Documentation

#### 3.4.3.1 `int Lista::pop ( int pozycja )`

Metoda zdejmujaca element z okreslonej pozycji listy <0,rozmiar>. Metoda dekrementuje rozmiar przy zdejmowaniu elementu.

Parameters

<i>pozycja</i>	- typu int, numer elementu ktory ma byc zdjety z listy.
----------------	---------------------------------------------------------

Returns

wartosc - typu int, wartosc zdejmowana z listy.

#### 3.4.3.2 `int Lista::pop ( ) [inline]`

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

Returns

wartosc - typu int, wartosc zdejmowana z listy.

#### 3.4.3.3 `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.

#### 3.4.3.4 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.
----------------	--------------------------------------------

#### 3.4.3.5 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](#)

## 3.5 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.*

### 3.5.1 Detailed Description

Klasa [Stos](#).

### 3.5.2 Constructor & Destructor Documentation

#### 3.5.2.1 Stos::Stos ( )

Konstruktor bezparametryczny. Konstruktor inicjalizujacy straznika stosu wartoscia NULL ,oraz rozmiar wartoscia 0.

### 3.5.2.2 Stos::~~Stos ( )

Destruktor bezparametryczny stosu.

## 3.5.3 Member Function Documentation

### 3.5.3.1 int Stos::pop ( )

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

#### Returns

wartosc - typu int, wartosc zdejmowana ze stosu.

### 3.5.3.2 void Stos::push ( int *wartosc* )

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

#### Parameters

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

### 3.5.3.3 int Stos::size ( )

Metoda zwracajaca wielkosc stosu.

#### Returns

rozmiar - typu int,rozmiar stosu.

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

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

## Chapter 4

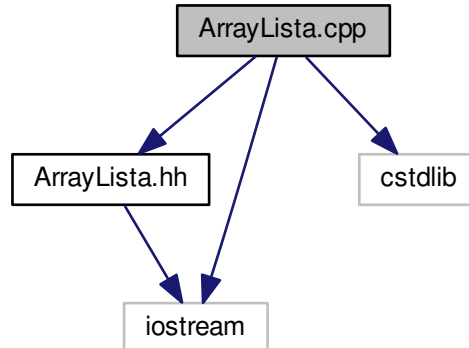
# File Documentation

### 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](#).

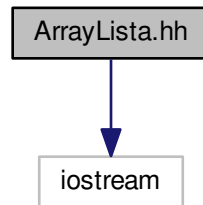
Plik zawiera metody klasy [ArrayList](#).

### 4.2 ArrayList.hh File Reference

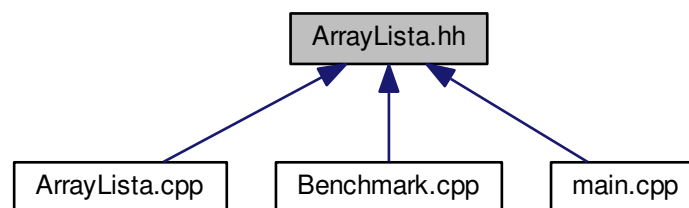
Definicja klasy [ArrayList](#).

```
#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](#).

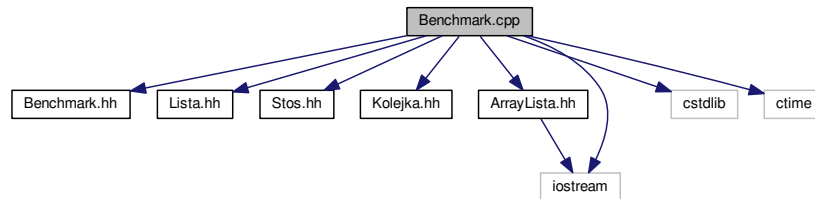
Plik zawiera definicje klasy modulujacej pojecie listy jednokierunkowej opartej na tablicy dynamicznej.

## 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 <cstdlib>
#include <iostream>
#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](#).

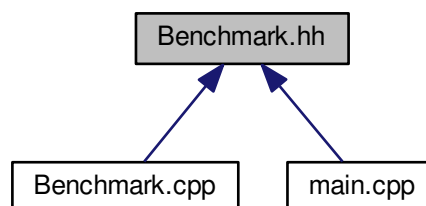
### 4.3.2 Macro Definition Documentation

#### 4.3.2.1 `#define BENCHMARK_CPP`

## 4.4 Benchmark.hh File Reference

Definicja klasy [Benchmark](#).

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



## Classes

- class [Benchmark](#)

Klasa [Benchmark](#).

### 4.4.1 Detailed Description

Definicja klasy [Benchmark](#).

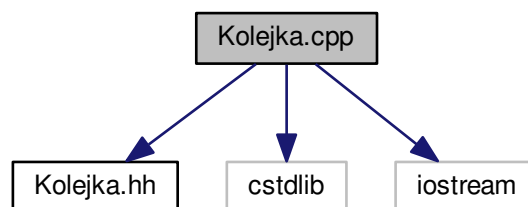
Plik zawiera definicje klasy [Benchmark](#).

## 4.5 Kolejka.cpp File Reference

Metody klasy [Kolejka](#).

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

Include dependency graph for Kolejka.cpp:



### 4.5.1 Detailed Description

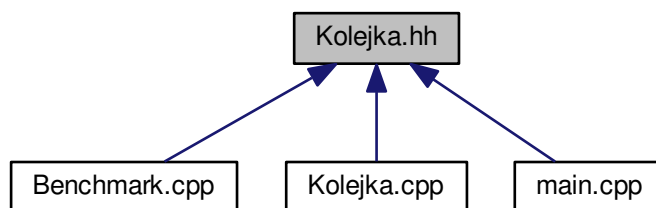
Metody klasy [Kolejka](#).

Plik zawiera metody klasy [Kolejka](#).

## 4.6 Kolejka.hh File Reference

Definicja klasy [Kolejka](#).

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



## Classes

- class [Kolejka](#)  
*Klasa [Kolejka](#).*

### 4.6.1 Detailed Description

Definicja klasy [Kolejka](#).

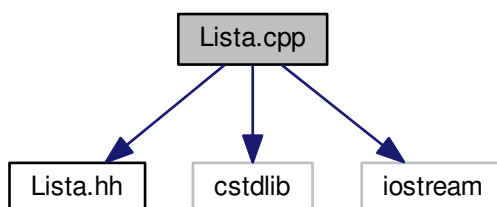
Plik zawiera definicje klasy [Kolejka](#).

## 4.7 Lista.cpp File Reference

Metody klasy [Lista](#).

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

Include dependency graph for `Lista.cpp`:



### 4.7.1 Detailed Description

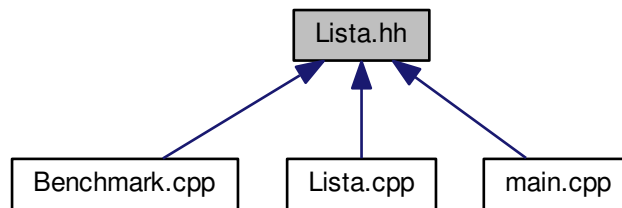
Metody klasy [Lista](#).

Plik zawiera metody klasy [Lista](#).

## 4.8 Lista.hh File Reference

Definicja klasy [Lista](#).

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



### Classes

- class [Lista](#)

*Klasa [Lista](#).*

### 4.8.1 Detailed Description

Definicja klasy [Lista](#).

Plik zawiera definicje klasy modulujacej pojecie listy jednokierunkowej.

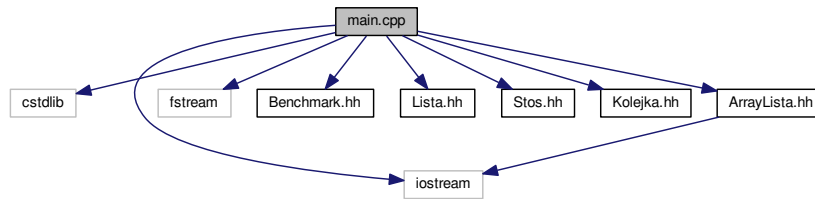
## 4.9 main.cpp File Reference

Modul glowny.

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



Include dependency graph for main.cpp:



## Functions

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

*Funkcja glowna programu.*

### 4.9.1 Detailed Description

Modul glowny.

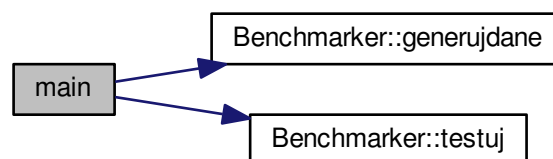
Plik zawiera funkcje main.

### 4.9.2 Function Documentation

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

Funkcja glowna programu.

Here is the call graph for this function:



## 4.10 Stos.cpp File Reference

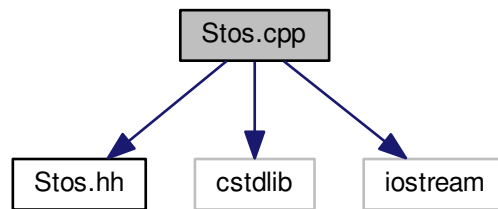
Metody klasy [Stos](#).

```

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

```

Include dependency graph for Stos.cpp:



#### 4.10.1 Detailed Description

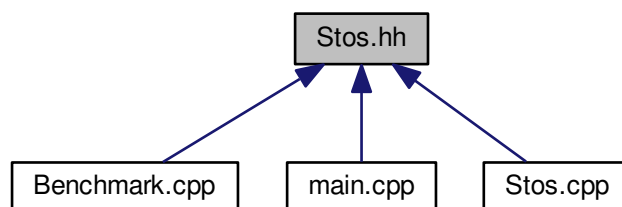
Metody klasy [Stos](#).

Plik zawiera metody klasy [Stos](#).

### 4.11 Stos.hh File Reference

Definicja klasy [Stos](#).

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



#### Classes

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

#### 4.11.1 Detailed Description

Definicja klasy [Stos](#).

Plik zawiera definicje klasy [Stos](#).

# Index

- ~ArrayList
  - ArrayList, 6
- ~Kolejka
  - Kolejka, 8
- ~Lista
  - Lista, 10
- ~Stos
  - Stos, 11
- ArrayList, 5
  - ~ArrayList, 6
  - ArrayList, 5
  - ile\_elementow, 6
  - pop, 6
  - push, 6
  - size, 6
- ArrayList.cpp, 13
- ArrayList.hh, 13
- BENCHMARK\_CPP
  - Benchmark.cpp, 15
- Benchmark.cpp, 14
  - BENCHMARK\_CPP, 15
- Benchmark.hh, 15
- Benchmark, 7
  - generujdane, 7
  - testuj, 7
- generujdane
  - Benchmark, 7
- ile\_elementow
  - ArrayList, 6
- Kolejka, 8
  - ~Kolejka, 8
  - Kolejka, 8
  - pop, 9
  - push, 9
  - size, 9
- Kolejka.cpp, 16
- Kolejka.hh, 16
- Lista, 9
  - ~Lista, 10
  - Lista, 10
  - pop, 10
  - push, 10, 11
  - size, 11
- Lista.cpp, 17
- Lista.hh, 18
- main
  - main.cpp, 19
- main.cpp, 18
  - main, 19
- pop
  - ArrayList, 6
  - Kolejka, 9
  - Lista, 10
  - Stos, 12
- push
  - ArrayList, 6
  - Kolejka, 9
  - Lista, 10, 11
  - Stos, 12
- size
  - ArrayList, 6
  - Kolejka, 9
  - Lista, 11
  - Stos, 12
- Stos, 11
  - ~Stos, 11
  - pop, 12
  - push, 12
  - size, 12
  - Stos, 11
- Stos.cpp, 19
- Stos.hh, 20
- testuj
  - Benchmark, 7