

## Zadanie 7

Generated by Doxygen 1.8.9.1

Wed May 20 2015 13:51:08



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	Algorytm< T > Class Template Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Constructor & Destructor Documentation . . . . .	7
4.1.2.1	~Algorytm . . . . .	7
4.1.3	Member Function Documentation . . . . .	7
4.1.3.1	alokujdane . . . . .	7
4.1.3.2	wykonajalgorytm . . . . .	8
4.2	Algorytm1 Class Reference . . . . .	8
4.2.1	Detailed Description . . . . .	9
4.2.2	Constructor & Destructor Documentation . . . . .	9
4.2.2.1	~Algorytm1 . . . . .	9
4.2.3	Member Function Documentation . . . . .	9
4.2.3.1	alokujdane . . . . .	9
4.2.3.2	wykonajalgorytm . . . . .	9
4.3	Algorytm2 Class Reference . . . . .	10
4.3.1	Detailed Description . . . . .	11
4.3.2	Member Function Documentation . . . . .	11
4.3.2.1	alokujdane . . . . .	11
4.3.2.2	wykonajalgorytm . . . . .	11
4.4	Algorytm3 Class Reference . . . . .	11
4.4.1	Detailed Description . . . . .	12
4.4.2	Member Function Documentation . . . . .	12

4.4.2.1	alokujdane	12
4.4.2.2	wykonajalgorytm	13
4.5	Algorytm4 Class Reference	13
4.5.1	Detailed Description	14
4.5.2	Constructor & Destructor Documentation	14
4.5.2.1	~Algorytm4	14
4.5.3	Member Function Documentation	14
4.5.3.1	alokujdane	15
4.5.3.2	mergesort	16
4.5.3.3	scal	16
4.5.3.4	wykonajalgorytm	16
4.6	Algorytm5 Class Reference	17
4.6.1	Detailed Description	18
4.6.2	Member Function Documentation	18
4.6.2.1	alokujdane	18
4.6.2.2	wykonajalgorytm	18
4.7	Algorytm6 Class Reference	19
4.7.1	Detailed Description	19
4.7.2	Constructor & Destructor Documentation	19
4.7.2.1	~Algorytm6	19
4.7.3	Member Function Documentation	20
4.7.3.1	alokujdane	20
4.7.3.2	wykonajalgorytm	20
4.8	ArrayLista Class Reference	20
4.8.1	Detailed Description	21
4.8.2	Constructor & Destructor Documentation	21
4.8.2.1	ArrayLista	21
4.8.2.2	ArrayLista	22
4.8.2.3	~ArrayLista	22
4.8.3	Member Function Documentation	22
4.8.3.1	pop	22
4.8.3.2	push	22
4.8.3.3	push	22
4.8.3.4	size	22
4.9	Benchmark< T > Class Template Reference	23
4.9.1	Detailed Description	23
4.9.2	Member Function Documentation	24
4.9.2.1	powiadam	24
4.9.2.2	testuj	24
4.10	HaszTab Class Reference	24

4.10.1 Detailed Description . . . . .	26
4.10.2 Constructor & Destructor Documentation . . . . .	26
4.10.2.1 HaszTab . . . . .	26
4.10.2.2 ~HaszTab . . . . .	26
4.10.3 Member Function Documentation . . . . .	26
4.10.3.1 mieszaj . . . . .	26
4.10.3.2 odczytaj . . . . .	26
4.10.3.3 pop . . . . .	27
4.10.3.4 pop . . . . .	27
4.10.3.5 push . . . . .	27
4.10.3.6 push . . . . .	28
4.10.3.7 size . . . . .	28
4.10.3.8 size_k1 . . . . .	28
4.10.3.9 size_k2 . . . . .	28
4.11 Kolejka Class Reference . . . . .	29
4.11.1 Detailed Description . . . . .	30
4.11.2 Constructor & Destructor Documentation . . . . .	30
4.11.2.1 Kolejka . . . . .	30
4.11.2.2 ~Kolejka . . . . .	30
4.11.3 Member Function Documentation . . . . .	30
4.11.3.1 pop . . . . .	30
4.11.3.2 push . . . . .	30
4.11.3.3 size . . . . .	30
4.12 Lista Class Reference . . . . .	31
4.12.1 Detailed Description . . . . .	32
4.12.2 Constructor & Destructor Documentation . . . . .	32
4.12.2.1 Lista . . . . .	32
4.12.2.2 ~Lista . . . . .	32
4.12.3 Member Function Documentation . . . . .	32
4.12.3.1 pop . . . . .	32
4.12.3.2 pop . . . . .	32
4.12.3.3 push . . . . .	32
4.12.3.4 push . . . . .	33
4.12.3.5 size . . . . .	33
4.13 Obserwator Class Reference . . . . .	33
4.13.1 Detailed Description . . . . .	34
4.13.2 Member Function Documentation . . . . .	34
4.13.2.1 odswiez . . . . .	34
4.14 ObserwatorZapisujacy Class Reference . . . . .	34
4.14.1 Detailed Description . . . . .	35

4.14.2	Member Function Documentation	35
4.14.2.1	odswiez	35
4.15	Obserwowany Class Reference	35
4.15.1	Detailed Description	36
4.15.2	Member Function Documentation	36
4.15.2.1	dodaj	37
4.15.2.2	usun	38
4.15.3	Member Data Documentation	38
4.15.3.1	obserwatorzy	38
4.16	Stos Class Reference	38
4.16.1	Detailed Description	39
4.16.2	Constructor & Destructor Documentation	39
4.16.2.1	Stos	39
4.16.2.2	~Stos	39
4.16.3	Member Function Documentation	39
4.16.3.1	pop	39
4.16.3.2	push	40
4.16.3.3	size	40
4.17	Zasobnik< T > Class Template Reference	40
4.17.1	Detailed Description	40
4.17.2	Constructor & Destructor Documentation	41
4.17.2.1	~Zasobnik	41
4.17.3	Member Function Documentation	41
4.17.3.1	pop	41
4.17.3.2	push	41
4.17.3.3	size	41
<b>5</b>	<b>File Documentation</b>	<b>43</b>
5.1	Algorytm1.cpp File Reference	43
5.1.1	Detailed Description	43
5.2	Algorytm1.hh File Reference	44
5.2.1	Detailed Description	45
5.3	Algorytm2.cpp File Reference	45
5.3.1	Detailed Description	45
5.4	Algorytm2.hh File Reference	45
5.4.1	Detailed Description	46
5.5	Algorytm3.cpp File Reference	47
5.5.1	Detailed Description	47
5.6	Algorytm3.hh File Reference	47
5.6.1	Detailed Description	48

5.7	Algorytm4.cpp File Reference . . . . .	49
5.7.1	Detailed Description . . . . .	49
5.8	Algorytm4.hh File Reference . . . . .	49
5.8.1	Detailed Description . . . . .	50
5.9	Algorytm5.cpp File Reference . . . . .	51
5.9.1	Detailed Description . . . . .	51
5.10	Algorytm5.hh File Reference . . . . .	51
5.10.1	Detailed Description . . . . .	52
5.11	Algorytm6.cpp File Reference . . . . .	53
5.11.1	Detailed Description . . . . .	53
5.12	Algorytm6.hh File Reference . . . . .	53
5.12.1	Detailed Description . . . . .	54
5.13	AlgorytmAbs.hh File Reference . . . . .	55
5.13.1	Detailed Description . . . . .	55
5.14	ArrayLista.cpp File Reference . . . . .	56
5.14.1	Detailed Description . . . . .	56
5.15	ArrayLista.hh File Reference . . . . .	56
5.15.1	Detailed Description . . . . .	57
5.16	Benchmark.cpp File Reference . . . . .	57
5.16.1	Detailed Description . . . . .	58
5.17	Benchmark.hh File Reference . . . . .	58
5.17.1	Detailed Description . . . . .	59
5.18	GeneratoryDanych.cpp File Reference . . . . .	59
5.18.1	Detailed Description . . . . .	60
5.18.2	Function Documentation . . . . .	60
5.18.2.1	generujdane . . . . .	60
5.18.2.2	generujdane . . . . .	60
5.19	GeneratoryDanych.hh File Reference . . . . .	60
5.19.1	Detailed Description . . . . .	61
5.19.2	Function Documentation . . . . .	61
5.19.2.1	generujdane . . . . .	61
5.20	HaszTab.cpp File Reference . . . . .	61
5.20.1	Detailed Description . . . . .	62
5.21	HaszTab.hh File Reference . . . . .	62
5.21.1	Detailed Description . . . . .	62
5.22	Kolejka.cpp File Reference . . . . .	63
5.22.1	Detailed Description . . . . .	63
5.23	Kolejka.hh File Reference . . . . .	63
5.23.1	Detailed Description . . . . .	64
5.24	Lista.cpp File Reference . . . . .	64

5.24.1 Detailed Description . . . . .	65
5.25 Lista.hh File Reference . . . . .	65
5.25.1 Detailed Description . . . . .	66
5.26 main.cpp File Reference . . . . .	66
5.26.1 Detailed Description . . . . .	67
5.26.2 Function Documentation . . . . .	67
5.26.2.1 main . . . . .	67
5.27 Obserwator.hh File Reference . . . . .	67
5.27.1 Detailed Description . . . . .	67
5.28 ObserwatorZapisujacy.cpp File Reference . . . . .	68
5.28.1 Detailed Description . . . . .	68
5.29 ObserwatorZapisujacy.hh File Reference . . . . .	68
5.29.1 Detailed Description . . . . .	69
5.30 Obserwowany.hh File Reference . . . . .	69
5.30.1 Detailed Description . . . . .	70
5.31 Stos.cpp File Reference . . . . .	70
5.31.1 Detailed Description . . . . .	71
5.32 Stos.hh File Reference . . . . .	71
5.32.1 Detailed Description . . . . .	72
5.33 Zasobnik.hh File Reference . . . . .	72
5.33.1 Detailed Description . . . . .	73
<b>Index</b>	<b>75</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Algorytm< T > . . . . .	7
Algorytm< int > . . . . .	7
Algorytm1 . . . . .	8
Algorytm2 . . . . .	10
Algorytm4 . . . . .	13
Algorytm5 . . . . .	17
Algorytm< string > . . . . .	7
Algorytm3 . . . . .	11
Algorytm6 . . . . .	19
Obserwator . . . . .	33
ObserwatorZapisujacy . . . . .	34
Obserwowany . . . . .	35
Benchmark< T > . . . . .	23
Zasobnik< T > . . . . .	40
Zasobnik< int > . . . . .	40
ArrayLista . . . . .	20
Kolejka . . . . .	29
Lista . . . . .	31
Stos . . . . .	38
Zasobnik< string > . . . . .	40
HaszTab . . . . .	24



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Algorytm&lt; T &gt;</a>	
Szablon klasy <a href="#">Algorytm</a> . . . . .	7
<a href="#">Algorytm1</a>	
Klasa <a href="#">Algorytm1</a> . . . . .	8
<a href="#">Algorytm2</a>	
Klasa <a href="#">Algorytm2</a> . . . . .	10
<a href="#">Algorytm3</a>	
Klasa <a href="#">Algorytm3</a> . . . . .	11
<a href="#">Algorytm4</a>	
Klasa <a href="#">Algorytm4</a> . . . . .	13
<a href="#">Algorytm5</a>	
Klasa <a href="#">Algorytm5</a> . . . . .	17
<a href="#">Algorytm6</a>	
Klasa <a href="#">Algorytm6</a> . . . . .	19
<a href="#">ArrayLista</a>	
Klasa <a href="#">ArrayLista</a> . . . . .	20
<a href="#">Benchmark&lt; T &gt;</a>	
Szablon klasy <a href="#">Benchmark</a> . . . . .	23
<a href="#">HaszTab</a>	
Klasa <a href="#">HaszTab</a> . . . . .	24
<a href="#">Kolejka</a>	
Klasa <a href="#">Kolejka</a> . . . . .	29
<a href="#">Lista</a>	
Klasa <a href="#">Lista</a> . . . . .	31
<a href="#">Obserwator</a>	
Klasa <a href="#">Obserwator</a> . . . . .	33
<a href="#">ObserwatorZapisujacy</a>	
Klasa <a href="#">ObserwatorZapisujacy</a> . . . . .	34
<a href="#">Obserwowany</a>	
Szablon klasy <a href="#">Obserwowany</a> . . . . .	35
<a href="#">Stos</a>	
Klasa <a href="#">Stos</a> . . . . .	38
<a href="#">Zasobnik&lt; T &gt;</a>	
Szablon klasy <a href="#">Zasobnik</a> . . . . .	40



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">Algorytm1.cpp</a>		
Metody klasy <a href="#">Algorytm1</a>	43	
<a href="#">Algorytm1.hh</a>		
Definicja klasy <a href="#">Algorytm1</a>	44	
<a href="#">Algorytm2.cpp</a>		
Metody klasy <a href="#">Algorytm2</a>	45	
<a href="#">Algorytm2.hh</a>		
Definicja klasy <a href="#">Algorytm2</a>	45	
<a href="#">Algorytm3.cpp</a>		
Metody klasy <a href="#">Algorytm3</a>	47	
<a href="#">Algorytm3.hh</a>		
Definicja klasy <a href="#">Algorytm3</a>	47	
<a href="#">Algorytm4.cpp</a>		
Metody klasy <a href="#">Algorytm4</a>	49	
<a href="#">Algorytm4.hh</a>		
Definicja klasy <a href="#">Algorytm4</a>	49	
<a href="#">Algorytm5.cpp</a>		
Metody klasy <a href="#">Algorytm5</a>	51	
<a href="#">Algorytm5.hh</a>		
Definicja klasy <a href="#">Algorytm5</a>	51	
<a href="#">Algorytm6.cpp</a>		
Metody klasy <a href="#">Algorytm6</a>	53	
<a href="#">Algorytm6.hh</a>		
Definicja klasy <a href="#">Algorytm6</a>	53	
<a href="#">AlgorytmAbs.hh</a>		
Definicja szablonu klasy abstrakcyjnej <a href="#">Algorytm</a>	55	
<a href="#">ArrayLista.cpp</a>		
Metody klasy <a href="#">ArrayLista</a>	56	
<a href="#">ArrayLista.hh</a>		
Definicja klasy <a href="#">ArrayLista</a>	56	
<a href="#">Benchmark.cpp</a>		
Metody klasy <a href="#">Benchmark</a>	57	
<a href="#">Benchmark.hh</a>		
Definicja szablonu klasy <a href="#">Benchmark</a>	58	
<a href="#">GeneratoryDanych.cpp</a>		
Funkcje generacji danych	59	
<a href="#">GeneratoryDanych.hh</a>		
Szablon funkcji generacji danych	60	

<a href="#">HaszTab.cpp</a>	
Metody klasy <a href="#">HaszTab</a>	61
<a href="#">HaszTab.hh</a>	
Definicja klasy <a href="#">HaszTab</a>	62
<a href="#">Kolejka.cpp</a>	
Metody klasy <a href="#">Kolejka</a>	63
<a href="#">Kolejka.hh</a>	
Definicja klasy <a href="#">Kolejka</a>	63
<a href="#">Lista.cpp</a>	
Metody klasy <a href="#">Lista</a>	64
<a href="#">Lista.hh</a>	
Definicja klasy <a href="#">Lista</a>	65
<a href="#">main.cpp</a>	
Modul glowny	66
<a href="#">Obserwator.hh</a>	
Definicja klasy <a href="#">Obserwator</a>	67
<a href="#">ObserwatorZapisujacy.cpp</a>	
Metody klasy <a href="#">ObserwatorZapisujacy</a>	68
<a href="#">ObserwatorZapisujacy.hh</a>	
Definicja klasy <a href="#">ObserwatorZapisujacy</a>	68
<a href="#">Obserwowany.hh</a>	
Definicja szablonu klasy abstrakcyjnej <a href="#">Obserwowany</a>	69
<a href="#">Stos.cpp</a>	
Metody klasy <a href="#">Stos</a>	70
<a href="#">Stos.hh</a>	
Definicja klasy <a href="#">Stos</a>	71
<a href="#">Zasobnik.hh</a>	
Definicja szablonu klasy abstrakcyjnej <a href="#">Zasobnik</a>	72

## Chapter 4

# Class Documentation

### 4.1 Algorytm< T > Class Template Reference

Szablon klasy [Algorytm](#).

```
#include <AlgorytmAbs.hh>
```

#### Public Member Functions

- virtual [~Algorytm](#) ()  
*Destruktor wirtualny algorytmu.*
- virtual void [alokuj dane](#) ([Zasobnik](#)< T > \*Tab, T \*dane, int liczba\_danych)=0  
*Metoda alokujaca na zasobniku dane.*
- virtual void [wykonaj algorytm](#) ([Zasobnik](#)< T > \*Tab, T \*dane, int liczba\_danych)=0  
*Metoda wykonujaca konkretny algorytm.*

#### 4.1.1 Detailed Description

```
template<typename T>class Algorytm< T >
```

Szablon klasy [Algorytm](#).

#### 4.1.2 Constructor & Destructor Documentation

4.1.2.1 `template<typename T> virtual Algorytm< T >::~~Algorytm ( ) [inline], [virtual]`

Destruktor wirtualny algorytmu.

#### 4.1.3 Member Function Documentation

4.1.3.1 `template<typename T> virtual void Algorytm< T >::alokuj dane ( Zasobnik< T > * Tab, T * dane, int liczba_danych ) [pure virtual]`

Metoda alokujaca na zasobniku dane.

## Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implemented in [Algorytm1](#), [Algorytm2](#), [Algorytm3](#), [Algorytm4](#), [Algorytm5](#), and [Algorytm6](#).

4.1.3.2 `template<typename T> virtual void Algorytm< T >::wykonajalgorytm ( Zasobnik< T > * Tab, T * dane, int liczba_danych ) [pure virtual]`

Metoda wykonujaca konkretny algorytm.

## Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implemented in [Algorytm1](#), [Algorytm2](#), [Algorytm3](#), [Algorytm4](#), [Algorytm5](#), and [Algorytm6](#).

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

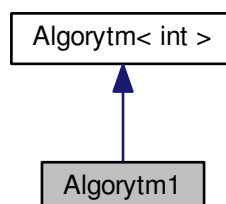
- [AlgorytmAbs.hh](#)

## 4.2 Algorytm1 Class Reference

Klasa [Algorytm1](#).

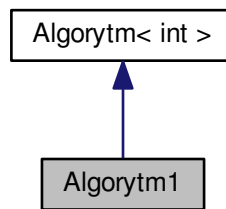
```
#include <Algorytm1.hh>
```

Inheritance diagram for Algorytm1:





Collaboration diagram for Algorytm1:



## Public Member Functions

- `~Algorytm1 ()`
- `void alokujdane (Zasobnik< int > *, int *, int)`  
*Metoda alokujaca na zasobniku dane.*
- `void wykonajalgorytm (Zasobnik< int > *, int *, int)`  
*Metoda wykonujaca konkretny algorytm.*

### 4.2.1 Detailed Description

Klasa `Algorytm1`.

### 4.2.2 Constructor & Destructor Documentation

4.2.2.1 `Algorytm1::~~Algorytm1 ( ) [inline]`

### 4.2.3 Member Function Documentation

4.2.3.1 `void Algorytm1::alokujdane ( Zasobnik< int > * Tab, int * dane, int liczba_danych ) [virtual]`

Metoda alokujaca na zasobniku dane.

Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implements `Algorytm< int >`.

4.2.3.2 `void Algorytm1::wykonajalgorytm ( Zasobnik< int > * Tab, int * dane, int liczba_danych ) [virtual]`

Metoda wykonujaca konkretny algorytm.

## Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implements [Algorytm< int >](#).

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

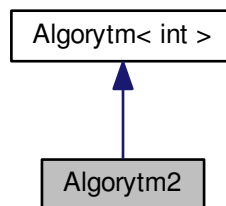
- [Algorytm1.hh](#)
- [Algorytm1.cpp](#)

## 4.3 Algorytm2 Class Reference

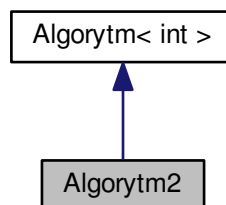
Klasa [Algorytm2](#).

```
#include <Algorytm2.hh>
```

Inheritance diagram for Algorytm2:



Collaboration diagram for Algorytm2:



### Public Member Functions

- void [alokujdane](#) ([Zasobnik< int >\\*](#) Tab, int \*dane, int liczba\_danych)  
*Metoda alokująca na zasobniku dane.*

- void [wykonajalgorytm](#) ([Zasobnik](#)< int > \*Tab, int \*dane, int liczba\_danych)

*Metoda wykonująca konkretny algorytm.*

### 4.3.1 Detailed Description

Klasa [Algorytm2](#).

### 4.3.2 Member Function Documentation

4.3.2.1 void [Algorytm2::alokujdane](#) ( [Zasobnik](#)< int > \* Tab, int \* dane, int liczba\_danych ) [virtual]

Metoda alokująca na zasobniku dane.

Parameters

<i>Tab</i>	- typu <a href="#">Zasobnik</a> <T>*, implementacja zasobnika
<i>dane</i>	- typu T*, dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu int, liczba danych dla zasobnika

Implements [Algorytm](#)< int >.

4.3.2.2 void [Algorytm2::wykonajalgorytm](#) ( [Zasobnik](#)< int > \* Tab, int \* dane, int liczba\_danych ) [virtual]

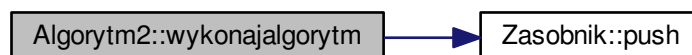
Metoda wykonująca konkretny algorytm.

Parameters

<i>Tab</i>	- typu <a href="#">Zasobnik</a> <T>*, implementacja zasobnika
<i>dane</i>	- typu T*, dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu int, liczba danych dla zasobnika

Implements [Algorytm](#)< int >.

Here is the call graph for this function:



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

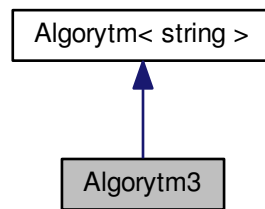
- [Algorytm2.hh](#)
- [Algorytm2.cpp](#)

## 4.4 Algorytm3 Class Reference

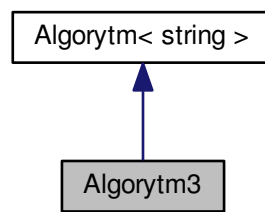
Klasa [Algorytm3](#).

```
#include <Algorytm3.hh>
```

Inheritance diagram for Algorytm3:



Collaboration diagram for Algorytm3:



## Public Member Functions

- void [alokujdane](#) ([Zasobnik](#)< string > \*Tab, string \*dane, int liczba\_danych)  
*Metoda alokujaca na zasobniku dane.*
- void [wykonajalgorytm](#) ([Zasobnik](#)< string > \*Tab, string \*dane, int liczba\_danych)  
*Metoda wykonujaca konkretny algorytm.*

### 4.4.1 Detailed Description

Klasa [Algorytm3](#).

### 4.4.2 Member Function Documentation

4.4.2.1 void `Algorytm3::alokujdane ( Zasobnik< string > * Tab, string * dane, int liczba_danych )` [virtual]

Metoda alokujaca na zasobniku dane.

## Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implements [Algorytm< string >](#).

Here is the call graph for this function:



4.4.2.2 `void Algorytm3::wykonajalgorytm ( Zasobnik< string > \* Tab, string * dane, int liczba_danych ) [virtual]`

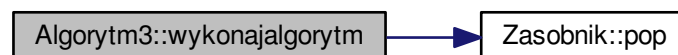
Metoda wykonująca konkretny algorytm.

## Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implements [Algorytm< string >](#).

Here is the call graph for this function:



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

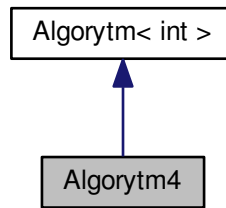
- [Algorytm3.hh](#)
- [Algorytm3.cpp](#)

## 4.5 Algorytm4 Class Reference

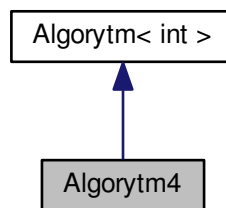
Klasa [Algorytm4](#).

```
#include <Algorytm4.hh>
```

Inheritance diagram for Algorytm4:



Collaboration diagram for Algorytm4:



## Public Member Functions

- [~Algorytm4](#) ()
- void [alokujdane](#) ([Zasobnik](#)< int > \*, int \*, int)
- void [wykonajalgorytm](#) ([Zasobnik](#)< int > \*, int \*, int)
- int \* [mergesort](#) (int \*, int)
- int \* [scal](#) (int \*, int \*, int, int \*, int)

### 4.5.1 Detailed Description

Klasa [Algorytm4](#).

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 [Algorytm4::~Algorytm4](#) ( ) [inline]

### 4.5.3 Member Function Documentation

4.5.3.1 void Algorytm4::alokuj dane ( Zasobnik< int > \* Tab, int \* dane, int liczba\_danych ) [virtual]

Metoda alokująca na zasobniku dane.

## Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implements [Algorytm< int >](#).

Here is the call graph for this function:



#### 4.5.3.2 `int * Algorytm4::mergesort ( int * Tab, int wielkosc )`

Here is the call graph for this function:



#### 4.5.3.3 `int * Algorytm4::scal ( int * Tab, int * tab_l, int size_l, int * tab_p, int size_p )`

#### 4.5.3.4 `void Algorytm4::wykonajalgorytm ( Zasobnik< int > * Tab, int * dane, int liczba_danych )` `[virtual]`

Metoda wykonująca konkretny algorytm.

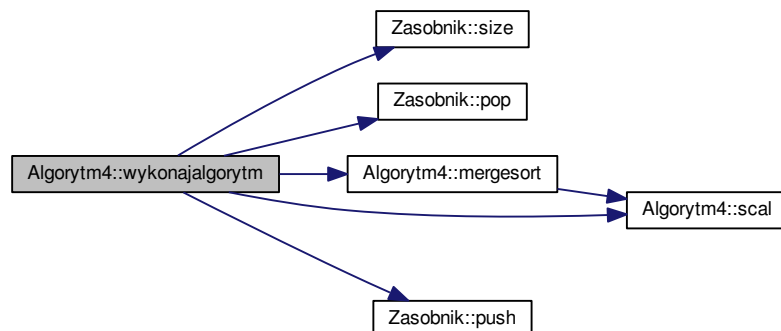
## Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implements [Algorytm< int >](#).



Here is the call graph for this function:



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

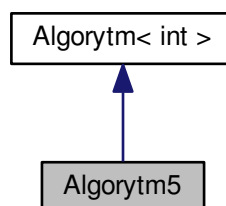
- [Algorytm4.hh](#)
- [Algorytm4.cpp](#)

## 4.6 Algorytm5 Class Reference

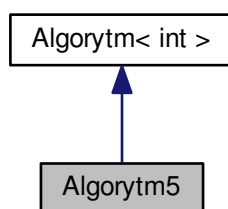
Klasa [Algorytm5](#).

```
#include <Algorytm5.hh>
```

Inheritance diagram for Algorytm5:



Collaboration diagram for Algorytm5:



## Public Member Functions

- void [alokujdane](#) ([Zasobnik<int> \\*](#), int \*, int)  
*Metoda alokujaca na zasobniku dane.*
- void [wykonajalgorytm](#) ([Zasobnik<int> \\*](#), int \*, int)  
*Metoda wykonujaca konkretny algorytm.*

### 4.6.1 Detailed Description

Klasa [Algorytm5](#).

### 4.6.2 Member Function Documentation

4.6.2.1 void [Algorytm5::alokujdane](#) ( [Zasobnik<int> \\*](#) *Tab*, int \* *dane*, int *liczba\_danych* ) [virtual]

Metoda alokujaca na zasobniku dane.

Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implements [Algorytm<int>](#).

4.6.2.2 void [Algorytm5::wykonajalgorytm](#) ( [Zasobnik<int> \\*](#) *Tab*, int \* *dane*, int *liczba\_danych* ) [virtual]

Metoda wykonujaca konkretny algorytm.

Parameters

<i>Tab</i>	- typu <code>Zasobnik&lt;T&gt;*</code> , implementacja zasobnika
<i>dane</i>	- typu <code>T*</code> , dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu <code>int</code> , liczba danych dla zasobnika

Implements [Algorytm<int>](#).

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

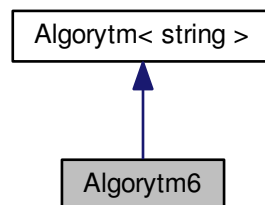
- [Algorytm5.hh](#)
- [Algorytm5.cpp](#)

## 4.7 Algorytm6 Class Reference

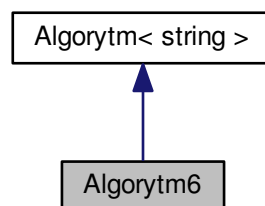
Klasa [Algorytm6](#).

```
#include <Algorytm6.hh>
```

Inheritance diagram for Algorytm6:



Collaboration diagram for Algorytm6:



### Public Member Functions

- [~Algorytm6](#) ()
- void [alokujdane](#) ([Zasobnik](#)< string > \*, string \*, int)  
*Metoda alokujaca na zasobniku dane.*
- void [wykonajalgorytm](#) ([Zasobnik](#)< string > \*, string \*, int)  
*Metoda wykonujaca konkretny algorytm.*

### 4.7.1 Detailed Description

Klasa [Algorytm6](#).

### 4.7.2 Constructor & Destructor Documentation

#### 4.7.2.1 [Algorytm6::~~Algorytm6](#) ( ) `[inline]`

### 4.7.3 Member Function Documentation

4.7.3.1 `void Algorytm6::alokuj dane ( Zasobnik< string > * Tab, string * dane, int liczba_danych ) [virtual]`

Metoda alokująca na zasobniku dane.

Parameters

<i>Tab</i>	- typu Zasobnik<T>*, implementacja zasobnika
<i>dane</i>	- typu T*, dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu int, liczba danych dla zasobnika

Implements [Algorytm< string >](#).

4.7.3.2 `void Algorytm6::wykonajalgorytm ( Zasobnik< string > * Tab, string * dane, int liczba_danych ) [virtual]`

Metoda wykonująca konkretny algorytm.

Parameters

<i>Tab</i>	- typu Zasobnik<T>*, implementacja zasobnika
<i>dane</i>	- typu T*, dane wygenerowane dla implementacji
<i>liczba_danych</i>	- typu int, liczba danych dla zasobnika

Implements [Algorytm< string >](#).

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

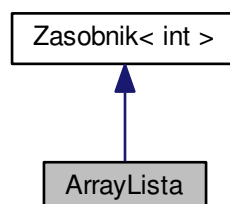
- [Algorytm6.hh](#)
- [Algorytm6.cpp](#)

## 4.8 ArrayLista Class Reference

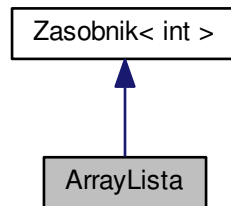
Klasa [ArrayLista](#).

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

Inheritance diagram for ArrayLista:



Collaboration diagram for ArrayLista:



## 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 Listy.*

- 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.*

### 4.8.1 Detailed Description

Klasa [ArrayLista](#).

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 ArrayLista::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.

#### 4.8.2.2 ArrayLista::ArrayLista ( int *wielkosc* )

#### 4.8.2.3 ArrayLista::~~ArrayLista ( )

Destruktor bezparametryczny Listy.

### 4.8.3 Member Function Documentation

#### 4.8.3.1 int ArrayLista::pop ( ) [virtual]

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

wartosc - typu int, wartosc zdejmowana ze stosu.

Implements [Zasobnik< int >](#).

#### 4.8.3.2 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, wartosc umieszczana na stosie.

#### 4.8.3.3 void ArrayLista::push ( int *wartosc* ) [inline],[virtual]

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

##### Parameters

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

Implements [Zasobnik< int >](#).

#### 4.8.3.4 int ArrayLista::size ( ) [virtual]

Metoda zwracająca rozmiar tablicy na której oparta jest lista.

##### Returns

rozmiar - typu int, rozmiar tablicy listy.

Implements [Zasobnik< int >](#).

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

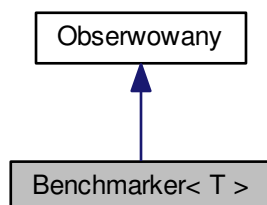
- [ArrayLista.hh](#)
- [ArrayLista.cpp](#)

## 4.9 Benchmark< T > Class Template Reference

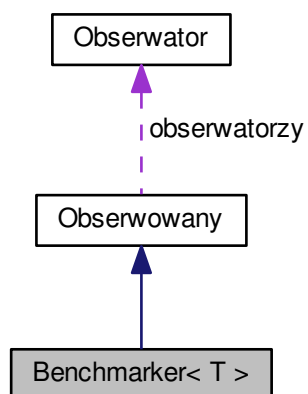
Szablon klasy [Benchmark](#).

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

Inheritance diagram for Benchmark< T >:



Collaboration diagram for Benchmark< T >:



### Public Member Functions

- void [testuj](#) ([Zasobnik](#)< T > \*, [Algorytm](#)< T > \*, T \*, int, int)  
*Szablon metody przeprowadzająca sprawdzenie czasu działania funkcji. Typy: [Lista](#) , [Stos](#) , [Kolejka](#), [HaszTab](#).*
- void [powiadom](#) (int, long int)  
*Metoda powiadamiająca obserwatora o czasie wykonania.*

### Additional Inherited Members

#### 4.9.1 Detailed Description

```
template<typename T>class Benchmark< T >
```

Szablon klasy [Benchmark](#).

## 4.9.2 Member Function Documentation

4.9.2.1 `template<typename T > void Benchmark< T >::powiadom ( int iteracja, long int czas_sredni )`

Metoda powiadamiająca obserwatora o czasie wykonania.

Parameters

<i>iteracja</i>	- typu int, liczba danych - identyfikator iteracji
<i>czas_sredni</i>	- typu long int, czas wykonania operacji

4.9.2.2 `template<typename T > template void Benchmark< T >::testuj ( Zasobnik< T > *, Algorytm< T > *, T *, int, int )`

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

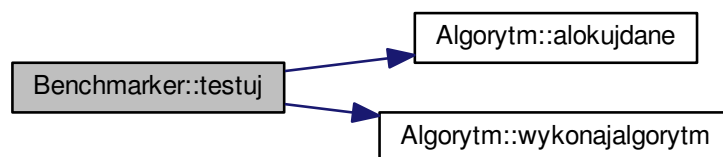
Template Parameters

<i>Tab</i>	- typu T*, wskaznik na zaimplementowany stos/liste/kolejke/tablice haszującą.
<i>dane</i>	- 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 działania funkcji.

Here is the call graph for this function:



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

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

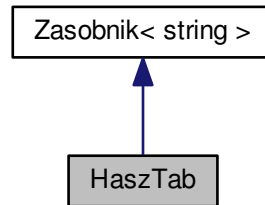
## 4.10 HaszTab Class Reference

Klasa [HaszTab](#).

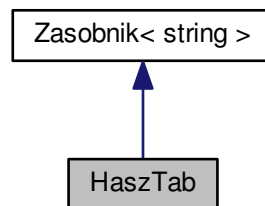


```
#include <HaszTab.hh>
```

Inheritance diagram for HaszTab:



Collaboration diagram for HaszTab:



## 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.*
- string [pop](#) (string klucz\_szukany)
- void [push](#) (string wartosc)  
*Metoda umieszczająca element na zasobniku.*
- string [pop](#) ()  
*Metoda zdejmująca element z zasobnika.*
- int [size](#) ()  
*Metoda zwracająca rozmiar zasobnika.*
- 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 której oparta jest tablica haszująca.*

- int [size\\_k2](#) ()

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

#### 4.10.1 Detailed Description

Klasa [HaszTab](#).

#### 4.10.2 Constructor & Destructor Documentation

##### 4.10.2.1 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.

##### 4.10.2.2 HaszTab::~~HaszTab ( )

Destruktor bezparametryczny tablicy haszującej.

#### 4.10.3 Member Function Documentation

##### 4.10.3.1 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.

##### 4.10.3.2 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 czytowania.

###### 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:



4.10.3.3 `string HaszTab::pop ( string klucz_szukany )`

4.10.3.4 `string HaszTab::pop ( ) [virtual]`

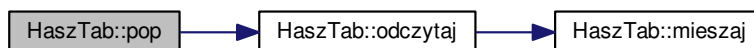
Metoda zdejmująca element z zasobnika.

## Returns

wartosc - typu T, wartosc zdejmowana z zasobnika.

Implements [Zasobnik< string >](#).

Here is the call graph for this function:



4.10.3.5 `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:



#### 4.10.3.6 void HaszTab::push ( string wartosc ) [virtual]

Metoda umieszczająca element na zasobniku.

##### Parameters

<i>wartosc</i>	- typu T, wartosc umieszczana na zasobniku.
----------------	---

Implements [Zasobnik< string >](#).

#### 4.10.3.7 int HaszTab::size ( ) [virtual]

Metoda zwracająca rozmiar zasobnika.

##### Returns

rozmiar - typu int, rozmiar zasobnika.

Implements [Zasobnik< string >](#).

#### 4.10.3.8 int HaszTab::size\_k1 ( )

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

##### Returns

rozmiar - typu int, rozmiar tablicy pierwszej.

#### 4.10.3.9 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:

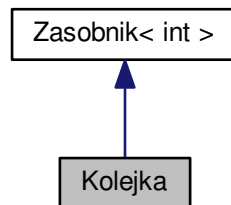
- [HaszTab.hh](#)
- [HaszTab.cpp](#)

## 4.11 Kolejka Class Reference

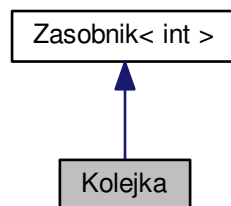
Klasa [Kolejka](#).

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

Inheritance diagram for Kolejka:



Collaboration diagram for Kolejka:



### 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.*

### 4.11.1 Detailed Description

Klasa [Kolejka](#).

### 4.11.2 Constructor & Destructor Documentation

#### 4.11.2.1 `Kolejka::Kolejka ( )`

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

#### 4.11.2.2 `Kolejka::~~Kolejka ( )`

Destruktor bezparametryczny kolejki.

### 4.11.3 Member Function Documentation

#### 4.11.3.1 `int Kolejka::pop ( )` [virtual]

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

##### Returns

wartosc - typu int, wartosc zdejmowana z kolejki.

Implements [Zasobnik< int >](#).

#### 4.11.3.2 `void Kolejka::push ( int wartosc )` [virtual]

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.
----------------	---

Implements [Zasobnik< int >](#).

#### 4.11.3.3 `int Kolejka::size ( )` [virtual]

Metoda zwracająca wielkość kolejki.

##### Returns

rozmiar - typu int, rozmiar kolejki.

Implements [Zasobnik< int >](#).

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

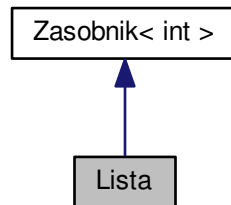
- [Kolejka.hh](#)
- [Kolejka.cpp](#)

## 4.12 Lista Class Reference

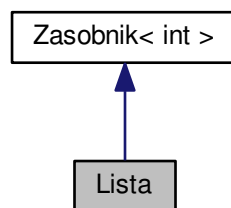
Klasa [Lista](#).

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

Inheritance diagram for Lista:



Collaboration diagram for Lista:



### 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 rozmiar listy.*
- int [pop](#) ()  
*Przeciążenie operacji pop dla listy. Pobiera domyślnie element listy z pozycji 1. Następuje dekrementacja rozmiar listy.*

- `int size ()`

*Metoda zwracajaca wielkosc listy.*

#### 4.12.1 Detailed Description

Klasa [Lista](#).

#### 4.12.2 Constructor & Destructor Documentation

##### 4.12.2.1 `Lista::Lista ( )`

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

##### 4.12.2.2 `Lista::~~Lista ( )`

Destruktor bezparametryczny listy.

#### 4.12.3 Member Function Documentation

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

Metoda zdejmująca element z określonej pozycji listy  $\langle 0, \text{rozmiar} \rangle$ . Metoda dekrementuje rozmiar przy zdejmowaniu elementu.

###### Parameters

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

###### Returns

wartosc - typu `int`, wartosc zdejmowana z listy.

##### 4.12.3.2 `int Lista::pop ( )` `[inline]`, `[virtual]`

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

###### Returns

wartosc - typu `int`, wartosc zdejmowana z listy.

Implements [Zasobnik< int >](#).

##### 4.12.3.3 `void Lista::push ( int wartosc, int pozycja )`

Metoda umieszczająca element określonej pozycji na liście  $\langle 0, \text{rozmiar} \rangle$ . Metoda inkrementuje rozmiar podczas umieszczania elementu na liście.

###### Parameters



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

#### 4.12.3.4 void Lista::push ( int *wartosc* ) [inline],[virtual]

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

##### Parameters

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

Implements [Zasobnik< int >](#).

#### 4.12.3.5 int Lista::size ( ) [virtual]

Metoda zwracajaca wielkosc listy.

##### Returns

rozmiar - typu int,rozmiar listy.

Implements [Zasobnik< int >](#).

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

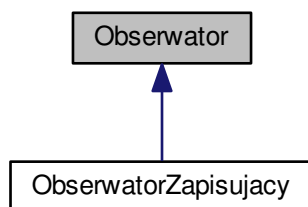
- [Lista.hh](#)
- [Lista.cpp](#)

## 4.13 Obserwator Class Reference

Klasa [Obserwator](#).

```
#include <Obserwator.hh>
```

Inheritance diagram for Obserwator:



### Public Member Functions

- virtual void [odswiez](#) (int k, long int sredni\_czas)=0  
*Metoda odswiezajaca obserwatora.*

### 4.13.1 Detailed Description

Klasa [Obserwator](#).

### 4.13.2 Member Function Documentation

4.13.2.1 `virtual void Obserwator::odswiez ( int k, long int sredni_czas )` `[pure virtual]`

Metoda odswiezajaca obserwatora.

Parameters

<i>k</i>	-typu int, ilosc danych na obserwowany obiekcie
<i>sredni_czas</i>	- typu long int, sredni czas wykonania operacji

Implemented in [ObserwatorZapisujacy](#).

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

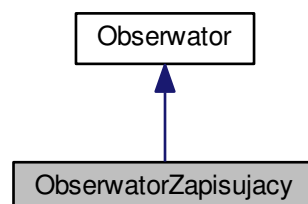
- [Obserwator.hh](#)

## 4.14 ObserwatorZapisujacy Class Reference

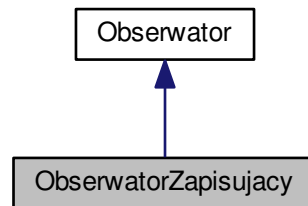
Klasa [ObserwatorZapisujacy](#).

```
#include <ObserwatorZapisujacy.hh>
```

Inheritance diagram for ObserwatorZapisujacy:



Collaboration diagram for ObserwatorZapisujacy:



### Public Member Functions

- void [odswiez](#) (int, long int)

*Metoda odswiezajaca obserwatora.*

#### 4.14.1 Detailed Description

Klasa [ObserwatorZapisujacy](#).

#### 4.14.2 Member Function Documentation

4.14.2.1 void `ObserwatorZapisujacy::odswiez ( int k, long sredni_czas )` `[virtual]`

Metoda odswiezajaca obserwatora.

Parameters

<i>k</i>	-typu int, ilosc danych na obserwowany obiekcie
<i>sredni_czas</i>	- typu long int, sredni czas wykonania operacji

Implements [Obserwator](#).

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

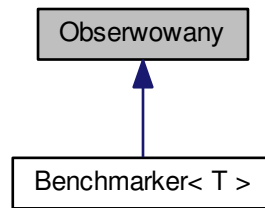
- [ObserwatorZapisujacy.hh](#)
- [ObserwatorZapisujacy.cpp](#)

## 4.15 Obserwowany Class Reference

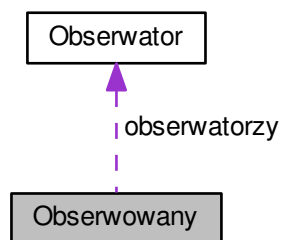
Szablon klasy [Obserwowany](#).

```
#include <Obserwowany.hh>
```

Inheritance diagram for Obserwowany:



Collaboration diagram for Obserwowany:



## Public Member Functions

- void `dodaj` (`Obserwator` \*Obs)  
*Metoda dodajaca obserwatora do obiektu.*
- void `usun` (`Obserwator` \*Obs)  
*Metoda usuwajaca obserwatora z obiektu.*

## Protected Attributes

- `Obserwator` \* `obserwatorzy`

### 4.15.1 Detailed Description

Szablon klasy `Obserwowany`.

### 4.15.2 Member Function Documentation

4.15.2.1 void Obserwowany::dodaj( **Obserwator** \* *Obs* ) [inline]

Metoda dodająca obserwatora do obiektu.

## Parameters

<i>Obs</i>	- typu Obserwator*, wskaznik na danego obserwatora
------------	--

4.15.2.2 void Obserwowany::usun ( Obserwator \* *Obs* ) [inline]

Metoda usuwajaca obserwatora z obiektu.

## Parameters

<i>Obs</i>	- typu Obserwator*, wskaznik na danego obserwatora
------------	--

### 4.15.3 Member Data Documentation

4.15.3.1 Obserwator\* Obserwowany::obserwatorzy [protected]

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

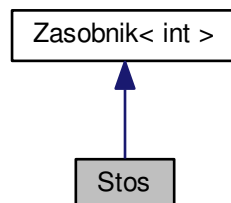
- [Obserwowany.hh](#)

## 4.16 Stos Class Reference

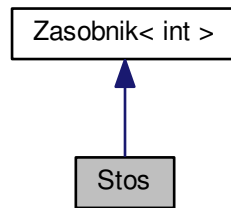
Klasa [Stos](#).

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

Inheritance diagram for Stos:



Collaboration diagram for Stos:



## Public Member Functions

- `Stos ()`  
*Konstruktor bezparametryczny. Konstruktor inicjalizujący strażnika stosu wartością NULL ,oraz rozmiar wartością 0.*
- `~Stos ()`  
*Destruktor bezparametryczny stosu.*
- `void push (int)`  
*Metoda umieszczająca element na stosie. Metoda inkrementuje rozmiar podczas umieszczania elementu na stosie.*
- `int pop ()`  
*Metoda zdejmująca element ze stosu. Metoda dekrementuje rozmiar przy zdejmowaniu ze stosu.*
- `int size ()`  
*Metoda zwracająca wielkość stosu.*

### 4.16.1 Detailed Description

Klasa `Stos`.

### 4.16.2 Constructor & Destructor Documentation

#### 4.16.2.1 `Stos::Stos ( )`

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

#### 4.16.2.2 `Stos::~~Stos ( )`

Destruktor bezparametryczny stosu.

### 4.16.3 Member Function Documentation

#### 4.16.3.1 `int Stos::pop ( ) [virtual]`

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

**Returns**

wartosc - typu int, wartosc zdejmowana ze stosu.

Implements [Zasobnik< int >](#).

**4.16.3.2 void Stos::push ( int wartosc ) [virtual]**

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

**Parameters**

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

Implements [Zasobnik< int >](#).

**4.16.3.3 int Stos::size ( ) [virtual]**

Metoda zwracajaca wielkosc stosu.

**Returns**

rozmiar - typu int, rozmiar stosu.

Implements [Zasobnik< int >](#).

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

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

**4.17 Zasobnik< T > Class Template Reference**

Szablon klasy [Zasobnik](#).

```
#include <Zasobnik.hh>
```

**Public Member Functions**

- virtual [~Zasobnik](#) ()  
*Destruktor wirtualny.*
- virtual void [push](#) (T wartosc)=0  
*Metoda umieszczajaca element na zasobniku.*
- virtual T [pop](#) ()=0  
*Metoda zdejmujaca element z zasobnika.*
- virtual int [size](#) ()=0  
*Metoda zwracajaca rozmiar zasobnika.*

**4.17.1 Detailed Description**

```
template<typename T>class Zasobnik< T >
```

Szablon klasy [Zasobnik](#).



## 4.17.2 Constructor & Destructor Documentation

4.17.2.1 `template<typename T> virtual Zasobnik< T >::~~Zasobnik ( ) [inline],[virtual]`

Destruktor wirtualny.

## 4.17.3 Member Function Documentation

4.17.3.1 `template<typename T> virtual T Zasobnik< T >::pop ( ) [pure virtual]`

Metoda zdejmujaca element z zasobnika.

### Returns

wartosc - typu T, wartosc zdejmowana z zasobnika.

Implemented in [Lista](#), [ArrayLista](#), [HaszTab](#), [Kolejka](#), and [Stos](#).

4.17.3.2 `template<typename T> virtual void Zasobnik< T >::push ( T wartosc ) [pure virtual]`

Metoda umieszczajaca element na zasobniku.

### Parameters

<i>wartosc</i>	- typu T, wartosc umieszczana na zasobniku.
----------------	---

Implemented in [Lista](#), [HaszTab](#), [ArrayLista](#), [Kolejka](#), and [Stos](#).

4.17.3.3 `template<typename T> virtual int Zasobnik< T >::size ( ) [pure virtual]`

Metoda zwracajaca rozmiar zasobnika.

### Returns

rozmiar - typu int,rozmiar zasobnika.

Implemented in [Lista](#), [ArrayLista](#), [HaszTab](#), [Kolejka](#), and [Stos](#).

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

- [Zasobnik.hh](#)



## Chapter 5

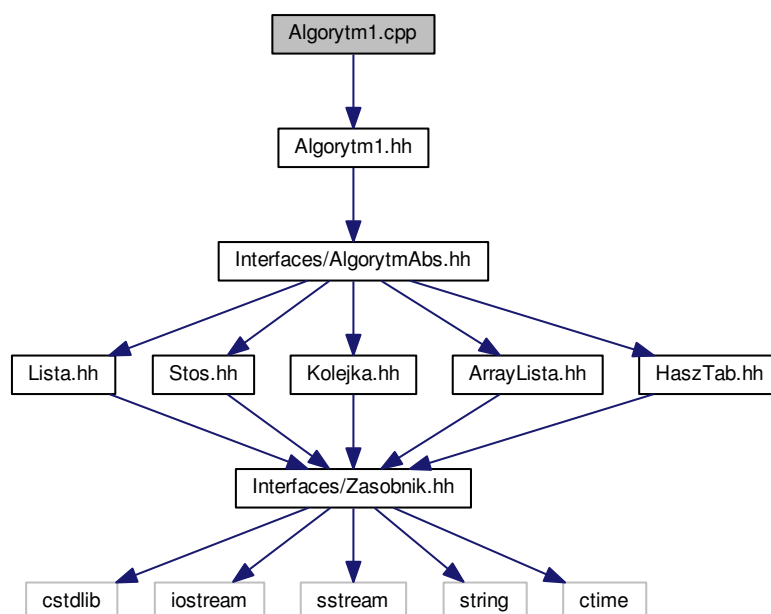
# File Documentation

### 5.1 Algorytm1.cpp File Reference

Metody klasy [Algorytm1](#).

```
#include "Algorytm1.hh"
```

Include dependency graph for Algorytm1.cpp:



#### 5.1.1 Detailed Description

Metody klasy [Algorytm1](#).

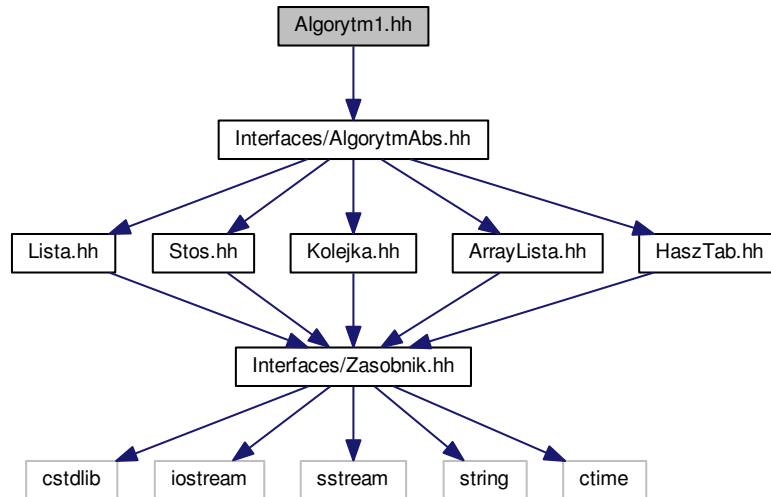
Plik zawiera metody klasy [Algorytm1](#).

## 5.2 Algorytm1.hh File Reference

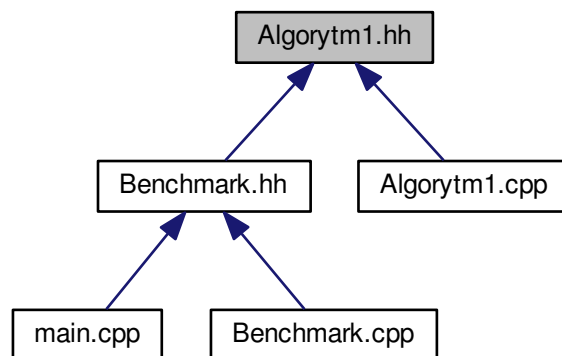
Definicja klasy [Algorytm1](#).

```
#include "Interfaces/AlgorytmAbs.hh"
```

Include dependency graph for Algorytm1.hh:



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



### Classes

- class [Algorytm1](#)

*Klasa [Algorytm1](#).*

### 5.2.1 Detailed Description

Definicja klasy [Algorytm1](#).

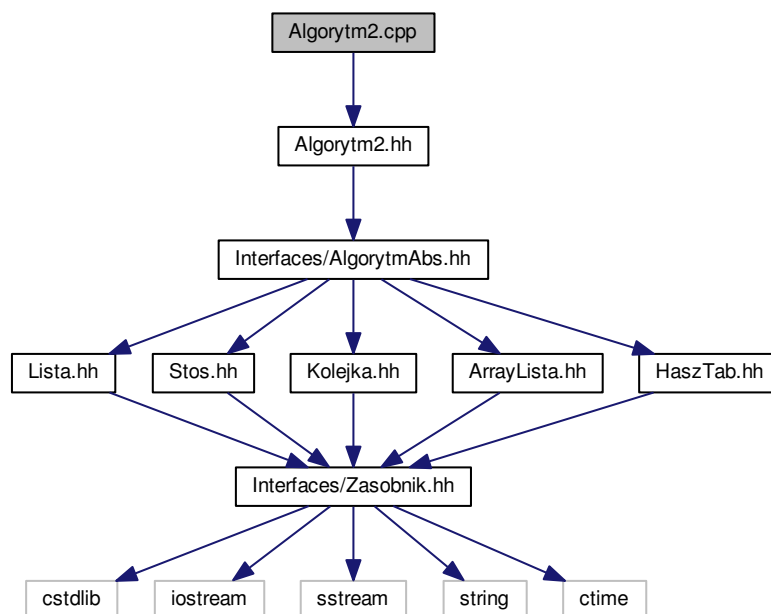
Plik zawiera definicje klasy [Algorytm1](#).

## 5.3 Algorytm2.cpp File Reference

Metody klasy [Algorytm2](#).

```
#include "Algorytm2.hh"
```

Include dependency graph for Algorytm2.cpp:



### 5.3.1 Detailed Description

Metody klasy [Algorytm2](#).

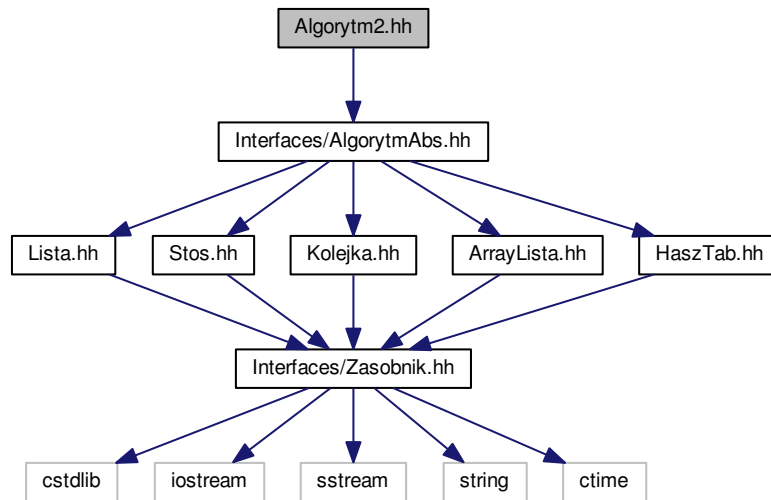
Plik zawiera metody klasy [Algorytm2](#).

## 5.4 Algorytm2.hh File Reference

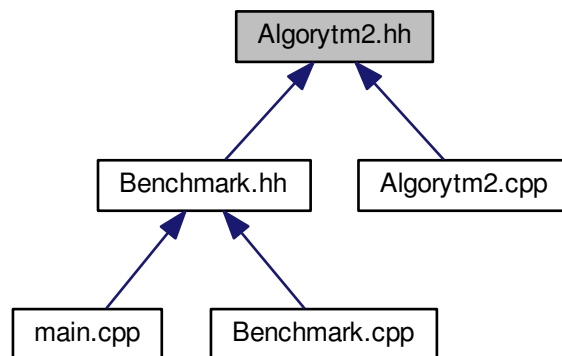
Definicja klasy [Algorytm2](#).

```
#include "Interfaces/AlgorytmAbs.hh"
```

Include dependency graph for Algorytm2.hh:



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



## Classes

- class [Algorytm2](#)  
*Klasa [Algorytm2](#).*

### 5.4.1 Detailed Description

Definicja klasy [Algorytm2](#).

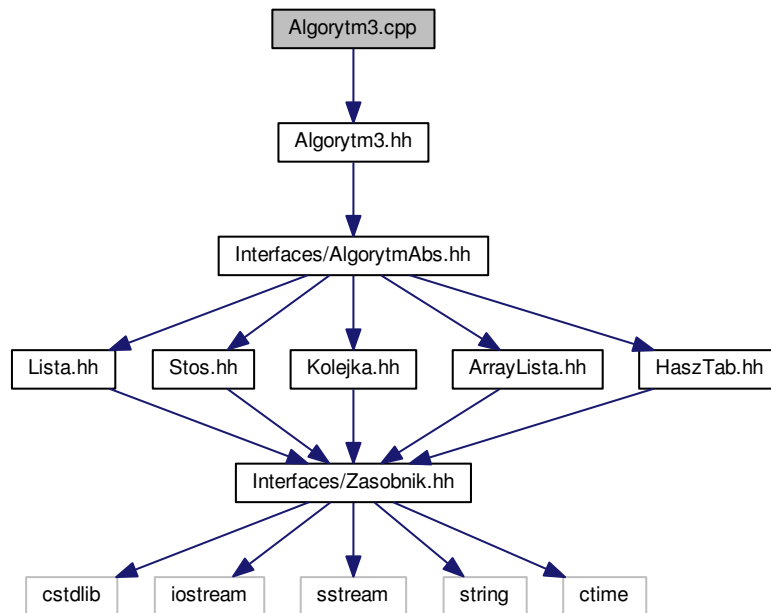
Plik zawiera definicje klasy [Algorytm2](#).

## 5.5 Algorytm3.cpp File Reference

Metody klasy [Algorytm3](#).

```
#include "Algorytm3.hh"
```

Include dependency graph for Algorytm3.cpp:



### 5.5.1 Detailed Description

Metody klasy [Algorytm3](#).

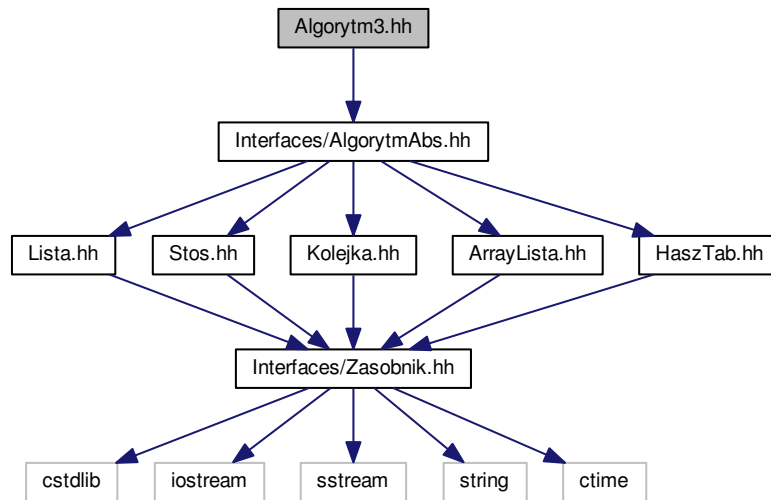
Plik zawiera metody klasy [Algorytm3](#).

## 5.6 Algorytm3.hh File Reference

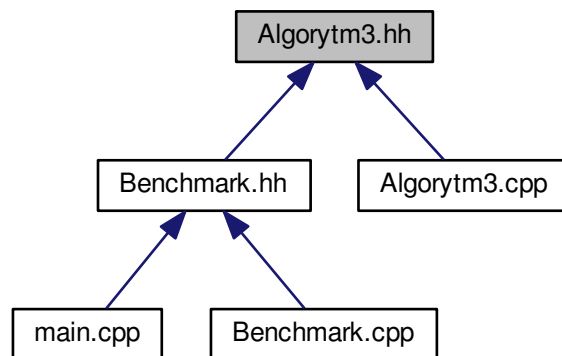
Definicja klasy [Algorytm3](#).

```
#include "Interfaces/AlgorytmAbs.hh"
```

Include dependency graph for Algorytm3.hh:



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



## Classes

- class [Algorytm3](#)  
Klasa [Algorytm3](#).

### 5.6.1 Detailed Description

Definicja klasy [Algorytm3](#).

Plik zawiera definicje klasy [Algorytm3](#).

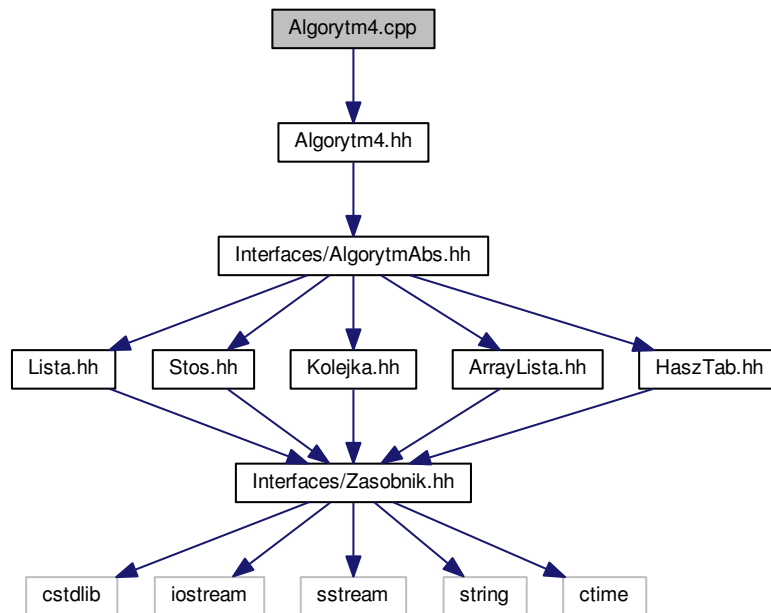


## 5.7 Algorytm4.cpp File Reference

Metody klasy [Algorytm4](#).

```
#include "Algorytm4.hh"
```

Include dependency graph for Algorytm4.cpp:



### 5.7.1 Detailed Description

Metody klasy [Algorytm4](#).

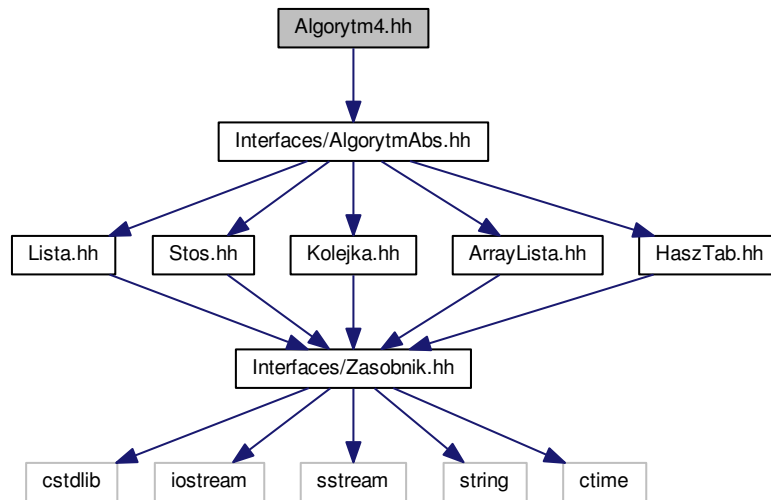
Plik zawiera metody klasy [Algorytm4](#).

## 5.8 Algorytm4.hh File Reference

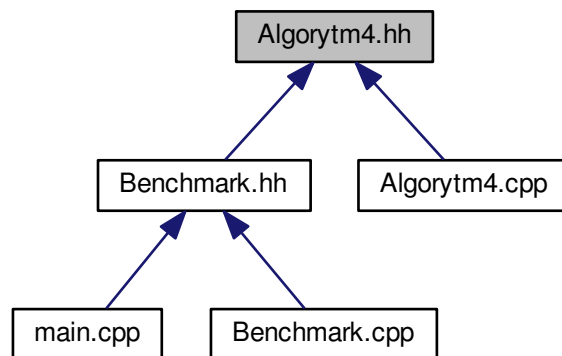
Definicja klasy [Algorytm4](#).

```
#include "Interfaces/AlgorytmAbs.hh"
```

Include dependency graph for Algorytm4.hh:



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



## Classes

- class [Algorytm4](#)  
*Klasa [Algorytm4](#).*

### 5.8.1 Detailed Description

Definicja klasy [Algorytm4](#).

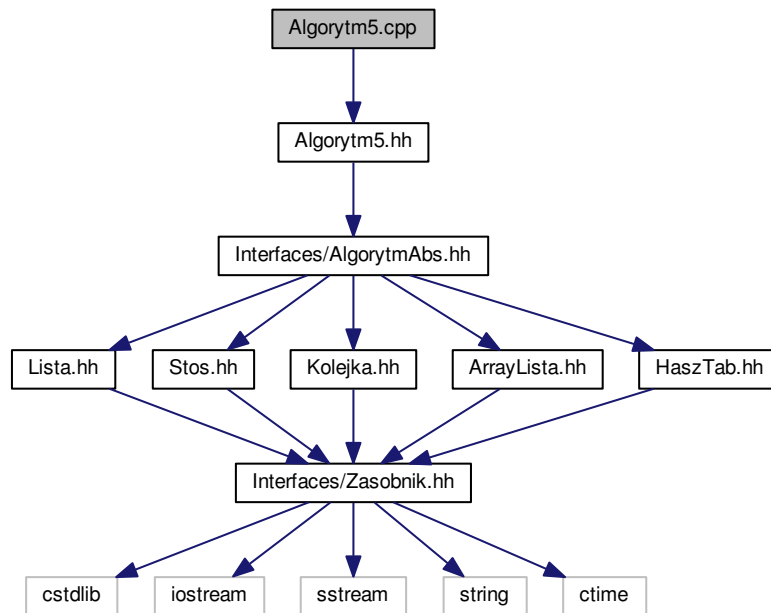
Plik zawiera definicje klasy [Algorytm4](#).

## 5.9 Algorytm5.cpp File Reference

Metody klasy [Algorytm5](#).

```
#include "Algorytm5.hh"
```

Include dependency graph for Algorytm5.cpp:



### 5.9.1 Detailed Description

Metody klasy [Algorytm5](#).

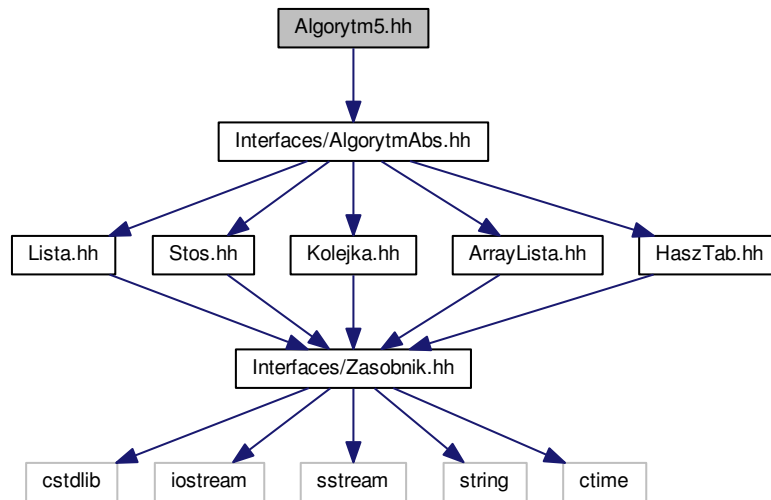
Plik zawiera metody klasy [Algorytm5](#).

## 5.10 Algorytm5.hh File Reference

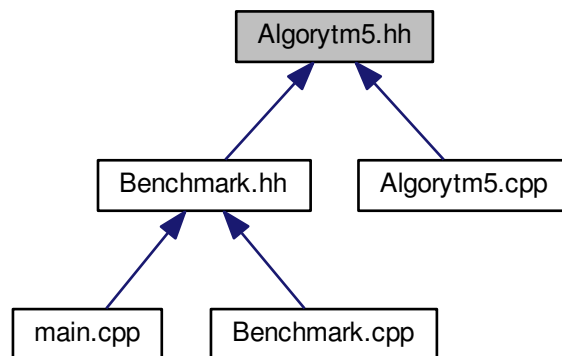
Definicja klasy [Algorytm5](#).

```
#include "Interfaces/AlgorytmAbs.hh"
```

Include dependency graph for Algorytm5.hh:



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



## Classes

- class [Algorytm5](#)  
*Klasa [Algorytm5](#).*

### 5.10.1 Detailed Description

Definicja klasy [Algorytm5](#).

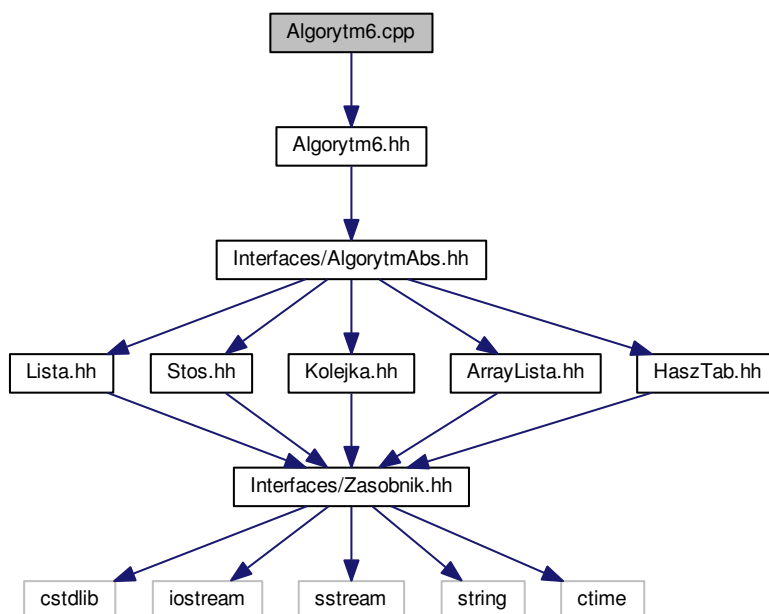
Plik zawiera definicje klasy [Algorytm5](#).

## 5.11 Algorytm6.cpp File Reference

Metody klasy [Algorytm6](#).

```
#include "Algorytm6.hh"
```

Include dependency graph for Algorytm6.cpp:



### 5.11.1 Detailed Description

Metody klasy [Algorytm6](#).

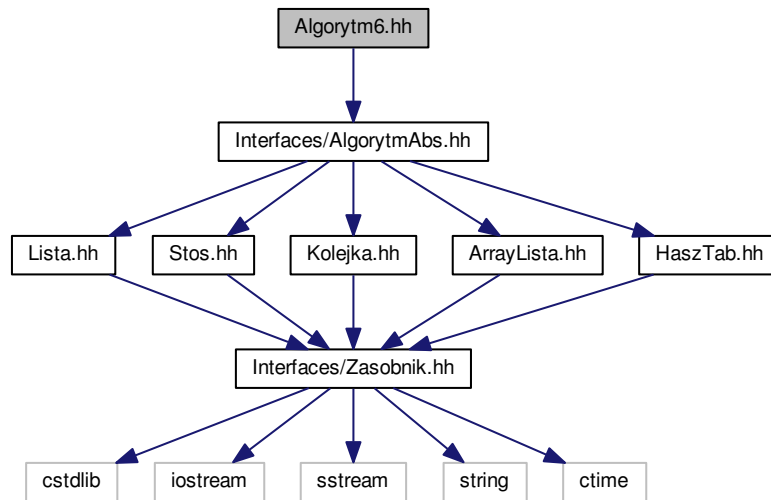
Plik zawiera metody klasy [Algorytm6](#).

## 5.12 Algorytm6.hh File Reference

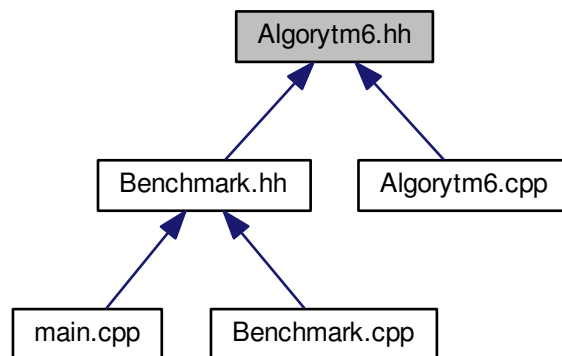
Definicja klasy [Algorytm6](#).

```
#include "Interfaces/AlgorytmAbs.hh"
```

Include dependency graph for Algorytm6.hh:



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



## Classes

- class [Algorytm6](#)  
*Klasa [Algorytm6](#).*

### 5.12.1 Detailed Description

Definicja klasy [Algorytm6](#).

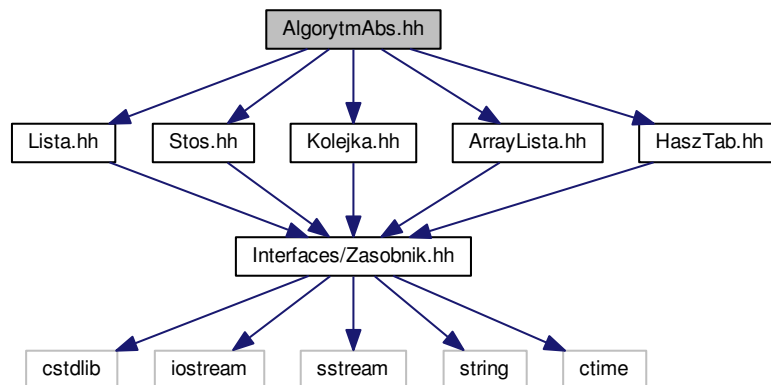
Plik zawiera definicje klasy [Algorytm6](#).

## 5.13 AlgorytmAbs.hh File Reference

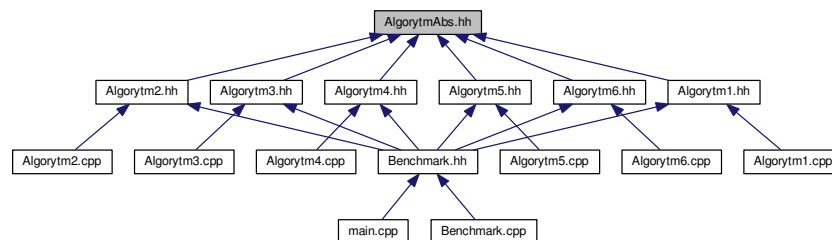
Definicja szablonu klasy abstrakcyjnej [Algorytm](#).

```
#include "Lista.hh"
#include "Stos.hh"
#include "Kolejka.hh"
#include "ArrayLista.hh"
#include "HaszTab.hh"
```

Include dependency graph for AlgorytmAbs.hh:



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



### Classes

- class [Algorytm](#)< T >  
Szablon klasy [Algorytm](#).

### 5.13.1 Detailed Description

Definicja szablonu klasy abstrakcyjnej [Algorytm](#).

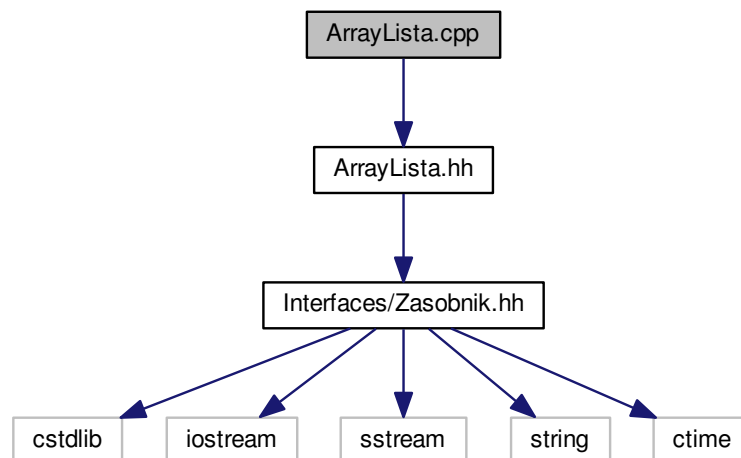
Plik zawiera definicje szablonu klasy abstrakcyjnej [Algorytm](#).

## 5.14 ArrayLista.cpp File Reference

Metody klasy [ArrayLista](#).

```
#include "ArrayLista.hh"
```

Include dependency graph for ArrayLista.cpp:



### 5.14.1 Detailed Description

Metody klasy [ArrayLista](#).

Plik zawiera metody klasy [ArrayLista](#).

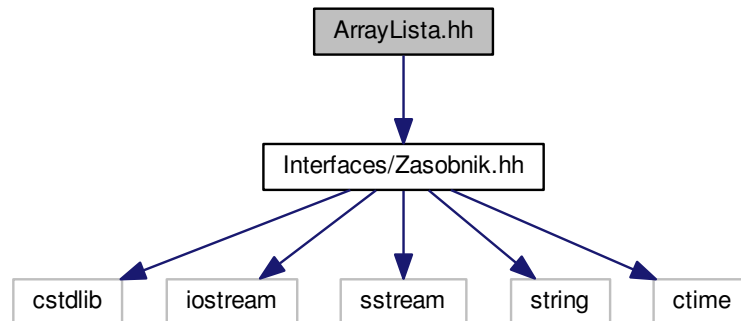
## 5.15 ArrayLista.hh File Reference

Definicja klasy [ArrayLista](#).

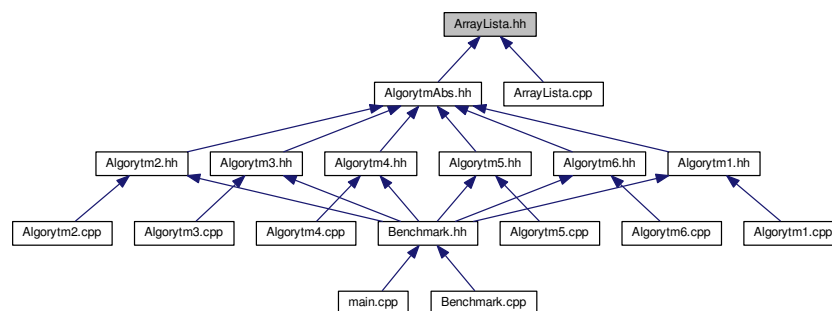


```
#include "Interfaces/Zasobnik.hh"
```

Include dependency graph for ArrayLista.hh:



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



## Classes

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

### 5.15.1 Detailed Description

Definicja klasy [ArrayLista](#).

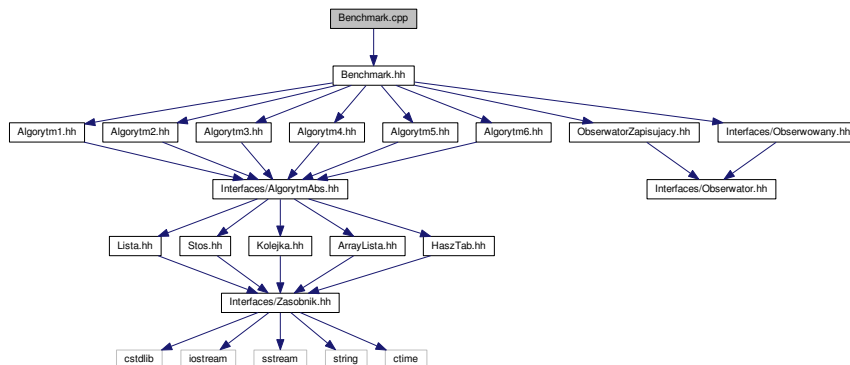
Plik zawiera definicje klasy modulujacej pojecie listy jednokierunkowej opartej na tablicy dynamicznej.

## 5.16 Benchmark.cpp File Reference

Metody klasy [Benchmark](#).

```
#include "Benchmark.hh"
```

Include dependency graph for Benchmark.cpp:



### 5.16.1 Detailed Description

Metody klasy [Benchmark](#).

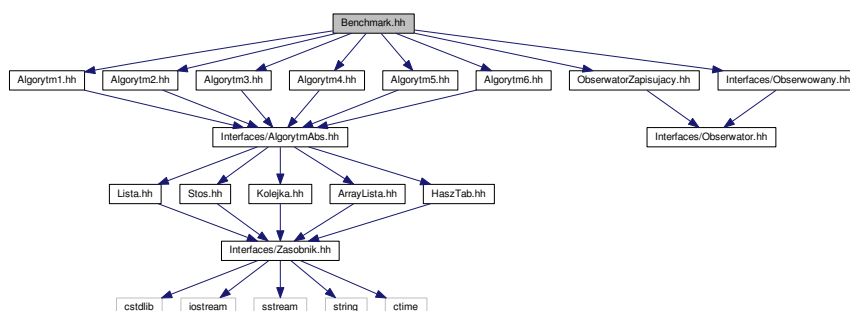
Plik zawiera metody klasy [Benchmark](#).

## 5.17 Benchmark.hh File Reference

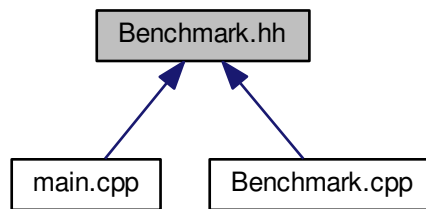
Definicja szablonu klasy [Benchmark](#).

```
#include "Algorytm1.hh"
#include "Algorytm2.hh"
#include "Algorytm3.hh"
#include "Algorytm4.hh"
#include "Algorytm5.hh"
#include "Algorytm6.hh"
#include "ObserwatorZapisujacy.hh"
#include "Interfaces/Obserwowany.hh"
```

Include dependency graph for Benchmark.hh:



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



## Classes

- class [Benchmark< T >](#)  
Szablon klasy [Benchmark](#).

### 5.17.1 Detailed Description

Definicja szablonu klasy [Benchmark](#).

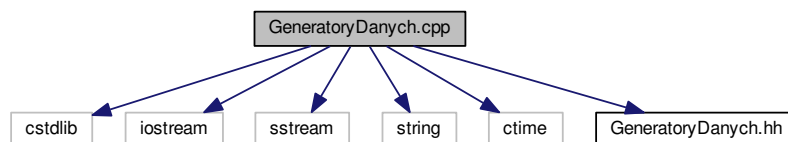
Plik zawiera definicje szablonu klasy [Benchmark](#).

## 5.18 GeneratoryDanych.cpp File Reference

Funkcje generacji danych.

```
#include <cstdlib>
#include <iostream>
#include <sstream>
#include <string>
#include <ctime>
#include "GeneratoryDanych.hh"
```

Include dependency graph for GeneratoryDanych.cpp:



## Functions

- `template<>`  
`int * generujdane (int l_danych)`

*Szablon metody generujacej wartosci losowe danego typu.*

- `template<>`  
`string * generujdane (int l_danych)`

*Szablon metody generujacej wartosci losowe danego typu.*

### 5.18.1 Detailed Description

Funkcje generacji danych.

Plik zawiera funkcje generacji danych .

### 5.18.2 Function Documentation

#### 5.18.2.1 `template<> int* generujdane ( int l_danych )`

Szablon metody generujacej wartosci losowe danego typu.

Parameters

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

Returns

T\* -wskaznik na dany typ, wskaznik na tablice z wygenerowanymi danymi.

#### 5.18.2.2 `template<> string* generujdane ( int l_danych )`

Szablon metody generujacej wartosci losowe danego typu.

Parameters

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

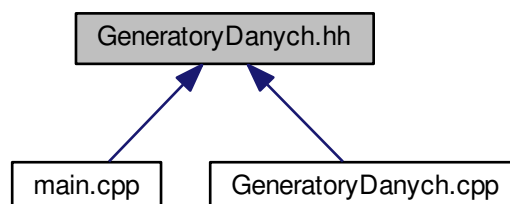
Returns

T\* -wskaznik na dany typ, wskaznik na tablice z wygenerowanymi danymi.

## 5.19 GeneratoryDanych.hh File Reference

Szablon funkcji generacji danych.

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



## Functions

- `template<typename T>`  
`T * generujdane (int l_danych)`

*Szablon metody generujacej wartosci losowe danego typu.*

### 5.19.1 Detailed Description

Szablon funkcji generacji danych.

Plik zawiera szablon funkcji generacji danych.

### 5.19.2 Function Documentation

#### 5.19.2.1 `template<typename T> T* generujdane ( int l_danych )`

Szablon metody generujacej wartosci losowe danego typu.

##### Parameters

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

##### Returns

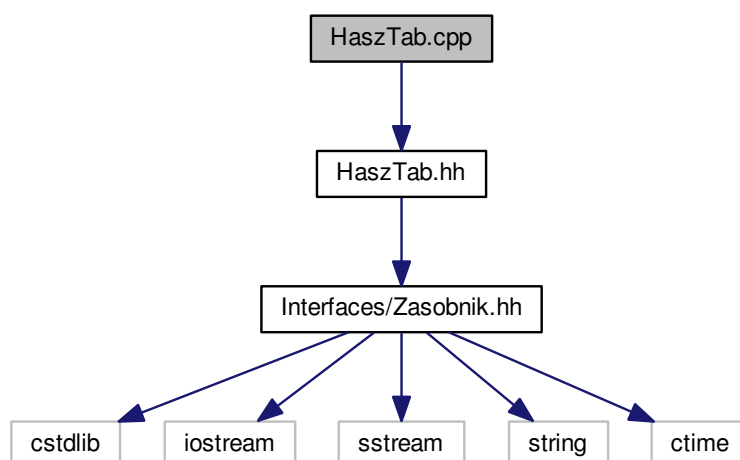
T\* -wskaznik na dany typ, wskaznik na tablice z wygenerowanymi danymi.

## 5.20 HaszTab.cpp File Reference

Metody klasy [HaszTab](#).

```
#include "HaszTab.hh"
```

Include dependency graph for HaszTab.cpp:



### 5.20.1 Detailed Description

Metody klasy [HaszTab](#).

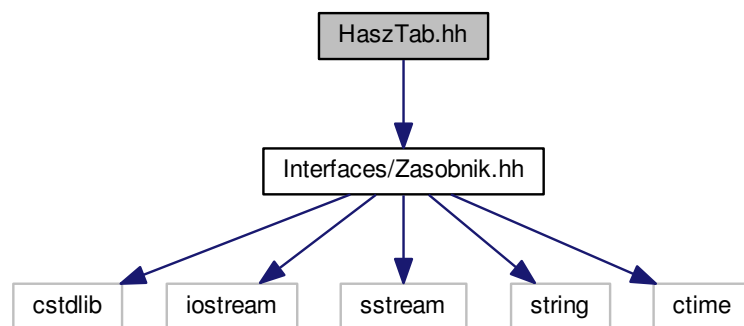
Plik zawiera metody klasy [HaszTab](#).

## 5.21 HaszTab.hh File Reference

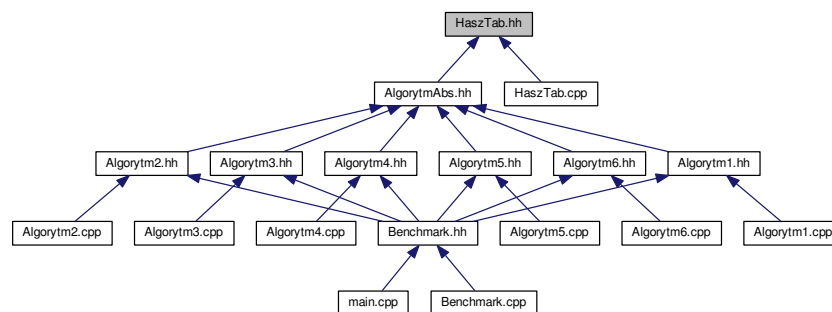
Definicja klasy [HaszTab](#).

```
#include "Interfaces/Zasobnik.hh"
```

Include dependency graph for HaszTab.hh:



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



## Classes

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

### 5.21.1 Detailed Description

Definicja klasy [HaszTab](#).

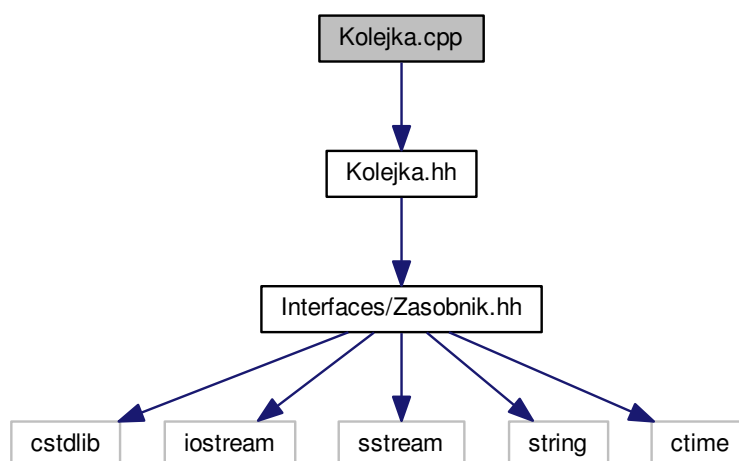
Plik zawiera definicje klasy modulujacej pojecie tablicy haszujacej.

## 5.22 Kolejka.cpp File Reference

Metody klasy [Kolejka](#).

```
#include "Kolejka.hh"
```

Include dependency graph for Kolejka.cpp:



### 5.22.1 Detailed Description

Metody klasy [Kolejka](#).

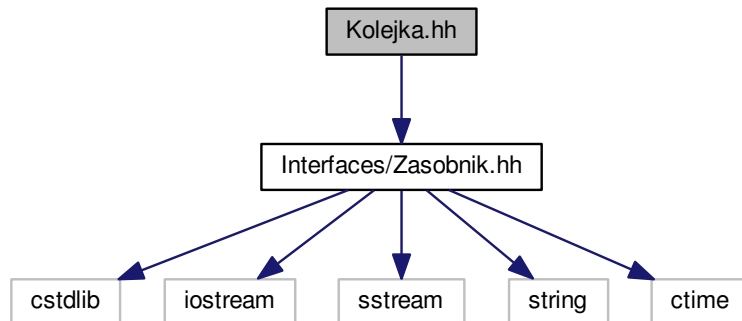
Plik zawiera metody klasy [Kolejka](#).

## 5.23 Kolejka.hh File Reference

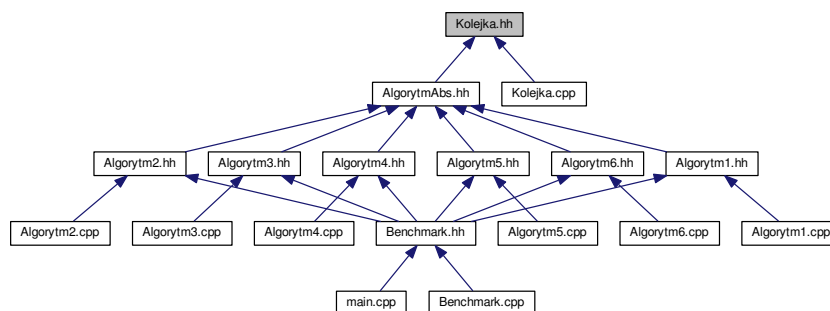
Definicja klasy [Kolejka](#).

```
#include "Interfaces/Zasobnik.hh"
```

Include dependency graph for Kolejka.hh:



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



## Classes

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

### 5.23.1 Detailed Description

Definicja klasy [Kolejka](#).

Plik zawiera definicje klasy [Kolejka](#).

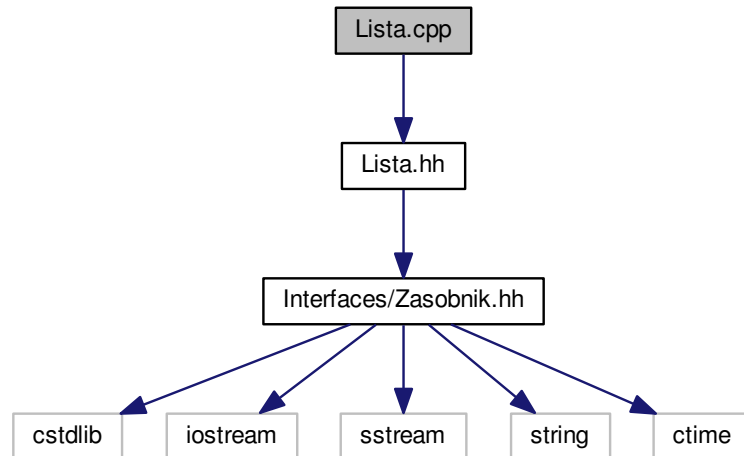
## 5.24 Lista.cpp File Reference

Metody klasy [Lista](#).



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

Include dependency graph for Lista.cpp:



### 5.24.1 Detailed Description

Metody klasy [Lista](#).

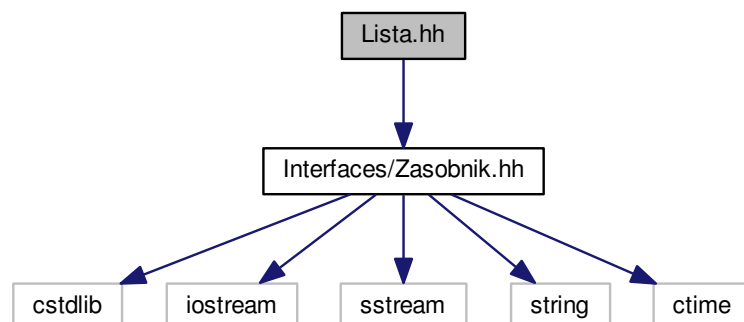
Plik zawiera metody klasy [Lista](#).

## 5.25 Lista.hh File Reference

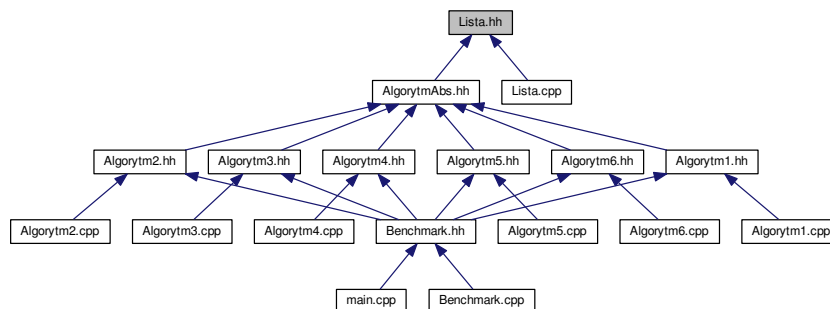
Definicja klasy [Lista](#).

```
#include "Interfaces/Zasobnik.hh"
```

Include dependency graph for Lista.hh:



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



## Classes

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

### 5.25.1 Detailed Description

Definicja klasy [Lista](#).

Plik zawiera definicje klasy modulujacej pojecie listy jednokierunkowej.

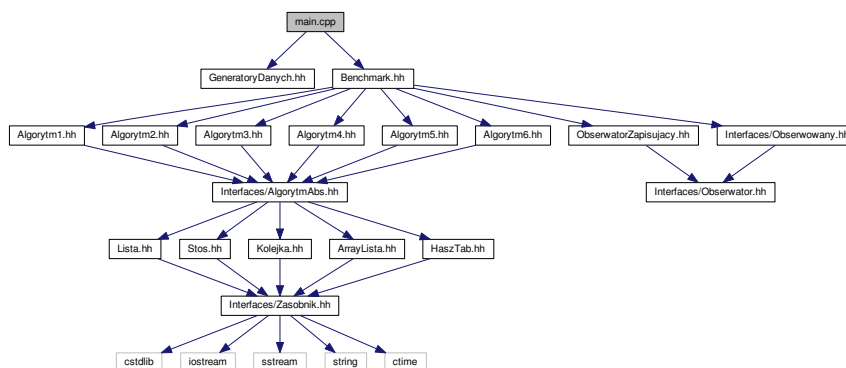
## 5.26 main.cpp File Reference

Modul glowny.

```
#include "GeneratoryDanych.hh"
```

```
#include "Benchmark.hh"
```

Include dependency graph for main.cpp:



## Functions

- int [main](#) (int argc, char \*argv[])  
*Funkcja glowna programu.*

### 5.26.1 Detailed Description

Modul glowny.

Plik zawiera funkcje main.

### 5.26.2 Function Documentation

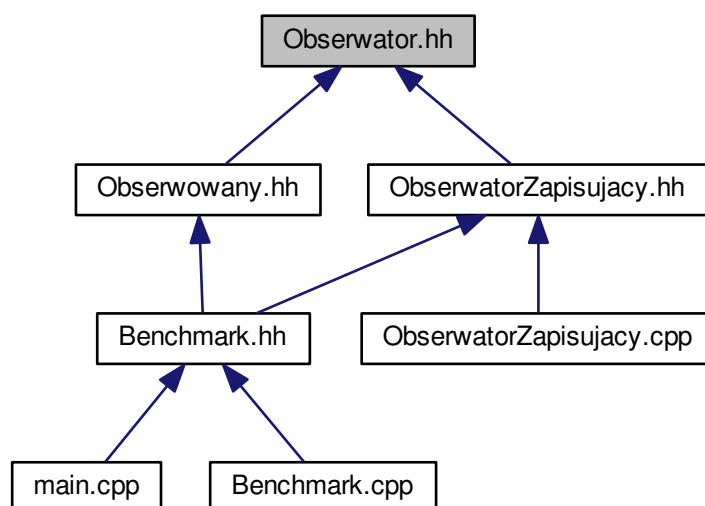
#### 5.26.2.1 `int main ( int argc, char * argv[] )`

Funkcja glowna programu.

## 5.27 Obserwator.hh File Reference

Definicja klasy [Obserwator](#).

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



### Classes

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

### 5.27.1 Detailed Description

Definicja klasy [Obserwator](#).

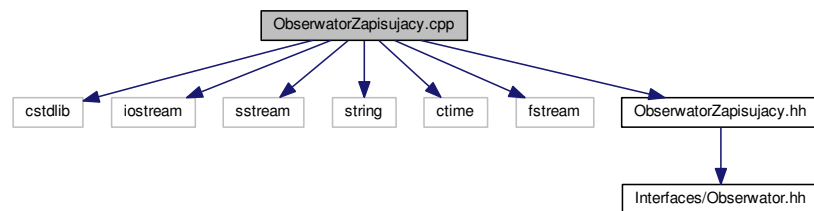
Plik zawiera definicje klasy [Obserwator](#).

## 5.28 ObserwatorZapisujacy.cpp File Reference

Metody klasy [ObserwatorZapisujacy](#).

```
#include <cstdlib>
#include <iostream>
#include <sstream>
#include <string>
#include <ctime>
#include <fstream>
#include "ObserwatorZapisujacy.hh"
```

Include dependency graph for ObserwatorZapisujacy.cpp:



### 5.28.1 Detailed Description

Metody klasy [ObserwatorZapisujacy](#).

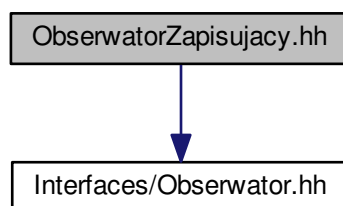
Plik zawiera metody klasy [ObserwatorZapisujacy](#).

## 5.29 ObserwatorZapisujacy.hh File Reference

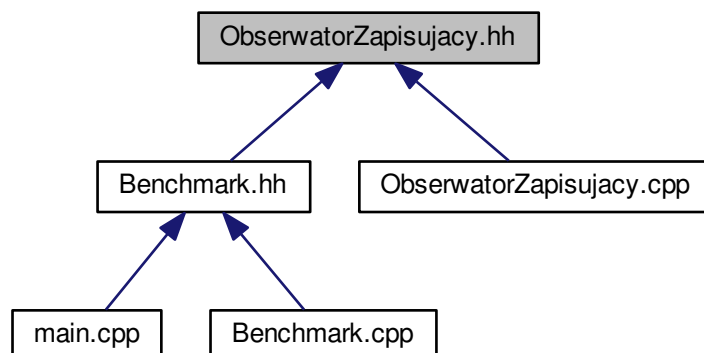
Definicja klasy [ObserwatorZapisujacy](#).

```
#include "Interfaces/Obserwator.hh"
```

Include dependency graph for ObserwatorZapisujacy.hh:



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



## Classes

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

### 5.29.1 Detailed Description

Definicja klasy [ObserwatorZapisujacy](#).

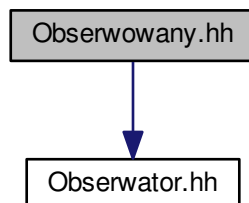
Plik zawiera definicje klasy [ObserwatorZapisujacy](#).

## 5.30 Obserwowany.hh File Reference

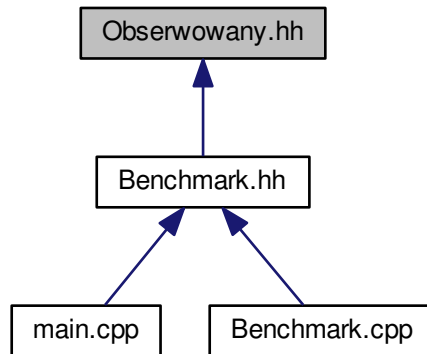
Definicja szablonu klasy abstrakcyjnej [Obserwowany](#).

```
#include "Obserwator.hh"
```

Include dependency graph for `Obserwowany.hh`:



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



## Classes

- class [Obserwowany](#)

*Szablon klasy [Obserwowany](#).*

### 5.30.1 Detailed Description

Definicja szablonu klasy abstrakcyjnej [Obserwowany](#).

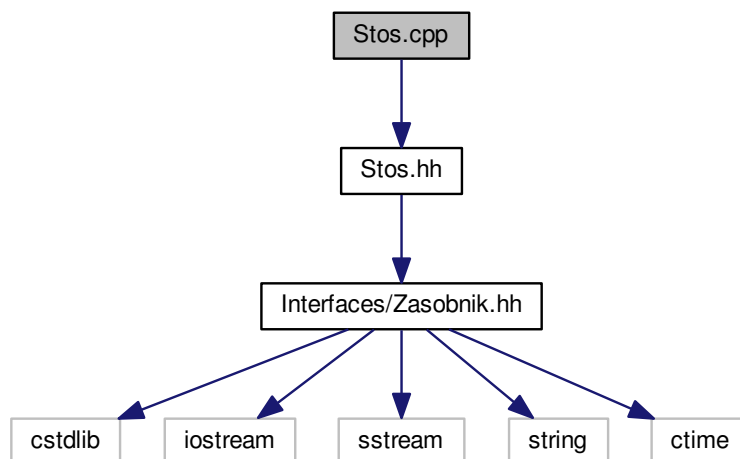
Plik zawiera definicje szablonu klasy abstrakcyjnej [Obserwowany](#).

## 5.31 Stos.cpp File Reference

Metody klasy [Stos](#).

```
#include "Stos.hh"
```

Include dependency graph for Stos.cpp:



### 5.31.1 Detailed Description

Metody klasy [Stos](#).

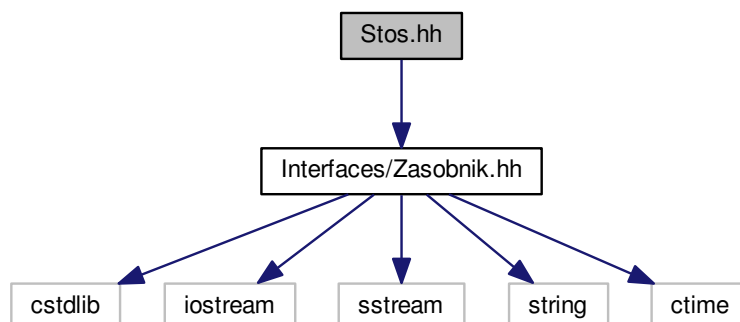
Plik zawiera metody klasy [Stos](#).

## 5.32 Stos.hh File Reference

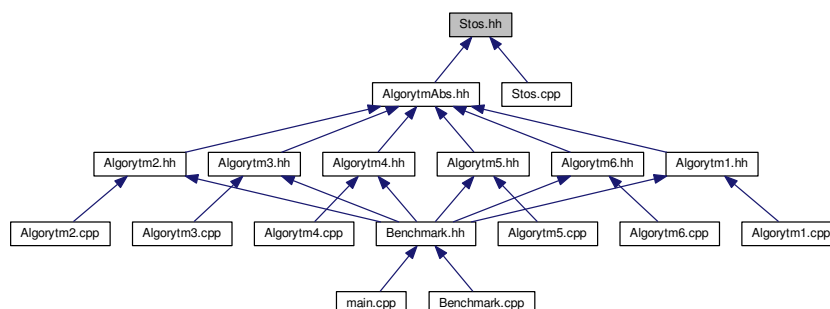
Definicja klasy [Stos](#).

```
#include "Interfaces/Zasobnik.hh"
```

Include dependency graph for Stos.hh:



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



## Classes

- class [Stos](#)

Klasa [Stos](#).

### 5.32.1 Detailed Description

Definicja klasy [Stos](#).

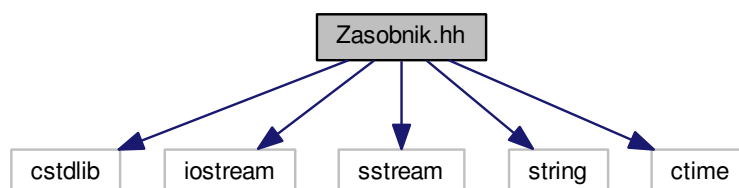
Plik zawiera definicje klasy [Stos](#).

## 5.33 Zasobnik.hh File Reference

Definicja szablonu klasy abstrakcyjnej [Zasobnik](#).

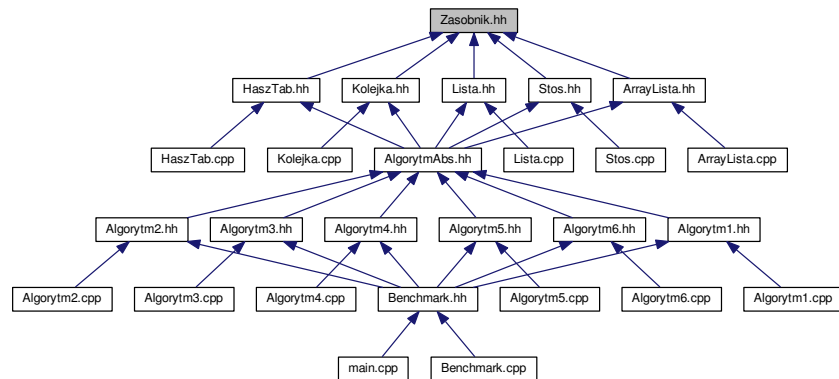
```
#include <cstdlib>
#include <iostream>
#include <sstream>
#include <string>
#include <ctime>
```

Include dependency graph for Zasobnik.hh:





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



## Classes

- class [Zasobnik< T >](#)  
Szablon klasy [Zasobnik](#).

### 5.33.1 Detailed Description

Definicja szablonu klasy abstrakcyjnej [Zasobnik](#).

Plik zawiera definicje szablonu klasy abstrakcyjnej [Zasobnik](#).



# Index

- ~Algorytm
  - Algorytm, 7
- ~Algorytm1
  - Algorytm1, 9
- ~Algorytm4
  - Algorytm4, 14
- ~Algorytm6
  - Algorytm6, 19
- ~ArrayLista
  - ArrayLista, 22
- ~HaszTab
  - HaszTab, 26
- ~Kolejka
  - Kolejka, 30
- ~Lista
  - Lista, 32
- ~Stos
  - Stos, 39
- ~Zasobnik
  - Zasobnik, 41
- Algorytm
  - ~Algorytm, 7
  - alokujdane, 7
  - wykonajalgorytm, 8
- Algorytm < T >, 7
- Algorytm1, 8
  - ~Algorytm1, 9
  - alokujdane, 9
  - wykonajalgorytm, 9
- Algorytm1.cpp, 43
- Algorytm1.hh, 44
- Algorytm2, 10
  - alokujdane, 11
  - wykonajalgorytm, 11
- Algorytm2.cpp, 45
- Algorytm2.hh, 45
- Algorytm3, 11
  - alokujdane, 12
  - wykonajalgorytm, 13
- Algorytm3.cpp, 47
- Algorytm3.hh, 47
- Algorytm4, 13
  - ~Algorytm4, 14
  - alokujdane, 14
  - mergesort, 16
  - scal, 16
  - wykonajalgorytm, 16
- Algorytm4.cpp, 49
- Algorytm4.hh, 49
- Algorytm5, 17
  - alokujdane, 18
  - wykonajalgorytm, 18
- Algorytm5.cpp, 51
- Algorytm5.hh, 51
- Algorytm6, 19
  - ~Algorytm6, 19
  - alokujdane, 20
  - wykonajalgorytm, 20
- Algorytm6.cpp, 53
- Algorytm6.hh, 53
- AlgorytmAbs.hh, 55
- alokujdane
  - Algorytm, 7
  - Algorytm1, 9
  - Algorytm2, 11
  - Algorytm3, 12
  - Algorytm4, 14
  - Algorytm5, 18
  - Algorytm6, 20
- ArrayLista, 20
  - ~ArrayLista, 22
  - ArrayLista, 21
  - pop, 22
  - push, 22
  - size, 22
- ArrayLista.cpp, 56
- ArrayLista.hh, 56
- Benchmark.cpp, 57
- Benchmark.hh, 58
- Benchmarker
  - powiadom, 24
  - testuj, 24
- Benchmarker < T >, 23
- dodaj
  - Obserwowany, 36
- GeneratoryDanych.cpp, 59
  - generujdane, 60
- GeneratoryDanych.hh, 60
  - generujdane, 61
- generujdane
  - GeneratoryDanych.cpp, 60
  - GeneratoryDanych.hh, 61
- HaszTab, 24
  - ~HaszTab, 26
  - HaszTab, 26

- mieszaj, [26](#)
  - odczytaj, [26](#)
  - pop, [27](#)
  - push, [27](#), [28](#)
  - size, [28](#)
  - size\_k1, [28](#)
  - size\_k2, [28](#)
- HaszTab.cpp, [61](#)
- HaszTab.hh, [62](#)
- Kolejka, [29](#)
  - ~Kolejka, [30](#)
  - Kolejka, [30](#)
  - pop, [30](#)
  - push, [30](#)
  - size, [30](#)
- Kolejka.cpp, [63](#)
- Kolejka.hh, [63](#)
- Lista, [31](#)
  - ~Lista, [32](#)
  - Lista, [32](#)
  - pop, [32](#)
  - push, [32](#), [33](#)
  - size, [33](#)
- Lista.cpp, [64](#)
- Lista.hh, [65](#)
- main
  - main.cpp, [67](#)
- main.cpp, [66](#)
  - main, [67](#)
- mergesort
  - Algorytm4, [16](#)
- mieszaj
  - HaszTab, [26](#)
- Obserwator, [33](#)
  - odswiez, [34](#)
- Obserwator.hh, [67](#)
- ObserwatorZapisujacy, [34](#)
  - odswiez, [35](#)
- ObserwatorZapisujacy.cpp, [68](#)
- ObserwatorZapisujacy.hh, [68](#)
- obserwatorzy
  - Obserwowany, [38](#)
- Obserwowany, [35](#)
  - dodaj, [36](#)
  - obserwatorzy, [38](#)
  - usun, [38](#)
- Obserwowany.hh, [69](#)
- odczytaj
  - HaszTab, [26](#)
- odswiez
  - Obserwator, [34](#)
  - ObserwatorZapisujacy, [35](#)
- pop
  - ArrayLista, [22](#)
- HaszTab, [27](#)
- Kolejka, [30](#)
- Lista, [32](#)
- Stos, [39](#)
- Zasobnik, [41](#)
- powiadom
  - Benchmark, [24](#)
- push
  - ArrayLista, [22](#)
  - HaszTab, [27](#), [28](#)
  - Kolejka, [30](#)
  - Lista, [32](#), [33](#)
  - Stos, [40](#)
  - Zasobnik, [41](#)
- scal
  - Algorytm4, [16](#)
- size
  - ArrayLista, [22](#)
  - HaszTab, [28](#)
  - Kolejka, [30](#)
  - Lista, [33](#)
  - Stos, [40](#)
  - Zasobnik, [41](#)
- size\_k1
  - HaszTab, [28](#)
- size\_k2
  - HaszTab, [28](#)
- Stos, [38](#)
  - ~Stos, [39](#)
  - pop, [39](#)
  - push, [40](#)
  - size, [40](#)
  - Stos, [39](#)
- Stos.cpp, [70](#)
- Stos.hh, [71](#)
- testuj
  - Benchmark, [24](#)
- usun
  - Obserwowany, [38](#)
- wykonajalgorytm
  - Algorytm, [8](#)
  - Algorytm1, [9](#)
  - Algorytm2, [11](#)
  - Algorytm3, [13](#)
  - Algorytm4, [16](#)
  - Algorytm5, [18](#)
  - Algorytm6, [20](#)
- Zasobnik
  - ~Zasobnik, [41](#)
  - pop, [41](#)
  - push, [41](#)
  - size, [41](#)
- Zasobnik< T >, [40](#)
- Zasobnik.hh, [72](#)