Zadanie 8

Generated by Doxygen 1.8.9.1

Wed May 27 2015 22:46:25

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

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 alokujdane (Zasobnik< T > *Tab, T *dane, int liczba_danych)=0

  Metoda alokujaca na zasobniku dane.
- virtual void wykonajalgorytm (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 >::alokujdane (Zasobnik< T > * *Tab*, T * *dane*, int *liczba_danych*) [pure virtual]

Metoda alokujaca na zasobniku dane.

Parameters

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

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

Implemented in Algorytm1, Algorytm2, Algorytm3, Algorytm4, Algorytm5, and Algorytm6.

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

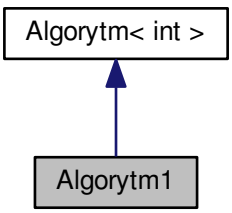
- AlgorytmAbs.hh

4.2 Algorytm1 Class Reference

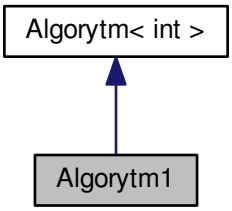
Klasa Algorytm1.

`#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

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

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

Implements Algorytm< int >.

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

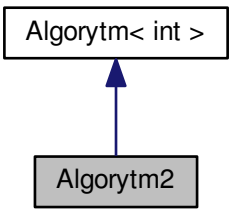
- Algorytm1.hh
- Algorytm1.cpp

4.3 Algorytm2 Class Reference

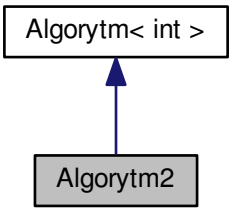
Klasa Algorytm2.

`#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 alokujaca na zasobniku dane.

- void wykonajalgorytm (Zasobnik< int > *Tab, int *dane, int liczba_danych)

  Metoda wykonujaca 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 alokujaca na zasobniku dane.

Parameters

Tab	- typu Zasobnik<T>*, implementacja zasobnika
dane	- typu T*, dane wygenerowane dla implementacji
liczba_danych	- 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]

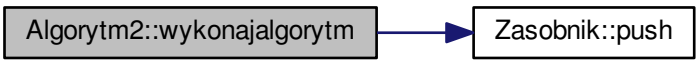
Metoda wykonujaca konkretny algorytm.

Parameters

Tab	- typu Zasobnik<T>*, implementacja zasobnika
dane	- typu T*, dane wygenerowane dla implementacji
liczba_danych	- 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:

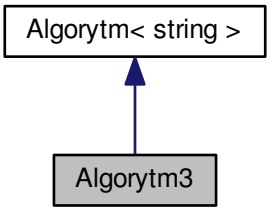
- Algorytm2.hh
- Algorytm2.cpp

4.4 Algorytm3 Class Reference

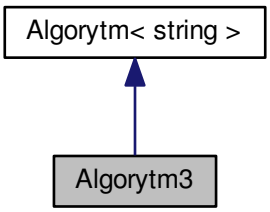
Klasa Algorytm3.

`#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]

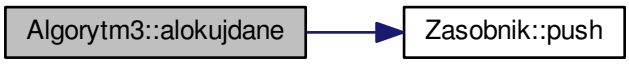
Metoda alokujaca na zasobniku dane.

Parameters

Tab	- typu Zasobnik<T>*, implementacja zasobnika
dane	- typu T*, dane wygenerowane dla implementacji
liczba_danych	- typu int, 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 wykonujaca konkretny algorytm.

Parameters

Tab	- typu Zasobnik<T>*, implementacja zasobnika
dane	- typu T*, dane wygenerowane dla implementacji
liczba_danych	- typu int, 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:

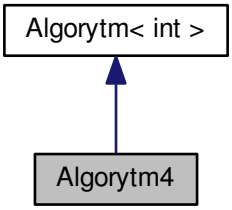
- Algorytm3.hh
- Algorytm3.cpp

4.5 Algorytm4 Class Reference

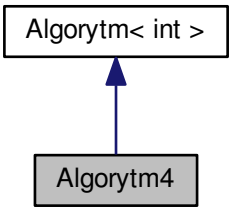
Klasa Algorytm4.

`#include <Algorytm4.hh>`

Inheritance diagram for Algorytm4:



Collaboration diagram for Algorytm4:



Public Member Functions

- ~Algorytm4 ()
- void alokujdane (Zasobnik< int > *, int *, int)

  Metoda alokujaca na zasobniku dane.
- void wykonajalgorytm (Zasobnik< int > *, int *, int)

  Metoda wykonujaca konkretny algorytm.
- 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::alokujdane (Zasobnik< int > * *Tab,* int * *dane,* int *liczba_danych*) [virtual]

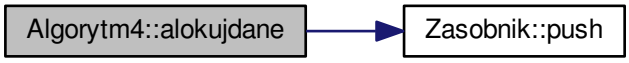
Metoda alokujaca na zasobniku dane.

Parameters

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

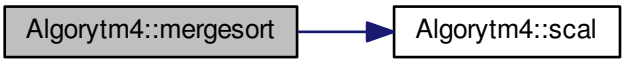
Implements Algorytm< int >.

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]

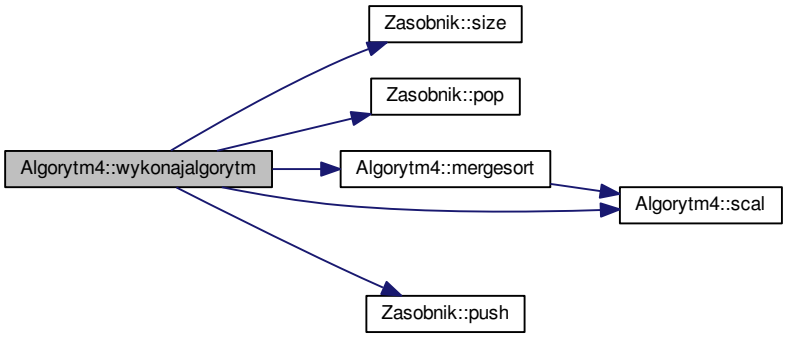
Metoda wykonujaca konkretny algorytm.

Parameters

Tab	- typu Zasobnik<T>∗, implementacja zasobnika
dane	- typu T∗, dane wygenerowane dla implementacji
liczba_danych	- 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:

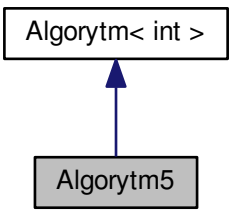
- Algorytm4.hh
- Algorytm4.cpp

4.6 Algorytm5 Class Reference

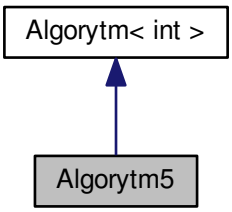
Klasa Algorytm5.

`#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]

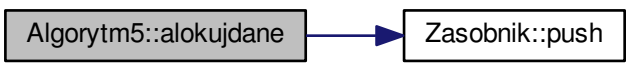
Metoda alokujaca na zasobniku dane.

Parameters

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

Implements Algorytm< int >.

Here is the call graph for this function:



4.6.2.2 void Algorytm5::wykonajalgorytm (Zasobnik< int > * *Tab,* int * *dane,* int *liczba_danych*) [virtual]

Metoda wykonujaca konkretny algorytm.

Parameters

Tab	- typu Zasobnik<T>*, implementacja zasobnika
dane	- typu T*, dane wygenerowane dla implementacji
liczba_danych	- 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:

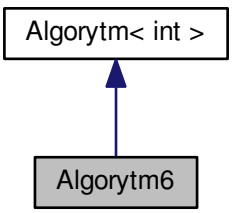
- Algorytm5.hh
- Algorytm5.cpp

4.7 Algorytm6 Class Reference

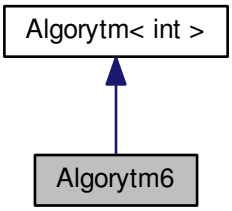
Klasa Algorytm6.

`#include <Algorytm6.hh>`

Inheritance diagram for Algorytm6:



Collaboration diagram for Algorytm6:



Public Member Functions

- ~Algorytm6 ()
- void alokujdane (Zasobnik< int > *, int *, int)

  Metoda alokujaca na zasobniku dane.
- void wykonajalgorytm (Zasobnik< int > *, int *, 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::alokujdane (Zasobnik< int > * *Tab*, int * *dane*, int *liczba_danych*) [virtual]

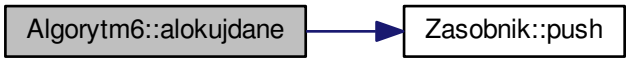
Metoda alokujaca na zasobniku dane.

Parameters

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

Implements Algorytm< int >.

Here is the call graph for this function:



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

Metoda wykonujaca konkretny algorytm.

Parameters

Tab	- typu Zasobnik<T>*, implementacja zasobnika
dane	- typu T*, dane wygenerowane dla implementacji
liczba_danych	- 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:

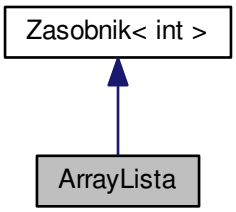
- Algorytm6.hh
- Algorytm6.cpp

4.8 ArrayLista Class Reference

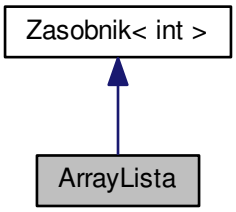
Klasa ArrayLista.

`#include <ArrayLista.hh>`

Inheritance diagram for ArrayLista:



Collaboration diagram for ArrayLista:



Public Member Functions

- ArrayLista ()

  Konstruktor bezparametryczny. Konstruktor inicjalizujacy 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 umieszczajaca element na liscie Metoda inkrementuje rozmiar podczas umieszczania elementu na liscie.
- void push (int wartosc)

  Przeciazenie operacji push. Powieksza liste podczas dodawania elementu do 200%. Nastepuje inkrementacja rozmiar listy.
- int pop ()

  Metoda zdejmujaca element z listy. Metoda dekrementuje ilosc_elementow przy zdejmowaniu z listy. Tablica listy jest zmniejszana podczas zdejmowania elementu o polowe gdy ilosc elementow znajdujacych sie na niej jest rowna polowie jej rozmiaru.
- int size ()

  Metoda zwracajaca rozmiar tablicy na ktorej 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 inicjalizujacy 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 zdejmujaca element z listy. Metoda dekrementuje ilosc_elementow przy zdejmowaniu z listy. Tablica listy jest zmniejszana podczas zdejmowania elementu o polowe gdy ilosc elementow znajdujacych sie na niej jest rowna polowie jej rozmiaru.

Returns

wartosc - typu int, wartosc zdejmowana ze stosu.

Implements Zasobnik< int >.

4.8.3.2 void ArrayLista::push (int *wartosc,* int *zwiekszanie*)

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

Parameters

zwiekszanie	- typu int, mnoznik rozszerzania tablicy podczas dodawania elementow listy .
wartosc	- typu int, wartosc umieszczana na stosie.

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

Przeciazenie operacji push. Powieksza liste podczas dodawania elementu do 200%. Nastepuje inkrementacja rozmiar listy.

Parameters

wartosc	- typu int, wartosc umieszczana na liscie.

Implements Zasobnik< int >.

4.8.3.4 int ArrayLista::size ( ) [virtual]

Metoda zwracajaca rozmiar tablicy na ktorej oparta jest lista.

Returns

rozmiar - typu int,rozmiar tablicy listy.

Implements Zasobnik< int >.

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

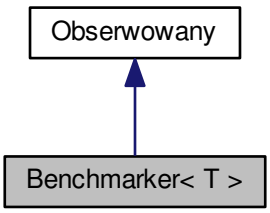
- ArrayLista.hh
- ArrayLista.cpp

4.9 Benchmarker< T > Class Template Reference

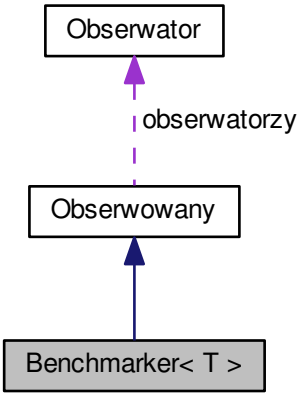
Szablon klasy Benchmarker.

`#include <Benchmark.hh>`

Inheritance diagram for Benchmarker< T >:



Collaboration diagram for Benchmarker< T >:



Public Member Functions

- void testuj (Zasobnik< T > *, Algorytm< T > *, T *, int, int)

  Szablon metody przeprowadzajaca sprawdzenie czasu dzialania funkcji. Typy: Lista , Stos , Kolejka, HaszTab.
- void powiadom (int, long int)

  Metoda powiadamiajaca obserwatora o czasie wykonania.

Additional Inherited Members

4.9.1 Detailed Description

template<typename T>class Benchmarker< T >

Szablon klasy Benchmarker.

4.9.2 Member Function Documentation

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

Metoda powiadamiajaca obserwatora o czasie wykonania.

Parameters

iteracja	- typu int, liczba danych - identyfikator iteracji
czas_sredni	- typu long int, czas wykonania operacji

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

Szablon metody przeprowadzajaca sprawdzenie czasu dzialania funkcji. Typy: Lista , Stos , Kolejka, HaszTab.

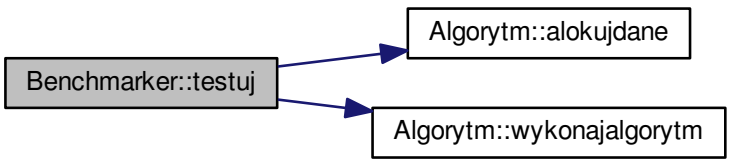
Template Parameters

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

Returns

czas_calkowity_usredniony - typu long int, czas sredni dzialania funkcji.

Here is the call graph for this function:



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

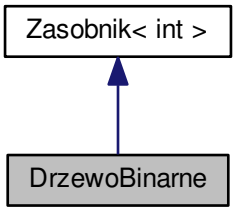
- Benchmark.hh
- Benchmark.cpp

4.10 DrzewoBinarne Class Reference

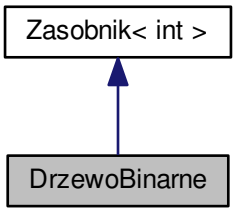
Klasa DrzewoBinarne.

`#include <DrzewoBinarne.hh>`

Inheritance diagram for DrzewoBinarne:



Collaboration diagram for DrzewoBinarne:



Public Member Functions

- DrzewoBinarne ()

  Konstruktor bezparametryczny. Konstruktor inicjalizujacy drzewo binarne. korzen - inicjalizowany wartoscia NULL rozmiar - inicjalizowany wartoscia 0.
- ~DrzewoBinarne ()

  Destruktor bezparametryczny drzewa binarnego.
- void push (int wartosc)

  Metoda umieszczajaca element na drzewie binarnym.
- int pop ()

Metoda zdejmujaca element z drzewa binarnego.

- int pop (int szukana)

  Metoda zdejmujaca dany element z drzewa binarnego.
- int size ()

  Metoda zwracajaca ilosc wezlow drzewa binarnego.
- void wypisz_pelne ()

  Metoda wypisujaca cale drzewo. Wywoluje kolejne wypisz dla lewego oraz prawego subdrzewa korzenia.
- void wypisz (element ∗iter, int a)

  Metoda wypisujaca subdrzewo.
- element ∗ obroc_l (element ∗pivot)

  Metoda obracajaca w lewo dla elementu.
- element ∗ obroc_p (element ∗pivot)

  Metoda obracajaca w prawo dla elementu.
- element ∗ rownowaz (element ∗tmp)

  Metoda rownowazaca drzewo binarne.

4.10.1 Detailed Description

Klasa DrzewoBinarne.

4.10.2 Constructor & Destructor Documentation

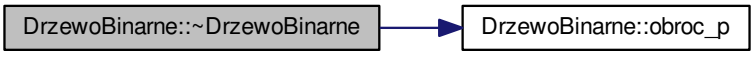
4.10.2.1 DrzewoBinarne::DrzewoBinarne ( )

Konstruktor bezparametryczny. Konstruktor inicjalizujacy drzewo binarne. korzen - inicjalizowany wartoscia NULL rozmiar - inicjalizowany wartoscia 0.

4.10.2.2 DrzewoBinarne::~DrzewoBinarne ( )

Destruktor bezparametryczny drzewa binarnego.

Here is the call graph for this function:



4.10.3 Member Function Documentation

4.10.3.1 DrzewoBinarne::element ∗ DrzewoBinarne::obroc_l (element ∗ *pivot*)

Metoda obracajaca w lewo dla elementu.

Parameters

pivot	- typu element∗ , zadany wskaznik na element , dla ktorego wystapi obrocenie.

Returns

pivot - typu element∗, wskaznik na element , ktory wskoczyl za dany.

4.10.3.2 DrzewoBinarne::element ∗ DrzewoBinarne::obroc_p (element ∗ *pivot*)

Metoda obracajaca w prawo dla elementu.

Parameters

pivot	- typu element∗ , zadany wskaznik na element , dla ktorego wystapi obrocenie.

Returns

pivot - typu element∗, wskaznik na element , ktory wskoczyl za dany.

4.10.3.3 int DrzewoBinarne::pop ( ) [virtual]

Metoda zdejmujaca element z drzewa binarnego.

Returns

wartosc - typu int, wartosc zdejmowana z drzewa binarnego.

Implements Zasobnik< int >.

4.10.3.4 int DrzewoBinarne::pop (int *szukana*) [virtual]

Metoda zdejmujaca dany element z drzewa binarnego.

Parameters

szukana	- typu int, wartosc szukana na drzewie binarnym.

Returns

wartosc - typu int, wartosc zdejmowana z drzewa binarnego.

Implements Zasobnik< int >.

4.10.3.5 void DrzewoBinarne::push (int *wartosc*) [virtual]

Metoda umieszczajaca element na drzewie binarnym.

Parameters

wartosc	- typu int, wartosc umieszczana na drzewie binarnym.

Implements Zasobnik< int >.

Here is the call graph for this function:

4.10.3.6 DrzewoBinarne::element * DrzewoBinarne::rownowaz (element * *tmp*)

Metoda rownowazaca drzewo binarne.

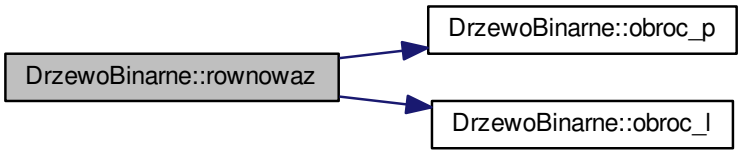
Parameters

tmp	- typu element* , zadany wskaznik , najczesciej korzen.

Returns

tmp - typu element*, wskaznik na korzen.

Here is the call graph for this function:



4.10.3.7 int DrzewoBinarne::size ( ) [virtual]

Metoda zwracajaca ilosc wezlow drzewa binarnego.

Returns

rozmiar - typu int, ilosc wezlow drzewa binarnego.

Implements Zasobnik< int >.

4.10.3.8 void DrzewoBinarne::wypisz (element * *iter,* int *a*)

Metoda wypisujaca subdrzewo.

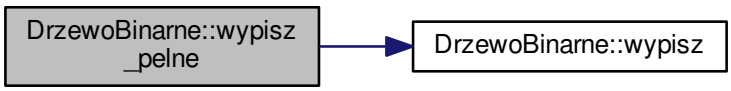
Parameters

iter	- typu element∗ , zadany wskaznik na subdrzewo.
a	- typu int, poziom wyisywanego subdrzewa.

4.10.3.9 void DrzewoBinarne::wypisz_pelne ( )

Metoda wypisujaca cale drzewo. Wywoluje kolejne wypisz dla lewego oraz prawego subdrzewa korzenia.

Here is the call graph for this function:



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

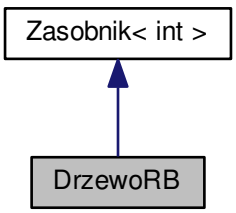
- DrzewoBinarne.hh
- DrzewoBinarne.cpp

4.11 DrzewoRB Class Reference

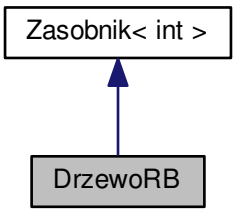
Klasa DrzewoRB.

`#include <DrzewoRB.hh>`

Inheritance diagram for DrzewoRB:



Collaboration diagram for DrzewoRB:



Public Member Functions

- DrzewoRB ()

  Konstruktor bezparametryczny. Konstruktor inicjalizujacy drzewo binarne. korzen - inicjalizowany wartoscia NULL rozmiar - inicjalizowany wartoscia 0.
- ~DrzewoRB ()

  Destruktor bezparametryczny drzewa RB.
- void push (int wartosc)

  Metoda umieszczajaca element na drzewie RB.
- int pop ()

  Metoda zdejmujaca element z drzewa RB.
- int pop (int szukana)

  Metoda zdejmujaca dany element z drzewa RB.
- int size ()

  Metoda zwracajaca ilosc wezlow drzewa RB.
- void wypisz_pelne ()

  Metoda wypisujaca cale drzewo. Wywoluje kolejne wypisz dla lewego oraz prawego subdrzewa korzenia.
- void wypisz (element *iter, int a)

  Metoda wypisujaca subdrzewo.
- element * obroc_l (element *pivot)

  Metoda obracajaca w lewo dla elementu.
- element * obroc_p (element *pivot)

  Metoda obracajaca w prawo dla elementu.
- element * ukladaj (element *pivot)

  Metoda ukladajaca element na drzewie RB.

4.11.1 Detailed Description

Klasa DrzewoRB.

4.11.2 Constructor & Destructor Documentation

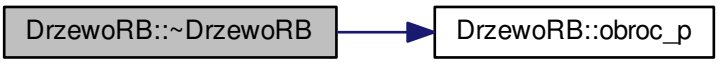
4.11.2.1 DrzewoRB::DrzewoRB ( )

Konstruktor bezparametryczny. Konstruktor inicjalizujacy drzewo binarne. korzen - inicjalizowany wartoscia NULL rozmiar - inicjalizowany wartoscia 0.

4.11.2.2 DrzewoRB::~DrzewoRB ( )

Destruktor bezparametryczny drzewa RB.

Here is the call graph for this function:



4.11.3 Member Function Documentation

4.11.3.1 DrzewoRB::element * DrzewoRB::obroc_l (element * *pivot*)

Metoda obracajaca w lewo dla elementu.

Parameters

pivot	- typu element* , zadany wskaznik na element , dla ktorego wystapi obrocenie.

Returns

pivot - typu element*, wskaznik na element , ktory wskoczyl za dany.

4.11.3.2 DrzewoRB::element * DrzewoRB::obroc_p (element * *pivot*)

Metoda obracajaca w prawo dla elementu.

Parameters

pivot	- typu element* , zadany wskaznik na element , dla ktorego wystapi obrocenie.

Returns

pivot - typu element*, wskaznik na element , ktory wskoczyl za dany.

4.11.3.3 int DrzewoRB::pop ( ) [virtual]

Metoda zdejmujaca element z drzewa RB.

Returns

wartosc - typu int, wartosc zdejmowana z drzewa RB.

Implements Zasobnik< int >.

4.11.3.4 int DrzewoRB::pop (int *szukana*) [virtual]

Metoda zdejmujaca dany element z drzewa RB.

Parameters

szukana	- typu int, wartosc szukana na drzewie RB.

Returns

wartosc - typu int, wartosc zdejmowana z drzewa RB.

Implements Zasobnik< int >.

4.11.3.5 void DrzewoRB::push (int *wartosc*) [virtual]

Metoda umieszczajaca element na drzewie RB.

Parameters

wartosc	- typu int, wartosc umieszczana na drzewie RB.

Implements Zasobnik< int >.

Here is the call graph for this function:

4.11.3.6 int DrzewoRB::size ( ) [virtual]

Metoda zwracajaca ilosc wezlow drzewa RB.

Returns

rozmiar - typu int, ilosc wezlow drzewa RB.

Implements Zasobnik< int >.

4.11.3.7 DrzewoRB::element * DrzewoRB::ukladaj (element * *pivot*)

Metoda ukladajaca element na drzewie RB.

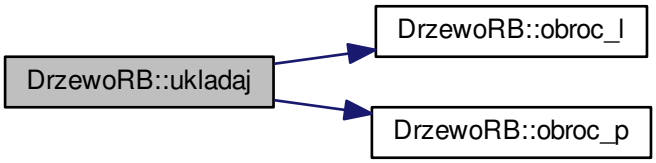
Parameters

pivot	- typu element* , zadany wskaznik , najczesciej korzen.

Returns

pivot - typu element∗, wskaznik na element.

Here is the call graph for this function:



4.11.3.8 void DrzewoRB::wypisz (element ∗ *iter,* int *a*)

Metoda wypisujaca subdrzewo.

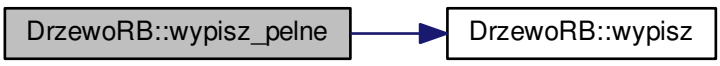
Parameters

iter	- typu element∗ , zadany wskaznik na subdrzewo.
a	- typu int, poziom wyisywanego subdrzewa.

4.11.3.9 void DrzewoRB::wypisz_pelne ( )

Metoda wypisujaca cale drzewo. Wywoluje kolejne wypisz dla lewego oraz prawego subdrzewa korzenia.

Here is the call graph for this function:



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

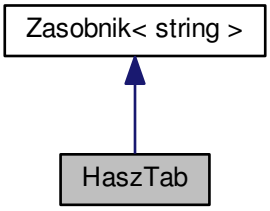
- DrzewoRB.hh
- DrzewoRB.cpp

4.12 HaszTab Class Reference

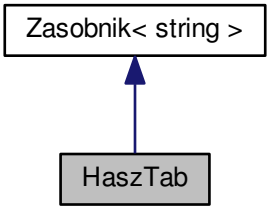
Klasa HaszTab.

`#include <HaszTab.hh>`

Inheritance diagram for HaszTab:



Collaboration diagram for HaszTab:



Public Member Functions

- HaszTab ()

  Konstruktor bezparametryczny. Konstruktor inicjalizujacy tablice haszującą. rozmiar_k1 - wielkosc tablicy pierwszej. rozmiar_k2 - wielkosc tablicy drugiej. Tworzona jest maciecz elementow (string wartosc,string klucz) o zadanej wielkości.
- ~HaszTab ()

  Destruktor bezparametryczny tablicy haszującej.
- void push (string wartosc, string klucz)

  Przeciazenie operacji push. Metoda umieszczająca element o zadanej wartości i kluczu na tablicy haszującej.
- string pop (string klucz_szukany)

  Metoda zdejmujaca element z tablicy haszujacej.
- void push (string wartosc)

  Metoda umieszczajaca element na tablicy haszujacej.
- string pop ()

  Metoda zdejmujaca element z tablicy haszujacej.
- int size ()

  Metoda zwracajaca rozmiar Tablicy Haszujacej.
- 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 zczytująca element z tablicy haszującej. Metoda zczytuje element o określonym kluczu. Wartości oraz klucze nie są usuwane podczas zczytywania.

- int size_k1 ()

  Metoda zwracajaca rozmiar tablicy pierwszej na ktorej oparta jest tablica haszująca.

- int size_k2 ()

  Metoda zwracajaca rozmiar tablicy drugiej na ktorej oparta jest tablica haszująca.

4.12.1 Detailed Description

Klasa HaszTab.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 HaszTab::HaszTab ( )

Konstruktor bezparametryczny. Konstruktor inicjalizujacy tablice haszującą. rozmiar_k1 - wielkosc tablicy pierwszej. rozmiar_k2 - wielkosc tablicy drugiej. Tworzona jest maciecz elementow (string wartosc,string klucz) o zadanej wielkości.

4.12.2.2 HaszTab::~HaszTab ( )

Destruktor bezparametryczny tablicy haszującej.

4.12.3 Member Function Documentation

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

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

Parameters

klucz_umieszczany	- typu string, zadany klucz.
modulacja	- typu int, wielkość tablicy która moduluje klucz.

Returns

indeks - typu int, zmodulowany indeks elementu.

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

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

Parameters

klucz_szukany	- typu string, szukany klucz.

Returns

wartosc - typu int, wartosc zdejmowana ze stosu.

Here is the call graph for this function:

4.12.3.3 string HaszTab::pop (string *klucz_szukany*) [virtual]

Metoda zdejmujaca element z tablicy haszujacej.

Returns

klucz_szukany - typu string, klucz wartosci zdejmowanej z tablicy haszujacej.

Implements Zasobnik< string >.

4.12.3.4 string HaszTab::pop ( ) [virtual]

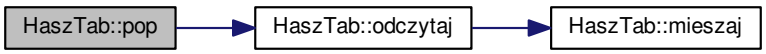
Metoda zdejmujaca element z tablicy haszujacej.

Returns

wartosc - typu string, wartosc zdejmowana z tablicy haszujacej.

Implements Zasobnik< string >.

Here is the call graph for this function:



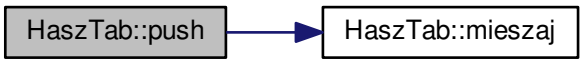
4.12.3.5 void HaszTab::push (string *wartosc*, string *klucz*)

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

Parameters

wartosc	- typu string,zadana wartość.
klucz	- typu string, zadany klucz.

Here is the call graph for this function:



4.12.3.6 void HaszTab::push (string *wartosc*) [virtual]

Metoda umieszczajaca element na tablicy haszujacej.

Parameters

wartosc	- typu string, wartosc umieszczana na tablicy haszujacej.

Implements Zasobnik< string >.

4.12.3.7 int HaszTab::size ( ) [virtual]

Metoda zwracajaca rozmiar Tablicy Haszujacej.

Returns

rozmiar - typu int,rozmiar Tablicy Haszujacej.

Implements Zasobnik< string >.

4.12.3.8 int HaszTab::size_k1 ( )

Metoda zwracajaca rozmiar tablicy pierwszej na ktorej oparta jest tablica haszująca.

Returns

rozmiar - typu int,rozmiar tablicy pierwszej.

4.12.3.9 int HaszTab::size_k2 ( )

Metoda zwracajaca rozmiar tablicy drugiej na ktorej oparta jest tablica haszująca.

Returns

rozmiar - typu int,rozmiar tablicy drugiej.

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

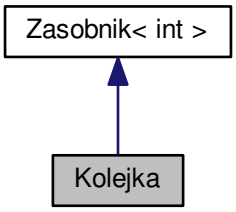
- HaszTab.hh
- HaszTab.cpp

4.13 Kolejka Class Reference

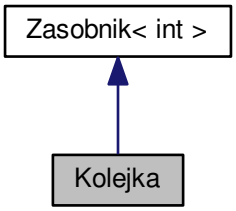
Klasa Kolejka.

`#include <Kolejka.hh>`

Inheritance diagram for Kolejka:



Collaboration diagram for Kolejka:



Public Member Functions

- Kolejka ()

  Konstruktor bezparametryczny. Konstruktor inicjalizujacy straznika_poczatek i straznika_koniec kolejki wartosciami NULL , oraz rozmiar kolejki wartoscia 0.
- ~Kolejka ()

  Destruktor bezparametryczny kolejki.
- void push (int wartosc)

  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 pop (int a)

  Metoda zdejmujaca element z poczatku kolejki. Metoda dekrementuje rozmiar przy zdejmowaniu elementu.
- int size ()

  Metoda zwracajaca wielkosc kolejki.

4.13.1 Detailed Description

Klasa Kolejka.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 Kolejka::Kolejka ( )

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

4.13.2.2 Kolejka::~Kolejka ( )

Destruktor bezparametryczny kolejki.

4.13.3 Member Function Documentation

4.13.3.1 int Kolejka::pop ( ) [virtual]

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

Returns

wartosc - typu int, wartosc zdejmowana z kolejki.

Implements Zasobnik< int >.

4.13.3.2 int Kolejka::pop (int *a*) [inline], [virtual]

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

Returns

wartosc - typu int, wartosc zdejmowana z kolejki.

Implements Zasobnik< int >.

4.13.3.3 void Kolejka::push (int *wartosc*) [virtual]

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

Parameters

wartosc	- typu int, wartosc umieszczana na koncu kolejki.

Implements Zasobnik< int >.

4.13.3.4 int Kolejka::size ( ) [virtual]

Metoda zwracajaca wielkosc kolejki.

Returns

rozmiar - typu int,rozmiar kolejki.

Implements Zasobnik< int >.

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

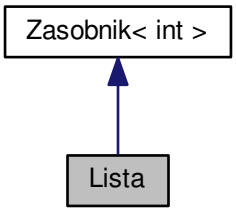
- Kolejka.hh
- Kolejka.cpp

4.14 Lista Class Reference

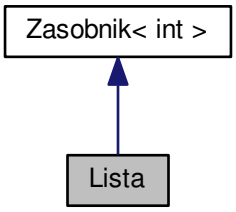
Klasa Lista.

`#include <Lista.hh>`

Inheritance diagram for Lista:



Collaboration diagram for Lista:



Public Member Functions

- Lista ()

  Konstruktor bezparametryczny. Konstruktor inicjalizujacy straznika listy wartoscia NULL oraz rozmiar wartoscia 0.
- ~Lista ()

  Destruktor bezparametryczny listy.

- void push (int, int)

  Metoda umieszczajaca element okreslonej pozycji na liscie <0,rozmiar>. Metoda inkrementuje rozmiar podczas umieszczania elementuna liscie.
- int pop (int)

  Metoda zdejmujaca element z okreslonej pozycji listy <0,rozmiar>. Metoda dekrementuje rozmiar przy zdejmowaniu elementu.
- void push (int wartosc)

  Przeciazenie operacji push. Umieszcza element domyslnie na pozycji 1. Nastepuje inkrementacja rozmiar listy.
- int pop ()

  Przeciazenie operacji pop dla listy. Pobiera domyslnie element listy z pozycji 1.Nastepuje dekrementacja rozmiar listy.
- int size ()

  Metoda zwracajaca wielkosc listy.

4.14.1 Detailed Description

Klasa Lista.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 Lista::Lista ( )

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

4.14.2.2 Lista::~Lista ( )

Destruktor bezparametryczny listy.

4.14.3 Member Function Documentation

4.14.3.1 int Lista::pop (int *pozycja*) [virtual]

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

Parameters

pozycja	- typu int, numer elementu ktory ma byc zdjety z listy.

Returns

wartosc - typu int, wartosc zdejmowana z listy.

Implements Zasobnik< int >.

4.14.3.2 int Lista::pop ( ) [inline], [virtual]

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

Returns

wartosc - typu int, wartosc zdejmowana z listy.

Implements Zasobnik< int >.

4.14.3.3 void Lista::push (int *wartosc,* int *pozycja*)

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

Parameters

wartosc	- typu int, wartosc umieszczana na liscie.
pozycja	- typu int, pozycja na ktorej jest umieszczana wartosc.

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

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

Parameters

wartosc	- typu int, wartosc umieszczana na liscie.

Implements Zasobnik< int >.

4.14.3.5 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:

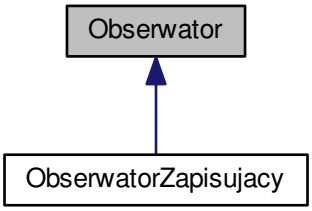
- Lista.hh
- Lista.cpp

4.15 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.

4.15.1 Detailed Description

Klasa Obserwator.

4.15.2 Member Function Documentation

4.15.2.1 virtual void Obserwator::odswiez (int *k,* long int *sredni_czas*) [pure virtual]

Metoda odswiezajaca obserwatora.

Parameters

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

Implemented in ObserwatorZapisujacy.

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

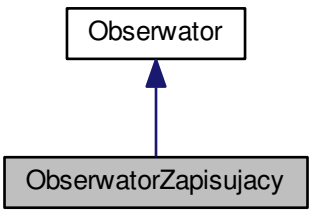
- Obserwator.hh

4.16 ObserwatorZapisujacy Class Reference

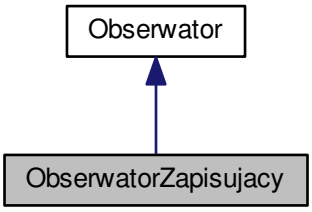
Klasa ObserwatorZapisujacy.

`#include <ObserwatorZapisujacy.hh>`

Inheritance diagram for ObserwatorZapisujacy:



Collaboration diagram for ObserwatorZapisujacy:



Public Member Functions

- void odswiez (int, long int)

  Metoda odswiezajaca obserwatora.

4.16.1 Detailed Description

Klasa ObserwatorZapisujacy.

4.16.2 Member Function Documentation

4.16.2.1 void ObserwatorZapisujacy::odswiez (int *k,* long *sredni_czas*) [virtual]

Metoda odswiezajaca obserwatora.

Parameters

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

Implements Obserwator.

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

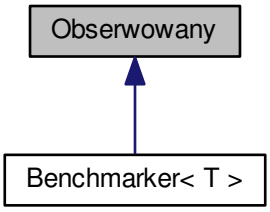
- ObserwatorZapisujacy.hh
- ObserwatorZapisujacy.cpp

4.17 Obserwowany Class Reference

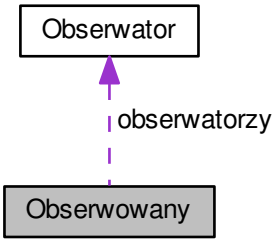
Szablon klasy Obserwowany.

`#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

4.17.1 Detailed Description

Szablon klasy Obserwowany.

4.17.2 Member Function Documentation

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

Metoda dodajaca obserwatora do obiektu.

Parameters

Obs	- typu Obserwator∗, wskaznik na danego obserwatora

4.17.2.2 void Obserwowany::usun (Obserwator ∗ *Obs*) [inline]

Metoda usuwajaca obserwatora z obiektu.

Parameters

Obs	- typu Obserwator∗, wskaznik na danego obserwatora

4.17.3 Member Data Documentation

4.17.3.1 Obserwator∗ Obserwowany::obserwatorzy [protected]

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

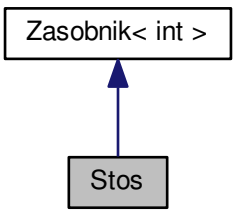
- Obserwowany.hh

4.18 Stos Class Reference

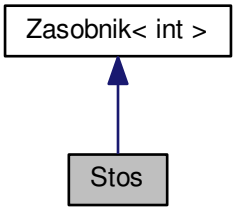
Klasa Stos.

`#include <Stos.hh>`

Inheritance diagram for Stos:



Collaboration diagram for Stos:



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 pop (int a)

  Metoda zdejmujaca element ze stosu. Metoda dekrementuje rozmiar przy zdejmowaniu ze stosu.
- int size ()

  Metoda zwracajaca wielkosc stosu.

4.18.1 Detailed Description

Klasa Stos.

4.18.2 Constructor & Destructor Documentation

4.18.2.1 Stos::Stos ( )

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

4.18.2.2 Stos::∼Stos ( )

Destruktor bezparametryczny stosu.

4.18.3 Member Function Documentation

4.18.3.1 int Stos::pop ( ) [virtual]

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

Returns

wartosc - typu int, wartosc zdejmowana ze stosu.

Implements Zasobnik< int >.

4.18.3.2 int Stos::pop (int *a*) [inline], [virtual]

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

Returns

wartosc - typu int, wartosc zdejmowana ze stosu.

Implements Zasobnik< int >.

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

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

Parameters

wartosc	- typu int, wartosc umieszczana na stosie.

Implements Zasobnik< int >.

4.18.3.4 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

4.19 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 T pop (T szukana)=0

Metoda zdejmujaca dany element z zasobnika.

- virtual int size ()=0

  Metoda zwracajaca rozmiar zasobnika.

4.19.1 Detailed Description

template<typename T>class Zasobnik< T >

Szablon klasy Zasobnik.

4.19.2 Constructor & Destructor Documentation

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

Destruktor wirtualny.

4.19.3 Member Function Documentation

4.19.3.1 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 HaszTab, Lista, ArrayLista, DrzewoRB, DrzewoBinarne, Kolejka, and Stos.

4.19.3.2 template<typename T> virtual T Zasobnik< T >::pop (T *szukana*) [pure virtual]

Metoda zdejmujaca dany element z zasobnika.

Parameters

szukana	- typu T, wartosc szukana w zasobniku.

Returns

wartosc - typu T, wartosc zdejmowana z zasobnika.

Implemented in DrzewoRB, HaszTab, DrzewoBinarne, Kolejka, Stos, and Lista.

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

Metoda umieszczajaca element na zasobniku.

Parameters

wartosc	- typu T, wartosc umieszczana na zasobniku.

Implemented in HaszTab, Lista, ArrayLista, DrzewoRB, DrzewoBinarne, Kolejka, and Stos.

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

Metoda zwracajaca rozmiar zasobnika.

Returns

rozmiar - typu int,rozmiar zasobnika.

Implemented in HaszTab, Lista, ArrayLista, DrzewoRB, DrzewoBinarne, Kolejka, and Stos.

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

- Zasobnik.hh

Chapter 5

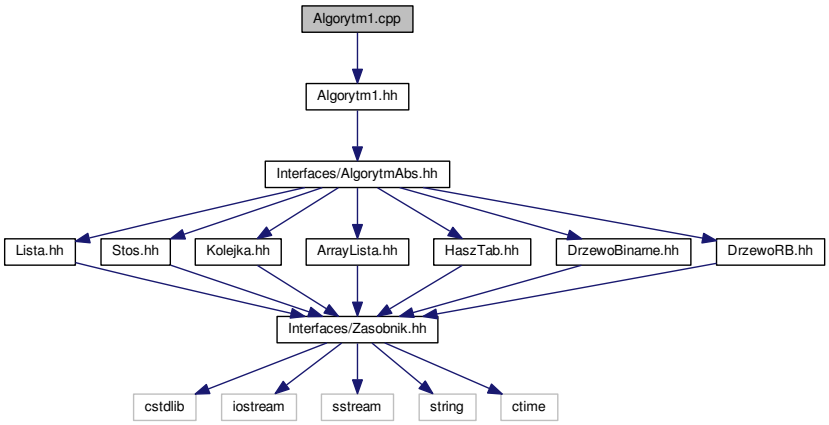
File Documentation

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.

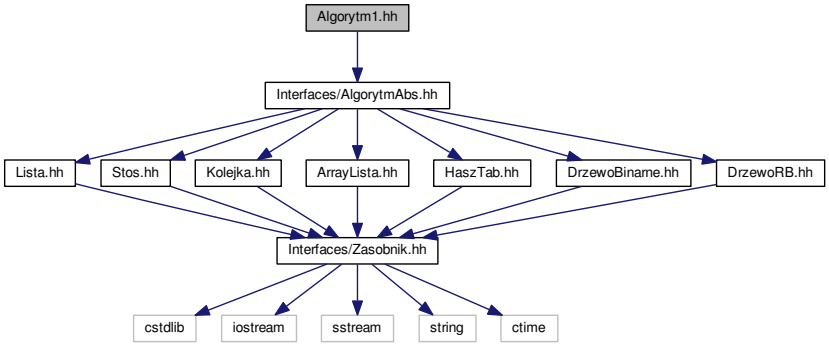
Plik zawiera metody klasy Algorytm1.

5.2 Algorytm1.hh File Reference

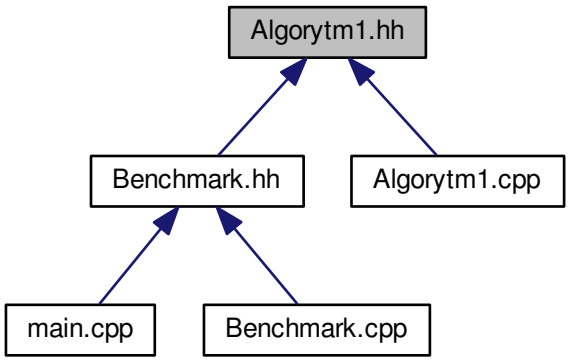
Definicja klasy Algorytm1.

```
#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

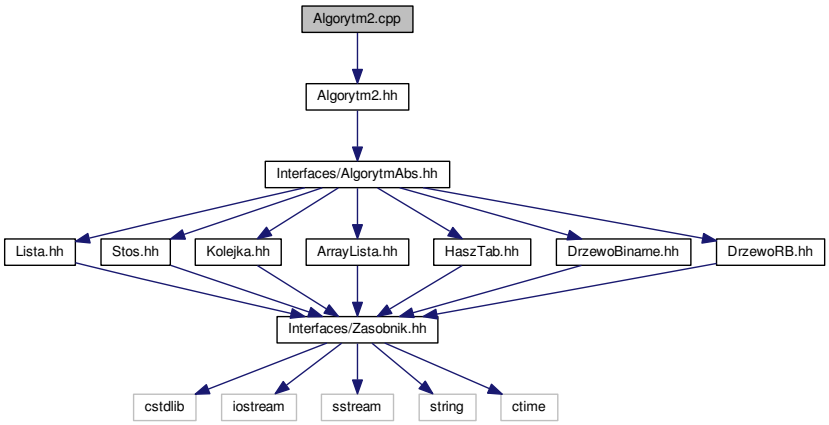
Definicja klasy Algorytm1.

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

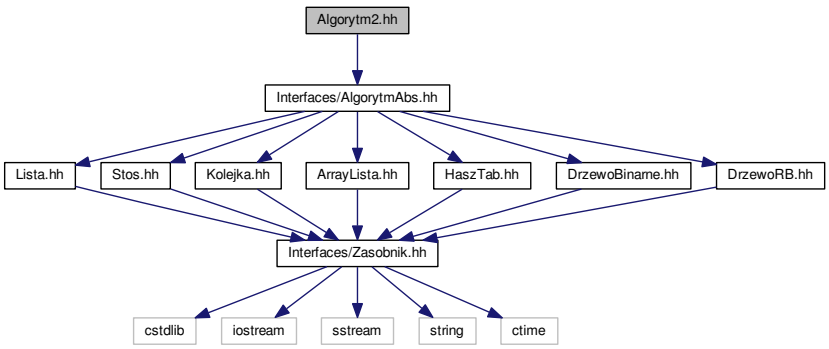
Metody klasy Algorytm2.

Plik zawiera metody klasy Algorytm2.

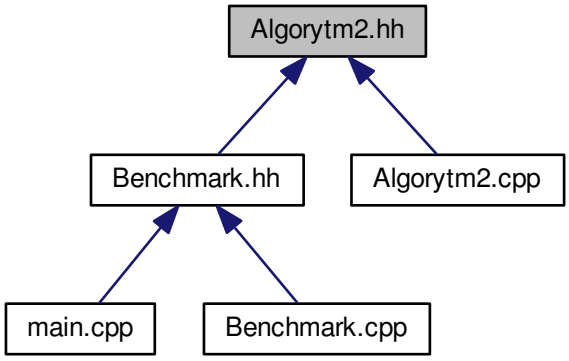
5.4 Algorytm2.hh File Reference

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

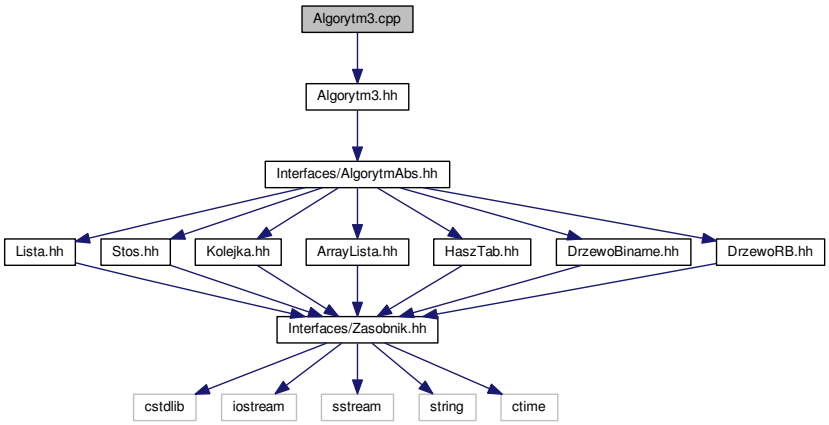
Definicja klasy Algorytm2.

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

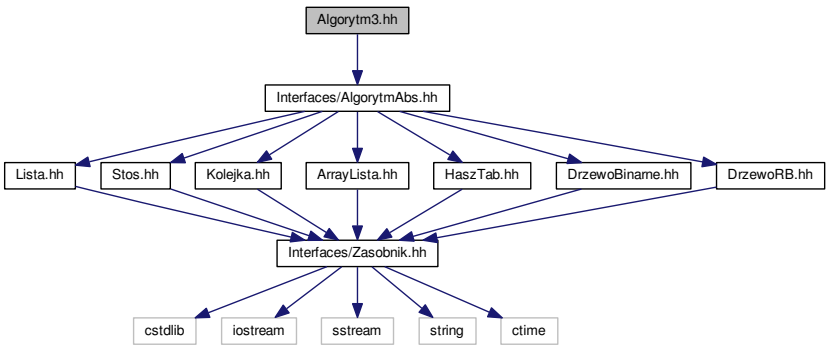
Metody klasy Algorytm3.

Plik zawiera metody klasy Algorytm3.

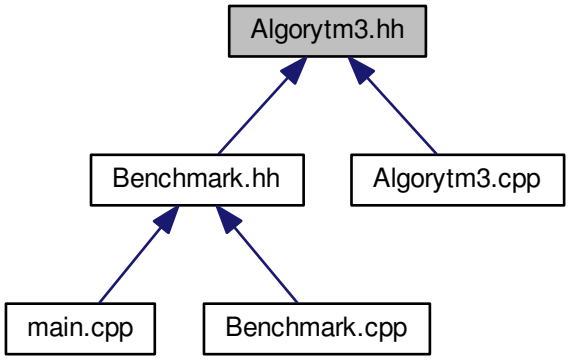
5.6 Algorytm3.hh File Reference

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

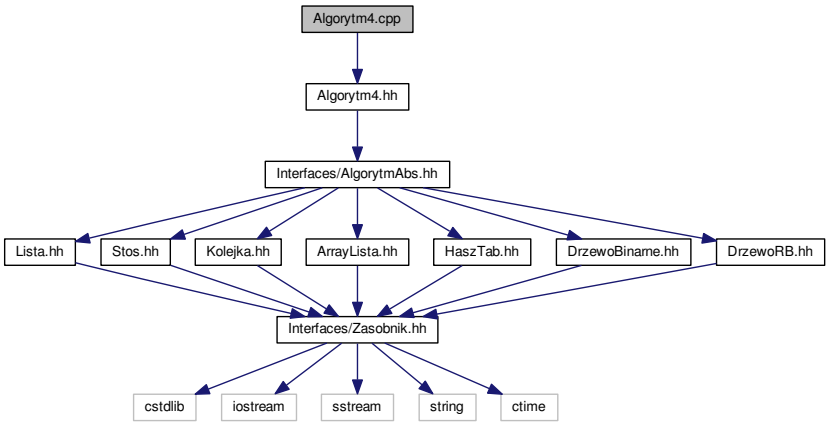
Definicja klasy Algorytm3.

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

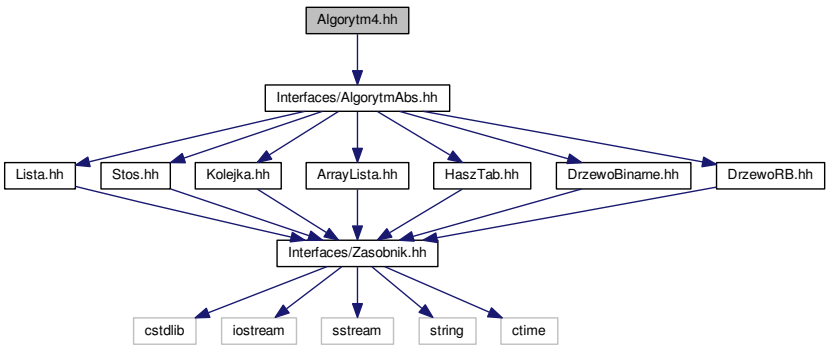
Metody klasy Algorytm4.

Plik zawiera metody klasy Algorytm4.

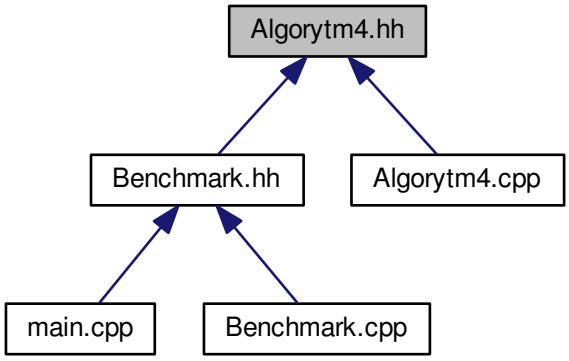
5.8 Algorytm4.hh File Reference

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

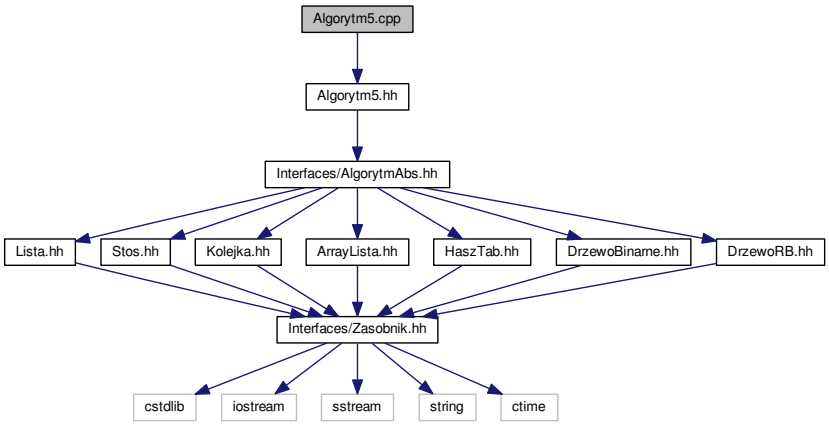
Definicja klasy Algorytm4.

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

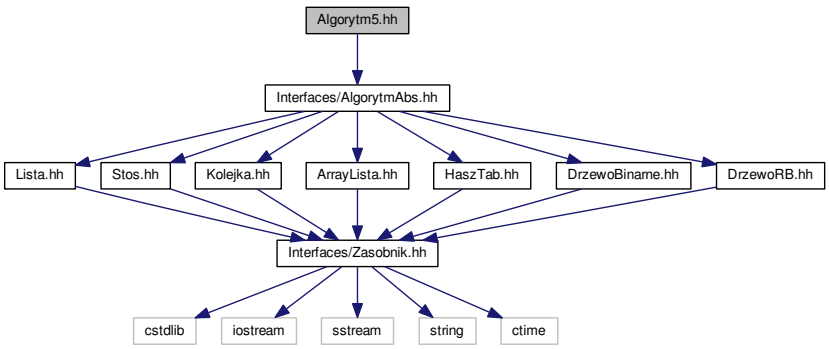
Metody klasy Algorytm5.

Plik zawiera metody klasy Algorytm5.

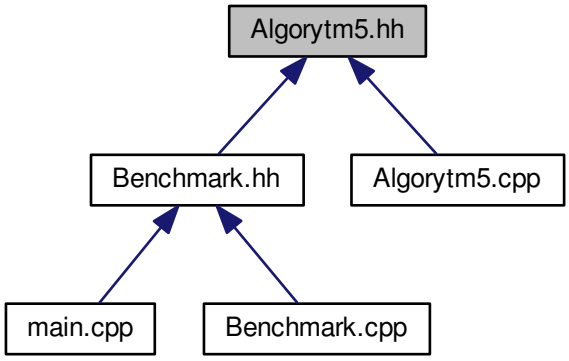
5.10 Algorytm5.hh File Reference

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

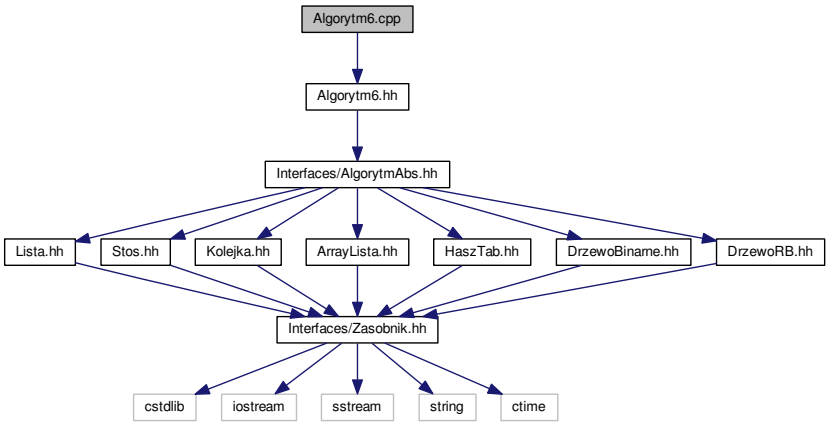
Definicja klasy Algorytm5.

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

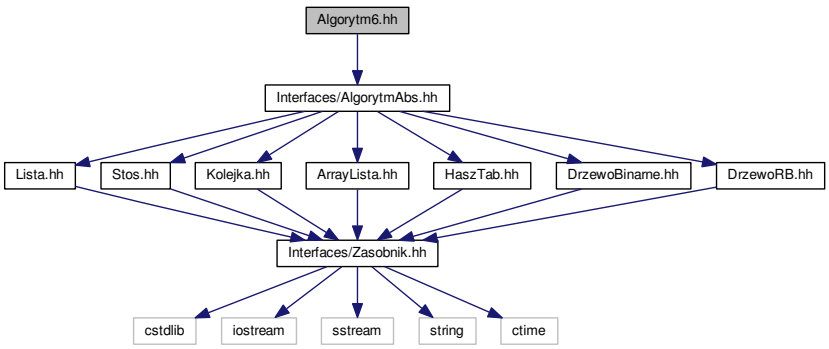
Metody klasy Algorytm6.

Plik zawiera metody klasy Algorytm6.

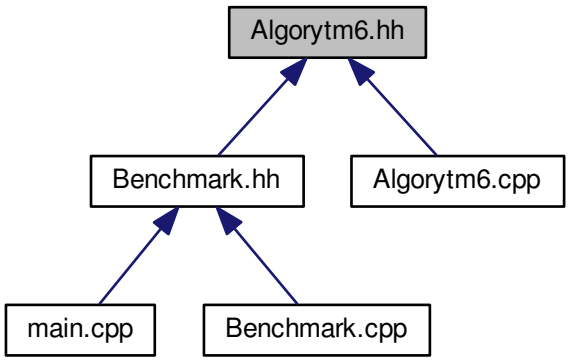
5.12 Algorytm6.hh File Reference

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.

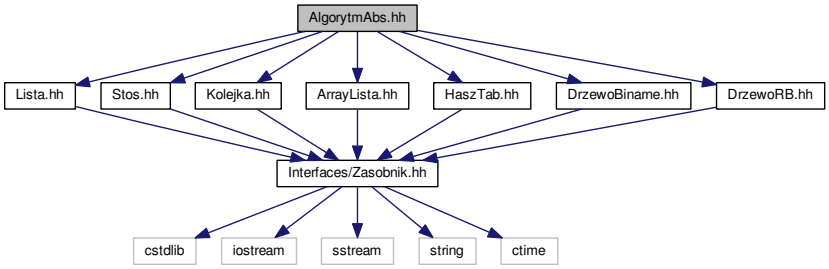
Plik zawiera definicje klasy Algorytm6.

5.13 AlgorytmAbs.hh File Reference

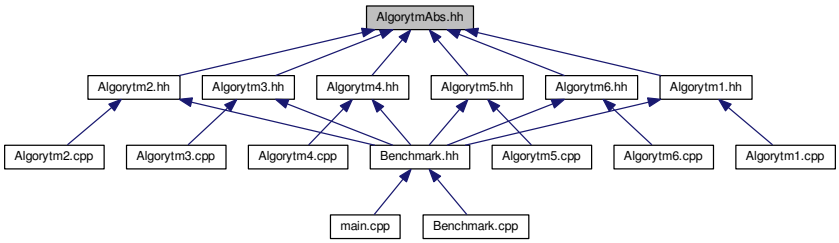
Definicja szablonu klasy abstrakcyjnej Algorytm.

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

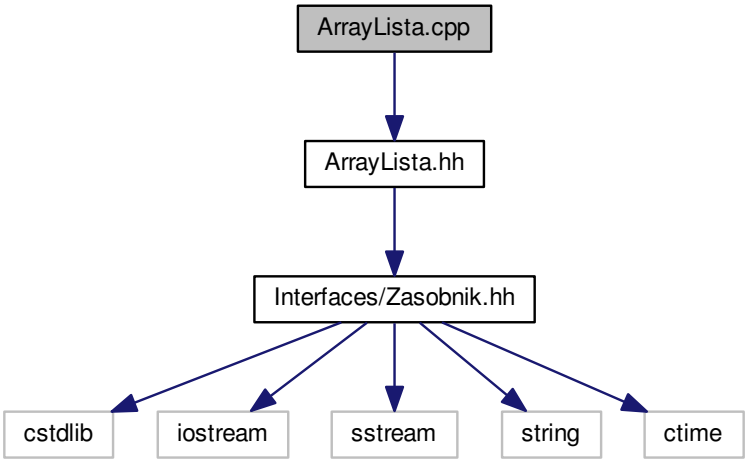
Definicja szablonu klasy abstrakcyjnej Algorytm.

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

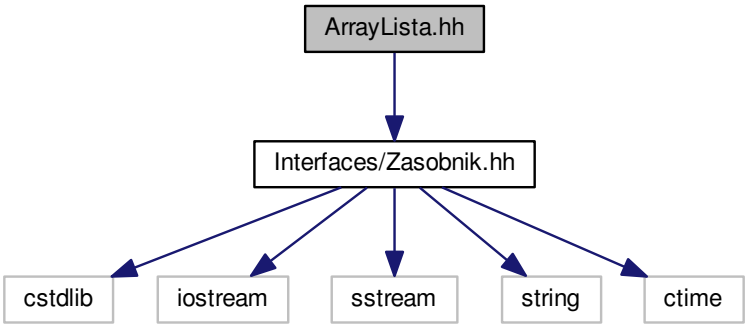
Metody klasy ArrayLista.

Plik zawiera metody klasy ArrayLista.

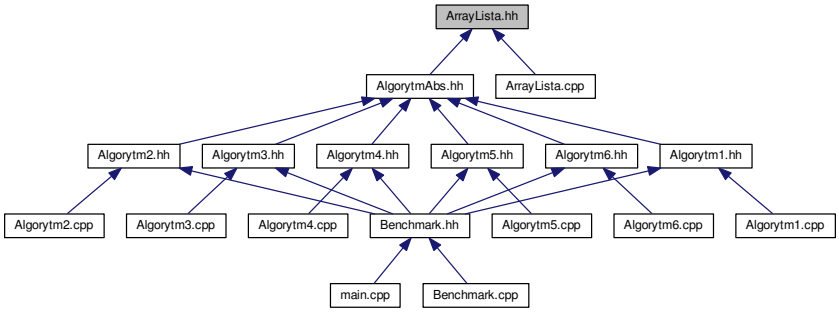
5.15 ArrayLista.hh File Reference

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.

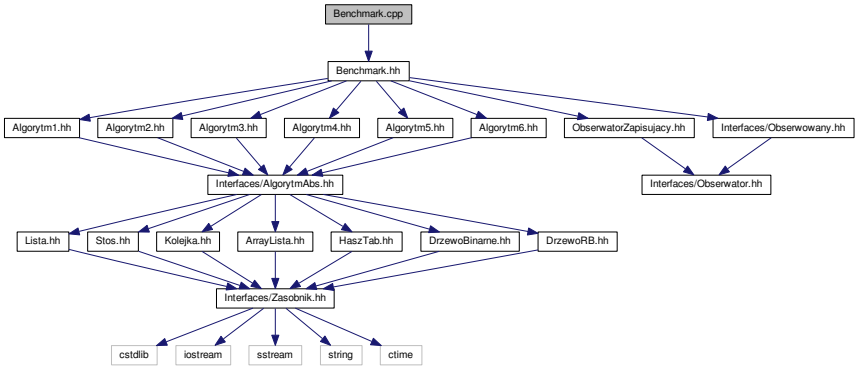
Plik zawiera definicje klasy modulujacej pojecie listy jednokierunkowej opartej na tablicy dynamicznej.

5.16 Benchmark.cpp File Reference

Metody klasy Benchmarker.

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

Include dependency graph for Benchmark.cpp:



5.16.1 Detailed Description

Metody klasy Benchmarker.

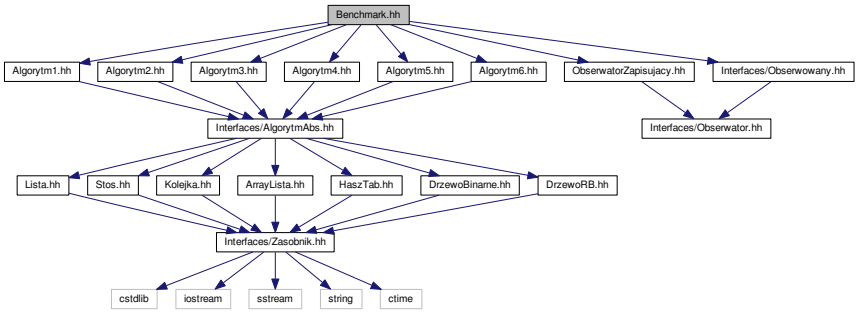
Plik zawiera metody klasy Benchmarker.

5.17 Benchmark.hh File Reference

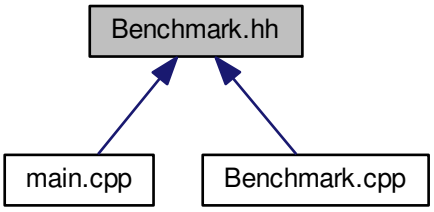
Definicja szablonu klasy Benchmarker.

```
#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 Benchmarker< T >

  Szablon klasy Benchmarker.

5.17.1 Detailed Description

Definicja szablonu klasy Benchmarker.

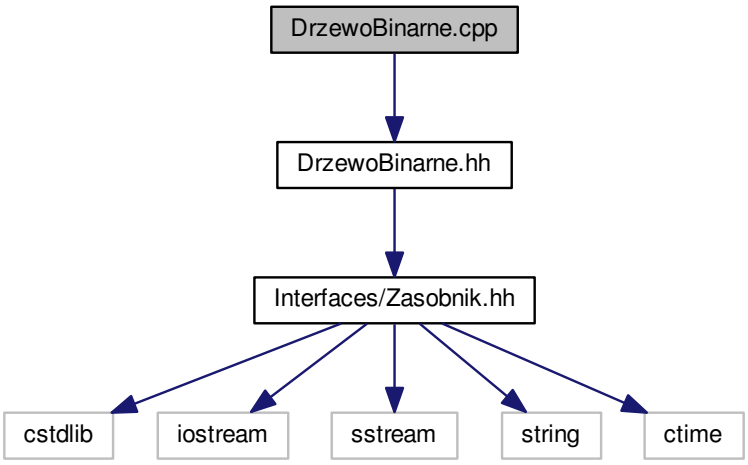
Plik zawiera definicje szablonu klasy Benchmarker.

5.18 DrzewoBinarne.cpp File Reference

Metody klasy DrzewoBinarne.

```
#include "DrzewoBinarne.hh"
```

Include dependency graph for DrzewoBinarne.cpp:



5.18.1 Detailed Description

Metody klasy DrzewoBinarne.

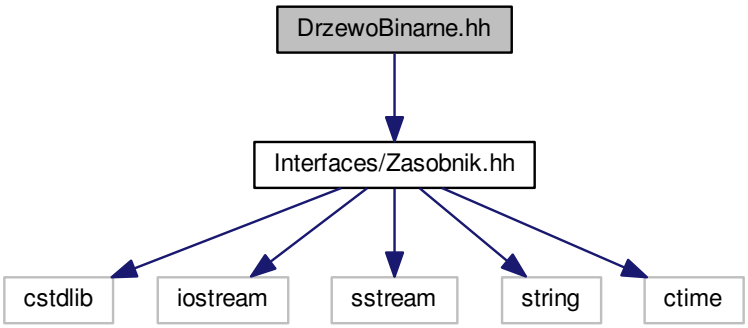
Plik zawiera metody klasy DrzewoBinarne.

5.19 DrzewoBinarne.hh File Reference

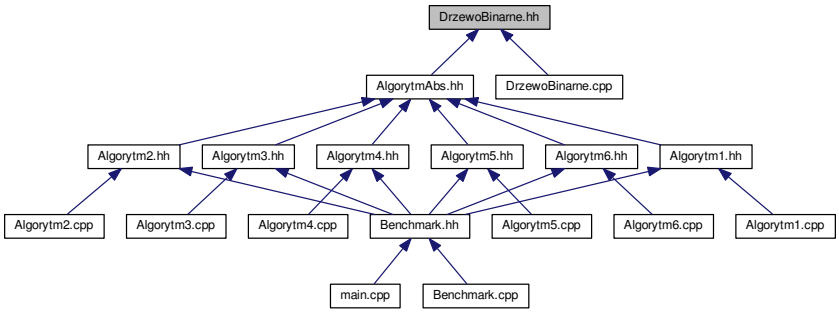
Definicja klasy DrzewoBinarne.

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

Include dependency graph for DrzewoBinarne.hh:



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



Classes

- class DrzewoBinarne

  Klasa DrzewoBinarne.

5.19.1 Detailed Description

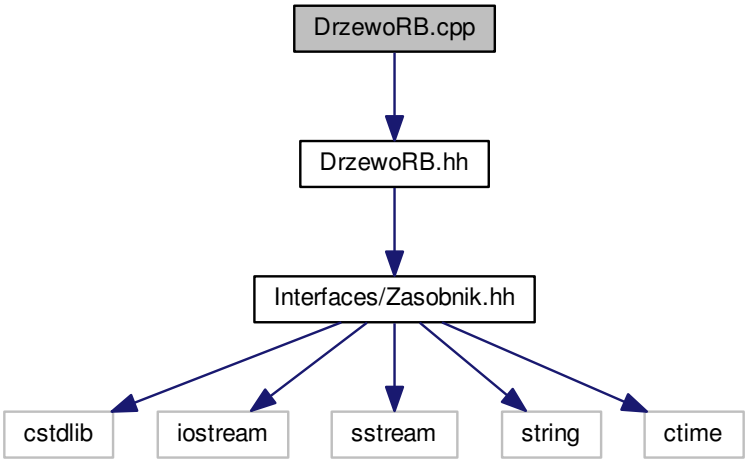
Definicja klasy DrzewoBinarne.

Plik zawiera definicje klasy DrzewoBinarne.

5.20 DrzewoRB.cpp File Reference

Metody klasy DrzewoRB.

```
#include "DrzewoRB.hh"
```
Include dependency graph for DrzewoRB.cpp:



5.20.1 Detailed Description

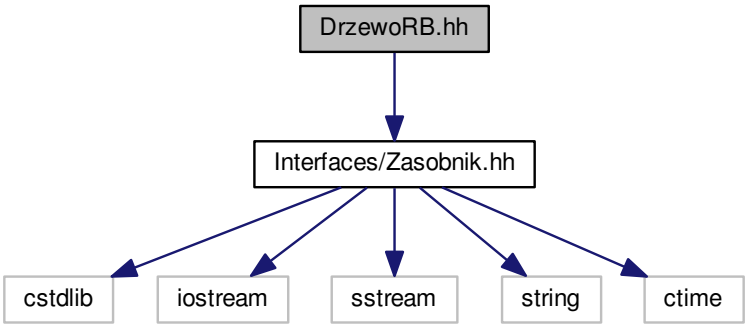
Metody klasy DrzewoRB.

Plik zawiera metody klasy DrzewoRB.

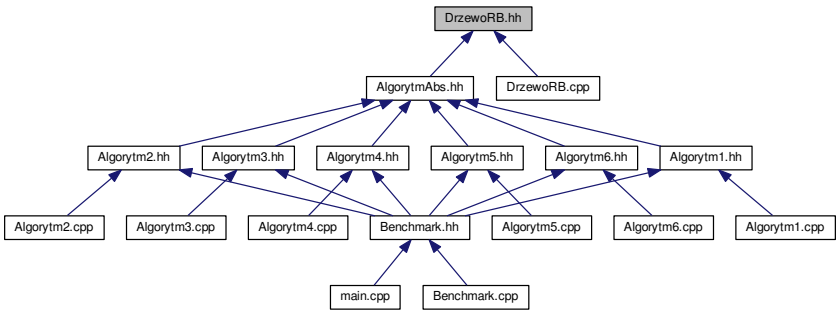
5.21 DrzewoRB.hh File Reference

Definicja klasy DrzewoRB.

```
#include "Interfaces/Zasobnik.hh"
```
Include dependency graph for DrzewoRB.hh:



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



Classes

- class DrzewoRB

  Klasa DrzewoRB.

5.21.1 Detailed Description

Definicja klasy DrzewoRB.

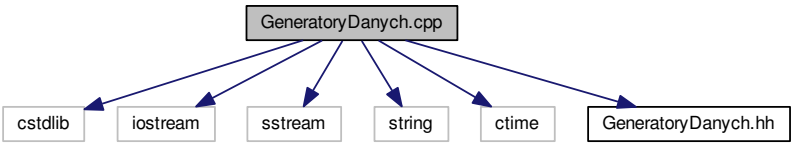
Plik zawiera definicje klasy DrzewoRB.

5.22 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.

5.22.1 Detailed Description

Funkcje generacji danych.

Plik zawiera funkcje generacji danych .

5.22.2 Function Documentation

5.22.2.1 template<> int* generujdane (int *l_danych*)

Szablon metody generujacej wartosci losowe danego typu.

Parameters

l_danych	- typu int, liczba generowanych danych.

Returns

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

5.22.2.2 template<> string* generujdane (int *l_danych*)

Szablon metody generujacej wartosci losowe danego typu.

Parameters

l_danych	- typu int, liczba generowanych danych.

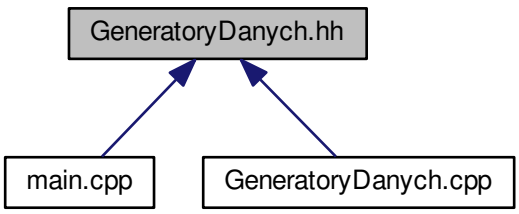
Returns

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

5.23 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.

5.23.1 Detailed Description

Szablon funkcji generacji danych.

Plik zawiera szablon funkcji generacji danych.

5.23.2 Function Documentation

5.23.2.1 template<typename T > T* generujdane (int *l_danych*)

Szablon metody generujacej wartosci losowe danego typu.

Parameters

l_danych	- typu int, liczba generowanych danych.

Returns

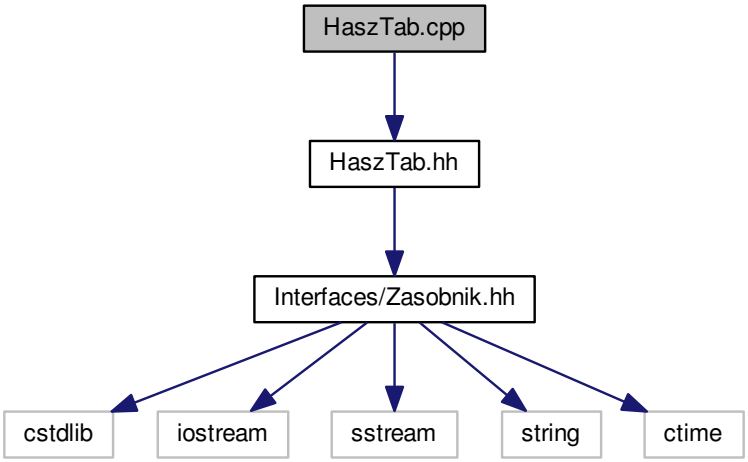
T* -wskaznik na dany typ, wskaznik na tablice z wygenerowanymi danymi.

5.24 HaszTab.cpp File Reference

Metody klasy HaszTab.

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

Include dependency graph for HaszTab.cpp:



5.24.1 Detailed Description

Metody klasy HaszTab.

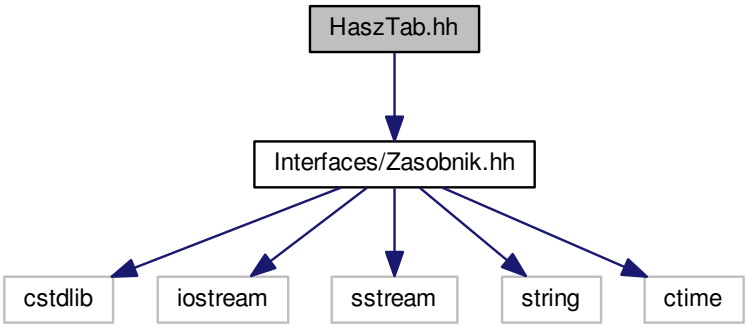
Plik zawiera metody klasy HaszTab.

5.25 HaszTab.hh File Reference

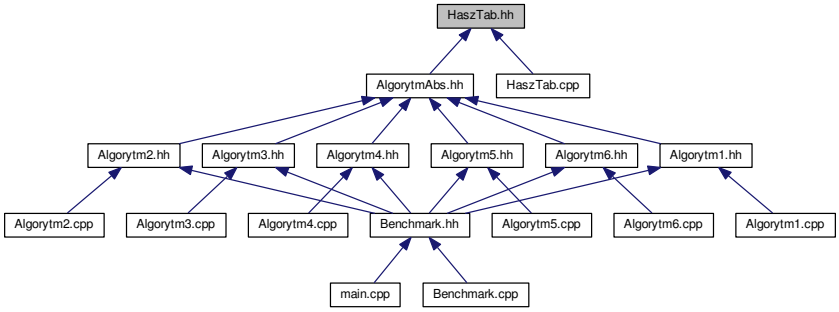
Definicja klasy HaszTab.

```
#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.

5.25.1 Detailed Description

Definicja klasy HaszTab.

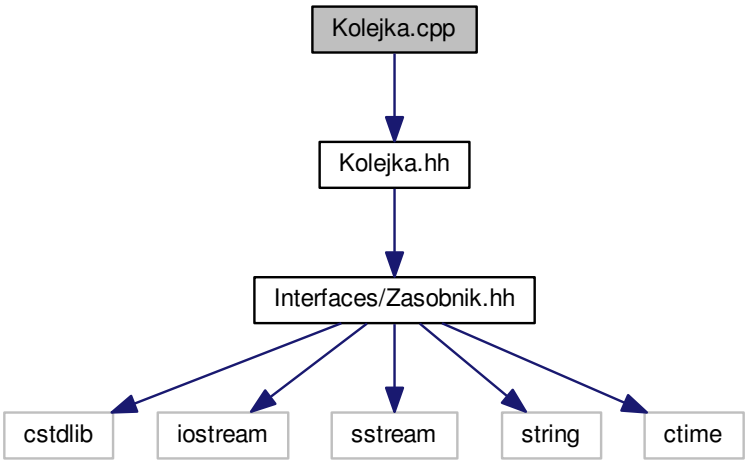
Plik zawiera definicje klasy modulujacej pojecie tablicy haszującej.

5.26 Kolejka.cpp File Reference

Metody klasy Kolejka.

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

Include dependency graph for Kolejka.cpp:



5.26.1 Detailed Description

Metody klasy Kolejka.

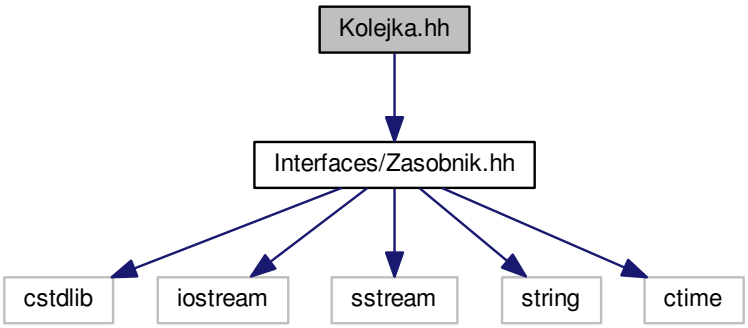
Plik zawiera metody klasy Kolejka.

5.27 Kolejka.hh File Reference

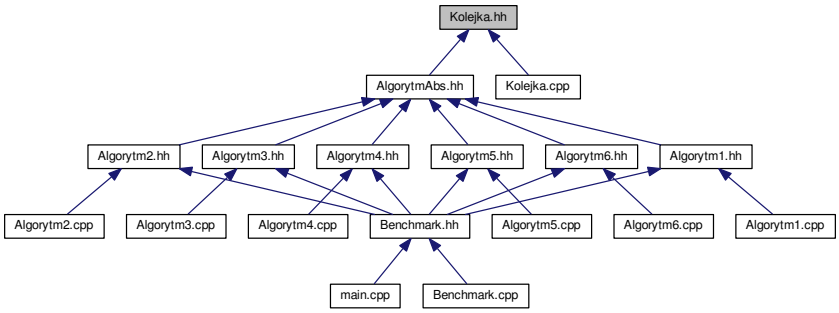
Definicja klasy Kolejka.

```
#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.

5.27.1 Detailed Description

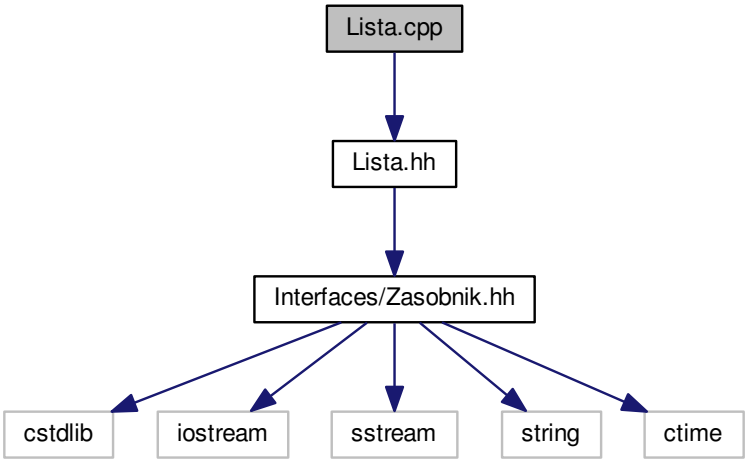
Definicja klasy Kolejka.

Plik zawiera definicje klasy Kolejka.

5.28 Lista.cpp File Reference

Metody klasy Lista.

```
#include "Lista.hh"
```
Include dependency graph for Lista.cpp:



5.28.1 Detailed Description

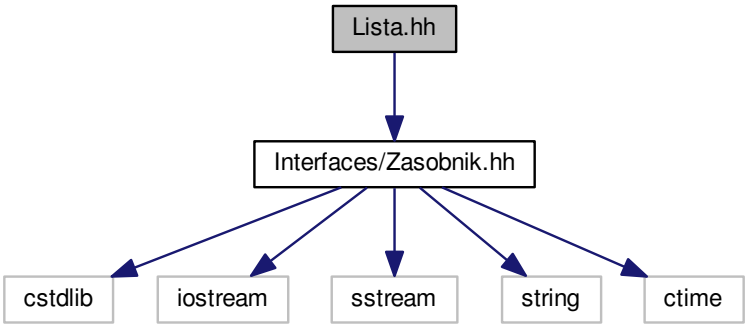
Metody klasy Lista.

Plik zawiera metody klasy Lista.

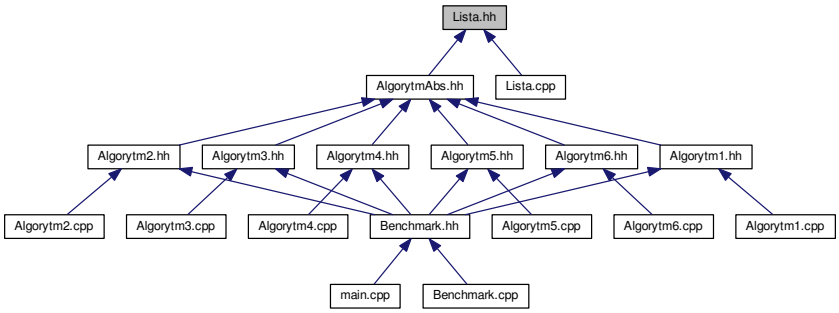
5.29 Lista.hh File Reference

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.

5.29.1 Detailed Description

Definicja klasy Lista.

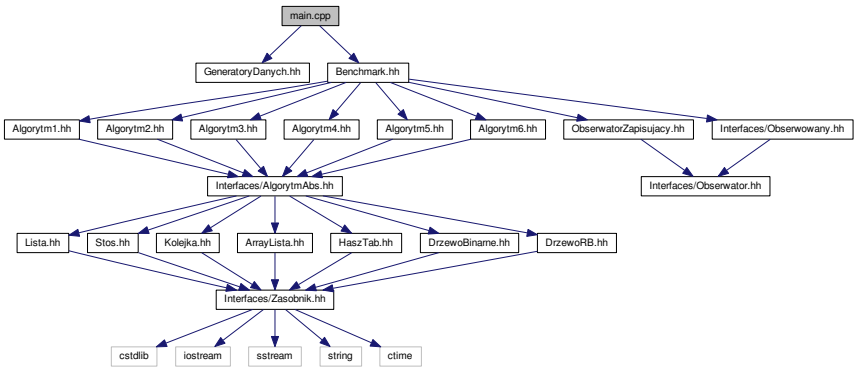
Plik zawiera definicje klasy modulujacej pojecie listy jednokierunkowej.

5.30 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.

5.30.1 Detailed Description

Modul glowny.

Plik zawiera funkcje main.

5.30.2 Function Documentation

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

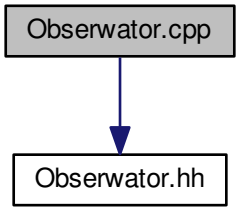
Funkcja glowna programu.

5.31 Obserwator.cpp File Reference

Metody klasy Obserwator.

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

Include dependency graph for Obserwator.cpp:



Functions

- void odswiez ()

5.31.1 Detailed Description

Metody klasy Obserwator.

Plik zawiera metody klasy Obserwator.

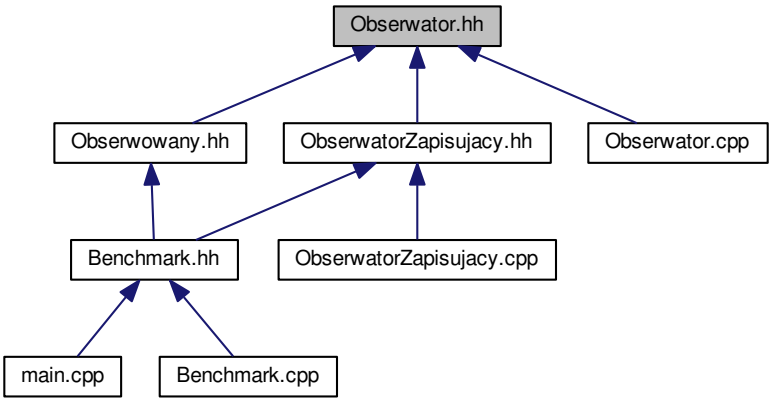
5.31.2 Function Documentation

5.31.2.1 void odswiez ( )

5.32 Obserwator.hh File Reference

Definicja klasy Obserwator.

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



Classes

- class Obserwator

  Klasa Obserwator.

5.32.1 Detailed Description

Definicja klasy Obserwator.

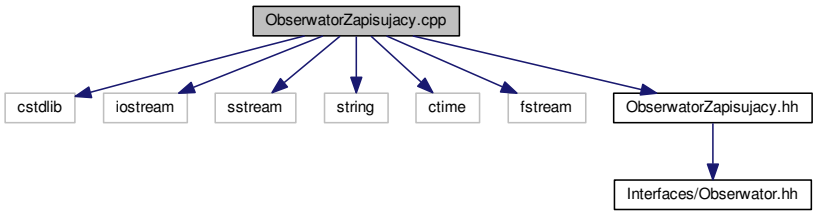
Plik zawiera definicje klasy Obserwator.

5.33 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:



5.33.1 Detailed Description

Metody klasy ObserwatorZapisujacy.

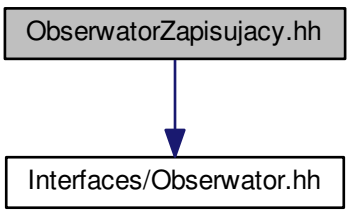
Plik zawiera metody klasy ObserwatorZapisujacy.

5.34 ObserwatorZapisujacy.hh File Reference

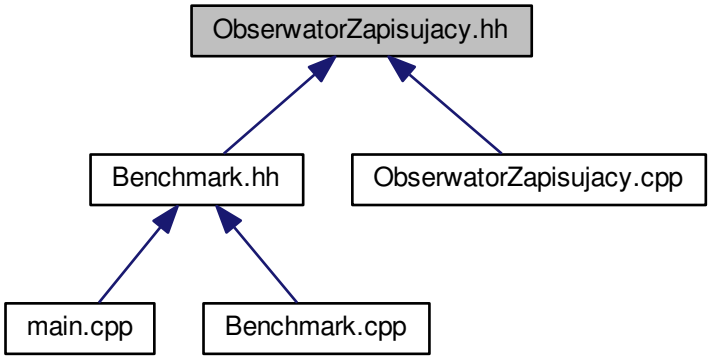
Definicja klasy ObserwatorZapisujacy.

```
#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.

5.34.1 Detailed Description

Definicja klasy ObserwatorZapisujacy.

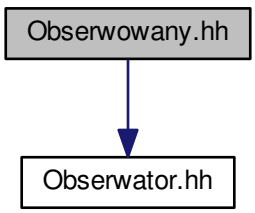
Plik zawiera definicje klasy ObserwatorZapisujacy.

5.35 Obserwowany.hh File Reference

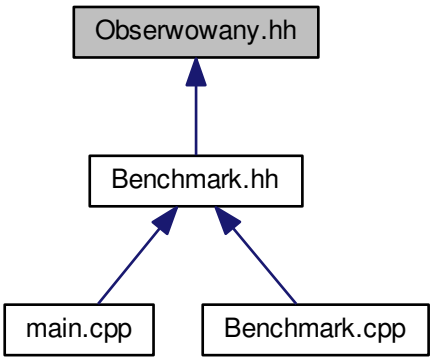
Definicja szablonu klasy abstrakcyjnej Obserwowany.

```
#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.

5.35.1 Detailed Description

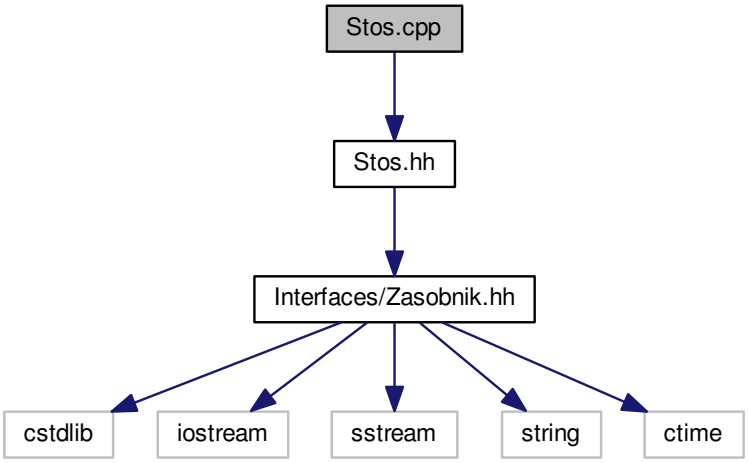
Definicja szablonu klasy abstrakcyjnej Obserwowany.

Plik zawiera definicje szablonu klasy abstrakcyjnej Obserwowany.

5.36 Stos.cpp File Reference

Metody klasy Stos.

```
#include "Stos.hh"
```
Include dependency graph for Stos.cpp:



5.36.1 Detailed Description

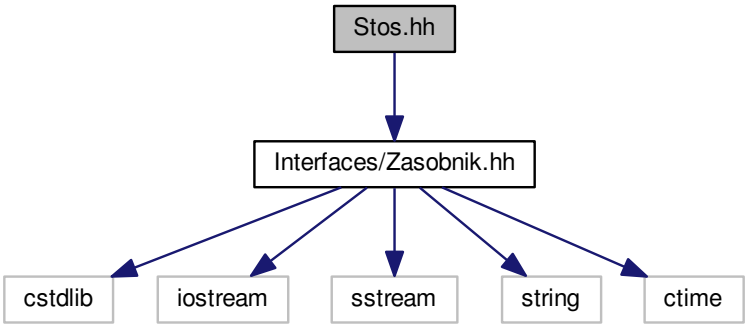
Metody klasy Stos.

Plik zawiera metody klasy Stos.

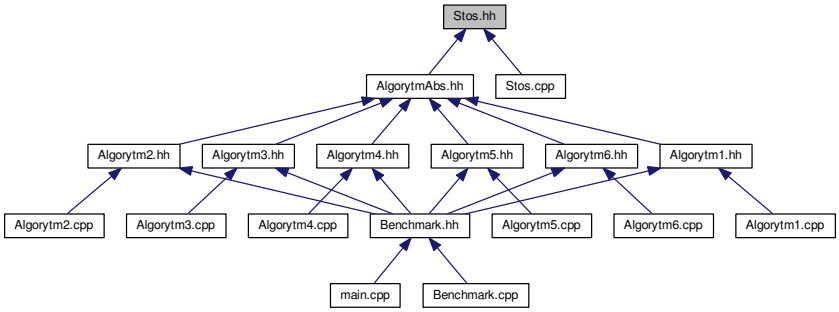
5.37 Stos.hh File Reference

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.

5.37.1 Detailed Description

Definicja klasy Stos.

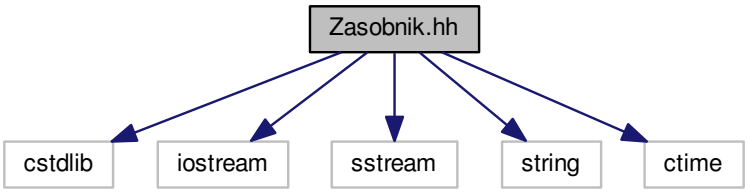
Plik zawiera definicje klasy Stos.

5.38 Zasobnik.hh File Reference

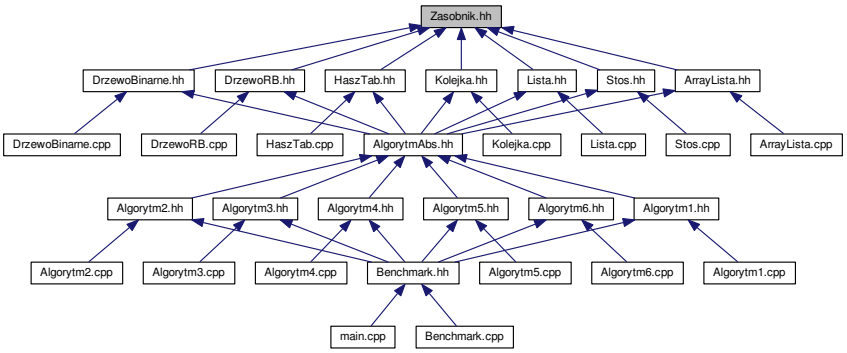
Definicja szablonu klasy abstrakcyjnej Zasobnik.

```
#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.

5.38.1 Detailed Description

Definicja szablonu klasy abstrakcyjnej Zasobnik.

Plik zawiera definicje szablonu klasy abstrakcyjnej Zasobnik.

Index