

PAMSI

6.01

Wygenerowano przez Doxygen 1.8.9.1

N, 17 kwi 2016 21:35:28

Spis treści

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

lista2s< typ >::bad_index	??
tablica::bad_index	??
Dictionary	??
tablica::empty	??
lista2s< typ >::empty	??
inKolejka	??
kolejka	??
inLista< typ >	??
lista2s< typ >	??
inLista< int >	??
lista	??
lista2s< int >	??
listatest	??
inLista< std::string >	??
lista2s< std::string >	??
inStos	??
stos	??
inTablica	??
tablica	??
kolejka	??
lista	??
stos	??
tablica	??
iRunnable	??
listatest	??
tablica	??
TabZHaszTest	??
ITabAsoc	??
TabAsoc	??
ITabZHasz	??
TabZHasz	??
TabZHaszTest	??
stoper	??
StrListyT< typ >	??
StrListyT< int >	??
StrListyT< std::string >	??

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

lista2s< typ >::bad_index	??
tablica::bad_index	
Obiekt zwracany podczas wystąpienia wyjątku występującego gdy nastąpi odwołanie do danych o złym indeksie	??
Dictionary	??
tablica::empty	
Obiekt zwracany podczas wystąpienia wyjątku występującego gdy nastąpi próba odzyskiwania na dane których kontener nie posiada	??
lista2s< typ >::empty	??
inKolejka	??
inLista< typ >	??
inStos	??
inTablica	??
iRunnable	??
ITabAsoc	
/file /brief plik zawiera interfejs wykorzystywany do tablicy asocjacyjnej	??
ITabZHasz	
/file /brief plik zawiera interfejs wykorzystywany do tablicy z haszowaniem	??
kolejka	
Klasa kolejka - jeden z abstrakcyjnych typów danych Klasa ma w swoim składzie metody służące do zarządzania kolejka, wszystkie operacje poza dodaniem nowej danej do pustej kolejki zgłaszają wyjątek	??
lista	??
lista2s< typ >	??
listatest	??
stoper	??
stos	??
StrListyT< typ >	??
TabAsoc	??
tablica	
Klasa tablica Klasa ma w swoim składzie metody służące do zarządzania tablica dynamiczna używane przez inne typy danych, może być też używana samodzielnie	??
TabZHasz	??
TabZHaszTest	??

Rozdział 3

Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

Dictionary.h	...	??
inkolejka.hh	...	??
inlista.hh		
	Plik zawiera interfejs używany w listach	??
instos.hh	...	??
intablica.hh	...	??
irunnable.hh		
	Zawiera interfejs do testowania zaimplementowanych algorytmów	??
itabasoc.hh	...	??
itabzhasz.hh	...	??
kolejka.hh		
	Plik zawiera definicje klasy kolejka	??
lista.hh		
	Plik zawiera klasę lista, która jest oparta na jednowymiarowej tablicy dynamicznej	??
lista2s.hh		
	Plik zawiera szablon klasy lista2s (str. ??), która jest zbudowana jak klasyczna lista	??
listatest.hh		
	Plik zawiera klasę listatest odpowiadającą za testowanie listy	??
sort.hh		
	Plik zawiera funkcje z metodami sortowania	??
stoper.hh	...	??
stos.hh		
	Plik zawiera definicje klasy stos	??
strukturalistyT.hh		
	Plik zawiera strukturę potrzebną do działania listy	??
tabasoc.hh	...	??
tablica.hh		
	Plik zawiera klasę tablica	??
tablicatest.hh		
	Plik zawiera klasę tablicatest	??
tabzhasz.hh	...	??
tabzhasztest.hh	...	??

Rozdział 4

Dokumentacja klas

4.1 Dokumentacja klasy `lista2s< typ >::bad_index`

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `lista2s.hh`

4.2 Dokumentacja klasy `tablica::bad_index`

obiekt zwracany podczas wystąpienia wyjątku występującego gdy nastąpi odwołanie do danych o złym indeksie

```
#include <tablica.hh>
```

4.2.1 Opis szczegółowy

obiekt zwracany podczas wystąpienia wyjątku występującego gdy nastąpi odwołanie do danych o złym indeksie

obiekt zwracany podczas wystąpienia wyjątku występującego gdy nastąpi próba oddziaływania na dane których kontener nie posiada

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `tablica.hh`

4.3 Dokumentacja klasy `Dictionary`

Metody publiczne

- void **setWords** ()
- string **getWords** (int i)
- int **countLines** ()
- string **RandomWords** ()
- int **RandomNumber** ()

4.3.1 Dokumentacja funkcji składowych

4.3.1.1 int `Dictionary::countLines` ()

funkcja zliczająca linie w pliku (ilosc slow w slowniku)

Zwraca

lines

4.3.1.2 string Dictionary::getWords (int *i*)

funkcja zwracająca słowo z tablicy z wyrazami ze słownika

Parametry

<i>i</i>	- numer indeksu zwracanego słowa
----------	----------------------------------

Zwraca

words_[*i*]

4.3.1.3 int Dictionary::RandomNumber ()

funkcja generująca losowe liczby

Zwraca

4.3.1.4 string Dictionary::RandomWords ()

funkcja generująca losowe słowa na podstawie słownika

Zwraca

4.3.1.5 void Dictionary::setWords ()

funkcja wczytująca słownik do tablicy

Dokumentacja dla tej klasy została wygenerowana z plików:

- Dictionary.h
- Dictionary.cpp

4.4 Dokumentacja klasy tablica::empty

obiekt zwracany podczas wystąpienia wyjątku występującego gdy nastąpi próba oddziaływania na dane których kontener nie posiada

```
#include <tablica.hh>
```

4.4.1 Opis szczegółowy

obiekt zwracany podczas wystąpienia wyjątku występującego gdy nastąpi próba oddziaływania na dane których kontener nie posiada

metoda sprawdza czy kontener jest pusty

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `tablica.hh`

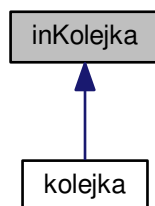
4.5 Dokumentacja klasy lista2s< typ >::empty

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `lista2s.hh`

4.6 Dokumentacja klasy inKolejka

Diagram dziedziczenia dla inKolejka



Metody publiczne

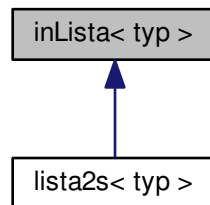
- virtual void **remove** ()=0
- virtual void **add** (int)=0
- virtual int **last** ()=0
- virtual int **first** ()=0

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `inkolejka.hh`

4.7 Dokumentacja szablonu klasy inLista< typ >

Diagram dziedziczenia dla inLista< typ >



Metody publiczne

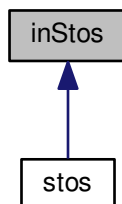
- virtual void **remove** (int)=0
usuwa dana o podanym indeksie, a nastepnie przestawia zmienne
- virtual void **remove** ()=0
usuwa ostatnia dana w kolejce
- virtual void **add** (typ, int)=0
dodaje dana na miejscu o podanym indeksie, jezeli nie moze byc tam wstawiona zglosi wyjatek, gdy miejsce jest zajete przestawia dane
- virtual void **add** (typ)=0
dodaje dana na koniec listy
- virtual typ **get** (int)=0
zwraca wartosc na podanym indeksie, jezeli nie moze tego zrobic, zwraci wyjatek
- virtual bool **isEmpty** ()=0
zwraca wartosc true gdy lista jest pusta lub false gdy jest na niej przynajmniej jeden element
- virtual int **size** ()=0
zwraca liczbe elementow na stosie

Dokumentacja dla tej klasy została wygenerowana z pliku:

- **inlista.hh**

4.8 Dokumentacja klasy inStos

Diagram dziedziczenia dla inStos



Metody publiczne

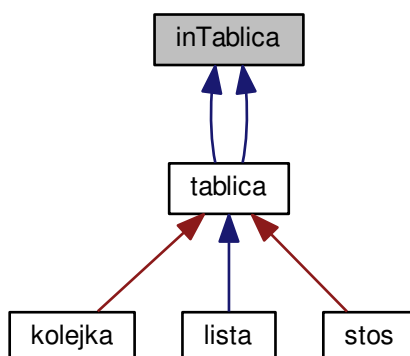
- virtual void **remove** ()=0
- virtual void **add** (int)=0
- virtual int **get** ()=0

Dokumentacja dla tej klasy została wygenerowana z pliku:

- instos.hh

4.9 Dokumentacja klasy inTablica

Diagram dziedziczenia dla inTablica



Metody publiczne

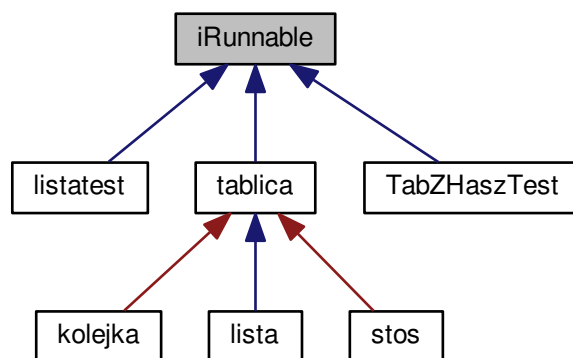
- virtual void **push** (int)=0
- virtual void **przypisz** (int, int)=0
- virtual void **zdejmij** (int)=0
- virtual int **size** ()=0
- virtual int **ind** ()=0
- virtual bool **isEmpty** ()=0

Dokumentacja dla tej klasy została wygenerowana z pliku:

- intablica.hh

4.10 Dokumentacja klasy iRunnable

Diagram dziedziczenia dla iRunnable



Metody publiczne

- virtual void **run** (int)=0
uruchamia testy dla zadanego algorytmu
- void **prepare** (int, int)

Atrybuty chronione

- int **ileOkrazen** =10
- int **ileDanych** =100000000

Dokumentacja dla tej klasy została wygenerowana z pliku:

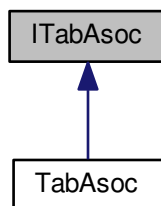
- irunnable.hh

4.11 Dokumentacja klasy ITabAsoc

/file /brief plik zawiera interfejs wykorzystywany do tablicy asocjacyjnej

```
#include <itabasoc.hh>
```

Diagram dziedziczenia dla ITabAsoc



Metody publiczne

- virtual int **search** (std::string)=0

szukanie wartosci dla frazy

- virtual void **insert** (std::string, int)=0

wstawienie wyrazu, o przypisanej pewnej wartosci calkowitej

4.11.1 Opis szczegółowy

/file /brief plik zawiera interfejs wykorzystywany do tablicy asocjacyjnej

Dokumentacja dla tej klasy została wygenerowana z pliku:

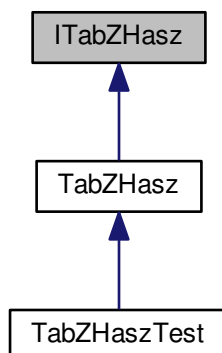
- itabasoc.hh

4.12 Dokumentacja klasy ITabZHasz

/file /brief plik zawiera interfejs wykorzystywany do tablicy z haszowaniem

```
#include <itabzhasz.hh>
```

Diagram dziedziczenia dla ITabZHasz



Metody publiczne

- virtual **TabAsoc kluczowanie** (std::string)=0

metoda ktora przypisuje kluczowi odpowiedni indeks po którym następuje późniejsze przeszukiwanie

- virtual **TabAsoc operator[]** (std::string)=0

przeciążenie operatora

4.12.1 Opis szczegółowy

/file /brief plik zawiera interfejs wykorzystywany do tablicy z haszowaniem

Dokumentacja dla tej klasy została wygenerowana z pliku:

- itabzhasz.hh

4.13 Dokumentacja klasy kolejka

Klasa kolejka - jeden z abstrakcyjnych typów danych Klasa ma w swoim składzie metody służące do zarządzania kolejką, wszystkie operacje poza dodaniem nowej danej do pustej kolejki zgłaszają wyjątek.

```
#include <kolejka.hh>
```

Diagram dziedziczenia dla kolejka

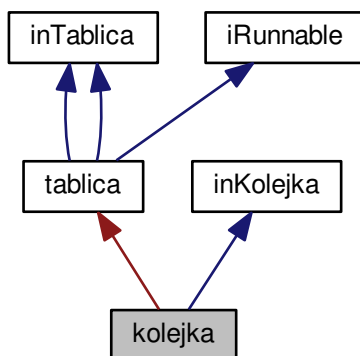
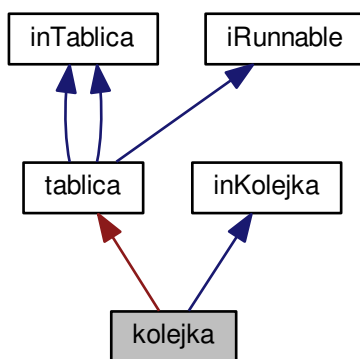


Diagram współpracy dla kolejka:



Metody publiczne

- void **remove** ()
usuwa pierwsza dana w kolejce
- void **add** (int)
dodaje dana na koniec kolejki
- int **first** ()
zwraca wartosc pierwszej danej

4.13.1 Opis szczegółowy

Klasa kolejka - jeden z abstrakcyjnych typów danych Klasa ma w swoim składzie metody służące do zarządzania kolejką, wszystkie operacje poza dodaniem nowej danej do pustej kolejki zgłaszają wyjątek.

Dokumentacja dla tej klasy została wygenerowana z plików:

- **kolejka.hh**
- kolejka.cpp

4.14 Dokumentacja klasy lista

Diagram dziedziczenia dla lista

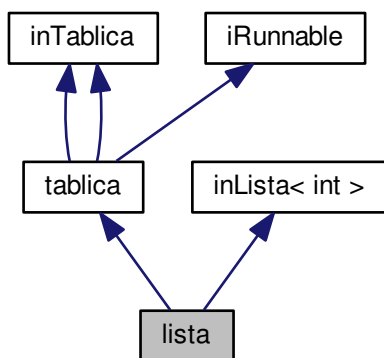
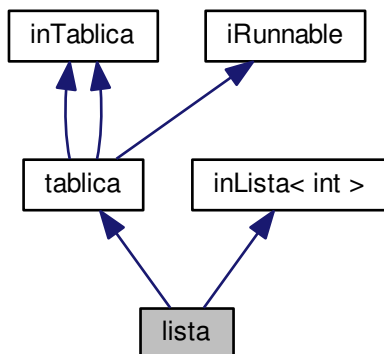


Diagram współpracy dla lista:



Metody publiczne

- void **remove** (int)
konstruktor bezparametryczny
- void **remove** ()
usuwa ostatnia dana w kolejce
- void **add** (int, int)
dodaje dana na miejscu o podanym indeksie, jezeli nie moze byc tam wstawiona zglosi wyjatek, gdy miejsce jest zajete przestawia dane
- void **add** (int)
dodaje dana na koniec listy
- int **get** (int)
zwraca wartosc na podanym indeksie, jezeli nie moze tego zrobic, zwroci wyjatek
- bool **isEmpty** ()
zwraca wartosc true gdy lista jest pusta lub false gdy jest na niej przynajmniej jeden element
- int **size** ()
zwraca liczbe elementow na stosie

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- lista.hh
- lista.cpp

4.15 Dokumentacja szablonu klasy lista2s< typ >

Diagram dziedziczenia dla lista2s< typ >

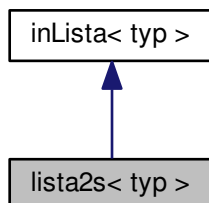
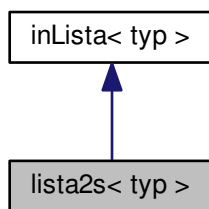


Diagram współpracy dla lista2s< typ >:



Komponenty

- class **bad_index**
- class **empty**

Metody publiczne

- **lista2s** ()
konstruktor bezparametryczny
- void **remove** (int)
usuwa dana o podanym indeksie, a nastepnie przestawia zmienne
- void **remove** ()
usuwa ostatnia dana w kolejce
- void **add** (typ, int)
dodaje dana na miejscu o podanym indeksie, jezeli nie moze byc tam wstawiona zglosi wyjatek, gdy miejsce jest zajete przestawia dane
- void **add** (typ)
dodaje dana na koniec listy
- typ **get** (int)
zwraca wartosc na podanym indeksie, jezeli nie moze tego zrobic, zwraci wyjatek
- bool **isEmpty** ()
zwraca wartosc true gdy lista jest pusta lub false gdy jest na niej przynajmniej jeden element
- int **size** ()
zwraca liczbe elementow na stosie
- int **find** (typ)

Atrybuty chronione

- **StrListyT**< typ > * **FIRST**
- int **rozmiar** =0

Dokumentacja dla tej klasy została wygenerowana z pliku:

- **lista2s.hh**

4.16 Dokumentacja klasy listatest

Diagram dziedziczenia dla listatest

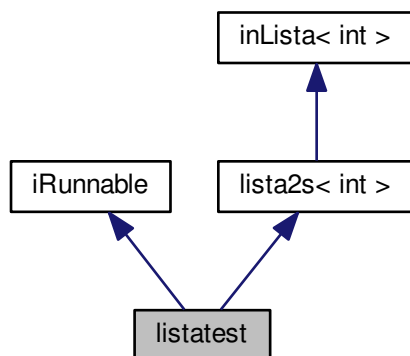
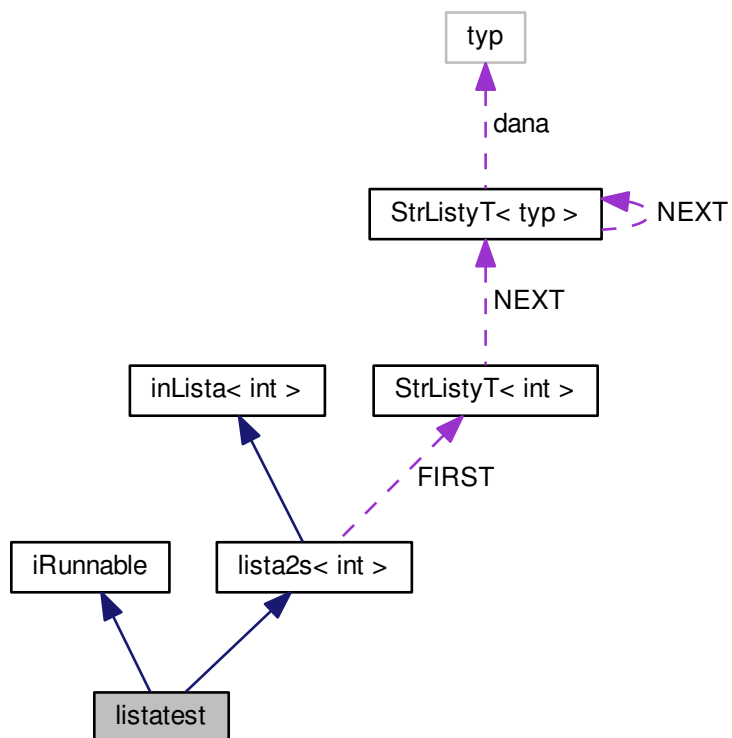


Diagram współpracy dla listatest:



Metody publiczne

- void **run** (int)
uruchamia testy dla zadanego algorytmu

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- listatest.hh
- listatest.cpp

4.17 Dokumentacja klasy stoper

Metody publiczne

- void **start** ()
- void **stop** ()
- std::chrono::duration< double > **getElapsedTime** ()
- std::chrono::duration< double > **getTime** ()
- bool **dumpToFile** (std::string)

Dokumentacja dla tej klasy została wygenerowana z plików:

- stoper.hh
- stoper.cpp

4.18 Dokumentacja klasy stos

Diagram dziedziczenia dla stos

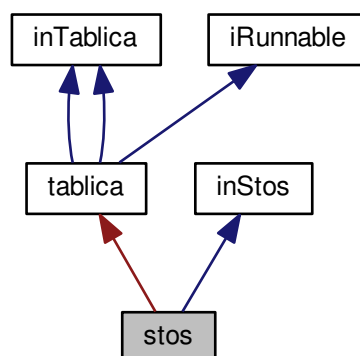
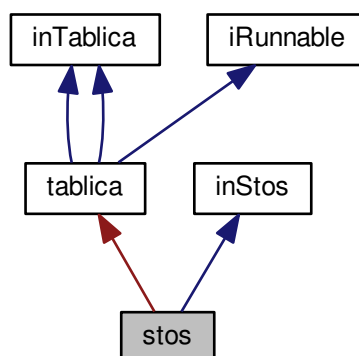


Diagram współpracy dla stos:



Metody publiczne

- **stos ()**

Klasa stos - jeden z abstrakcyjnych typow danych.

- void **remove ()**

konstruktor bezparametryczny

- void **add (int)**

*metoda **remove()** (str. ??)- nie przyjmuje wartosci, usuwa najwyzej polozona na stosie dana*

- int **get ()**

*metoda **add(int)** (str. ??)- przyjmuje wartosc int, bedaca nowa liczba do dodania na szczyt stosu*

4.18.1 Dokumentacja konstruktora i destruktora

4.18.1.1 stos::stos ()

Klasa stos - jeden z abstrakcyjnych typow danych.

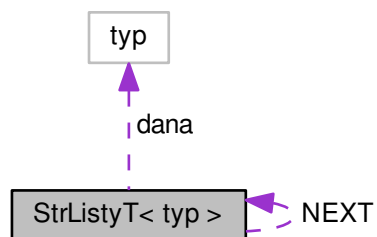
Klasa ma w swoim skladzie metody sluzace do zarzadzania stosem, mozliwe jest jedynie manipulowanie najwyzej polozonym elementem na stosie, inne operacje poza dodaniem nowej danej do niego na pustym stosie zgłaszają wyjątek

Dokumentacja dla tej klasy została wygenerowana z plików:

- **stos.hh**
- **stos.cpp**

4.19 Dokumentacja szablonu struktury StrListyT< typ >

Diagram współpracy dla StrListyT< typ >:



Atrybuty publiczne

- typ **dana**
- **StrListyT * NEXT**

Dokumentacja dla tej struktury została wygenerowana z pliku:

- **strukturalistyT.hh**

4.20 Dokumentacja klasy TabAsoc

Diagram dziedziczenia dla TabAsoc

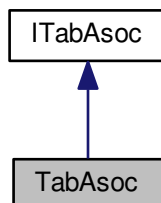
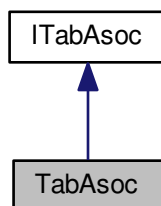


Diagram współpracy dla TabAsoc:



Metody publiczne

- int **search** (std::string)
szukanie wartosci dla frazy
- void **insert** (std::string, int)
wstawienie wyrazu, o przypisanej pewnej wartosci calkowitej

Dokumentacja dla tej klasy została wygenerowana z plików:

- tabasoc.hh
- tabasoc.cpp

4.21 Dokumentacja klasy tablica

Klasa tablica Klasa ma w swoim składzie metody sluzace do zarzadzania tablica dynamiczna uzywane przez inne typy danych, moze byc tez uzywana samodzielnie.

```
#include <tablica.hh>
```

Diagram dziedziczenia dla tablica

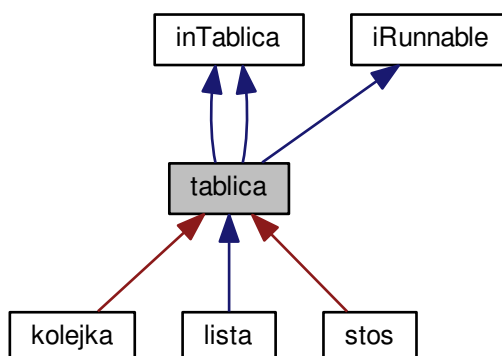
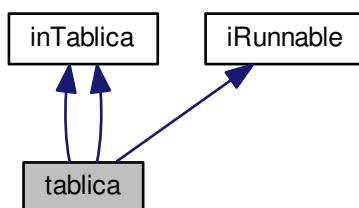


Diagram współpracy dla tablica:



Komponenty

- class **bad_index**
obiekt zwracany podczas wystąpienia wyjątku występującego gdy nastąpi odwołanie do danych o złym indeksie
- class **empty**
obiekt zwracany podczas wystąpienia wyjątku występującego gdy nastąpi próba oddziaływania na dane których kontener nie posiada

Metody publiczne

- void **zmniejsz** ()
zmniejsza ilość zaalokowanej pamięci
- **tablica** ()
konstruktor bezparametryczny, tworzy tablice o 10 polach
- **tablica** (int n)
- int **wyswietl** (int n)
konstruktor, tworzy tablice o podanej przez użytkownika ilości pol
- void **push** (int)
metoda dodająca dana na ostatnie wolne miejsce tablicy
- void **przypisz** (int, int)
przypisuje podana dana na konkretnym miejscu tablicy, pierwszym parametrem jest dana, a drugim miejsce na którym ma być zapisana
- void **zdejmij** (int)
usuwa ostatni element tablicy
- void **bubblesort** ()
sortowanie bombelkowe
- int **size** ()
metoda zwraca ilość zaalokowanego miejsca
- int **ind** ()
metoda zwraca indeks na którym jest ostatnia dana w kontenerze
- bool **isEmpty** ()
metoda sprawdza czy kontener jest pusty
- void **zmniejsz** ()
- **tablica** ()
zwiększa ilość zaalokowanej pamięci
- **tablica** (int n)

konstruktor bezparametryczny, tworzy tablice o 10 polach

- void **push** (int)

konstruktor, tworzy tablice o podanej przez uzytkownika ilosci pol

- void **przypisz** (int, int)

metoda dodajaca dana na ostatnie wolne miejsce tablicy

- void **zdejmij** (int)

przypisuje podana dana na konkretnym miejscu tablicy, pierwszym parametrem jest dana, a drugim miejsce na ktorym ma byc zapisana

- void **bubblesort** ()

usuwa ostatni element tablicy

- int **size** ()

sortowanie bombelkowe

- int **ind** ()

metoda zwraca ilosc zaalokowanego miejsca

- bool **isEmpty** ()

metoda zwraca indeks na ktorym jest ostatnia dana w kontenerze

- void **run** ()

obiekt zwracany podczas wystapienia wyjatku wystepujacego gdy nastapi odwołanie do danych o zlym indeksie

Atrybuty publiczne

- int **ile_elem**

Dodatkowe Dziedziczone Składowe

4.21.1 Opis szczegółowy

Klasa tablica Klasa ma w swoim składzie metody sluzace do zarzadzania tablica dynamiczna uzywane przez inne typy danych, moze byc tez uzywana samodzielnie.

Dokumentacja dla tej klasy została wygenerowana z plików:

- **tablica.hh**
- **tablicatest.hh**
- **tablica.cpp**
- **tablicatest.cpp**

4.22 Dokumentacja klasy TabZHasz

Diagram dziedziczenia dla TabZHasz

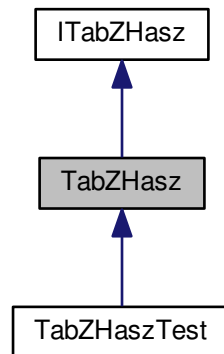
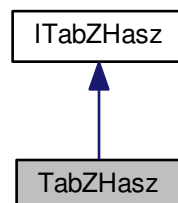


Diagram współpracy dla TabZHasz:



Metody publiczne

- **TabAsoc operator[]** (std::string)
przeciążenie operatora
- **TabZHasz ()**
konstruktor bezparametryczny, zakłada że liczba kontenerów = 10
- **TabZHasz (int ile)**
konstruktor z jednym parametrem określającym ilość kontenerów z danymi

Dokumentacja dla tej klasy została wygenerowana z plików:

- tabzhasz.hh
- tabzhasz.cpp

4.23 Dokumentacja klasy TabZHaszTest

Diagram dziedziczenia dla TabZHaszTest

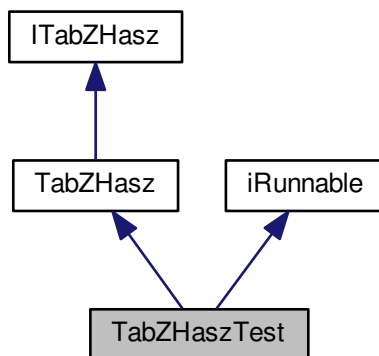
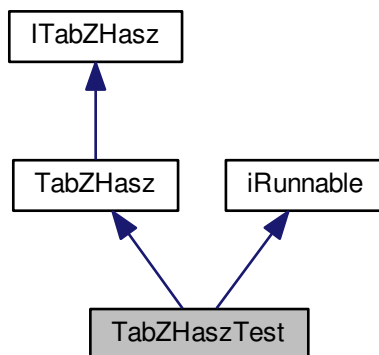


Diagram współpracy dla TabZHaszTest:



Metody publiczne

- void **run** (int)
uruchamia testy dla zadanego algorytmu

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- tabzhasztest.hh

- tabzhasztest.cpp

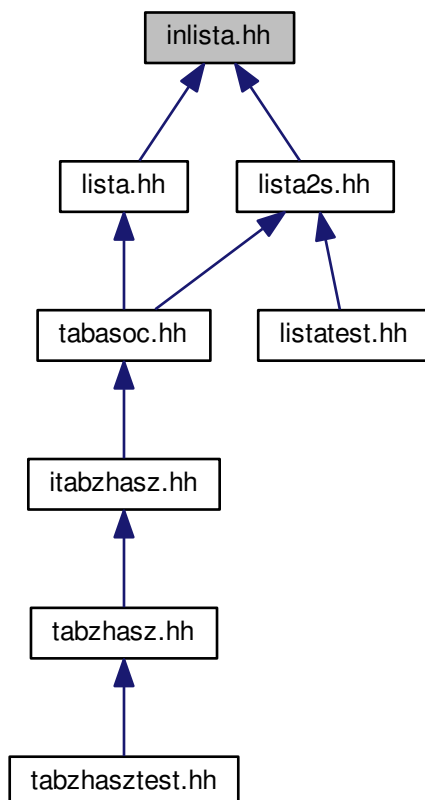
Rozdział 5

Dokumentacja plików

5.1 Dokumentacja pliku inlista.hh

plik zawiera interfejs używany w listach

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class **inLista**< **typ** >

5.1.1 Opis szczegółowy

plik zawiera interfejs używany w listach

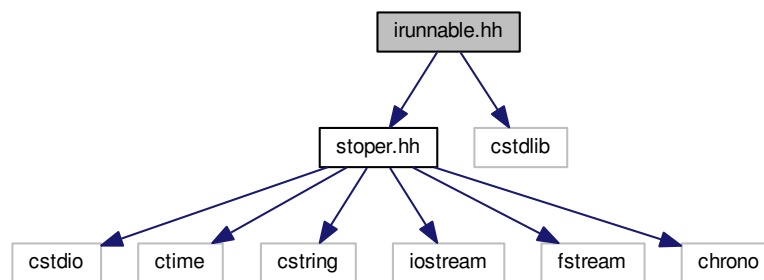
5.2 Dokumentacja pliku irunnable.hh

zawiera interfejs do testowania zaimplementowanych algorytmów

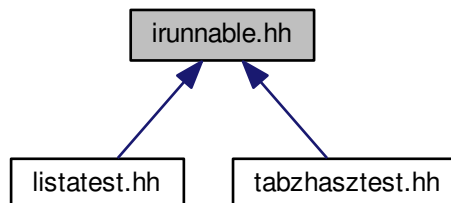
```
#include "stoper.hh"
```

```
#include <cstdlib>
```

Wykres zależności załączania dla irunnable.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class **iRunnable**

5.2.1 Opis szczegółowy

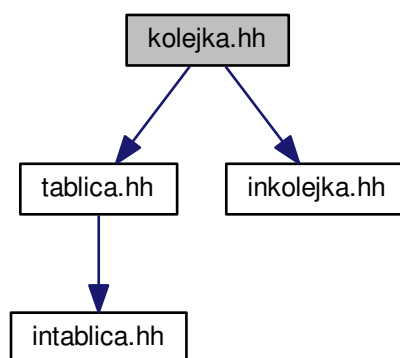
zawiera interfejs do testowania zaimplementowanych algorytmów

5.3 Dokumentacja pliku kolejka.hh

plik zawiera definicje klasy kolejka

```
#include "tablica.hh"  
#include "inkolejka.hh"
```

Wykres zależności załączania dla kolejka.hh:



Komponenty

- class **kolejka**

Klasa kolejka - jeden z abstrakcyjnych typów danych. Klasa ma w swoim składzie metody służące do zarządzania kolejką, wszystkie operacje poza dodaniem nowej danej do pustej kolejki zgłaszają wyjątek.

5.3.1 Opis szczegółowy

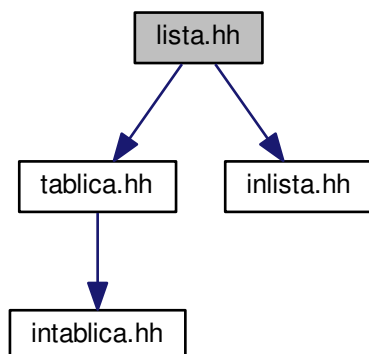
plik zawiera definicje klasy kolejka

5.4 Dokumentacja pliku lista.hh

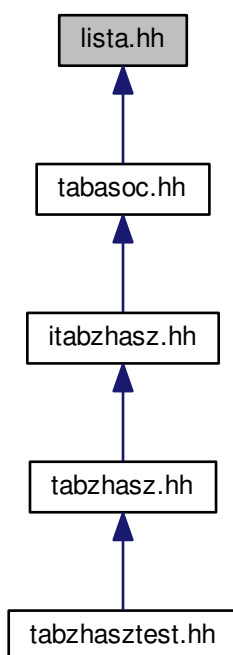
plik zawiera klasę lista, która jest oparta na jednowymiarowej tablicy dynamicznej

```
#include "tablica.hh"  
#include "inlista.hh"
```

Wykres zależności załączania dla lista.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class **lista**

5.4.1 Opis szczegółowy

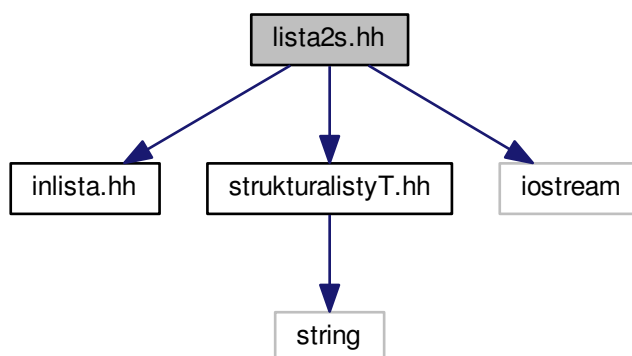
plik zawiera klasę `lista`, która jest oparta na jednowymiarowej tablicy dynamicznej

5.5 Dokumentacja pliku lista2s.hh

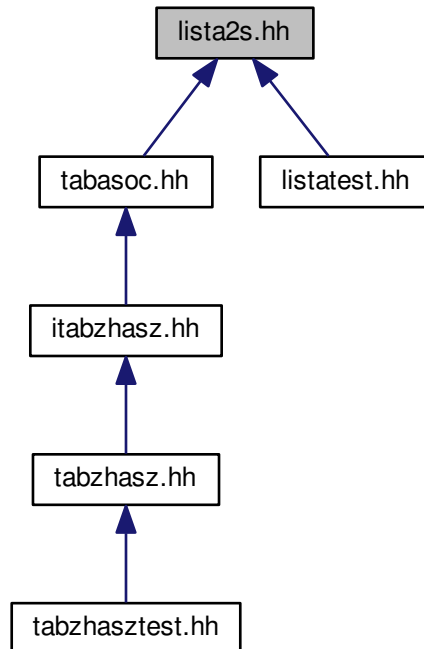
plik zawiera szablon klasy **lista2s** (str. ??), która jest zbudowana jak klasyczna lista

```
#include "inlista.hh"
#include "strukturalistyT.hh"
#include <iostream>
```

Wykres zależności załączania dla lista2s.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `lista2s< typ >`
- class `lista2s< typ >::bad_index`
- class `lista2s< typ >::empty`

5.5.1 Opis szczegółowy

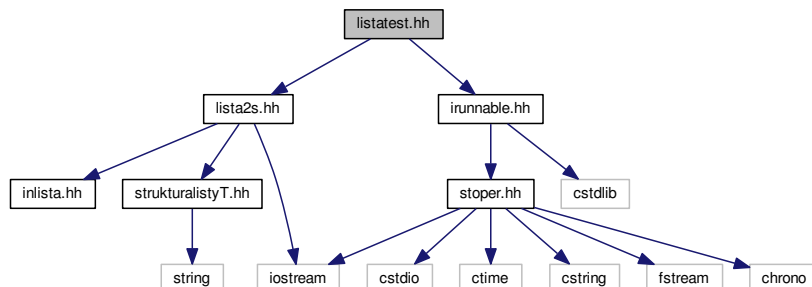
plik zawiera szablon klasy **lista2s** (str. ??), która jest zbudowana jak klasyczna lista

5.6 Dokumentacja pliku listatest.hh

plik zawiera klasę `listatest` odpowiadającą za testowanie listy

```
#include "lista2s.hh"
#include "irunnable.hh"
```

Wykres zależności załączania dla listatest.hh:



Komponenty

- class **listatest**

5.6.1 Opis szczegółowy

plik zawiera klasę listatest odpowiadającą za testowanie listy

5.7 Dokumentacja pliku sort.hh

plik zawiera funkcje z metodami sortowania

Funkcje

- void **bubblesort** (int tab[], int size)

5.7.1 Opis szczegółowy

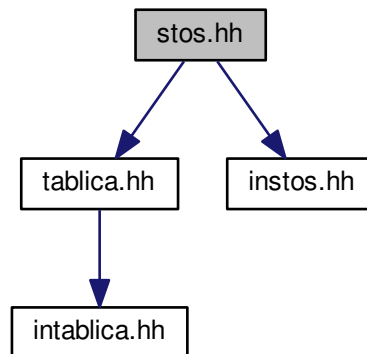
plik zawiera funkcje z metodami sortowania

5.8 Dokumentacja pliku stos.hh

plik zawiera definicję klasy stos

```
#include "tablica.hh"
#include "instos.hh"
```

Wykres zależności załączania dla stos.hh:



Komponenty

- class **stos**

5.8.1 Opis szczegółowy

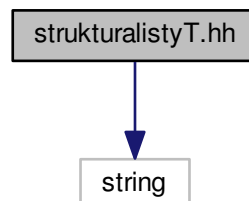
plik zawiera definicje klasy stos

5.9 Dokumentacja pliku strukturalistyT.hh

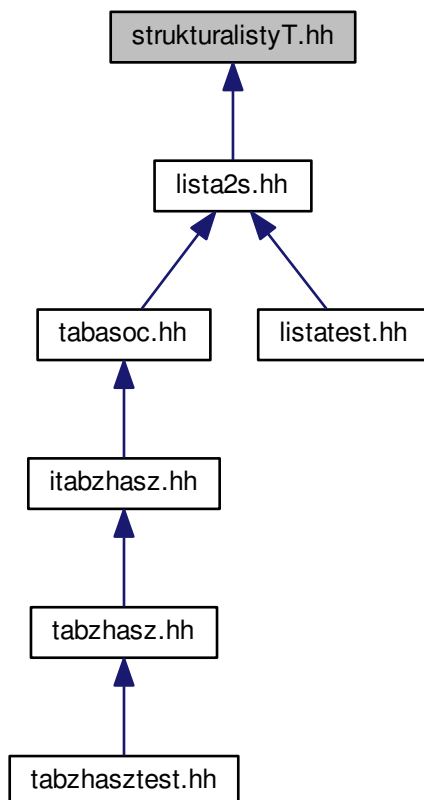
plik zawiera strukture potrzebna do działania listy

```
#include <string>
```

Wykres zależności załączania dla strukturalistyT.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- struct **StrListyT**< typ >

5.9.1 Opis szczegółowy

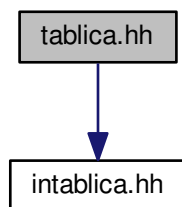
plik zawiera strukture potrzebna do dzialania listy

5.10 Dokumentacja pliku tablica.hh

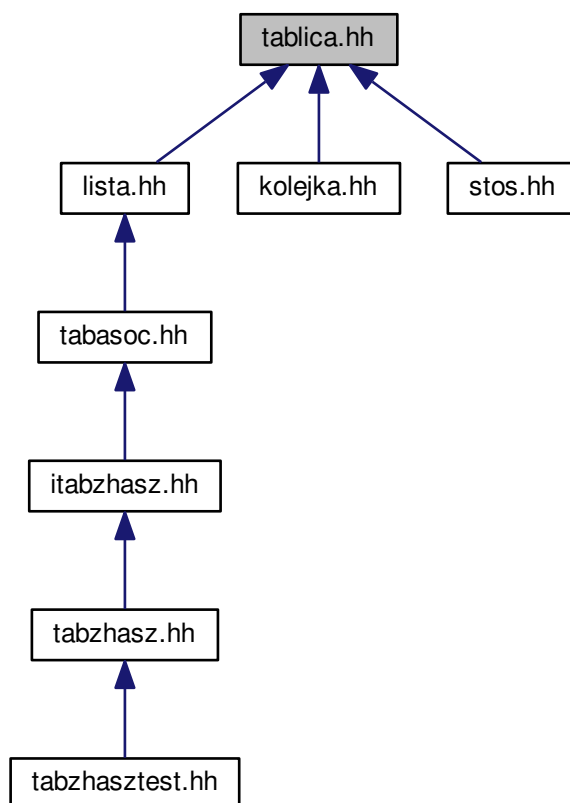
plik zawiera klase tablica

```
#include "intablica.hh"
```

Wykres zależności załączania dla tablica.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class **tablica**

Klasa tablica Klasa ma w swoim składzie metody sluzace do zarzadzania tablica dynamiczna uzywane przez inne typy danych, moze byc tez uzywana samodzielnie.

- class **tablica::empty**

obiekt zwracany podczas wystapienia wyjatku wystepujacego gdy nastapi proba odzialywania na dane ktorych kontener nie posiada

- class **tablica::bad_index**

obiekt zwracany podczas wystapienia wyjatku wystepujacego gdy nastapi odwołanie do danych o złym indeksie

5.10.1 Opis szczegółowy

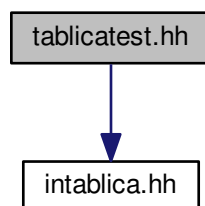
plik zawiera klase tablica

5.11 Dokumentacja pliku tablicatest.hh

plik zawiera klase tablicatest

```
#include "intablica.hh"
```

Wykres zależności załączania dla tablicatest.hh:



Komponenty

- class **tablica**

Klasa tablica Klasa ma w swoim składzie metody sluzace do zarzadzania tablica dynamiczna uzywane przez inne typy danych, moze byc tez uzywana samodzielnie.

- class **tablica::empty**

obiekt zwracany podczas wystapienia wyjatku wystepujacego gdy nastapi proba odzialywania na dane ktorych kontener nie posiada

- class **tablica::bad_index**

obiekt zwracany podczas wystapienia wyjatku wystepujacego gdy nastapi odwołanie do danych o złym indeksie

5.11.1 Opis szczegółowy

plik zawiera klase tablicatest

