

Lista, Stos, Kolejka
0.1

Wygenerowano przez Doxygen 1.8.6

N, 20 mar 2016 21:05:56

Spis treści

1 Indeks hierarchiczny	1
1.1 Hierarchia klas	1
2 Indeks klas	3
2.1 Lista klas	3
3 Indeks plików	5
3.1 Lista plików	5
4 Dokumentacja klas	7
4.1 Dokumentacja szablonu klasy <code>IList< Typ ></code>	7
4.1.1 Opis szczegółowy	7
4.1.2 Dokumentacja funkcji składowych	7
4.1.2.1 <code>add</code>	7
4.1.2.2 <code>get</code>	9
4.1.2.3 <code>isEmpty</code>	9
4.1.2.4 <code>remove</code>	9
4.1.2.5 <code>size</code>	9
4.2 Dokumentacja szablonu klasy <code>IQueue< Typ ></code>	10
4.2.1 Opis szczegółowy	10
4.2.2 Dokumentacja funkcji składowych	10
4.2.2.1 <code>dequeue</code>	10
4.2.2.2 <code>enqueue</code>	11
4.2.2.3 <code>front</code>	11
4.2.2.4 <code>isEmpty</code>	11
4.2.2.5 <code>size</code>	11
4.3 Dokumentacja szablonu klasy <code>IStack< Typ ></code>	11
4.3.1 Opis szczegółowy	12
4.3.2 Dokumentacja funkcji składowych	12
4.3.2.1 <code>isEmpty</code>	12
4.3.2.2 <code>pop</code>	12
4.3.2.3 <code>push</code>	12

4.3.2.4	size	13
4.3.2.5	top	13
4.4	Dokumentacja klasy List	13
4.4.1	Opis szczegółowy	14
4.5	Dokumentacja klasy Queue	14
4.5.1	Opis szczegółowy	15
4.6	Dokumentacja klasy Stack	15
4.6.1	Opis szczegółowy	16
5	Dokumentacja plików	17
5.1	Dokumentacja pliku IList.hh	17
5.1.1	Opis szczegółowy	18
5.2	Dokumentacja pliku IQueue.hh	18
5.2.1	Opis szczegółowy	19
5.3	Dokumentacja pliku IStack.hh	19
5.3.1	Opis szczegółowy	19
5.4	Dokumentacja pliku List.hh	20
5.4.1	Opis szczegółowy	20
5.5	Dokumentacja pliku Queue.hh	20
5.5.1	Opis szczegółowy	21
5.6	Dokumentacja pliku Stack.hh	21
5.6.1	Opis szczegółowy	22
Indeks		23

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

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

ICollection< Typ >	7
ICollection< string >	7
List	13
ICollection< Typ >	10
ICollection< double >	10
Queue	14
IStack< Typ >	11
IStack< double >	11
Stack	15

Rozdział 2

Indeks klas

2.1 Lista klas

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

IList< Typ >	
Szablon listy	7
IQueue< Typ >	
Szablon kolejki	10
IStack< Typ >	
Szablon stosu	11
List	13
Queue	14
Stack	15

Rozdział 3

Indeks plików

3.1 Lista plików

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

IList.hh	Definicja klasy IList	17
IQueue.hh	Definicja klasy IQueue	18
IStack.hh	Definicja klasy IStack	19
List.hh	Lista ze stringami	20
main.cpp	??
Queue.hh	Kolejka z elementami typu double	20
Stack.hh	Stos z elementami typu double	21

Rozdział 4

Dokumentacja klas

4.1 Dokumentacja szablonu klasy IList< Typ >

Szablon listy.

```
#include <IList.hh>
```

Metody publiczne

- void **add** (Typ item, int index)
Zapisywanie na liście.
- Typ **remove** (int index) throw (EmptyListException)
Ściąganie z listy.
- int **size** ()
Rozmiar listy.
- bool **isEmpty** ()
Czy pusty?
- Typ **get** (int index)
Element listy.

Atrybuty chronione

- Itablica1D< Typ > **Tablica**

4.1.1 Opis szczegółowy

```
template<typename Typ>class IList< Typ >
```

Na liście możemy wpisywać na każde miejsce i usuwać z każdego miejsca

Definicja w linii 19 pliku IList.hh.

4.1.2 Dokumentacja funkcji składowych

4.1.2.1 `template<typename Typ> void IList< Typ >::add (Typ item, int index) [inline]`

Wkłada element na dowolne miejsce na liście, jeżeli int index jest większy minimum o 2 od indexu ostatniego elementu to przestrzeń między elementami zostaje uzupełniona przez wartość 0. Jeżeli int index wskazuje na

miejsce, gdzie znajduje się element to zostaje on przepisany na kolejną pozycję, a na jego miejsce zostaje wpisany `int item`

Parametry

in	item	- element, który chcemy umieścić na liście
in	index	- miejsce, gdzie chcemy zapisać element (liczony od 0)

Definicja w linii 35 pliku IList.hh.

4.1.2.2 `template<typename Typ> Typ IList< Typ >::get (int index) [inline]`

Dostęp do dowolnego elementu listy

Wyjątki

<i>EmptyListException</i>	wyjątek pustej listy, wyrzucany gdy chcemy odczytać element o indexie którego nie ma na liście
---------------------------	--

Parametry

in	index	- Numer elementu, który chcemy odczytać, gdzie index 0 to pierwszy element
----	-------	--

Zwraca

Zwraca element, bez jego usuwania

Definicja w linii 85 pliku IList.hh.

4.1.2.3 `template<typename Typ> bool IList< Typ >::isEmpty () [inline]`

Sprawdza czy na liście znajdują się elementy

Zwracane wartości

<i>true</i>	- lista pusty
<i>false</i>	- na liście są elementy

Definicja w linii 71 pliku IList.hh.

4.1.2.4 `template<typename Typ> Typ IList< Typ >::remove (int index) throw EmptyListException) [inline]`

Usuwa element z listy

Wyjątki

<i>EmptyListException</i>	wyjątek pustej listy, wyrzucany gdy chcemy usunąć element o indexie którego nie ma na liście
---------------------------	--

Zwraca

Zwraca usunięty element

Definicja w linii 47 pliku IList.hh.

4.1.2.5 `template<typename Typ> int IList< Typ >::size () [inline]`

Rozmiar listy jest liczbą całkowitą liczoną od 0, gdy lista pusty,

Zwraca

Zwraca liczbę elementów zapisanych na liście

Definicja w linii 62 pliku IList.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [IList.hh](#)

4.2 Dokumentacja szablonu klasy IQueue< Typ >

Szablon kolejki.

```
#include <IQueue.hh>
```

Metody publiczne

- int [size](#) ()
Rozmiar kolejki.
- bool [isEmpty](#) ()
Czy pusty?
- void [enqueue](#) (Typ item)
Wkładanie do kolejki.
- Typ [dequeue](#) () throw (EmptyQueueException)
Wychodzenie z kolejki.
- Typ [front](#) () throw (EmptyQueueException)
Front kolejki.

Atrybuty chronione

- Itablica1D< Typ > **Tablica**

4.2.1 Opis szczegółowy

```
template<typename Typ>class IQueue< Typ >
```

Kolejka jest strukturą typu FIFO (First-In-Fist-Out), zaimplementowana na tablicy dynamicznej

Definicja w linii 19 pliku IQueue.hh.

4.2.2 Dokumentacja funkcji składowych

4.2.2.1 `template<typename Typ> Typ IQueue< Typ >::dequeue () throw EmptyQueueException)` `[inline]`

Usuwa element z początku kolejki

Wyjątki

<i>EmptyQueueException</i>	wyjątek pustej kolejki
----------------------------	------------------------

Zwraca

Zwraca usunięty element

Definicja w linii 63 pliku IQueue.hh.

4.2.2.2 `template<typename Typ> void IQueue< Typ >::enqueue (Typ item) [inline]`

Umieszcza element na końcu kolejki

Parametry

<i>in</i>	<i>item</i>	- element, który chcemy umieścić w kolejce
-----------	-------------	--

Definicja w linii 52 pliku IQueue.hh.

4.2.2.3 `template<typename Typ> Typ IQueue< Typ >::front () throw EmptyQueueException) [inline]`

Sprawdza co znajduje się na przodzie kolejki

Wyjątki

<i>EmptyQueueException</i>	wyjątek pustej kolejki
----------------------------	------------------------

Zwraca

Zwraca pierwszy element w kolejce, bez jego usuwania

Definicja w linii 78 pliku IQueue.hh.

4.2.2.4 `template<typename Typ> bool IQueue< Typ >::isEmpty () [inline]`

Sprawdza czy w kolejce znajdują się elementy

Zwracane wartości

<i>true</i>	- kolejka pusta
<i>false</i>	- w kolejce są elementy

Definicja w linii 40 pliku IQueue.hh.

4.2.2.5 `template<typename Typ> int IQueue< Typ >::size () [inline]`

Rozmiar kolejki jest liczbą całkowitą liczoną od 0, gdy stos pusty,

Zwraca

Zwraca liczbę elementów zapisanych w kolejce

Definicja w linii 31 pliku IQueue.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [IQueue.hh](#)

4.3 Dokumentacja szablonu klasy IStack< Typ >

Szablon stosu.

```
#include <IStack.hh>
```

Metody publiczne

- int [size](#) ()

- *Rozmiar stosu.*
- bool `isEmpty` ()
- *Czy pusty?*
- void `push` (Typ *item*)
- *Wkładanie na stos.*
- Typ `pop` () throw (EmptyStackException)
- *Ściąganie ze stosu.*
- Typ `top` () throw (EmptyStackException)
- *Szczyt stosu.*

Atrybuty chronione

- Itablica1D< Typ > **Tablica**

4.3.1 Opis szczegółowy

`template<typename Typ>class IStack< Typ >`

Stos jest strukturą typu LIFO (Last-In-Fist-Out), zaimplementowany na tablicy dynamicznej

Definicja w linii 20 pliku IStack.hh.

4.3.2 Dokumentacja funkcji składowych

4.3.2.1 `template<typename Typ> bool IStack< Typ >::isEmpty () [inline]`

Sprawdza czy na stosie znajdują się elementy

Zwracane wartości

<i>true</i>	- stos pusty
<i>false</i>	- na stosie są elementy

Definicja w linii 42 pliku IStack.hh.

4.3.2.2 `template<typename Typ> Typ IStack< Typ >::pop () throw EmptyStackException) [inline]`

Usuwa element ze szczytu stosu

Wyjątki

<i>EmptyStackException</i>	wyjątek pustego stosu
----------------------------	-----------------------

Zwraca

Zwraca usunięty element

Definicja w linii 66 pliku IStack.hh.

4.3.2.3 `template<typename Typ> void IStack< Typ >::push (Typ item) [inline]`

Wkłada element na szczyt stosu

Parametry

<i>in</i>	<i>item</i>	- element, który chcemy umieścić na stosie
-----------	-------------	--

Definicja w linii 54 pliku IStack.hh.

4.3.2.4 `template<typename Typ> int IStack< Typ >::size () [inline]`

Rozmiar stosu jest liczbą całkowitą liczoną od 0, gdy stos pusty,

Zwraca

Zwraca liczbę elementów zapisanych na stosie

Definicja w linii 32 pliku IStack.hh.

4.3.2.5 `template<typename Typ> Typ IStack< Typ >::top () throw EmptyStackException) [inline]`

Element na szczycie stosu

Wyjątki

<i>EmptyStackException</i>	wyjątek pustego stosu
----------------------------	-----------------------

Zwraca

Zwraca element ze szczytu, bez jego usuwania

Definicja w linii 81 pliku IStack.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [IStack.hh](#)

4.4 Dokumentacja klasy List

Diagram dziedziczenia dla List

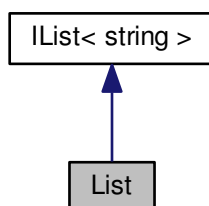
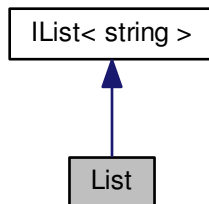


Diagram współpracy dla List:



Dodatkowe Dziedziczone Składowe

4.4.1 Opis szczegółowy

Definicja w linii 11 pliku `List.hh`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [List.hh](#)

4.5 Dokumentacja klasy Queue

Diagram dziedziczenia dla Queue

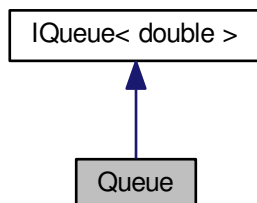
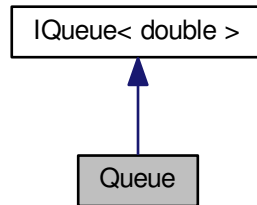


Diagram współpracy dla Queue:



Dodatkowe Dziedziczone Składowe

4.5.1 Opis szczegółowy

Definicja w linii 11 pliku `Queue.hh`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Queue.hh](#)

4.6 Dokumentacja klasy Stack

Diagram dziedziczenia dla Stack

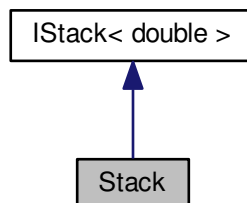
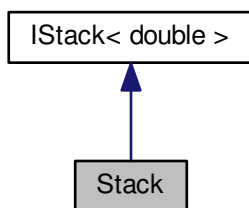


Diagram współpracy dla Stack:



Dodatkowe Dziedziczone Składowe

4.6.1 Opis szczegółowy

Definicja w linii 10 pliku Stack.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [Stack.hh](#)

Rozdział 5

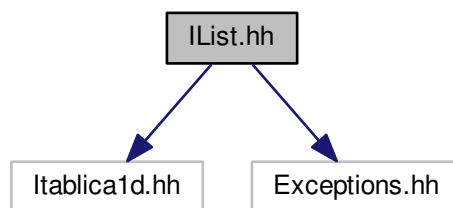
Dokumentacja plików

5.1 Dokumentacja pliku IList.hh

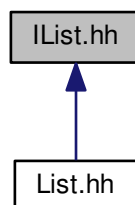
Definicja klasy [IList](#).

```
#include "Itablica1d.hh"  
#include "Exceptions.hh"
```

Wykres zależności załączania dla IList.hh:



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



Komponenty

- class `IList< Typ >`

Szablon listy.

5.1.1 Opis szczegółowy

Plik zawiera definicje szablonu klasy `IList`, który jest interfejsem listy

Definicja w pliku `IList.hh`.

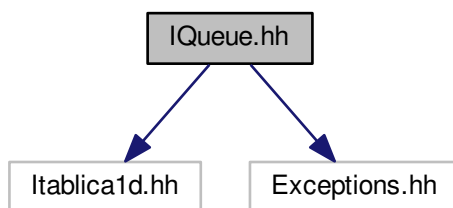
5.2 Dokumentacja pliku `IQueue.hh`

Definicja klasy `IQueue`.

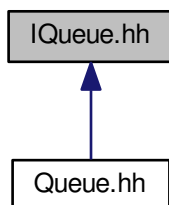
```
#include "Itablica1d.hh"
```

```
#include "Exceptions.hh"
```

Wykres zależności załączania dla `IQueue.hh`:



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



Komponenty

- class `IQueue< Typ >`

Szablon kolejki.

5.2.1 Opis szczegółowy

Plik zawiera definicje szablonu klasy [IQueue](#), który jest interfejsem kolejki

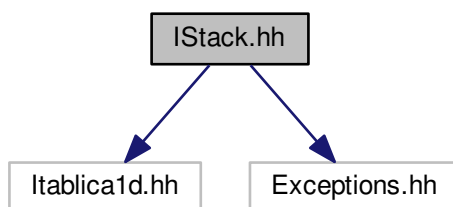
Definicja w pliku [IQueue.hh](#).

5.3 Dokumentacja pliku IStack.hh

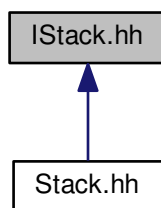
Definicja klasy [IStack](#).

```
#include "Itablica1d.hh"  
#include "Exceptions.hh"
```

Wykres zależności załączania dla IStack.hh:



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



Komponenty

- class [IStack< Typ >](#)
Szablon stosu.

5.3.1 Opis szczegółowy

Plik zawiera definicje szablonu klasy [IStack](#), który jest interfejsem stosu

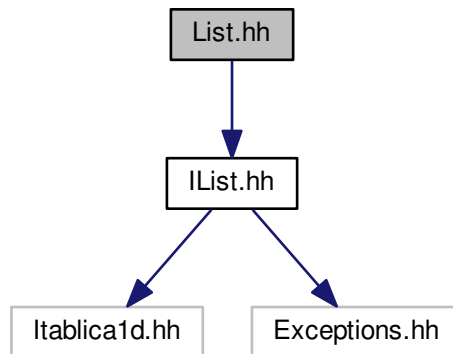
Definicja w pliku [IStack.hh](#).

5.4 Dokumentacja pliku List.hh

Lista ze stringami.

```
#include "IList.hh"
```

Wykres zależności załączania dla List.hh:



Komponenty

- class [List](#)

5.4.1 Opis szczegółowy

Plik zawiera przykład uszczegóławiania listy

Definicja w pliku [List.hh](#).

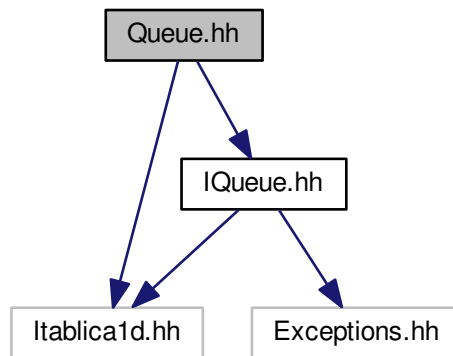
5.5 Dokumentacja pliku Queue.hh

Kolejka z elementami typu double.

```
#include "Itablica1d.hh"
```

```
#include "IQueue.hh"
```


Wykres zależności załączania dla Queue.hh:



Komponenty

- class [Queue](#)

5.5.1 Opis szczegółowy

Plik zawiera przykład uszczegóławiania kolejki

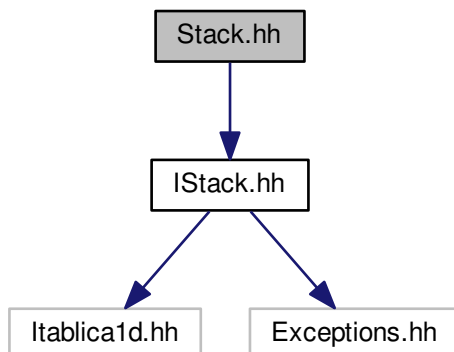
Definicja w pliku [Queue.hh](#).

5.6 Dokumentacja pliku Stack.hh

Stos z elementami typu double.

```
#include "IStack.hh"
```

Wykres zależności załączania dla Stack.hh:



Komponenty

- class [Stack](#)

5.6.1 Opis szczegółowy

Plik zawiera przykład uszczegóławiania stosu

Definicja w pliku [Stack.hh](#).

Skorowidz

add
 IList, [7](#)

dequeue
 IQueue, [10](#)

enqueue
 IQueue, [10](#)

front
 IQueue, [11](#)

get
 IList, [9](#)

IList
 add, [7](#)
 get, [9](#)
 isEmpty, [9](#)
 remove, [9](#)
 size, [9](#)

IList< Typ >, [7](#)

IList.hh, [17](#)

IQueue
 dequeue, [10](#)
 enqueue, [10](#)
 front, [11](#)
 isEmpty, [11](#)
 size, [11](#)

IQueue< Typ >, [10](#)

IQueue.hh, [18](#)

IStack
 isEmpty, [12](#)
 pop, [12](#)
 push, [12](#)
 size, [13](#)
 top, [13](#)

IStack< Typ >, [11](#)

IStack.hh, [19](#)

isEmpty
 IList, [9](#)
 IQueue, [11](#)
 IStack, [12](#)

List, [13](#)

List.hh, [20](#)

pop
 IStack, [12](#)

push
 IStack, [12](#)

Queue, [14](#)

Queue.hh, [20](#)

remove
 IList, [9](#)

size
 IList, [9](#)
 IQueue, [11](#)
 IStack, [13](#)

Stack, [15](#)

Stack.hh, [21](#)

top
 IStack, [13](#)