

Branch and bound algorithms

Generated by Doxygen 1.8.7

Tue May 17 2016 14:50:02

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	Graph Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	Graph	8
4.1.2.2	~Graph	8
4.1.3	Member Function Documentation	8
4.1.3.1	add_edge	8
4.1.3.2	add_vertex	8
4.1.3.3	BBsearch	8
4.1.3.4	BBsearch_with_extended_list	8
4.1.3.5	get_neighbours	9
4.1.3.6	is_connected	9
4.1.3.7	remove_edge	9
4.1.3.8	remove_vertex	9
4.1.3.9	search_path_BFS	10
4.1.3.10	search_path_DFS	10
4.1.3.11	visit_DFS	10
4.2	IGraph Class Reference	10
4.2.1	Detailed Description	11
4.2.2	Constructor & Destructor Documentation	11
4.2.2.1	~IGraph	11
4.2.3	Member Function Documentation	11

4.2.3.1	add_edge	11
4.2.3.2	add_vertex	11
4.2.3.3	get_neighbours	11
4.2.3.4	is_connected	11
4.2.3.5	remove_edge	11
4.2.3.6	remove_vertex	11
4.3	IList< E > Class Template Reference	12
4.3.1	Detailed Description	12
4.3.2	Constructor & Destructor Documentation	12
4.3.2.1	~IList	12
4.3.3	Member Function Documentation	12
4.3.3.1	add	12
4.3.3.2	at	12
4.3.3.3	remove	12
4.3.3.4	size	13
4.4	IPriorityQueue< E > Class Template Reference	13
4.4.1	Detailed Description	13
4.4.2	Constructor & Destructor Documentation	13
4.4.2.1	~IPriorityQueue	13
4.4.3	Member Function Documentation	13
4.4.3.1	add	13
4.4.3.2	remove	14
4.4.3.3	size	14
4.5	IQueue< E > Class Template Reference	14
4.5.1	Detailed Description	14
4.5.2	Constructor & Destructor Documentation	14
4.5.2.1	~IQueue	14
4.5.3	Member Function Documentation	14
4.5.3.1	add	14
4.5.3.2	remove	15
4.5.3.3	size	15
4.6	IRunnable Class Reference	15
4.6.1	Detailed Description	15
4.6.2	Member Function Documentation	15
4.6.2.1	run	15
4.7	List< E > Class Template Reference	15
4.7.1	Detailed Description	16
4.7.2	Constructor & Destructor Documentation	16
4.7.2.1	List	16
4.7.2.2	~List	16

4.7.3	Member Function Documentation	16
4.7.3.1	add	16
4.7.3.2	at	16
4.7.3.3	remove	17
4.7.3.4	show_list	17
4.7.3.5	size	17
4.8	Main_timer Class Reference	17
4.8.1	Detailed Description	18
4.8.2	Constructor & Destructor Documentation	18
4.8.2.1	~Main_timer	18
4.8.3	Member Function Documentation	18
4.8.3.1	get_ms_time	18
4.8.3.2	return_time	18
4.8.3.3	tim_start	18
4.8.3.4	tim_stop	18
4.9	Node< E > Class Template Reference	19
4.9.1	Detailed Description	19
4.9.2	Friends And Related Function Documentation	19
4.9.2.1	List< E >	19
4.10	PQNode< E > Class Template Reference	19
4.10.1	Detailed Description	19
4.10.2	Friends And Related Function Documentation	19
4.10.2.1	PriorityQueue< E >	19
4.11	PriorityQueue< E > Class Template Reference	20
4.11.1	Detailed Description	20
4.11.2	Constructor & Destructor Documentation	20
4.11.2.1	PriorityQueue	20
4.11.2.2	~PriorityQueue	20
4.11.3	Member Function Documentation	20
4.11.3.1	add	20
4.11.3.2	remove	21
4.11.3.3	show_queue	21
4.11.3.4	size	21
4.12	QNode< E > Class Template Reference	21
4.12.1	Detailed Description	21
4.12.2	Friends And Related Function Documentation	22
4.12.2.1	Queue< E >	22
4.13	Queue< E > Class Template Reference	22
4.13.1	Detailed Description	22
4.13.2	Constructor & Destructor Documentation	22

4.13.2.1	Queue	22
4.13.2.2	~Queue	23
4.13.3	Member Function Documentation	23
4.13.3.1	add	23
4.13.3.2	remove	23
4.13.3.3	show_queue	23
4.13.3.4	size	23
4.14	Test Class Reference	23
4.14.1	Detailed Description	24
4.14.2	Member Function Documentation	24
4.14.2.1	run	24
4.15	Timer Class Reference	24
4.15.1	Detailed Description	25
4.15.2	Constructor & Destructor Documentation	25
4.15.2.1	~Timer	25
4.15.3	Member Function Documentation	25
4.15.3.1	get_ms_time	25
4.15.3.2	return_time	25
4.15.3.3	tim_start	25
4.15.3.4	tim_stop	25
5	File Documentation	27
5.1	graph.cpp File Reference	27
5.2	graph.hh File Reference	27
5.3	igraph.hh File Reference	27
5.4	ilist.hh File Reference	27
5.5	ipriorityqueue.hh File Reference	28
5.6	iqueue.hh File Reference	28
5.7	irunnable.hh File Reference	28
5.8	list.cpp File Reference	28
5.9	list.hh File Reference	28
5.10	main.cpp File Reference	29
5.10.1	Function Documentation	29
5.10.1.1	main	29
5.11	maintimer.hh File Reference	29
5.12	priorityqueue.hh File Reference	29
5.13	queue.hh File Reference	29
5.14	test.cpp File Reference	30
5.15	test.hh File Reference	30
5.16	timer.hh File Reference	30

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

IGraph	10
Graph	7
ICollection< E >	12
List< E >	15
ICollection< int >	12
List< int >	15
IPriorityQueue< E >	13
PriorityQueue< E >	20
IQueue< E >	14
Queue< E >	22
IRunnable	15
Test	23
Main_timer	17
Timer	24
Node< E >	19
Node< int >	19
PQNode< E >	19
QNode< E >	21

Chapter 2

Class Index

2.1 Class List

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

Graph	
Klasa grafu	7
IGraph	
Interfejs grafu	10
IList< E >	
Interfejs listy	12
IPriorityQueue< E >	
Interfejs kolejki	13
IQueue< E >	
Interfejs kolejki	14
IRunnable	
Interfejs klasy rozruchowej	15
List< E >	
Klasa listy	15
Main_timer	
Interfejs stopera	17
Node< E >	
Klasa węzła listy	19
PQNode< E >	
Klasa węzła kolejki	19
PriorityQueue< E >	
Klasa kolejki	20
QNode< E >	
Klasa węzła kolejki	21
Queue< E >	
Klasa kolejki	22
Test	
Klasa rozruchowa	23
Timer	
Klasa stopera	24

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

graph.cpp	27
graph.hh	27
igraph.hh	27
ilist.hh	27
ipriorityqueue.hh	28
iqueue.hh	28
irunnable.hh	28
list.cpp	28
list.hh	28
main.cpp	29
maintimer.hh	29
priorityqueue.hh	29
queue.hh	29
test.cpp	30
test.hh	30
timer.hh	30

Chapter 4

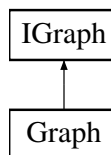
Class Documentation

4.1 Graph Class Reference

Klasa grafu.

```
#include <graph.hh>
```

Inheritance diagram for Graph:



Public Member Functions

- **Graph** (int vertices)
- **~Graph** ()
- void **add_vertex** (const int &v)
- void **add_edge** (const int &x, const int &y, const int &cost)
- void **remove_vertex** (const int &v)
- void **remove_edge** (const int &x, const int &y)
- **List**< int > **get_neighbours** (const int &v)
- bool **is_connected** (const int &x, const int &y)
- void **search_path_BFS** (const int &v)
- void **search_path_DFS** (const int &v)
- void **visit_DFS** (int i, const int &v)
- void **BBsearch** (const int &v)
- void **BBsearch_with_extended_list** (const int &v)

4.1.1 Detailed Description

Klasa grafu.

Zawiera metody umożliwiające operacje na grafie.

Definition at line 13 of file graph.hh.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Graph::Graph (int *vertices*)

Definition at line 4 of file graph.cpp.

4.1.2.2 Graph::~~Graph ()

Definition at line 21 of file graph.cpp.

4.1.3 Member Function Documentation

4.1.3.1 void Graph::add_edge (const int & *x*, const int & *y*, const int & *cost*) [virtual]

Metoda dodająca krawędź do grafu.

Parameters

<i>in</i>	<i>element</i>	typu int
<i>in</i>	<i>element</i>	typu int

Implements **IGraph** (p. 11).

Definition at line 55 of file graph.cpp.

4.1.3.2 void Graph::add_vertex (const int & *v*) [virtual]

Metoda dodająca wierzchołek do grafu. Ma zastosowanie w przypadku dodania dodatkowego wierzchołka po procedurze inicjacji całej struktury.

Parameters

<i>in</i>	<i>element</i>	typu int
-----------	----------------	----------

Implements **IGraph** (p. 11).

Definition at line 32 of file graph.cpp.

4.1.3.3 void Graph::BBsearch (const int & *v*)

Metoda przeszukująca graf wszerek wykorzystująca branch and bound

Parameters

<i>in</i>	<i>element</i>	typu int
-----------	----------------	----------

Returns

liczba przebytych ściezek

Definition at line 151 of file graph.cpp.

4.1.3.4 void Graph::BBsearch_with_extended_list (const int & *v*)

Metoda przeszukująca graf wszerek wykorzystująca branch and bound with extended list

Parameters

<i>in</i>	<i>element</i>	typu int
-----------	----------------	----------

Definition at line 213 of file graph.cpp.

4.1.3.5 `List< int > Graph::get_neighbours (const int & v) [virtual]`

Metoda zwracająca listę sąsiadów danego wierzchołka.

Parameters

<i>in</i>	<i>element</i>	typu int
-----------	----------------	----------

Returns

lista sąsiedztwa danego wierzchołka

Implements **IGraph** (p. 11).

Definition at line 76 of file graph.cpp.

4.1.3.6 `bool Graph::is_connected (const int & x, const int & y) [virtual]`

Metoda sprawdzająca istnienie krawędzi pomiędzy dwoma wierzchołkami.

Parameters

<i>in</i>	<i>element</i>	typu int
<i>in</i>	<i>element</i>	typu int

Returns

prawda lub fałsz

Implements **IGraph** (p. 11).

Definition at line 80 of file graph.cpp.

4.1.3.7 `void Graph::remove_edge (const int & x, const int & y) [virtual]`

Metoda usuwająca krawędź z grafu.

Parameters

<i>in</i>	<i>element</i>	typu int
<i>in</i>	<i>element</i>	typu int

Implements **IGraph** (p. 11).

Definition at line 70 of file graph.cpp.

4.1.3.8 `void Graph::remove_vertex (const int & v) [virtual]`

Metoda usuwająca wierzchołek z grafu.

Parameters

<i>in</i>	<i>element</i>	typu int
-----------	----------------	----------

Implements **IGraph** (p. 11).

Definition at line 64 of file graph.cpp.

4.1.3.9 void Graph::search_path_BFS (const int & v)

Metoda przeszukująca graf wszecz

Parameters

<i>in</i>	<i>element</i>	typu int
-----------	----------------	----------

Definition at line 89 of file graph.cpp.

4.1.3.10 void Graph::search_path_DFS (const int & v)

Metoda przeszukująca graf wglęb

Parameters

<i>in</i>	<i>element</i>	typu int
-----------	----------------	----------

Definition at line 136 of file graph.cpp.

4.1.3.11 void Graph::visit_DFS (int i, const int & v)

Metoda pomocnicza dla search_path_DFS

Parameters

<i>in</i>	<i>element</i>	typu int
-----------	----------------	----------

Definition at line 122 of file graph.cpp.

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

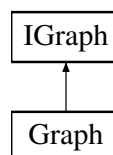
- **graph.hh**
- **graph.cpp**

4.2 IGraph Class Reference

Interfejs grafu.

```
#include <igraph.hh>
```

Inheritance diagram for IGraph:

**Public Member Functions**

- virtual void **add_vertex** (const int &v)=0

- virtual void **add_edge** (const int &x, const int &y, const int &cost)=0
- virtual void **remove_vertex** (const int &v)=0
- virtual void **remove_edge** (const int &x, const int &y)=0
- virtual **List**< int > **get_neighbours** (const int &v)=0
- virtual bool **is_connected** (const int &x, const int &y)=0
- virtual ~**IGraph** ()

4.2.1 Detailed Description

Interfejs grafu.

Zawiera metody umożliwiające operacje na grafie.

Definition at line 10 of file igraph.hh.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 virtual **IGraph**::~**IGraph** () [inline], [virtual]

Definition at line 20 of file igraph.hh.

4.2.3 Member Function Documentation

4.2.3.1 virtual void **IGraph**::**add_edge** (const int & x, const int & y, const int & cost) [pure virtual]

Implemented in **Graph** (p. 8).

4.2.3.2 virtual void **IGraph**::**add_vertex** (const int & v) [pure virtual]

Implemented in **Graph** (p. 8).

4.2.3.3 virtual **List**<int> **IGraph**::**get_neighbours** (const int & v) [pure virtual]

Implemented in **Graph** (p. 9).

4.2.3.4 virtual bool **IGraph**::**is_connected** (const int & x, const int & y) [pure virtual]

Implemented in **Graph** (p. 9).

4.2.3.5 virtual void **IGraph**::**remove_edge** (const int & x, const int & y) [pure virtual]

Implemented in **Graph** (p. 9).

4.2.3.6 virtual void **IGraph**::**remove_vertex** (const int & v) [pure virtual]

Implemented in **Graph** (p. 9).

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

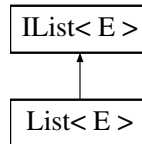
- **igraph.hh**

4.3 IList< E > Class Template Reference

Interfejs listy.

```
#include <ilist.hh>
```

Inheritance diagram for IList< E >:



Public Member Functions

- virtual void **add** (const E &elem, int i)=0
- virtual E **remove** (int i)=0
- virtual E **at** (int i)=0
- virtual int **size** ()=0
- virtual ~**IList** ()

4.3.1 Detailed Description

```
template<typename E>class IList< E >
```

Interfejs listy.

Zawiera metody umożliwiające operacje na liście.

Definition at line 10 of file ilist.hh.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 `template<typename E> virtual IList< E >::~~IList ()` `[inline]`, `[virtual]`

Definition at line 18 of file ilist.hh.

4.3.3 Member Function Documentation

4.3.3.1 `template<typename E> virtual void IList< E >::add (const E & elem, int i)` `[pure virtual]`

Implemented in **List< E >** (p. 16), and **List< int >** (p. 16).

4.3.3.2 `template<typename E> virtual E IList< E >::at (int i)` `[pure virtual]`

Implemented in **List< E >** (p. 16), and **List< int >** (p. 16).

4.3.3.3 `template<typename E> virtual E IList< E >::remove (int i)` `[pure virtual]`

Implemented in **List< E >** (p. 17), and **List< int >** (p. 17).

4.3.3.4 `template<typename E> virtual int IList< E >::size () [pure virtual]`

Implemented in **List< E >** (p. 17), and **List< int >** (p. 17).

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

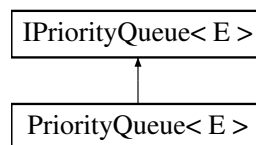
- **ilist.hh**

4.4 IPriorityQueue< E > Class Template Reference

Interfejs kolejki.

```
#include <ipriorityqueue.hh>
```

Inheritance diagram for IPriorityQueue< E >:



Public Member Functions

- virtual void **add** (const E &elem, const int &value)=0
- virtual E **remove** ()=0
- virtual int **size** ()=0
- virtual ~**IPriorityQueue** ()

4.4.1 Detailed Description

```
template<typename E>class IPriorityQueue< E >
```

Interfejs kolejki.

Zawiera metody umożliwiające operacje na kolejce.

Definition at line 10 of file ipriorityqueue.hh.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 `template<typename E> virtual IPriorityQueue< E >::~~IPriorityQueue () [inline], [virtual]`

Definition at line 17 of file ipriorityqueue.hh.

4.4.3 Member Function Documentation

4.4.3.1 `template<typename E> virtual void IPriorityQueue< E >::add (const E & elem, const int & value) [pure virtual]`

Implemented in **PriorityQueue< E >** (p. 20).

4.4.3.2 `template<typename E > virtual E IPriorityQueue< E >::remove () [pure virtual]`

Implemented in **PriorityQueue< E >** (p. 21).

4.4.3.3 `template<typename E > virtual int IPriorityQueue< E >::size () [pure virtual]`

Implemented in **PriorityQueue< E >** (p. 21).

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

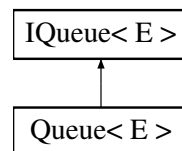
- **ipriorityqueue.hh**

4.5 IQueue< E > Class Template Reference

Interfejs kolejki.

```
#include <iqueue.hh>
```

Inheritance diagram for IQueue< E >:



Public Member Functions

- virtual void **add** (const E &elem)=0
- virtual E **remove** ()=0
- virtual int **size** ()=0
- virtual **~IQueue** ()

4.5.1 Detailed Description

```
template<typename E>class IQueue< E >
```

Interfejs kolejki.

Zawiera metody umożliwiające operacje na kolejce.

Definition at line 10 of file `iqueue.hh`.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 `template<typename E > virtual IQueue< E >::~IQueue () [inline],[virtual]`

Definition at line 17 of file `iqueue.hh`.

4.5.3 Member Function Documentation

4.5.3.1 `template<typename E > virtual void IQueue< E >::add (const E & elem) [pure virtual]`

Implemented in **Queue< E >** (p. 23).

4.5.3.2 `template<typename E> virtual E IQueue< E>::remove () [pure virtual]`

Implemented in **Queue< E >** (p. 23).

4.5.3.3 `template<typename E> virtual int IQueue< E>::size () [pure virtual]`

Implemented in **Queue< E >** (p. 23).

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

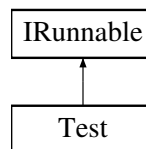
- **iqueue.hh**

4.6 IRunnable Class Reference

Interfejs klasy rozruchowej.

```
#include <irunnable.hh>
```

Inheritance diagram for IRunnable:



Public Member Functions

- virtual void **run** (int Argc, char *Argv[])=0

4.6.1 Detailed Description

Interfejs klasy rozruchowej.

Zawiera metodę umożliwiającą uruchomienie programu.

Definition at line 9 of file irunnable.hh.

4.6.2 Member Function Documentation

4.6.2.1 `virtual void IRunnable::run (int Argc, char * Argv[]) [pure virtual]`

Implemented in **Test** (p. 24).

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

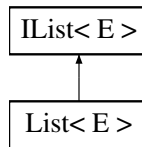
- **irunnable.hh**

4.7 List< E > Class Template Reference

Klasa listy.

```
#include <list.hh>
```

Inheritance diagram for List< E >:



Public Member Functions

- **List** ()
- **~List** ()
- void **add** (const E &elem, int i)
- E **remove** (int i)
- E **at** (int i)
- int **size** ()
- void **show_list** ()

4.7.1 Detailed Description

template<typename E>class List< E >

Klasa listy.

Zawiera metody umożliwiające operacje na liście.

Definition at line 11 of file list.hh.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 template<typename E> List< E >::List () [inline]

Definition at line 44 of file list.hh.

4.7.2.2 template<typename E> List< E >::~~List () [inline]

Definition at line 48 of file list.hh.

4.7.3 Member Function Documentation

4.7.3.1 template<typename E> void List< E >::add (const E & *elem*, int *i*) [virtual]

Funkcja dodająca element do listy

Parameters

in	<i>element</i>	typu E
in	<i>pozycja</i>	i

Implements **IList< E >** (p. 12).

Definition at line 98 of file list.hh.

4.7.3.2 template<typename E> int List< E >::at (int *i*) [virtual]

Funkcja zwracająca element listy na danej pozycji.

Parameters

<code>in</code>	<code>pozycja</code>	elementu
-----------------	----------------------	----------

Returns

Element typu E

Implements **IList**< **E** > (p. 12).

Definition at line 201 of file list.hh.

4.7.3.3 `template<typename E> int List< E >::remove (int i) [virtual]`

Funkcja usuwająca element z listy Wyrzuca wyjątek `EmptyListException` jeśli lista jest pusta oraz `WrongIndexException` jeśli wybrano zły indeks.

Returns

Element typu E

Implements **IList**< **E** > (p. 12).

Definition at line 139 of file list.hh.

4.7.3.4 `template<typename E> void List< E >::show_list ()`

Funkcja wyświetlająca listę

Definition at line 212 of file list.hh.

4.7.3.5 `template<typename E> int List< E >::size () [virtual]`

Funkcja zwracająca rozmiar listy

Returns

Rozmiar kolejki typu int

Implements **IList**< **E** > (p. 13).

Definition at line 196 of file list.hh.

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

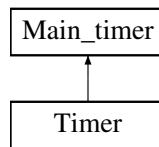
- **list.hh**
- **list.cpp**

4.8 Main_timer Class Reference

Interfejs stopera.

```
#include <maintimer.hh>
```

Inheritance diagram for Main_timer:



Public Member Functions

- virtual long double **get_ms_time** ()=0
- virtual void **tim_start** ()=0
- virtual void **tim_stop** ()=0
- virtual long double **return_time** ()=0
- virtual **~Main_timer** ()

4.8.1 Detailed Description

Interfejs stopera.

Zawiera metody umożliwiające mierzenie czasu.

Definition at line 9 of file maintimer.hh.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 virtual **Main_timer::~Main_timer** () [inline],[virtual]

Definition at line 16 of file maintimer.hh.

4.8.3 Member Function Documentation

4.8.3.1 virtual long double **Main_timer::get_ms_time** () [pure virtual]

Implemented in **Timer** (p. 25).

4.8.3.2 virtual long double **Main_timer::return_time** () [pure virtual]

Implemented in **Timer** (p. 25).

4.8.3.3 virtual void **Main_timer::tim_start** () [pure virtual]

Implemented in **Timer** (p. 25).

4.8.3.4 virtual void **Main_timer::tim_stop** () [pure virtual]

Implemented in **Timer** (p. 25).

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

- **maintimer.hh**

4.9 Node< E > Class Template Reference

Klasa węzła listy.

```
#include <list.hh>
```

Friends

- class **List**< E >

4.9.1 Detailed Description

```
template<typename E>class Node< E >
```

Klasa węzła listy.

Zawiera element węzła oraz wskaźnik na następny węzeł.

Definition at line 9 of file list.hh.

4.9.2 Friends And Related Function Documentation

4.9.2.1 `template<typename E> friend class List< E >` `[friend]`

Definition at line 21 of file list.hh.

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

- **list.hh**

4.10 PQNode< E > Class Template Reference

Klasa węzła kolejki.

```
#include <priorityqueue.hh>
```

Friends

- class **PriorityQueue**< E >

4.10.1 Detailed Description

```
template<typename E>class PQNode< E >
```

Klasa węzła kolejki.

Zawiera element węzła oraz wskaźnik na następny węzeł.

Definition at line 6 of file priorityqueue.hh.

4.10.2 Friends And Related Function Documentation

4.10.2.1 `template<typename E> friend class PriorityQueue< E >` `[friend]`

Definition at line 18 of file priorityqueue.hh.

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

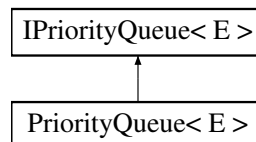
- **priorityqueue.hh**

4.11 PriorityQueue< E > Class Template Reference

Klasa kolejki.

```
#include <priorityqueue.hh>
```

Inheritance diagram for PriorityQueue< E >:



Public Member Functions

- **PriorityQueue** ()
- **~PriorityQueue** ()
- void **add** (const E &elem, const int &value)
- E **remove** ()
- int **size** ()
- void **show_queue** ()

4.11.1 Detailed Description

```
template<typename E>class PriorityQueue< E >
```

Klasa kolejki.

Zawiera metody umożliwiające operacje na kolejce.

Definition at line 8 of file priorityqueue.hh.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 `template<typename E> PriorityQueue< E >::PriorityQueue () [inline]`

Definition at line 41 of file priorityqueue.hh.

4.11.2.2 `template<typename E> PriorityQueue< E >::~~PriorityQueue () [inline]`

Definition at line 45 of file priorityqueue.hh.

4.11.3 Member Function Documentation

4.11.3.1 `template<typename E> void PriorityQueue< E >::add (const E &elem, const int &value) [virtual]`

Funkcja dodająca element do kolejki

Parameters

<i>in</i>	<i>element</i>	typu E
-----------	----------------	--------

Implements **IPriorityQueue< E >** (p. 13).

Definition at line 82 of file priorityqueue.hh.

4.11.3.2 `template<typename E> E PriorityQueue< E >::remove () [virtual]`

Funkcja usuwająca element z kolejki Wyrzuca wyjątek EmptyQueueException jeśli kolejka jest pusta.

Returns

Element typu E

Implements **IPriorityQueue< E >** (p. 14).

Definition at line 137 of file priorityqueue.hh.

4.11.3.3 `template<typename E> void PriorityQueue< E >::show_queue ()`

Funkcja wyświetlająca kolejkę

Definition at line 158 of file priorityqueue.hh.

4.11.3.4 `template<typename E> int PriorityQueue< E >::size () [virtual]`

Funkcja zwracająca rozmiar kolejki

Returns

Rozmiar kolejki typu int

Implements **IPriorityQueue< E >** (p. 14).

Definition at line 153 of file priorityqueue.hh.

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

- **priorityqueue.hh**

4.12 QNode< E > Class Template Reference

Klasa węzła kolejki.

```
#include <queue.hh>
```

Friends

- class **Queue< E >**

4.12.1 Detailed Description

```
template<typename E>class QNode< E >
```

Klasa węzła kolejki.

Zawiera element węzła oraz wskaźnik na następny węzeł.

Definition at line 6 of file queue.hh.

4.12.2 Friends And Related Function Documentation

4.12.2.1 `template<typename E> friend class Queue< E > [friend]`

Definition at line 18 of file queue.hh.

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

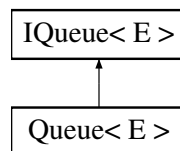
- **queue.hh**

4.13 Queue< E > Class Template Reference

Klasa kolejki.

```
#include <queue.hh>
```

Inheritance diagram for Queue< E >:



Public Member Functions

- **Queue** ()
- **~Queue** ()
- void **add** (const E &elem)
- E **remove** ()
- int **size** ()
- void **show_queue** ()

4.13.1 Detailed Description

```
template<typename E> class Queue< E >
```

Klasa kolejki.

Zawiera metody umożliwiające operacje na kolejce.

Definition at line 8 of file queue.hh.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 `template<typename E> Queue< E >::Queue () [inline]`

Definition at line 40 of file queue.hh.

4.13.2.2 `template<typename E> Queue< E >::~~Queue () [inline]`

Definition at line 44 of file queue.hh.

4.13.3 Member Function Documentation

4.13.3.1 `template<typename E> void Queue< E >::add (const E & elem) [virtual]`

Funkcja dodająca element do kolejki

Parameters

<i>in</i>	<i>element</i>	typu E
-----------	----------------	--------

Implements **IQueue< E >** (p. 14).

Definition at line 81 of file queue.hh.

4.13.3.2 `template<typename E> E Queue< E >::remove () [virtual]`

Funkcja usuwająca element z kolejki Wyrzuca wyjątek EmptyQueueException jeśli kolejka jest pusta.

Returns

Element typu E

Implements **IQueue< E >** (p. 15).

Definition at line 99 of file queue.hh.

4.13.3.3 `template<typename E> void Queue< E >::show_queue ()`

Funkcja wyświetlająca kolejkę

Definition at line 120 of file queue.hh.

4.13.3.4 `template<typename E> int Queue< E >::size () [virtual]`

Funkcja zwracająca rozmiar kolejki

Returns

Rozmiar kolejki typu int

Implements **IQueue< E >** (p. 15).

Definition at line 115 of file queue.hh.

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

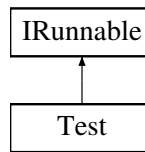
- **queue.hh**

4.14 Test Class Reference

Klasa rozruchowa.

```
#include <test.hh>
```

Inheritance diagram for Test:



Public Member Functions

- void **run** (int Argc, char *Argv[])

4.14.1 Detailed Description

Klasa rozruchowa.

Zawiera metodę umożliwiającą uruchomienie programu.

Definition at line 10 of file test.hh.

4.14.2 Member Function Documentation

4.14.2.1 void Test::run (int Argc, char * Argv[]) [virtual]

Implements **IRunnable** (p. 15).

Definition at line 7 of file test.cpp.

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

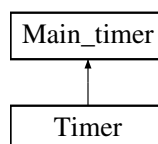
- **test.hh**
- **test.cpp**

4.15 Timer Class Reference

Klasa stopera.

```
#include <timer.hh>
```

Inheritance diagram for Timer:



Public Member Functions

- long double **get_ms_time** ()
- void **tim_start** ()
- void **tim_stop** ()
- long double **return_time** ()
- **~Timer** ()

4.15.1 Detailed Description

Klasa stopera.

Zawiera metody umożliwiające mierzenie czasu. Dokładny opis metod w dokumentacji projektu Lab2.

Definition at line 12 of file timer.hh.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 `Timer::~Timer ()` `[inline]`

Definition at line 22 of file timer.hh.

4.15.3 Member Function Documentation

4.15.3.1 `long double Timer::get_ms_time ()` `[virtual]`

Implements **Main_timer** (p. 18).

Definition at line 26 of file timer.hh.

4.15.3.2 `long double Timer::return_time ()` `[virtual]`

Implements **Main_timer** (p. 18).

Definition at line 48 of file timer.hh.

4.15.3.3 `void Timer::tim_start ()` `[virtual]`

Implements **Main_timer** (p. 18).

Definition at line 36 of file timer.hh.

4.15.3.4 `void Timer::tim_stop ()` `[virtual]`

Implements **Main_timer** (p. 18).

Definition at line 42 of file timer.hh.

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

- **timer.hh**

Chapter 5

File Documentation

5.1 graph.cpp File Reference

```
#include "graph.hh"
```

5.2 graph.hh File Reference

```
#include "igraph.hh"  
#include "list.hh"  
#include "queue.hh"  
#include "priorityqueue.hh"
```

Classes

- class **Graph**
Klasa grafu.

5.3 igraph.hh File Reference

```
#include "list.hh"
```

Classes

- class **IGraph**
Interfejs grafu.

5.4 ilist.hh File Reference

Classes

- class **IList< E >**
Interfejs listy.

5.5 ipriorityqueue.hh File Reference

Classes

- class **IPriorityQueue**< E >
Interfejs kolejki.

5.6 iqueue.hh File Reference

Classes

- class **IQueue**< E >
Interfejs kolejki.

5.7 irunnable.hh File Reference

Classes

- class **IRunnable**
Interfejs klasy rozruchowej.

5.8 list.cpp File Reference

```
#include "list.hh"  
#include <iostream>
```

5.9 list.hh File Reference

```
#include "ilist.hh"  
#include <cstddef>  
#include <cstring>  
#include <iostream>
```

Classes

- class **Node**< E >
Klasa węzła listy.
- class **List**< E >
Klasa listy.
- class **Node**< E >
Klasa węzła listy.
- class **List**< E >
Klasa listy.

5.10 main.cpp File Reference

```
#include "test.hh"  
#include <cstdlib>
```

Functions

- int **main** (int Argc, char *Argv[])

5.10.1 Function Documentation

5.10.1.1 int main (int Argc, char * Argv[])

Definition at line 4 of file main.cpp.

5.11 maintimer.hh File Reference

Classes

- class **Main_timer**
Interfejs stopera.

5.12 priorityqueue.hh File Reference

```
#include "ipriorityqueue.hh"
```

Classes

- class **PQNode**< E >
Klasa węzła kolejki.
- class **PriorityQueue**< E >
Klasa kolejki.
- class **PQNode**< E >
Klasa węzła kolejki.
- class **PriorityQueue**< E >
Klasa kolejki.

5.13 queue.hh File Reference

```
#include "iqueue.hh"
```

Classes

- class **QNode**< **E** >
Klasa węzła kolejki.
- class **Queue**< **E** >
Klasa kolejki.
- class **QNode**< **E** >
Klasa węzła kolejki.
- class **Queue**< **E** >
Klasa kolejki.

5.14 test.cpp File Reference

```
#include "test.hh"  
#include "graph.hh"  
#include "timer.hh"  
#include <cstdlib>  
#include <iostream>
```

5.15 test.hh File Reference

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

Classes

- class **Test**
Klasa rozruchowa.

5.16 timer.hh File Reference

```
#include <sys/time.h>  
#include <csddef>  
#include "maintimer.hh"
```

Classes

- class **Timer**
Klasa stopera.

Index

- ~Graph
 - Graph, 8
- ~IGraph
 - IGraph, 11
- ~IList
 - IList, 12
- ~IPriorityQueue
 - IPriorityQueue, 13
- ~IQueue
 - IQueue, 14
- ~List
 - List, 16
- ~Main_timer
 - Main_timer, 18
- ~PriorityQueue
 - PriorityQueue, 20
- ~Queue
 - Queue, 22
- ~Timer
 - Timer, 25
- add
 - IList, 12
 - IPriorityQueue, 13
 - IQueue, 14
 - List, 16
 - PriorityQueue, 20
 - Queue, 23
- add_edge
 - Graph, 8
 - IGraph, 11
- add_vertex
 - Graph, 8
 - IGraph, 11
- at
 - IList, 12
 - List, 16
- BBsearch
 - Graph, 8
- BBsearch_with_extended_list
 - Graph, 8
- get_ms_time
 - Main_timer, 18
 - Timer, 25
- get_neighbours
 - Graph, 9
 - IGraph, 11
- Graph, 7
- ~Graph, 8
- add_edge, 8
- add_vertex, 8
- BBsearch, 8
- BBsearch_with_extended_list, 8
- get_neighbours, 9
- Graph, 8
- is_connected, 9
- remove_edge, 9
- remove_vertex, 9
- search_path_BFS, 10
- search_path_DFS, 10
- visit_DFS, 10
- graph.cpp, 27
- graph.hh, 27
- IGraph, 10
 - ~IGraph, 11
 - add_edge, 11
 - add_vertex, 11
 - get_neighbours, 11
 - is_connected, 11
 - remove_edge, 11
 - remove_vertex, 11
- IList
 - ~IList, 12
 - add, 12
 - at, 12
 - remove, 12
 - size, 12
- IList< E >, 12
- IPriorityQueue
 - ~IPriorityQueue, 13
 - add, 13
 - remove, 13
 - size, 14
- IPriorityQueue< E >, 13
- IQueue
 - ~IQueue, 14
 - add, 14
 - remove, 14
 - size, 15
- IQueue< E >, 14
- IRunnable, 15
 - run, 15
- igraph.hh, 27
- ilist.hh, 27
- ipriorityqueue.hh, 28
- iqueue.hh, 28
- irunnable.hh, 28

- is_connected
 - Graph, 9
 - IGraph, 11
- List
 - ~List, 16
 - add, 16
 - at, 16
 - List, 16
 - remove, 17
 - show_list, 17
 - size, 17
- List< E >, 15
 - Node, 19
- list.cpp, 28
- list.hh, 28
- main
 - main.cpp, 29
- main.cpp, 29
 - main, 29
- Main_timer, 17
 - ~Main_timer, 18
 - get_ms_time, 18
 - return_time, 18
 - tim_start, 18
 - tim_stop, 18
- maintimer.hh, 29
- Node
 - List< E >, 19
- Node< E >, 19
- PQNode
 - PriorityQueue< E >, 19
- PQNode< E >, 19
- PriorityQueue
 - ~PriorityQueue, 20
 - add, 20
 - PriorityQueue, 20
 - remove, 21
 - show_queue, 21
 - size, 21
- PriorityQueue< E >, 20
 - PQNode, 19
- priorityqueue.hh, 29
- QNode
 - Queue< E >, 22
- QNode< E >, 21
- Queue
 - ~Queue, 22
 - add, 23
 - Queue, 22
 - remove, 23
 - show_queue, 23
 - size, 23
- Queue< E >, 22
 - QNode, 22
- queue.hh, 29
- remove
 - IList, 12
 - IPriorityQueue, 13
 - IQueue, 14
 - List, 17
 - PriorityQueue, 21
 - Queue, 23
- remove_edge
 - Graph, 9
 - IGraph, 11
- remove_vertex
 - Graph, 9
 - IGraph, 11
- return_time
 - Main_timer, 18
 - Timer, 25
- run
 - IRunnable, 15
 - Test, 24
- search_path_BFS
 - Graph, 10
- search_path_DFS
 - Graph, 10
- show_list
 - List, 17
- show_queue
 - PriorityQueue, 21
 - Queue, 23
- size
 - IList, 12
 - IPriorityQueue, 14
 - IQueue, 15
 - List, 17
 - PriorityQueue, 21
 - Queue, 23
- Test, 23
 - run, 24
- test.cpp, 30
- test.hh, 30
- tim_start
 - Main_timer, 18
 - Timer, 25
- tim_stop
 - Main_timer, 18
 - Timer, 25
- Timer, 24
 - ~Timer, 25
 - get_ms_time, 25
 - return_time, 25
 - tim_start, 25
 - tim_stop, 25
- timer.hh, 30
- visit_DFS
 - Graph, 10