Lista dwukierunkowa - Projekt C++

Wygenerowano za pomocą Doxygen 1.15.0

1 Struktura katalogów 1

1.1 Katalogi . . . 1

2 Indeks klas 3

2.1 Lista klas . . . 3

3 Indeks plików 5

3.1 Lista plików . . . 5

4 Dokumentacja katalogów 7

4.1 Dokumentacja katalogu include . . . 7

4.2 Dokumentacja katalogu src . . . 7

5 Dokumentacja klas 9

5.1 Dokumentacja szablonu klasy DoublyListIterator< T > . . . 9

5.1.1 Opis szczegółowy . . . 9

5.1.2 Dokumentacja składowych definicji typu . . . 10

5.1.2.1 iterator_category . . . 10

5.1.2.2 pointer . . . 10

5.1.2.3 reference . . . 10

5.1.2.4 value_type . . . 10

5.1.3 Dokumentacja konstruktora i destruktora . . . 10

5.1.3.1 DoublyListIterator() . . . 10

5.1.4 Dokumentacja funkcji składowych . . . 10

5.1.4.1 operator"!=() . . . 10

5.1.4.2 operator∗() . . . 10

5.1.4.3 operator++() [1/2] . . . 11

5.1.4.4 operator++() [2/2] . . . 11

5.1.4.5 operator--() [1/2] . . . 11

5.1.4.6 operator--() [2/2] . . . 11

5.1.4.7 operator->() . . . 11

5.1.4.8 operator==() . . . 11

5.2 Dokumentacja szablonu klasy DwukierunkowaLista< T > . . . 11

5.2.1 Opis szczegółowy . . . 12

5.2.2 Dokumentacja konstruktora i destruktora . . . 12

5.2.2.1 DwukierunkowaLista() . . . 12

5.2.2.2 ∼DwukierunkowaLista() . . . 12

5.2.3 Dokumentacja funkcji składowych . . . 13

5.2.3.1 begin() . . . 13

5.2.3.2 clear() . . . 13

5.2.3.3 display_forward() . . . 13

5.2.3.4 display_reverse() . . . 13

5.2.3.5 empty() . . . 13

5.2.3.6 end() . . . 13

5.2.3.7 getAt() . . . 13
5.2.3.8 getHead() . . . 13
5.2.3.9 getTail() . . . 14
5.2.3.10 insertAt() . . . 14
5.2.3.11 pop_back() . . . 14
5.2.3.12 pop_front() . . . 14
5.2.3.13 push_back() . . . 14
5.2.3.14 push_front() . . . 14
5.2.3.15 removeAt() . . . 14
5.2.3.16 size() . . . 15
5.3 Dokumentacja szablonu struktury Node< T > . . . 15
5.3.1 Opis szczegółowy . . . 15
5.3.2 Dokumentacja konstruktora i destruktora . . . 15
5.3.2.1 Node() . . . 15
5.3.3 Dokumentacja atrybutów składowych . . . 16
5.3.3.1 data . . . 16
5.3.3.2 next . . . 16
5.3.3.3 prev . . . 16
5.4 Dokumentacja szablonu klasy NodeFactory< T > . . . 16
5.4.1 Opis szczegółowy . . . 16
5.4.2 Dokumentacja funkcji składowych . . . 17
5.4.2.1 createNode() . . . 17
5.4.2.2 destroyNode() . . . 17

6 Dokumentacja plików **19**
6.1 Dokumentacja pliku include/DwukierunkowaLista.h . . . 19
6.1.1 Opis szczegółowy . . . 19
6.2 DwukierunkowaLista.h . . . 19
6.3 Dokumentacja pliku include/Iterator.h . . . 20
6.3.1 Opis szczegółowy . . . 20
6.4 Iterator.h . . . 21
6.5 Dokumentacja pliku include/Node.h . . . 21
6.5.1 Opis szczegółowy . . . 21
6.6 Node.h . . . 22
6.7 Dokumentacja pliku include/NodeFactory.h . . . 22
6.7.1 Opis szczegółowy . . . 22
6.8 NodeFactory.h . . . 22
6.9 Dokumentacja pliku src/DwukierunkowaLista.cpp . . . 23
6.9.1 Opis szczegółowy . . . 23
6.10 Dokumentacja pliku src/Iterator.cpp . . . 23
6.10.1 Opis szczegółowy . . . 23
6.11 Dokumentacja pliku src/main.cpp . . . 23

6.11.1 Opis szczegółowy 24
6.11.2 Dokumentacja funkcji 24
6.11.2.1 main() 24
6.12 Dokumentacja pliku src/Node.cpp 24
6.12.1 Opis szczegółowy 24
6.13 Dokumentacja pliku src/NodeFactory.cpp 24
6.13.1 Opis szczegółowy 24

7 Przykłady **25**
7.1 main.cpp 25

Skorowidz **27**

Rozdział 1

Struktura katalogów

1.1 Katalogi

include 7
DwukierunkowaLista.h 19
Iterator.h 20
Node.h 21
NodeFactory.h 22
src 7
DwukierunkowaLista.cpp 23
Iterator.cpp 23
main.cpp 23
Node.cpp 24
NodeFactory.cpp 24

Rozdział 2

Indeks klas

2.1 Lista klas

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

DoublyListIterator< T >
: Szablonowy iterator klasy DoublyLinkedList 9
DwukierunkowaLista< T >
: Szablonowa implementacja listy dwukierunkowej 11
Node< T >
: Szablonowa struktura węzła listy dwukierunkowej 15
NodeFactory< T >
: Szablonowa klasa fabryczna tworząca i usuwająca węzły listy 16

Rozdział 3

Indeks plików

3.1 Lista plików

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

include/DwukierunkowaLista.h
Deklaracja szablonowej klasy DwukierunkowaLista 19
include/Iterator.h
Deklaracja szablonowego iteratora dla listy dwukierunkowej 20
include/Node.h
Deklaracja szablonowej struktury Node – pojedynczego węzła listy dwukierunkowej 21
include/NodeFactory.h
Deklaracja klasy NodeFactory – wzorzec fabryki do tworzenia węzłów listy 22
src/DwukierunkowaLista.cpp
Implementacja metod szablonu klasy DwukierunkowaLista 23
src/Iterator.cpp
Implementacja iteratora dla listy dwukierunkowej 23
src/main.cpp
Program testowy listy dwukierunkowej DoublyLinkedList 23
src/Node.cpp
Implementacja konstruktora szablonu Node 24
src/NodeFactory.cpp
Implementacja metod fabryki węzłów listy 24

Rozdział 4

Dokumentacja katalogów

4.1 Dokumentacja katalogu include

Pliki

- plik DwukierunkowaLista.h

  Deklaracja szablonowej klasy DwukierunkowaLista.
- plik Iterator.h

  Deklaracja szablonowego iteratora dla listy dwukierunkowej.
- plik Node.h

  Deklaracja szablonowej struktury Node – pojedynczego węzła listy dwukierunkowej.
- plik NodeFactory.h

  Deklaracja klasy NodeFactory – wzorzec fabryki do tworzenia węzłów listy.

4.2 Dokumentacja katalogu src

Pliki

- plik DwukierunkowaLista.cpp

  Implementacja metod szablonu klasy DwukierunkowaLista.
- plik Iterator.cpp

  Implementacja iteratora dla listy dwukierunkowej.
- plik main.cpp

  Program testowy listy dwukierunkowej DoublyLinkedList.
- plik Node.cpp

  Implementacja konstruktora szablonu Node.
- plik NodeFactory.cpp

  Implementacja metod fabryki wêz^{3}*ów listy.*

Rozdział 5

Dokumentacja klas

5.1 Dokumentacja szablonu klasy DoublyListIterator< T >

Szablonowy iterator klasy DoublyLinkedList.

`#include <Iterator.h>`

Typy publiczne

- using iterator_category = std::bidirectional_iterator_tag
- using value_type = T
- using reference = T&
- using pointer = T*

Metody publiczne

- DoublyListIterator (Node< T > *ptr)
- reference operator* () const
- pointer operator-> () const
- DoublyListIterator & operator++ ()
- DoublyListIterator operator++ (int)
- DoublyListIterator & operator-- ()
- DoublyListIterator operator-- (int)
- bool operator== (const DoublyListIterator &other) const
- bool operator!= (const DoublyListIterator &other) const

5.1.1 Opis szczegółowy

template<typename T>
class DoublyListIterator< T >

Szablonowy iterator klasy DoublyLinkedList.

Parametry Szablonu

T	Typ danych przechowywanych w liście.

5.1.2 Dokumentacja składowych definicji typu

5.1.2.1 iterator_category

```
template<typename T>
using DoublyListIterator< T >::iterator_category = std::bidirectional_iterator_tag
```

5.1.2.2 pointer

```
template<typename T>
using DoublyListIterator< T >::pointer = T*
```

5.1.2.3 reference

```
template<typename T>
using DoublyListIterator< T >::reference = T&
```

5.1.2.4 value_type

```
template<typename T>
using DoublyListIterator< T >::value_type = T
```

5.1.3 Dokumentacja konstruktora i destruktora

5.1.3.1 DoublyListIterator()

```
template<typename T>
DoublyListIterator< T >::DoublyListIterator (
            Node< T > * ptr)  [explicit]
```

5.1.4 Dokumentacja funkcji składowych

5.1.4.1 operator"!=()

```
template<typename T>
bool DoublyListIterator< T >::operator!= (
            const DoublyListIterator< T > & other) const
```

5.1.4.2 operator*()

```
template<typename T>
DoublyListIterator< T >::reference DoublyListIterator< T >::operator* () const
```

5.1.4.3 operator++() [1/2]

```
template<typename T>
DoublyListIterator< T > & DoublyListIterator< T >::operator++ ()
```

5.1.4.4 operator++() [2/2]

```
template<typename T>
DoublyListIterator< T > DoublyListIterator< T >::operator++ (
            int )
```

5.1.4.5 operator--() [1/2]

```
template<typename T>
DoublyListIterator< T > & DoublyListIterator< T >::operator-- ()
```

5.1.4.6 operator--() [2/2]

```
template<typename T>
DoublyListIterator< T > DoublyListIterator< T >::operator-- (
            int )
```

5.1.4.7 operator->()

```
template<typename T>
DoublyListIterator< T >::pointer DoublyListIterator< T >::operator-> () const
```

5.1.4.8 operator==()

```
template<typename T>
bool DoublyListIterator< T >::operator== (
            const DoublyListIterator< T > & other) const
```

Dokumentacja dla tej klasy została wygenerowana z plików:

- include/Iterator.h
- src/Iterator.cpp

5.2 Dokumentacja szablonu klasy DwukierunkowaLista< T >

Szablonowa implementacja listy dwukierunkowej.

```
#include <DwukierunkowaLista.h>
```

Metody publiczne

- DwukierunkowaLista ()
- ~DwukierunkowaLista ()
- void push_front (const T &value)

  Dodaje element na początek listy.
- void push_back (const T &value)
- void pop_front ()
- void pop_back ()
- void clear ()
- bool empty () const
- size_t size () const
- void display_forward () const
- void display_reverse () const
- void insertAt (size_t index, const T &value)
- void removeAt (size_t index)
- T & getAt (size_t index)
- DoublyListIterator< T > begin ()
- DoublyListIterator< T > end ()
- Node< T > * getHead () const
- Node< T > * getTail () const

5.2.1 Opis szczegółowy

template<typename T>
class DwukierunkowaLista< T >

Szablonowa implementacja listy dwukierunkowej.

Parametry Szablonu

T	Typ danych przechowywanych w liście.

5.2.2 Dokumentacja konstruktora i destruktora

5.2.2.1 DwukierunkowaLista()

```
template<typename T>
DwukierunkowaLista< T >::DwukierunkowaLista ()
```

5.2.2.2 ~DwukierunkowaLista()

```
template<typename T>
DwukierunkowaLista< T >::~DwukierunkowaLista ()
```

5.2.3 Dokumentacja funkcji składowych

5.2.3.1 begin()

```
template<typename T>
DoublyListIterator< T > DwukierunkowaLista< T >::begin ()
```

5.2.3.2 clear()

```
template<typename T>
void DwukierunkowaLista< T >::clear ()
```

5.2.3.3 display_forward()

```
template<typename T>
void DwukierunkowaLista< T >::display_forward () const
```

5.2.3.4 display_reverse()

```
template<typename T>
void DwukierunkowaLista< T >::display_reverse () const
```

5.2.3.5 empty()

```
template<typename T>
bool DwukierunkowaLista< T >::empty () const
```

5.2.3.6 end()

```
template<typename T>
DoublyListIterator< T > DwukierunkowaLista< T >::end ()
```

5.2.3.7 getAt()

```
template<typename T>
T & DwukierunkowaLista< T >::getAt (
            size_t index)
```

5.2.3.8 getHead()

```
template<typename T>
Node< T > * DwukierunkowaLista< T >::getHead () const
```

5.2.3.9 getTail()

```
template<typename T>
Node< T > * DwukierunkowaLista< T >::getTail () const
```

5.2.3.10 insertAt()

```
template<typename T>
void DwukierunkowaLista< T >::insertAt (
            size_t index,
            const T & value)
```

5.2.3.11 pop_back()

```
template<typename T>
void DwukierunkowaLista< T >::pop_back ()
```

5.2.3.12 pop_front()

```
template<typename T>
void DwukierunkowaLista< T >::pop_front ()
```

5.2.3.13 push_back()

```
template<typename T>
void DwukierunkowaLista< T >::push_back (
            const T & value)
```

5.2.3.14 push_front()

```
template<typename T>
void DwukierunkowaLista< T >::push_front (
            const T & value)
```

Dodaje element na początek listy.

Parametry

value	Wartość do dodania.

5.2.3.15 removeAt()

```
template<typename T>
void DwukierunkowaLista< T >::removeAt (
            size_t index)
```

5.2.3.16 size()

```
template<typename T>
size_t DwukierunkowaLista< T >::size () const
```

Dokumentacja dla tej klasy została wygenerowana z plików:

- include/DwukierunkowaLista.h
- src/DwukierunkowaLista.cpp

5.3 Dokumentacja szablonu struktury Node< T >

Szablonowa struktura węzła listy dwukierunkowej.

```
#include <Node.h>
```

Metody publiczne

- Node (const T &value)

  Konstruktor inicjalizujący dane w węźle.

Atrybuty publiczne

- T data
- Node< T > * prev
- Node< T > * next

5.3.1 Opis szczegółowy

template<typename T>
struct Node< T >

Szablonowa struktura węzła listy dwukierunkowej.

Parametry Szablonu

T	Typ danych przechowywany w węźle.

5.3.2 Dokumentacja konstruktora i destruktora

5.3.2.1 Node()

```
template<typename T>
Node< T >::Node (
            const T & value)  [explicit]
```

Konstruktor inicjalizujący dane w węźle.

Parametry

value	Wartość przechowywana w węźle.

5.3.3 Dokumentacja atrybutów składowych

5.3.3.1 data

```
template<typename T>
T Node< T >::data
```

5.3.3.2 next

```
template<typename T>
Node<T>* Node< T >::next
```

5.3.3.3 prev

```
template<typename T>
Node<T>* Node< T >::prev
```

Dokumentacja dla tej struktury została wygenerowana z plików:

- include/Node.h
- src/Node.cpp

5.4 Dokumentacja szablonu klasy NodeFactory< T >

Szablonowa klasa fabryczna tworząca i usuwająca węzły listy.

```
#include <NodeFactory.h>
```

Statyczne metody publiczne

- static Node< T > * createNode (const T &value)

  Tworzy nowy węzeł z podaną wartością.
- static void destroyNode (Node< T > *node)

5.4.1 Opis szczegółowy

template<typename T>
class NodeFactory< T >

Szablonowa klasa fabryczna tworząca i usuwająca węzły listy.

Parametry Szablonu

T	Typ danych przechowywany w węźle.

5.4.2 Dokumentacja funkcji składowych

5.4.2.1 createNode()

```
template<typename T>
Node< T > * NodeFactory< T >::createNode (
            const T & value)  [static]
```

Tworzy nowy węzeł z podaną wartością.

Parametry

value	Wartość do zapisania w węźle.

Zwraca

Wskaźnik na nowy węzeł.

5.4.2.2 destroyNode()

```
template<typename T>
void NodeFactory< T >::destroyNode (
            Node< T > * node)  [static]
```

Dokumentacja dla tej klasy została wygenerowana z plików:

- include/NodeFactory.h
- src/NodeFactory.cpp

Rozdział 6

Dokumentacja plików

6.1 Dokumentacja pliku include/DwukierunkowaLista.h

Deklaracja szablonowej klasy DwukierunkowaLista.

```
#include "NodeFactory.h"
#include <iostream>
#include <stdexcept>
#include "Iterator.h"
```

Komponenty

- class DwukierunkowaLista< T >

  Szablonowa implementacja listy dwukierunkowej.

6.1.1 Opis szczegółowy

Deklaracja szablonowej klasy DwukierunkowaLista.

Data

2025-10-25

6.2 DwukierunkowaLista.h

Idź do dokumentacji tego pliku.

```
00001
00007
00008 #ifndef DWUKIERUNKOWA_LISTA_H
00009 #define  DWUKIERUNKOWA_LISTA_H
00010
00011 #include "NodeFactory.h"
00012 #include <iostream>
00013 #include <stdexcept>
00014 #include "Iterator.h"
00015
00020
```

```
00021 template<typename T>
00022 class DwukierunkowaLista {
00023 public:
00024     DwukierunkowaLista();
00025     ~DwukierunkowaLista();
00026
00027     void push_front(const T& value);
00028     void push_back(const T& value);
00029     void pop_front();
00030     void pop_back();
00031     void clear();
00032     bool empty() const;
00033     size_t size() const;
00034
00035     void display_forward() const;
00036     void display_reverse() const;
00037
00038     void insertAt(size_t index, const T& value);
00039     void removeAt(size_t index);
00040     T& getAt(size_t index);
00041
00042     DoublyListIterator<T> begin();
00043     DoublyListIterator<T> end();
00044     // dostęp do surowych wskaźników (potrzebne do testów / iteracji wstecz)
00045     Node<T>* getHead() const;
00046     Node<T>* getTail() const;
00047
00048 private:
00049     Node<T>* head;
00050     Node<T>* tail;
00051
00052     size_t sz;
00053 };
00054
00055 #endif
```

6.3 Dokumentacja pliku include/Iterator.h

Deklaracja szablonowego iteratora dla listy dwukierunkowej.

```
#include "Node.h"
#include <iterator>
```

Komponenty

- class DoublyListIterator< T >

  Szablonowy iterator klasy DoublyLinkedList.

6.3.1 Opis szczegółowy

Deklaracja szablonowego iteratora dla listy dwukierunkowej.

Data

2025-10-25

6.4 Iterator.h

Idź do dokumentacji tego pliku.

```
00001
00006
00007 #ifndef ITERATOR_H
00008 #define ITERATOR_H
00009
00010 #include "Node.h"
00011 #include <iterator>
00012
00017
00018 template<typename T>
00019 class DoublyListIterator {
00020 public:
00021     using iterator_category = std::bidirectional_iterator_tag;
00022     using value_type = T;
00023     using reference = T&;
00024     using pointer = T*;
00025
00026     explicit DoublyListIterator(Node<T>* ptr);
00027     reference operator*() const;
00028     pointer operator->() const;
00029
00030     DoublyListIterator& operator++();  // ++it
00031     DoublyListIterator operator++(int); // it++
00032     DoublyListIterator& operator--();  // --it
00033     DoublyListIterator operator--(int); // it--
00034
00035     bool operator==(const DoublyListIterator& other) const;
00036     bool operator!=(const DoublyListIterator& other) const;
00037
00038 private:
00039     Node<T>* current;
00040 };
00041
00042 #endif // ITERATOR_H
```

6.5 Dokumentacja pliku include/Node.h

Deklaracja szablonowej struktury Node – pojedynczego węzła listy dwukierunkowej.

Komponenty

- struct Node< T >

  Szablonowa struktura węzła listy dwukierunkowej.

6.5.1 Opis szczegółowy

Deklaracja szablonowej struktury Node – pojedynczego węzła listy dwukierunkowej.

Data

2025-10-25

Autor

Szymon Kraj

6.6 Node.h

Idź do dokumentacji tego pliku.

```
00001
00007
00008 #ifndef NODE_H
00009 #define NODE_H
00010
00015 template<typename T>
00016 struct Node {
00017     T data;
00018     Node<T>* prev;
00019     Node<T>* next;
00020
00025
00026     explicit Node(const T& value); // deklaracja konstruktora
00027 };
00028
00029 #endif // NODE_H
00030
```

6.7 Dokumentacja pliku include/NodeFactory.h

Deklaracja klasy NodeFactory – wzorzec fabryki do tworzenia węzłów listy.

```
#include "Node.h"
```

Komponenty

- class NodeFactory< T >

  Szablonowa klasa fabryczna tworząca i usuwająca węzły listy.

6.7.1 Opis szczegółowy

Deklaracja klasy NodeFactory – wzorzec fabryki do tworzenia węzłów listy.

Data

2025-10-25

6.8 NodeFactory.h

Idź do dokumentacji tego pliku.

```
00001
00006
00007 #ifndef NODE_FACTORY_H
00008 #define NODE_FACTORY_H
00009
00010 #include "Node.h"
00015 template<typename T>
00016 class NodeFactory {
00017 public:
00023
00024     static Node<T>* createNode(const T& value);
00025     static void destroyNode(Node<T>* node);
00030 };
00031
00032 #endif // NODE_FACTORY_H
00033
```

6.9 Dokumentacja pliku src/DwukierunkowaLista.cpp

Implementacja metod szablonu klasy DwukierunkowaLista.

```
#include "DwukierunkowaLista.h"
#include <string>
```

6.9.1 Opis szczegółowy

Implementacja metod szablonu klasy DwukierunkowaLista.

Data

2025-10-25

6.10 Dokumentacja pliku src/Iterator.cpp

Implementacja iteratora dla listy dwukierunkowej.

```
#include "Iterator.h"
#include <string>
```

6.10.1 Opis szczegółowy

Implementacja iteratora dla listy dwukierunkowej.

Data

2025-10-25

6.11 Dokumentacja pliku src/main.cpp

Program testowy listy dwukierunkowej DoublyLinkedList.

```
#include <iostream>
#include "DwukierunkowaLista.h"
#include <string>
```

Funkcje

- int main ()

6.11.1 Opis szczegółowy

Program testowy listy dwukierunkowej DoublyLinkedList.

Data

2025-10-25

6.11.2 Dokumentacja funkcji

6.11.2.1 main()

```
int main ()
```

6.12 Dokumentacja pliku src/Node.cpp

Implementacja konstruktora szablonu Node.

```
#include "Node.h"
#include <string>
```

6.12.1 Opis szczegółowy

Implementacja konstruktora szablonu Node.

Data

2025-10-25

6.13 Dokumentacja pliku src/NodeFactory.cpp

Implementacja metod fabryki węzłów listy.

```
#include "NodeFactory.h"
#include <string>
```

6.13.1 Opis szczegółowy

Implementacja metod fabryki węzłów listy.

Data

2025-10-25

Rozdział 7

Przykłady

7.1 main.cpp

Ten przyk^{3}ad demonstruje u¿ycie listy dwukierunkowej: dodawanie, usuwanie, iterowanie i czyszczenie elementów.

Skorowidz

~DwukierunkowaLista
  DwukierunkowaLista< T >, 12

begin
  DwukierunkowaLista< T >, 13

clear
  DwukierunkowaLista< T >, 13
createNode
  NodeFactory< T >, 17

data
  Node< T >, 16
destroyNode
  NodeFactory< T >, 17
display_forward
  DwukierunkowaLista< T >, 13
display_reverse
  DwukierunkowaLista< T >, 13
Dokumentacja katalogu include, 7
Dokumentacja katalogu src, 7
DoublyListIterator
  DoublyListIterator< T >, 10
DoublyListIterator< T >, 9
  DoublyListIterator, 10
  iterator_category, 10
  operator!=, 10
  operator++, 10, 11
  operator->, 11
  operator--, 11
  operator==, 11
  operator*, 10
  pointer, 10
  reference, 10
  value_type, 10
DwukierunkowaLista
  DwukierunkowaLista< T >, 12
DwukierunkowaLista< T >, 11
  ~DwukierunkowaLista, 12
  begin, 13
  clear, 13
  display_forward, 13
  display_reverse, 13
  DwukierunkowaLista, 12
  empty, 13
  end, 13
  getAt, 13
  getHead, 13
  getTail, 13
  insertAt, 14
  pop_back, 14
  pop_front, 14
  push_back, 14
  push_front, 14
  removeAt, 14
  size, 14

empty
  DwukierunkowaLista< T >, 13
end
  DwukierunkowaLista< T >, 13

getAt
  DwukierunkowaLista< T >, 13
getHead
  DwukierunkowaLista< T >, 13
getTail
  DwukierunkowaLista< T >, 13

include/DwukierunkowaLista.h, 19
include/Iterator.h, 20, 21
include/Node.h, 21, 22
include/NodeFactory.h, 22
insertAt
  DwukierunkowaLista< T >, 14
iterator_category
  DoublyListIterator< T >, 10

main
  main.cpp, 24
main.cpp
  main, 24

next
  Node< T >, 16
Node
  Node< T >, 15
Node< T >, 15
  data, 16
  next, 16
  Node, 15
  prev, 16
NodeFactory< T >, 16
  createNode, 17
  destroyNode, 17

operator!=
  DoublyListIterator< T >, 10
operator++
  DoublyListIterator< T >, 10, 11
operator->

DoublyListIterator< T >, 11
operator--
DoublyListIterator< T >, 11
operator==
DoublyListIterator< T >, 11
operator*
DoublyListIterator< T >, 10

pointer
DoublyListIterator< T >, 10
pop_back
DwukierunkowaLista< T >, 14
pop_front
DwukierunkowaLista< T >, 14
prev
Node< T >, 16
push_back
DwukierunkowaLista< T >, 14
push_front
DwukierunkowaLista< T >, 14

reference
DoublyListIterator< T >, 10
removeAt
DwukierunkowaLista< T >, 14

size
DwukierunkowaLista< T >, 14
src/DwukierunkowaLista.cpp, 23
src/Iterator.cpp, 23
src/main.cpp, 23
src/Node.cpp, 24
src/NodeFactory.cpp, 24

value_type
DoublyListIterator< T >, 10