

Lista jednokierunkowa

Generated by Doxygen 1.8.20

1 Class Index	1
1.1 Class List	1
2 Class Documentation	3
2.1 Czlowiek Class Reference	3
2.1.1 Detailed Description	3
2.1.2 Member Function Documentation	3
2.1.2.1 display()	3
2.1.2.2 wczytaj_z_pliku()	4
2.2 List_element< T > Class Template Reference	4
2.2.1 Detailed Description	4
2.2.2 Member Function Documentation	4
2.2.2.1 get_next()	5
2.2.2.2 get_value()	5
2.2.2.3 set_next()	5
2.2.2.4 set_value()	5
2.3 My_forward_list< T > Class Template Reference	6
2.3.1 Detailed Description	6
2.3.2 Member Function Documentation	6
2.3.2.1 clear()	7
2.3.2.2 erase()	7
2.3.2.3 find()	7
2.3.2.4 is_empty()	7
2.3.2.5 pop_back()	8
2.3.2.6 pop_front()	8
2.3.2.7 push_back()	8
2.3.2.8 push_front()	8
2.3.2.9 search_index()	8
2.3.2.10 show()	9
2.3.2.11 size()	9
2.3.2.12 sort()	9
2.3.2.13 wczytaj_baze()	10
2.3.2.14 wczytaj_binarnie()	10
2.3.2.15 zapisz_binarnie()	10
2.4 Samochod Class Reference	11
2.4.1 Detailed Description	11
2.4.2 Member Function Documentation	11
2.4.2.1 display()	11
2.4.2.2 wczytaj_z_pliku()	12
Index	13

Chapter 1

Class Index

1.1 Class List

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

Czlowiek	3
List_element< T >	4
My_forward_list< T >	6
Samochod	11

Chapter 2

Class Documentation

2.1 Czlowiek Class Reference

```
#include <Czlowiek.h>
```

Public Member Functions

- **Czlowiek** (string _imie, string _nazwisko, string _PESEL, int _wiek)
- void [display](#) (ostream &output=cout)
- void [wczytaj_z_pliku](#) (ifstream &input)

Public Attributes

- string **imie**
- string **nazwisko**
- string **PESEL**
- int **wiek**

2.1.1 Detailed Description

Klasa reprezentująca człowieka.

2.1.2 Member Function Documentation

2.1.2.1 display()

```
void Czlowiek::display (
    ostream & output = cout )
```

Metoda wyświetlająca atrybuty człowieka.

Parameters

<i>output</i>	Strumień wyjściowy
---------------	--------------------

2.1.2.2 wczytaj_z_pliku()

```
void Czlowiek::wczytaj_z_pliku (
    ifstream & input )
```

Metoda wczytująca atrybuty człowieka z podanego strumienia wejściowego.

Parameters

<i>Strumień</i>	wejściowy
-----------------	-----------

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

- Czlowiek.h

2.2 List_element< T > Class Template Reference

```
#include <Lista.h>
```

Public Member Functions

- **List_element** (T x=0)
- T [get_value](#) ()
- void [set_value](#) (T x)
- shared_ptr< [List_element](#)< T > > [get_next](#) ()
- void [set_next](#) (shared_ptr< [List_element](#)< T > > new_next)

2.2.1 Detailed Description

```
template<class T>
class List_element< T >
```

Szablon klasy reprezentującej element listy.

2.2.2 Member Function Documentation

2.2.2.1 get_next()

```
template<class T >
shared_ptr< List_element< T > > List_element< T >::get_next
```

Metoda zwracająca wskaźnik na następny element.

Returns

Wskaźnik na następny element

2.2.2.2 get_value()

```
template<class T >
T List_element< T >::get_value
```

Metoda zwracająca wartość elementu.

Returns

Wartość elementu

2.2.2.3 set_next()

```
template<class T >
void List_element< T >::set_next (
    shared_ptr< List_element< T > > new_next )
```

Metoda ustawiająca wskaźnik na następny element.

Parameters

<i>new_next</i>	Nowy wskaźnik na następny element
-----------------	-----------------------------------

2.2.2.4 set_value()

```
template<class T >
void List_element< T >::set_value (
    T x )
```

Metoda ustawiająca wartość elementu.

Parameters

<i>x</i>	Nowa wartość
----------	--------------

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

- Lista.h

2.3 My_forward_list< T > Class Template Reference

```
#include <Lista.h>
```

Public Member Functions

- **My_forward_list** (const [My_forward_list](#) &to_copy)
- **My_forward_list** (const [My_forward_list](#) &&to_move)
- [My_forward_list](#) & **operator=** (const [My_forward_list](#) &lista)
- [My_forward_list](#) & **operator=** ([My_forward_list](#) &&lista)
- void [push_front](#) (T x)
- void [push_back](#) (T x)
- void [pop_front](#) ()
- void [pop_back](#) ()
- void [show](#) (ostream &output=cout)
- int [size](#) ()
- void [erase](#) (int pos)
- shared_ptr< [List_element](#)< T > > [find](#) (int pos)
- int [search_index](#) (shared_ptr< [List_element](#)< T > > elem)
- bool [is_empty](#) ()
- void [sort](#) (bool choice, int criteria=1)
- void [wczytaj_binarnie](#) (string name)
- void [zapisz_binarnie](#) (string name)
- void [wczytaj_baze](#) (string name)
- void [clear](#) ()

2.3.1 Detailed Description

```
template<class T>
class My_forward_list< T >
```

Szablon klasy reprezentującej listę jednokierunkową.

2.3.2 Member Function Documentation

2.3.2.1 clear()

```
template<class T >
void My_forward_list< T >::clear
```

Metoda czyszcząca listę.

2.3.2.2 erase()

```
template<class T >
void My_forward_list< T >::erase (
    int pos )
```

Metoda usuwająca element z danej pozycji. Generuje stosowny komunikat, jeśli element jest poza zakresem.

Parameters

<i>pos</i>	Indeks elementu do usunięcia
------------	------------------------------

2.3.2.3 find()

```
template<class T >
shared_ptr< List_element< T > > My_forward_list< T >::find (
    int pos )
```

Metoda znajdująca element na podanej pozycji.

Parameters

<i>pos</i>	Indeks elementu do znalezienia
------------	--------------------------------

Returns

Wskaźnik na szukany element bądź nullptr

2.3.2.4 is_empty()

```
template<class T >
bool My_forward_list< T >::is_empty
```

Metoda zwracająca wartość TRUE, jeśli lista jest pusta.

Returns

Czy lista jest pusta?

2.3.2.5 pop_back()

```
template<class T >
void My_forward_list< T >::pop_back
```

Metoda usuwająca element z końca listy.

2.3.2.6 pop_front()

```
template<class T >
void My_forward_list< T >::pop_front
```

Metoda usuwająca element z początku listy.

2.3.2.7 push_back()

```
template<class T >
void My_forward_list< T >::push_back (
    T x )
```

Metoda dodająca element na koniec listy.

Parameters

x	Dodawany element
---	------------------

2.3.2.8 push_front()

```
template<class T >
void My_forward_list< T >::push_front (
    T x )
```

Metoda dodająca element na początek listy.

Parameters

x	Dodawany element
---	------------------

2.3.2.9 search_index()

```
template<class T >
int My_forward_list< T >::search_index (
    shared_ptr< List_element< T >> elem )
```

Metoda znajdująca indeks podanego elementu.

Parameters

<i>elem</i>	Wskaźnik na szukany element
-------------	-----------------------------

Returns

Indeks szukanego elementu

2.3.2.10 `show()`

```
template<class T >
void My_forward_list< T >::show (
    ostream & output = cout )
```

Metoda wyświetlająca zawartość listy.

Parameters

<i>output</i>	Strumień wyjściowy
---------------	--------------------

2.3.2.11 `size()`

```
template<class T >
int My_forward_list< T >::size
```

Metoda zwracająca rozmiar listy.

Returns

Rozmiar listy

2.3.2.12 `sort()`

```
template<class T >
void My_forward_list< T >::sort (
    bool choice,
    int criteria = 1 )
```

Metoda sortująca listę według podanego kryterium.

Parameters

<i>choice</i>	Jeśli 0, sortuje rosnąco, w przeciwnym wypadku malejąco
<i>criteria</i>	Kryterium sortowania (dla klasy Człowiek i Samochod)

2.3.2.13 wczytaj_baze()

```
template<class T >
void My_forward_list< T >::wczytaj_baze (
    string name )
```

Metoda wczytująca zawartość listy z podanego pliku.

Parameters

<i>Nazwa</i>	pliku
--------------	-------

2.3.2.14 wczytaj_binarnie()

```
template<class T >
void My_forward_list< T >::wczytaj_binarnie (
    string name )
```

Metoda wczytująca zawartość listy z podanego pliku binarnego.

Parameters

<i>name</i>	Nazwa pliku binarnego
-------------	-----------------------

2.3.2.15 zapisz_binarnie()

```
template<class T >
void My_forward_list< T >::zapisz_binarnie (
    string name )
```

Metoda zapisująca zawartość listy do pliku binarnego.

Parameters

<i>name</i>	Nazwa pliku binarnego
-------------	-----------------------

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

- Lista.h

2.4 Samochod Class Reference

```
#include <Samochod.h>
```

Public Member Functions

- **Samochod** (string _marka, string _model, string _VIN, int _przebieg)
- void [display](#) (ostream &output=cout)
- void [wczytaj_z_pliku](#) (ifstream &input)

Public Attributes

- string **marka**
- string **model**
- string **VIN**
- int **przebieg**

2.4.1 Detailed Description

Klasa reprezentująca samochód.

2.4.2 Member Function Documentation

2.4.2.1 display()

```
void Samochod::display (
    ostream & output = cout )
```

Metoda wyświetlająca atrybuty samochodu.

Parameters

<i>output</i>	Strumień wyjściowy
---------------	--------------------

2.4.2.2 wczytaj_z_pliku()

```
void Samochod::wczytaj_z_pliku (
    ifstream & input )
```

Metoda wczytująca atrybuty samochodu z podanego strumienia wejściowego.

Parameters

<i>Strumień</i>	wejściowy
-----------------	-----------

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

- Samochod.h

Index

clear
 My_forward_list< T >, 6
Czlowiek, 3
 display, 3
 wczytaj_z_pliku, 4

display
 Czlowiek, 3
 Samochod, 11

erase
 My_forward_list< T >, 7

find
 My_forward_list< T >, 7

get_next
 List_element< T >, 4

get_value
 List_element< T >, 5

is_empty
 My_forward_list< T >, 7

List_element< T >, 4
 get_next, 4
 get_value, 5
 set_next, 5
 set_value, 5

My_forward_list< T >, 6
 clear, 6
 erase, 7
 find, 7
 is_empty, 7
 pop_back, 7
 pop_front, 8
 push_back, 8
 push_front, 8
 search_index, 8
 show, 9
 size, 9
 sort, 9
 wczytaj_baze, 10
 wczytaj_binarnie, 10
 zapisz_binarnie, 10

pop_back
 My_forward_list< T >, 7

pop_front
 My_forward_list< T >, 8

push_back
 My_forward_list< T >, 8
push_front
 My_forward_list< T >, 8

Samochod, 11
 display, 11
 wczytaj_z_pliku, 11

search_index
 My_forward_list< T >, 8

set_next
 List_element< T >, 5

set_value
 List_element< T >, 5

show
 My_forward_list< T >, 9

size
 My_forward_list< T >, 9

sort
 My_forward_list< T >, 9

wczytaj_baze
 My_forward_list< T >, 10

wczytaj_binarnie
 My_forward_list< T >, 10

wczytaj_z_pliku
 Czlowiek, 4
 Samochod, 11

zapisz_binarnie
 My_forward_list< T >, 10