# Listy

[std::list - cppreference.com](https://en.cppreference.com/w/cpp/container/list)

Lista jest dynamiczną strukturą danych, która przechowuje elementy w kolejności wstawienia. Każdy element w liście jest połączony z poprzednim i następnym elementem za pomocą wskaźników.

Funkcje dostępne w bibliotece <list>:

1. push\_back(): Dodaje nowy element na końcu listy.

2. push\_front(): Dodaje nowy element na początku listy.

3. pop\_back(): Usuwa ostatni element z listy.

4. pop\_front(): Usuwa pierwszy element z listy.

5. insert(): Wstawia nowy element w określone miejsce listy.

6. erase(): Usuwa element z określonego miejsca listy.

7. clear(): Usuwa wszystkie elementy z listy.

8. size(): Zwraca liczbę elementów w liście.

9. empty(): Sprawdza, czy lista jest pusta.

10. front(): Zwraca pierwszy element listy.

11. back(): Zwraca ostatni element listy.

Inicjalizacja listy w C++:

#include <iostream>

#include <list>

using namespace std;

int main() {

list<int> myList; // Inicjalizacja pustej listy

myList.push\_back(5); // Dodanie elementu na końcu listy

myList.push\_front(10); // Dodanie elementu na początku listy

// Inne operacje na liście...

return 0;

}

**Zadania**

1. Zdefiniuj klasę Osoba:

- Utwórz plik nagłówkowy osoba.h.

- Zdefiniuj klasę Osoba zawierającą prywatne pole nazwisko typu tablicowego.

- Dodaj konstruktor, który umożliwia inicjalizację obiektu nazwiskiem lub pozostawia pole puste.

- Zaimplementuj funkcje zaprzyjaźnione do przeciążenia operatorów << dla obiektów Osoba i wskaźników na obiekty Osoba.

2. Zdefiniuj klasę Lista:

- Utwórz plik nagłówkowy lista.h.

- Zdefiniuj klasę Lista zawierającą prywatne pole elementy typu std::list przechowujące obiekty Osoba.

- Dodaj konstruktor klasy Lista.

- Zaimplementuj metody do dodawania, usuwania i wyświetlania elementów listy.

3. Stwórz plik źródłowy main.cpp:

- W tym pliku, zaimportuj pliki nagłówkowe osoba.h i lista.h.

- Utwórz obiekty klasy Osoba reprezentujące różnych muzyków, fizyków itp.

- Utwórz obiekt klasy Lista i dodaj do niego utworzone obiekty Osoba.

- Wykonaj operacje na obiekcie klasy Lista, takie jak dodawanie, usuwanie i wyświetlanie zawartości.

4. Skompiluj i uruchom program:

- Skorzystaj z kompilatora C++, aby skompilować plik źródłowy main.cpp wraz z plikami nagłówkowymi.

- Uruchom skompilowany program i sprawdź, czy działa zgodnie z oczekiwaniami.

5. Testuj różne przypadki:

- Przetestuj działanie programu dla różnych przypadków, takich jak dodawanie, usuwanie i wyświetlanie obiektów Osoba.

- Upewnij się, że program obsługuje poprawnie operacje na liście, takie jak dodawanie, usuwanie i wyświetlanie zawartości.