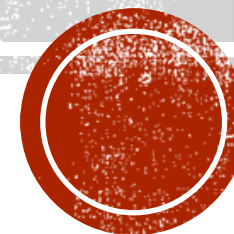


BIBLIOTEKA STL



Kamil Legierski

Wykorzystano materiały:

Vardan Grigoryan, Shunguang Wu, Expert C++. Packt, 2020.

<https://www.udemy.com/course/programowanie-objektowe-w-jezyku-cpp-oop-od-a-do-z/>

AGENDA

- Biblioteka STL - co to ?
- Rodzaje kontenerów
- Iteratory
- Algorytmy
- Vector
- List
- Deque
- Array
- Makra

BIBLIOTEKA STL - CO TO?

- Standard Template Library
- Zestaw szablonów klas, która udostępnia:
 - kontenery,
 - iteratory,
 - algorytmy (ściśle związane z kontenerami)

BIBLIOTEKA STL - RODZAJE KONTENERÓW

Kontenery sekwencyjne:

- vector
- list
- deque
- array
- forward_list

Adaptery kontenerów:

- queue
- priority_queue
- stack

BIBLIOTEKA STL - RODZAJE KONTENERÓW

Kontenery asocjacyjne:

- set
 - multiset
 - map
 - multimap
-
- posortowane struktury danych
 - wydajne pod względem przeszukiwania wartości

Nieuporządkowane kontenery asocjacyjne:

- unordered_set
- unordered_multiset
- unordered_map
- unordered_multimap

BIBLIOTEKA STL - ITERATORY

- Obiekt, który umożliwia:
 - dostęp sekwencyjny do elementów kontenerów
 - poruszanie się po elementach kontenerów

BIBLIOTEKA STL - ALGORYTMY

- Zbiór funkcji wykonujących różne operacje np. sortowania, wyszukiwania, zliczania itd.
- Przykładowe funkcje udostępnione przez bibliotekę algorithm
 - find()
 - count()
 - reverse()
 - sort()

BIBLIOTEKA STL - <VECTOR>

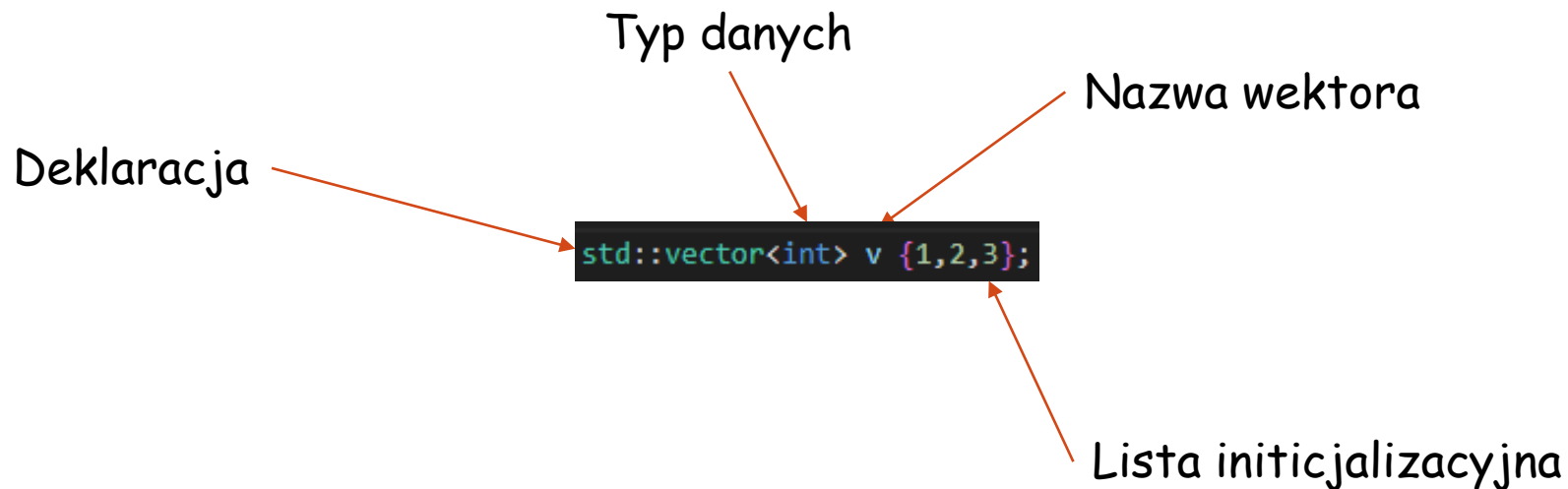
- Kontener
- Przechowuje wartości tego samego typu
- Podobny do tablicy
- Rozmiar może zmieniać się dynamicznie

Deklaracja Typ danych Nazwa wektora

```
std::vector<int> v;
```


BIBLIOTEKA STL - <VECTOR>

- Deklaracja z inicjalizacją



BIBLIOTEKA STL - <VECTOR>

- Najczęściej wykorzystywane metody:
 - Size() - zwraca liczbę elementów wektora
 - End() - wskazuje na pierwsze miejsce po ostatnim elemencie w wektorze
 - Begin() - wskazuje na pierwszy element w wektorze
 - [n] - wskazuje na n ty element w wektorze (pamiętaj, że iterujemy od 0)
 - Assign(n,m) - uzupełnia na nowo wektor n elementami o wartości m
 - Pop_back() - usuwa ostatni element wektora
 - Push_back() - dodaje nową wartość na koniec wektora
 - Clear() - usuwa wszystkie wartości wektora

BIBLIOTEKA STL - <LIST>

- Kontener sekwencyjny
- Nieciągła alokacja pamięci
- Poruszanie **wolniejsze** od poruszania po wektorze
- Wstawianie **szybsze** od wstawiania w wektorze
- Usuwanie elementów **szybsze** od usuwania w wektorze

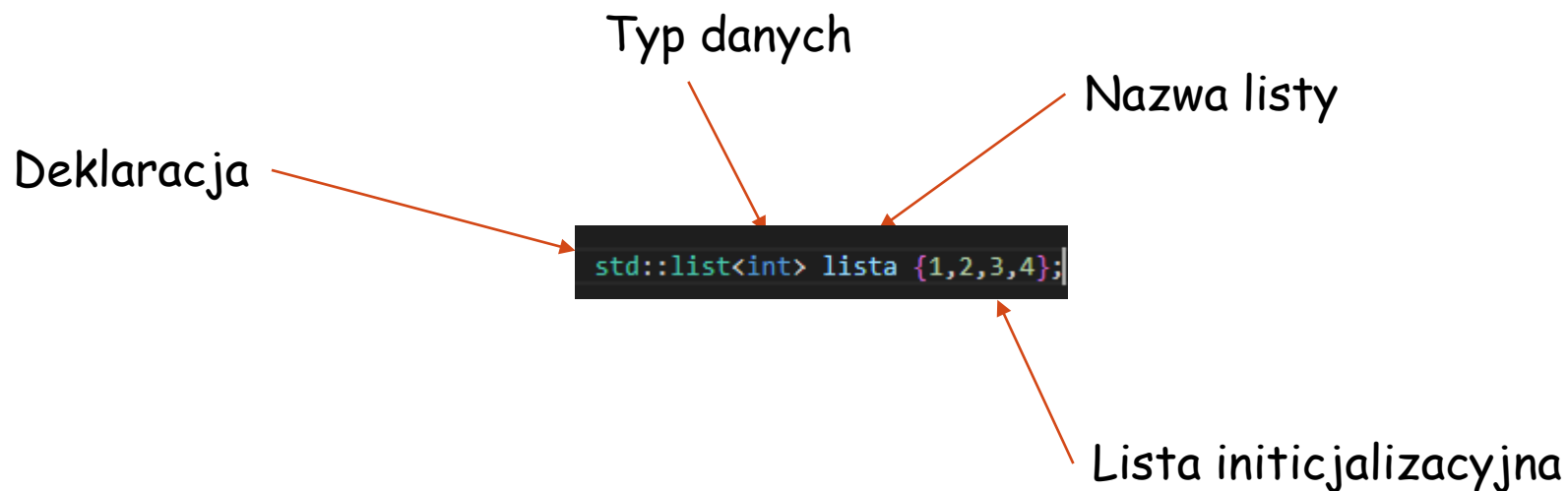
BIBLIOTEKA STL - <LIST>

- Kontener
- Przechowuje wartości tego samego typu
- Podobny do tablicy
- Rozmiar może zmieniać się dynamicznie



BIBLIOTEKA STL - <LIST>

- Deklaracja z inicjalizacją



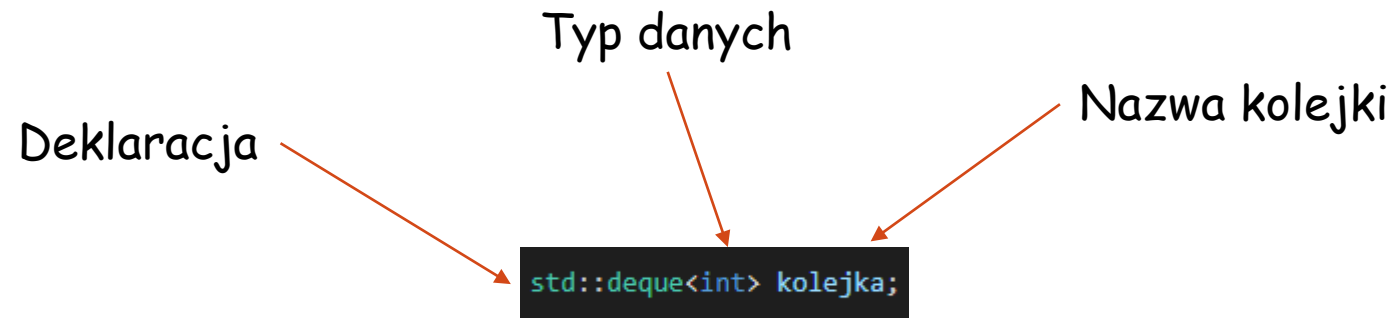
BIBLIOTEKA STL - <LIST>

- Najczęściej wykorzystywane metody:
 - `Push_front(value)` - dodaje element `value` na początek listy
 - `Push_back(value)` - dodaje element `value` na końcu listy
 - `Pop_back()` - usuwa ostatni element na liście
 - `Pop_front()` - usuwa pierwszy element na liście
 - `Remove(value)` - usuwa wszystkie wystąpienia `value` w liście
 - `Insert(pos, value)` - dodaje `value` na pozycji `pos`

BIBLIOTEKA STL - <DEQUE>

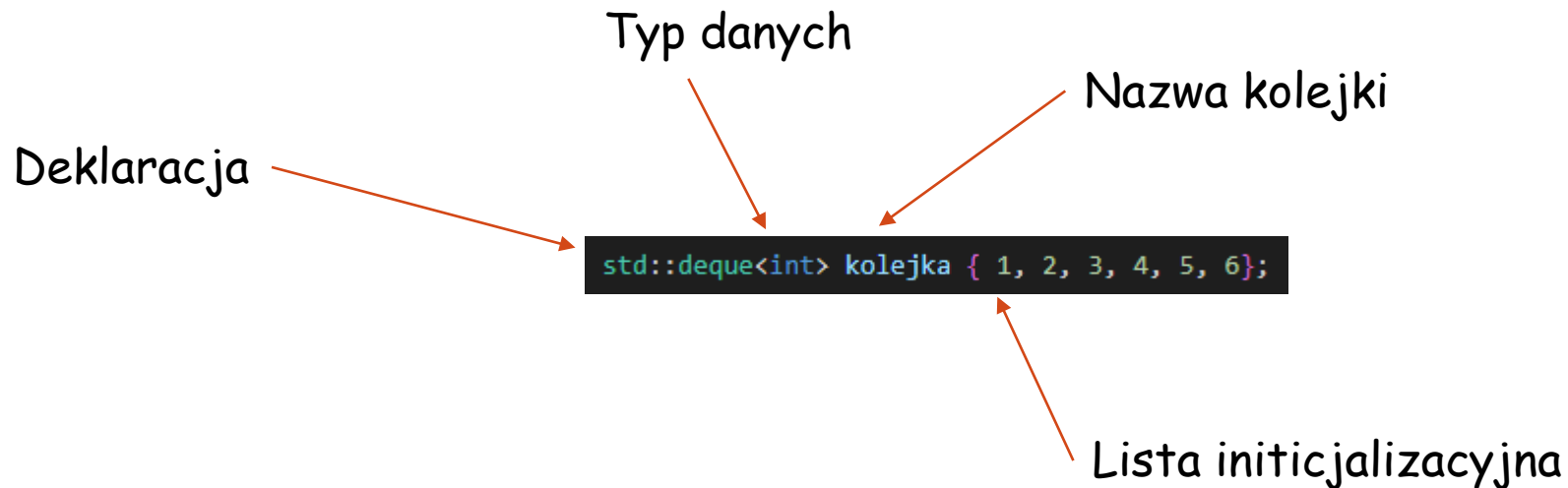
- Kontener sekwencyjny
- Kolejka dwustronna
- Nie gwarantuje ciągłej alokacji w pamięci

BIBLIOTEKA STL - <DEQUE>



BIBLIOTEKA STL - <DEQUE>

- Deklaracja z inicjalizacją



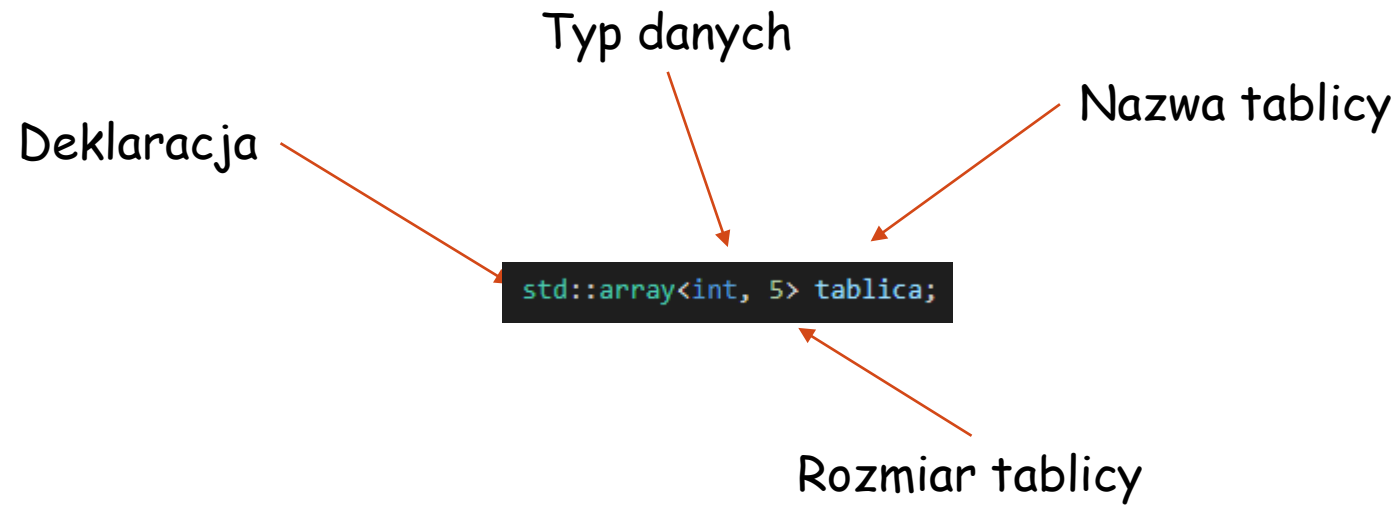
BIBLIOTEKA STL - <DEQUE>

- Najczęściej wykorzystywane metody:
 - `Push_back(value)` - dodaje element `value` na końcu kolejki
 - `Pop_front()` - usuwa pierwszy element kolejki

BIBLIOTEKA STL - <ARRAY>

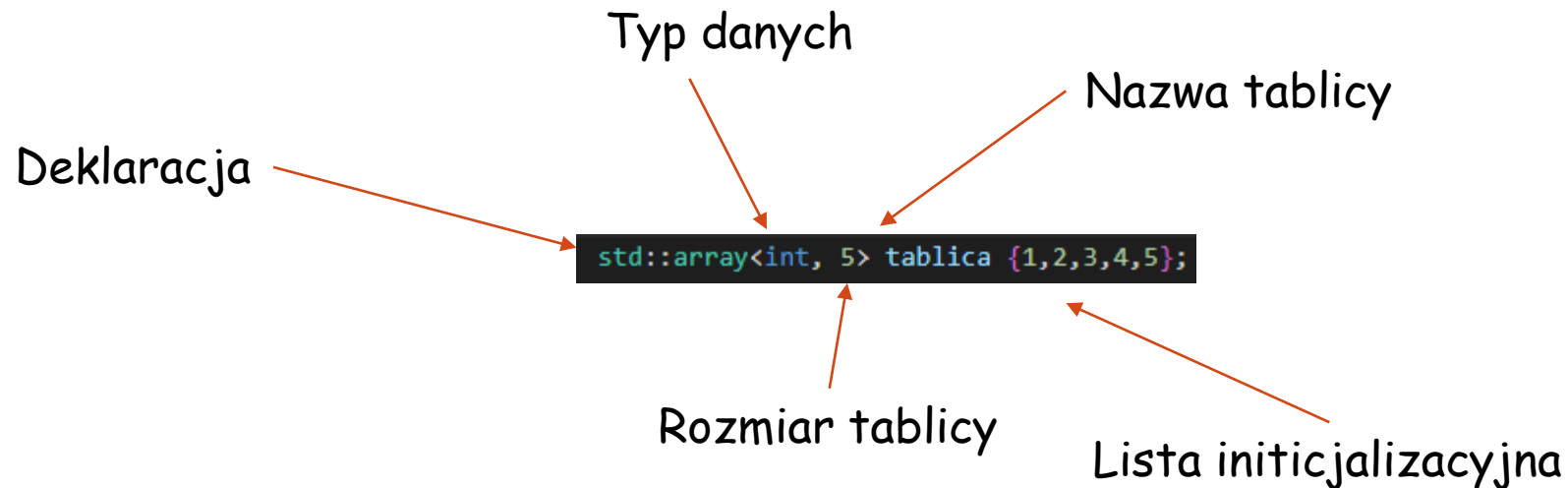
- Lepszy odpowiednik tablic z języka C
- Stały rozmiar
- Deklaracja wraz z rozmiarem

BIBLIOTEKA STL - <ARRAY>



BIBLIOTEKA STL - <ARRAY>

- Deklaracja z inicjalizacją



BIBLIOTEKA STL - <ARRAY>

- Najczęściej wykorzystywane metody:
 - `At(n)` - zwraca wartość n-tego element
 - `[n]` - zwraca wartość n-tego element
 - `Get<n>` - zwraca wartość n-tego element
 - `Fill(value)` - uzupełnia tablice wartościami value

ZADANIE 5

Napisz program, który będzie wczytywał od użytkownika liczby całkowite tak długo, aż ten nie wprowadzi znaku `e`, wtedy kończy się wpisywanie. Program analizuje wpisane przez użytkownika liczby oraz wyświetli statystyki wpisanych liczb.

Pamiętaj, że użytkownik może wprowadzić niepoprawny znak 😊

Dobierz kontener, który najbardziej odpowiada twoim potrzebą.

Pomyśl o użyciu kastowania (`static_cast`) 😊

BIBLIOTEKA STL - <QUEUE>

- Działa w kontekście FIFO - first in, first out

BIBLIOTEKA STL - <QUEUE>

Najczęściej wykorzystywane metody:

- `push(n)` - dodaje liczbę `n` do kolejki
- `empty()` - zwraca `true`, jeśli kolejka jest pusta oraz `false`, jeśli są w niej elementy
- `size()` - zwraca ilość elementów w kontenerze
- `swap(kolejka)` - uzupełnia kontener, na którym została wywołana metoda danymi z *kolejka*, dane z *kolejką* są usuwane.
- `front()` - zwraca wartość pierwszego elementu w kontenerze
- `back()` - zwraca wartość ostatniego elementu w kontenerze
- `pop()` - usuwa pierwszy element kolejki

BIBLIOTEKA STL - <PRIORITY_QUEUE>

- Działa w kontekście FIFO - first in, first out
- Pierwszy element jest największym ze wszystkich elementów znajdujących się w kontenerze

BIBLIOTEKA STL - < PRIORITY_QUEUE >

Najczęściej wykorzystywane metody:

- top() - zwraca wartość największego elementu

BIBLIOTEKA STL - <STACK>

- Działa w kontekście LIFO - last in, first out

BIBLIOTEKA STL - <SET>

- Kontener asocjacyjny
- Każdy element jest unikalny
- Elementy w set są stałe nie można ich modyfikować

BIBLIOTEKA STL - <SET>

- `insert(n)` - dodaje liczbę n do kontenera
- `lower_bound(n)` - zwraca największą wartość jaka znajduje się przed n
- `upper_bound(n)` - zwraca najmniejszy element, który znajduje się po n

BIBLIOTEKA STL - <MULTISET>

- Zestaw wielokrotny
- Kontener asocjacyjny
- Może zawierać element tej samej wartości
- Elementy w set są stałe nie można ich modyfikować

BIBLIOTEKA STL - <MAP>

- Kontener asocjacyjny
- Przechowuje wartość w postaci klucza i wartości

BIBLIOTEKA STL - <MULTIMAP>

- Kontener asocjacyjny
- Przechowuje wartość w postaci klucza i wartości
- Wiele elementów może mieć te same klucze

BIBLIOTEKA STL - MAKRO

- Blok kodu, któremu nadano nazwę
- Definiujemy przy pomocy dyrektywy `#define`
- Definiujemy z argumentami lub bez
- Działaniem przypomina funkcję
- W dobrej praktyce ograniczamy użycie makr

ZADANIE 6

Napisz program, który będzie wczytywał od użytkownika imiona oraz wiek tak długo, aż ten nie wprowadzi znaku `e`, wtedy kończy się wpisywanie. Program po przeanalizowaniu wyświetli poniższe statystyki:

- Średnia wieku
- Średnia wieku dla wieków nieparzystych
- Mediana
- Ilość takich samych imion -> uzględnij, że imiona mogą mieć różną wielkość
- Ilość osób w tym samym wieku

Dobierz kontener, który najbardziej odpowiada twoim potrzebą.



THANK YOU

Kamil Legierski

kamil.legierski@zeiss.com

www.zeiss.com

36