# Klasy pojemnikowe: Wektory

Standard Template Library (w skrócie STL) jest standardową biblioteką szablonów, wchodzącą w skład podstawowych bibliotek C++

Standard Template Library (STL) to zestaw szablonów klas i funkcji w języku C++, które dostarczają programistom gotowych, zoptymalizowanych narzędzi do wykonywania różnych operacji na strukturach danych, algorytmach i kontenerach. Jest to integralna część biblioteki standardowej języka C++, zaprojektowana w celu zapewnienia wydajnych i uniwersalnych narzędzi do programowania.

**Główne składniki STL to:**

1. **Kontenery**: Obejmują różne typy struktur danych, takie jak **wektory, listy, stosy, kolejki, mapy, zbiory** itp. Te kontenery zapewniają różne sposoby przechowywania i organizowania danych, umożliwiając łatwe manipulowanie nimi.

2. **Iteratory**: Są to obiekty, które umożliwiają nawigację po elementach kontenera. Iteratory działają w podobny sposób do wskaźników, pozwalając na przeglądanie, odczytywanie i modyfikowanie danych w kontenerze.

3. **Algorytmy**: STL dostarcza zestaw gotowych algorytmów, które można stosować do kontenerów. Obejmują one algorytmy sortowania, wyszukiwania, transformacji, łączenia itp. Te algorytmy są zoptymalizowane i ogólnie użyteczne, co pozwala programistom unikać ponownego wynajdywania koła.

Korzyści korzystania ze Standard Template Library (STL) obejmują:

- Wydajność: Algorytmy STL są często zoptymalizowane i zaprojektowane w taki sposób, aby działać efektywnie na różnych rodzajach danych.

- Uniwersalność: Dzięki zastosowaniu szablonów, STL jest niezależna od typów, co oznacza, że ​​może być stosowana do różnych typów danych.

- Gotowe rozwiązania: STL dostarcza gotowe, przetestowane rozwiązania dla wielu powszechnych problemów programistycznych, co pozwala oszczędzić czas i wysiłek programistów.

- Łatwość użycia: Dzięki wysokopoziomowym interfejsom kontenerów i algorytmów, korzystanie z STL jest zwykle intuicyjne i prostsze niż implementowanie tych funkcji od zera.

**Porównanie podstawowych struktur danych w C++**

1. Wektor (vector): Jest dynamiczną tablicą, która umożliwia efektywne wstawianie, usuwanie i dostęp do elementów w dowolnym miejscu, dzięki czemu jest często stosowany do przechowywania danych o znanym rozmiarze w kontenerach w C++.

2. Lista (list): Jest strukturą danych opartą na podwójnie wiązanej liście, co umożliwia szybkie wstawianie i usuwanie elementów w dowolnym miejscu, ale kosztem wydajności dostępu do elementów w porównaniu z wektorem.

3. Talia (deque): Jest to struktura danych podobna do wektora, ale umożliwia wstawianie i usuwanie elementów zarówno z przodu, jak i z tyłu kolekcji w czasie stałym.

4. Stos (stack): Jest to struktura danych LIFO (Last-In-First-Out), która umożliwia dodawanie i usuwanie elementów tylko z jednego końca. W C++ zazwyczaj implementowany jest przy użyciu wektora lub listy.

5. Kolejka (queue): Jest to struktura danych FIFO (First-In-First-Out), która umożliwia dodawanie elementów na końcu i usuwanie ich z początku. W C++ zazwyczaj implementowany jest przy użyciu listy lub talii.