



vector w C++

Krótkie wprowadzenie praktyczne

STL – Standard Template Library

- Potężny zestaw bibliotek standardowych dostępnych w języku C++
- Zawiera implementacje m.in.:
 - Algorytmów – sortowania, wyszukiwania...
 - Tablic dynamicznych – vector, deque...
 - Kolejek
 - Stosów
 - Map (słowników)
 - I wielu innych obiektów i nie tylko

Tablice dynamiczne

- W przeciwieństwie do klasycznych tablic (statycznych) ich rozmiar nie jest stały
- Można ustalić ich rozmiar, a potem go zmienić
- Można do nich dodawać nowe elementy, a także je usuwać
- Tablice dynamiczne w stl:
 - vector
 - deque



Tablice dynamiczne - użycie

- Tablice z biblioteki stl tworzymy używając następującej składni:

```
nazwa_klasy<typ_elementów> nazwa_zmiennej;
```

- Gdzie:
 - nazwa_klasy: nazwa typu z biblioteki stl, np. vector
 - typ_elementów: (prawie) dowolny typ danych, jakie chcemy przechowywać w tablicy, np. int
 - nazwa_zmiennej: nazwa naszej tablicy dynamicznej

Tablice dynamiczne - użycie

- Po utworzeniu tablicy i wypełnieniu jej elementami, możemy jej używać podobnie do tablic statycznych
- Aby odczytać lub zmodyfikować wskazany element używamy notacji nawiasów kwadratowych, np.:

```
tab[5] = 23;
```

- Poza klasycznym zastosowaniem, tablice dynamiczne z biblioteki stl posiadają także wiele własnych metod (funkcji), których możemy użyć, aby np.: dodać element do tablicy, odczytać obecny rozmiar tablicy, wyczyścić tablicę itp.

Vector

- Biblioteka: `vector`
`#include<vector>`
- Pozwala na szybkie dodawanie nowego elementu na koniec tablicy, a także jego usuwanie
- W konstrukcji każdy element w `vector` ma wskaźnik na następny element

Vector - inicjalizacja

- Pusta tablica przechowująca elementy typu `int`:

```
vector<int> tab;
```

- Tablica zawierająca 10 elementów o domyślnej wartości (dla `int` to 0)

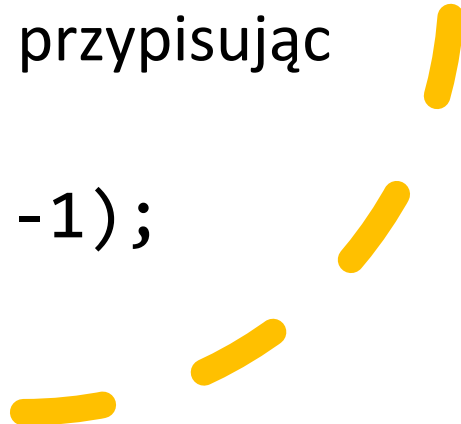
```
vector<int> tab(10);
```

- Tablica zawierająca 10 elementów o wartości -1

```
vector<int> tab(10, -1);
```

- Możemy też podobny efekt uzyskać przypisując nową wartość

```
tab = vector<int>(10, -1);
```



Vector – dodawanie elementów

- Dodanie elementu na koniec tablicy:
`tab.push_back(23);`
- Dodanie elementu na 5-tą pozycję:
`tab.insert(tab.begin()+5, 23);`



Vector — usuwanie elementów

- Usunięcie ostatniego elementu z tablicy:
`tab.pop_back();`
- Usunięcie elementu znajdującego się na 5-tej pozycji:
`tab.erase(tab.begin() + 5);`

