Вектор

Библиотека: vector

Синтаксис: vector <type> name;

Векторът е динамичен шаблонен масив[1], част от стандартната библиотека на C++ (STL). Също като обикновените масиви, векторите поддържат пряк достъп до елементите си, следователно се използват по сходен начин, но, за разлика от тях, могат да променят размера си според нуждите на потребителя.

Освен използвайки прекия достъп, друг начин за достъп до елементите на масива е чрез итератори, подобно на повечето структури от STL. Предпочитани са пред масивите заради своя променлив капацитет. Капацитетът на вектора[2] се удвоява всеки път при нужда – след преоразмеряване или добавяне на нов елемент в края.

Пример:

```
vector<int> v;                  //създава празен (с размер 0) вектор с елементи от тип int
vector<int> v1(4, 128);         //създава вектор с 4 елемента от тип int, всеки от тях със стойност 128
vector<int> v2(4);              //създава вектор с 4 елемента от тип int със стойност по подразбиране (в случая - 0)
vector<vector<int> >g;          //създава вектор от вектори (таблица) от тип int. В началото таблицата е празна.
vector<vector<int> >g1(4); //създава таблица с 4 празни реда от тип int.
```

Често използвани методи и оператори:

=	Оператор за присвояване.
.size()	Връща текущия размер (брой елементи) на вектора.
.begin()	Връща итератор, сочещ началото на вектора.
.end()	Връща итератор, сочещ края (след последния елемент) на вектора.
.resize(n)	Променя размера на вектора на n елемента.
.assign(n, x)	Присвоява на първите n елемента стойност x (ако е необходимо, преоразмерява).
.push_back(x)	Добавя елемент със стойност x в края на вектора.
.insert(pos, x)	Добавя (вмъква) на позиция pos (подадена чрез итератор) елемент със стойност x.
.pop_back()	Изтрива последния елемент от вектора.
.erase(pos)	Изтрива елемента на позиция pos (подадена чрез итератор).

[1] Шаблонен – може да бъде от произволен тип (int, string, vector<int>, …), който се задава при създаване на обекта; динамичен – може да съдържа променлив брой елементи.

[2] Капацитетът и размерът са различни неща.

.clear()	Изтрива всички елементи от вектора.
[n]	Оператор за пряк достъп до n-тия елемент.

Не толкова често използвани функции:

.capacity()	Връща текущия капацитет на вектора.
.emplace(pos, x)	Вмъква на позиция pos (подадена чрез итератор) елемента x. Връща итератор, сочещ към новия елемент. Важна оптимизация е, че, за разлика от insert, работи с референция на x.
.emplace_back(x)	Добавя в края на вектора елемента x. Важна оптимизация е, че, за разлика от push_back, работи с референция на x.

Допълнителни източници:

- Dynamic Arrays in C++ (std::vector)
- VECTOR/DYNAMIC ARRAY - Making DATA STRUCTURES in C++
- Optimizing the usage of std::vector in C++
- std:: vector – cplusplus.com
- C++ Vectors (With Examples)