Оглавление

[Общие понятия 1](#_Toc130244285)

[Vector 2](#_Toc130244286)

# Общие понятия

STL – Стандартная библиотека шаблонов

Большинство контейнеров STL – реализация динамических структур данных

Набор стандартных решений для частоиспользуемых задач

Контейнер – набор однотипных элементов

# Vector

Вектор – улучшенный динамический массив

Не нужно заботится о выделении и освобождении памяти. Всё происходит автоматически.

Выделяется новая память с запасом, когда добавляется новый элемент. Можно сказать, что вектор резервирует память под данные, которые будут занесены в будущем. (происходит при достижении максимального числа элементов в capacity )

Создание вектора выглядит следующим образом:

**std::vector <тип данных> название\_вектора**

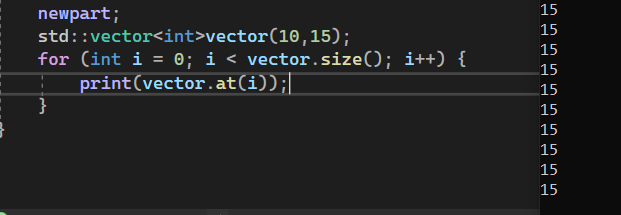


Как и с обычным динамическим массивом можно инициализировать его следующим образом



Если нужно инициализировать вектор числами X K-раз, то можно использовать следующую конструкцию

**std::vector<тип данных> имя\_вектора(K, X)**



**Методы вектора:**

* vector.push\_back(x) – добавление x в конце массива
* vector.pop\_back() – удаление последнего элемента
* vector.size() – возвращает количество элементов массива
* vector.capacity() – возвращает зарезервированные ячейки памяти
* vector.reserve(x) – резервирует x ячеек памяти.
* vector.shrink\_to\_fit() – убирает зарезервированную область памяти.
* vector.at – аналогично vector[x], только добавляет ограничение, чтобы не выйти из области доступной памяти (проверяет границы), но работает немного медленнее
* vector.clear() – очищаем массив
* vector.empty() – «пустой ли массив?» да = 1, нет = 0.
* vector.resize() – изменяет размер контейнера
* vector.insert() – вставка элемента на i-ую позицию.
* vector.erase() – удаление i-го элемента