

CMDK

Copyright © 2016-2017 by Marat Sungatullin

В этом документе даются сведения, необходимые для программирования игр на движке “CMDK”.

## Структуры данных

### Простой динамический массив Array

В движке существует простой динамический массив, элементы которого размещаются в статической памяти, если число хранимых элементов не превышает размера внутреннего статического буфера, который задается как второй параметр шаблона класса.

Шаблон класса простого динамического массива имеет вид:

template <class T, int N=1> class Array

Например, если задан массив Array<char, 512> и число элементов в нем никогда не превышало значения 512, то все данные будут размещаться в его внутреннем статическом массиве-буфере. Если же число элементов, которые нужно сохранить в массиве, превысит 512, то элементы будут размещаться во внутреннем динамическом массиве. Емкость массива не может уменьшаться при операциях.

**Интерфейс**

Класс имеет следующий интерфейс.

Конструктор по умолчанию. Создает пустой массив.

Конструктор, заполняющий массив числом заданных элементов (конструируемых своими конструкторами по умолчанию).

Конструктор копирования создает массив как копию другого массива.

Оператор копирования – копирует элементы из другого массива.

Метод begin возвращает указатель на начало массива (адрес элемента с нулевым индексом). Метод end возвращает указатель на память сразу за последним элементом массива. Оператор индексирования служит для доступа к произвольному элементу массива по индексу. Доступны константные и неконстантные версии методов begin, end и оператора индексирования.

Метод size возвращает число элементов в массиве.

Метод clear удаляет все элементы массива.

Метод setSize устанавливает новый размер массива. Если новый размер массива меньше текущего, вместо удаляемых элементов помещаются копии временного объекта, построенного конструктором по умолчанию.

Для добавления элемента в конец массива необходимо использовать метод add, в который нужно передать ссылку на добавляемый элемент. Метод подобен методу push\_back вектора библиотеки stl.

Для добавления элемента перед индексом (сместив элементы за индексом вправо к концу массива) необходимо использовать метод add, в который нужно передать индекс и ссылку на добавляемый элемент. Метод подобен методу insert вектора библиотеки stl, но в отличие от последнего не возвращает никаких значений и принимает целочисленный индекс, а не итератор.

Вообще данный класс не имеет и не использует никакие итераторы.

**Пример использования**

lang::Array<**char**, 8> chBuf1;

chBuf1.add('w');

chBuf1.add('o');

chBuf1.add('l');

chBuf1.add('d');

// Массив содержит “wold”

// Добавляем ‘r’, смещая “ld” к концу

chBuf1.add(2, 'r');

// Массив содержит “world”

chBuf1.add(0, ' ');

chBuf1.add(0, 'o');

chBuf1.add(0, 'l');

chBuf1.add(0, 'l');

chBuf1.add(0, 'e');

chBuf1.add(0, 'H');

chBuf1.add('!');

// Массив содержит “Hello world!”

**int** l\_count = 0;

**for** (**char**\* b = chBuf1.begin(); b != chBuf1.end(); b++)

{

**if** ((\*b) == 'l') l\_count++;

}

std::cout << "l character met " << l\_count << " times\n";

chBuf1.clear();