# строки

**Строки –** это массив символов.

### Правила определения строки

При помощи массива типа char

*Char cat[8]={‘a’, ‘d’, ‘e’, ‘c’, ’v’, ’g’, ’t’, ’y’ }; -не строка*

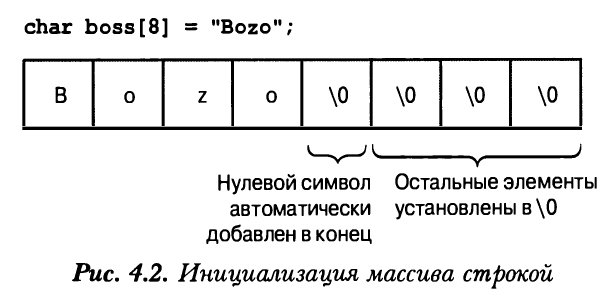
*Char dog[7]= ”adecvgty\0”; - строка*

*Для того чтобы получить строку она должна обязательно оканчиваться на \0 (конец строки)-иначе программа отображенния или вывода данных в консоли будет продолжаться до тех пор пока не встретит ноль в ячейках памяти.*

*char bird[11] = "Mr. Cheeps"; // наличие символа \0 подразумевается*

*char fish[] = "Bubbles"; // позволяет компилятору подсчитать количество символов*

### Конкатенация

Иногда строки могут оказаться слишком большими, чтобы удобно разместиться в одной строке кода. C++ позволяет выполнять конкатенацию строковых литералов — т.е. комбинировать две строки с двойными кавычками в одну.

*cout « "I'd give my right arm to be" " a great violinist. \n";*

*cout << "I'd give my right arm to be a great violinist. \n" ;*

*cout « "I'd give my right ar"*

*"m to be a great violinist. \n";*

Для того что бы программа сама подсчитала и выделила определенный обьем памяти для переменной нужно вместо указания количества символов оставить пустые скобки.

Для определения количества символов содержащихся в массиве применяется директива <cstring> strlen его оператор

На заметку!

При определении минимального размера массива, необходимого для хранения строки, не забудьте учесть при подсчете завершающий нулевой символ.

**Строчно-ориентированный ввод с помощью getline ()**

Функция getline () читает целую строку, используя символ новой строки, который передан клавишей <Enter>, для обозначения конца ввода. Этот метод инициируется вызовом функции cin.getline().

*cin.getline(name,20); //ограничение 20 элементами массива или вывод только 20 элементов*

Листинг 4.4. insrt2 . срр

*// insrt2.cpp -- чтение более одного слова с помощью getline*

*#include <iostream>*

*int main()*

*{*

*using namespace std;*

*const int ArSize = 20;*

*char name[ArSize];*

*char dessert[ArSize];*

*cout « "Enter your name:\n";*

*cin.getline(name, ArSize);*

*cout « "Enter your favorite dessert:\n";*

*cin.getline(dessert, ArSize);*

*cout « "I have some delicious " « dessert;*

*cout « " for you, " << name « " .\n";*

*return 0;*

*}*

**Строчно-ориентированный ввод с помощью get ()**

Теперь попробуем другой подход. Класс istream имеет функцию-член get () , которая доступна в различных вариантах. Один из них работает почти так же, как get line (). Он принимает те же аргументы, интерпретирует их аналогичным

образом, и читает до конца строки. Но вместо того, чтобы прочитать и отбросить символ новой строки, get () оставляет его во входной очереди. Предположим, что используются два вызова get () подряд:

*cin.get(name, ArSize);*

*cin.get(dessert, Arsize); // проблема*