

**КАЛУЖСКИЙ ФИЛИАЛ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ Н.Э. БАУМАНА
(национальный исследовательский университет)»**



Факультет «Информатика и управление»

Кафедра «Программное обеспечение ЭВМ, информационные технологии»

Высокоуровневое программирование

Лекция №9. «Строки»

Символы и С-строки

Символ – элементарная единица, некоторый набор которых несет определенный смысл. В языке программирования C++ предусмотрено использование символьных констант. Символьная константа – это целочисленное значение (типа `int`) представленное в виде символа, заключённого в одинарные кавычки, например `'a'`.

// объявления символьной переменной

```
char symbol = 'a';
```

// где `symbol` – имя переменной типа `char`

// `char` – тип данных для хранения символов

Строки в C++ представляются как массивы элементов типа **`char`**, заканчивающиеся нуль-терминатором `\0` называются ***С-строками*** или строками в стиле C.

`\0` — символ нуль-терминатора.

```
char string[5] = {'a', 'b', 'c', 'd', '\0'};
```

Инициализация С-строк

1. // пример объявления строки

```
char string[10];
```

// где string – имя строковой переменной

// 10 – размер массива, то есть в данной строке может поместиться 9 символов , последнее место отводится под нуль-терминатор.

2. //строка при объявлении может быть инициализирована начальным значением

```
char string[10] = "abcdefghf";
```

3. // посимвольная инициализация строки:

```
char string[10] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'f', '\0' };
```

// десятый символ это нуль-терминатор.

4. //инициализация строки без указания размера

```
char string[] = "abcdefghf";
```

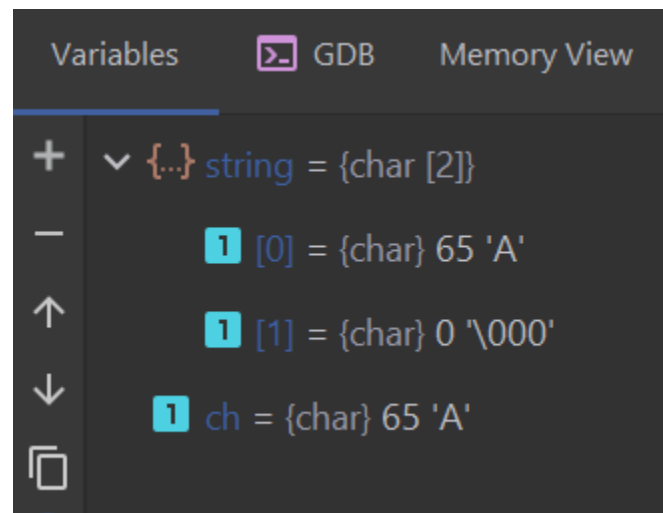
//всё то же самое только размер не указываем.

Вес символа и строки

Вес одного символа – 1 байт или 8 бит.

Задача. Каковы размеры `char string[] = "A"` и `char ch = 'A'`?

```
string: 2,  ch: 1
```



Массивы строк

Объявление массивов строк в языке С также возможно. Для этого используются двумерные массивы символов, что имеет следующий синтаксис:

```
char имя[количество][длина];
```

Первым размером матрицы указывается количество строк в массиве, а вторым — максимальная (с учетом завершающего нуля) длина каждой строки

При объявлении массивов строк можно производить инициализацию:

```
char имя[количество][длина] = { стр. литерал №1, ...,  
стр. литерал №N };
```

Считывание строк с клавиатуры

Проще всего считать строку с клавиатуры, создав массив, который примет эту строку с помощью инструкции `cin`. Считывание строки, введенной пользователем с клавиатуры, отображено в следующей программе.

```
#include <iostream>
using namespace std;
int main() {
    char str[80];
    cout << "Введите строку: ";
    cin >> str;
    cout << "Вот ваша строка: ";
    cout << str << endl;
    return 0;
}
```

Проблема при считывании строки

Но есть недостаток `cin`, он выведет только первое слово, а не весь текст

```
Input ur string: Vasya is good  
That's ur string: Vasya
```

Дело в том, что оператор "`>>`" (в инструкции `cin`) прекращает считывание строки, как только встречает символ пробела, табуляции или новой строки

getline()

Функция **getline** предназначена для ввода данных из потока, например, для ввода данных из консольного окна. Если формально описывать ее функционал, то она извлекает данные из входного потока до строкового разделителя, который не записывается в получившийся массив данных.

```
cin.getline(string, streamsize, separator);
```

где `string` – переменная типа `char`, в которую запишется строка, `streamsize` - максимально количество символов, которое может быть записано в строку, и `separator` – строковый разделитель, показывающий на конец строки.

Пример

```
#include <iostream>
using namespace std;
int main(){
    char str[15] = "";
    cout << "Input smth: " << endl;
    cin.getline(str, 14, '\n');
    cout << str;
    return 0;
}
```

Ввод: Hello, world!

Вывод: Hello, world!

Другие функции

Если подключить файл `<cstring>`, то можно использовать такие функции как:

strcpy()

strcat()

strlen()

strcmp()

и другие.

strcpy()

strcpy (str1, str2)

Выполняется побайтное копирование символов из строки str2 в строку str1. Копирование прекращается только в случае достижения символа '\0' (нуль-терминатор). Перед копированием необходимо проверить выполнение условия, что длина str2 меньше или равна длине str1. В противном случае возможно возникновение ошибок, связанных с наложением данных.

```
char str [20];
```

```
strcpy (str, “Проверка копирования”);
```

strlen()

strlen (s)

Функция strlen вычисляет количество символов в строке до первого вхождения символа конца строки. При этом символ конца строки не входит в подсчитанное количество символов.

```
char str[] = "0123456789"
```

```
int length = strlen(str);
```

```
cout << "length = " << length;
```

Завершающий нулевой символ в длину строки не входит.

strcat()

strcat(s1, s2)

Данная функция объединяет s2 и s1.

```
char str [80] = “”;
```

```
strcpy (str, “Для продолжения”);
```

```
strcat (str, “Нажмите клавишу”);
```

strcmp()

strcmp(str1, str2)

Функция имеет тип `int`, так как данная функция выполняет сравнение двух строк `str1` и `str2`, различая прописные и строчные буквы, в результате сравнения возвращает одно из следующих значений:

Если результат функции < 0 , то `str1 < str2`.

Если результат функции $= 0$, то строки эквивалентны

Если результат функции > 0 , то `str1 > str2`

```
str1 [ ] = «Borland»
```

```
str2 [ ] = «BORLAND»
```

```
int i;
```

```
i = strcmp (str1, str2); // i = 32
```

strchr()

strchr(s, 'symbol')

Производит поиск символа в строке, указывает место первого вхождения символа в строку. Если символ не найден, функция возвращает NULL.

```
char str [ ] = «абвгдеёжзийк»;  
pstr = strchr (str, 'ж');
```

Результат работы программы – «жзийк» в строке str.

strrchr() – возвращает указатель на последний, совпадающий с заданным символом, символ в строке. Если символ не найден, возвращается значение NULL.

И некоторые другие функции

`strlwr(s)` — преобразует прописные буквы в строчные

`strupr(s)` — преобразует строчные буквы в прописные

`strrev(s)` — реверсирование строки, т.е. изменяет порядок следования символов на обратный.

`stricmp(s)` — тоже самое, что и `strcmp()` только не берет в учет регистр.