

Строки C++

- Строка в C++ - это массив символов, оканчивающийся нулевым символом ('\0').
- Таким образом, можно определить строки двумя способами: как массив символов или как указатель на первый символ строки, например:
 - `char str1[10] = "string1";` // объявление строки с помощью массива символов
- `char *str` // указатель
- Различие между указателями `str1` и `str` заключается в том, что указатель `str1` является именованной константой, а указатель `str` – переменной.

Строки C++

- Длина строковой константы не может превышать 509 символов по стандарту. Однако, многие реализации допускают строки большей длины. При инициализации строк размерность массива лучше не указывать, это выполнит компилятор, подсчитав длину строки и добавив к ней единицу.
- Например, `char s1[] = "This is a string.";`
-

Объявление

- `char str[10] = "ABCDE";` создает массив из 10 символов
- `char str1[]={'A','B','C','D','E','\0'};`
- `char str2[11];`

Задана строка, скопировать ее в СИМВОЛЬНЫЙ МАССИВ.

- `#include<iostream.h>`
- `void main()`
- `{ char str1 [] = "1234567890", /* объявляем символьный массив str1 и инициализируем его */`
- `str2[11]; // объявляем символьный массив без инициализации`
- `/* в цикле пока не встретится конец строки присваиваем текущему элементу массива str2 символ из массива str1 */`
- `for(int i = 0; str1[i] != '\0'; i++) str2[i] = str1[i];`
- `str2 [i] = '\0';// копируем ноль-символ в str2.`
- `cout << str2 << '\n'; // вывод строки на экран`

;

Ввод и вывод

- элементы произвольного массива можно вводить с клавиатура и выводить на экран только по одному элементу в цикле, то в символьный массив можно ввести сразу всю строку, используя оператор ввода
- `cin >> Имя_массива;`
- и, аналогичным образом, вывести сразу всю строку на экран, используя оператор вывода
- `cout << Имя_массива;`

Пример

- `#include <iostream.h>`
- `void main()`
- `{`
- `char str[31];` // объявление символьного массива
- `cout<<"Enter a string (max 30 symbols):";`
- `cin>>str;` // ввод строки
- `cout<<"\nYou enter string:"<<str;` // вывод строки
- `}`

Пример

- `#include<iostream.h>`
- `void main()`
- `{ char str[] = "Здравствуй, мир!"; // объявляем символный массив`
- `char *b; // объявляем указатель на символ`
- `b = &str[12]; // теперь b указывает на 12-ый символ str`
- `*b = 'M'; // присваиваем первому элементу b символ 'M'`
- `cout << b; // выводим строку b на экран (Mир!)`
- `}`

strlen() (от слова length — длина)

- `#include <cstring>`
- `using namespace std;`
- `int main()`
- `{setlocale(LC_ALL, "rus");`
- `char ourStr[128] = "";` *// для сохранения строки*
- `cout << "Введите текст латиницей (не больше 128 СИМВОЛОВ):\n";`
- `cin.getline(ourStr, 128);`
- `cout << "Строка \"\" << ourStr << "\" состоит из "`
- `<< strlen(ourStr) << " символов!\n\n";`
- `return 0;}`

`strcat()` (от слова `concatenation` — соединение)

- `int main()`
- `{ char someText1[64] = "Сайт purecodespp.com!";`
- `char someText2[] = "Учите C++ с нами!";`
- `cout << "Строка someText1 -\\"" << someText1 <<" \n";`
- `cout << "Строка someText2 - \\"" << someText2 <<"\n";`
- `strcat(someText1 , someText2); // передаём
someText2 в функцию`
- `cout << "Строка someText1 после объединения с
someText2 -\n\\"" << someText1 << "\" \n\n";`
- `return 0;}`

`strcpy()` (от слова *copy* — копирование)

- `int main()`
- `{ char someText1[64] = "Сайт!";`
- `char someText2[] = "Основы C++";`
- `cout <<"Строка someText1 " <<someText1;`
- `cout << "Строка someText2 - \" <<someText2 ;`
- `strcpy(someText1, someText2);` *// передаём someText1 и someText2 в функцию*
- `cout << "Строка someText1 после копирования someText2 -\n\" << someText1 ;`
- `return 0;}`

strcmp() (от слова compare — сравнение)

- int main()
- {
- char someText1[] = "Сайт purecodespp.com!";
- char someText2[] = "Сайт purecodespp.com/";
- cout << "Строка someText1 - \" << someText1;
- cout << "Строка someText2 - \" <<someText2 ;
- cout <<strcmp(someText1, someText2) <<endl;
- return 0;
- }

