**Символы и строки в С++**

**Строка — последовательность (массив) символов**

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

**В таблице ASCII представлены символы и их целочисленные значения.**

|  |  |
| --- | --- |
| 1234 | **// объявления символьной переменной** **char symbol = 'a';****//  где symbol – имя переменной типа  char****//  char – тип данных для хранения символов** |

**Строки в С++ представляются как массивы элементов типа char, заканчивающиеся нуль-терминатором \0**

**называются С строками или строками в стиле С.**

**\0  — символ нуль-терминатора.**

**Символьные строки состоят из набора символьных констант заключённых в двойные кавычки. При объявлении строкового массива необходимо учитывать наличие в конце строки нуль-терминатора, и отводить дополнительный байт под него.**

|  |  |
| --- | --- |
| 1234 | **// 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.** |

**Строка при объявлении  может быть инициализирована**

**начальным значением, например, так:**

|  |  |
| --- | --- |
| **1** | **char string[10] = "abcdefghf";** |

**Если подсчитать кол-во символов в двойных кавычках после символа равно их окажется 9, а размер строки 10 символов,  последнее  место отводится под нуль–терминатор, причём компилятор сам добавит его в конец строки.**

|  |  |
| --- | --- |
| 123 | **// посимвольная инициализация строки:** **char string[10] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'f', '\0'};****// десятый  символ это  нуль-терминатор.** |

**При объявлении строки не обязательно указывать её размер, но при этом обязательно нужно её инициализировать начальным значением. Тогда размер строки определится автоматически и в конец строки добавится нуль-терминатор.**

|  |  |
| --- | --- |
| 123 | **//инициализация строки без указания размера** **char string[] = "abcdefghf";****//всё то же самое только размер не указываем.** |

**Строка может содержать символы, цифры и специальные знаки. В С++ строки заключаются в двойные кавычки. Имя строки является константным указателем на первый символ. Разработаем программу, с использованием строк.**

|  |  |
| --- | --- |
| 12345678910111213141516 |  **symbols.cpp//: определяет точку входа для консольного приложения.** **#include "stdafx.h"****#include <iostream>****using namespace std;****int main(int argc, char\* argv[])**{**char string[] = "this is string - "; // объявление и инициализация строки**    **cout << "Enter the string: ";**    **char in\_string[500]; // строковый массив для ввода** **gets(in\_string); // функция gets() считывает все введённые символы с пробелами до тех пор, пока не будет нажата клавиша Enter** **cout << string << in\_string << endl; // вывод строкового значения**    system("pause");    return 0;} |

В строке 12с помощью функции gets()считаются все введённые символы с пробелами до тех пор, пока во вводимом потоке не встретится код клавиши enter. Если использовать операцию cin то из всего введённого считается последовательность символов до первого пробела (см. Рисунок 1).

| Таблица 1 — Функции для работы со строками и символами |
| --- |
| **Функция** | **Пояснение** |
| [**strlen**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strlen/)**(**имя\_строки**)** | определяет длину указанной строки, без учёта нуль-символа |
| **Копирование строк** |
| [**strcpy**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strcpy/)**(**s1,s2**)** | выполняет побайтное копирование символов из строки  s2 в строку s1 |
| [**strncpy**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strncpy/)**(**s1,s2, n**)** | выполняет побайтное копирование n символов из строки  s2 в строку s1. возвращает значения s1 |
| **Конкатенация строк** |
| [**strcat(**s1,s2**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strcat/) | объединяет строку s2 со строкой s1. Результат сохраняется в s1 |
| [**strncat(**s1,s2,n**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strncat/) | объединяет n символов строки s2 со строкой s1. Результат сохраняется в s1 |
| **Сравнение строк** |
| [**strcmp(**s1,s2**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strcmp/) | сравнивает строку s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2,  <0  — если s1>s2 С учётом регистра |
| [**strncmp(**s1,s2,n**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strncmp/) | сравнивает n символов строки s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2,  <0  — если s1>s2 С учётом регистра |
| **stricmp(**s1,s2**)** | сравнивает строку s1 со строкой s2 и возвращает результат типа **int**: 0 –если строки эквивалентны, >0 – если s1<s2,  <0  — если s1>s2 Без учёта регистра |
| **strnicmp(**s1,s2,n**)** | сравнивает n символов строки s1 со строкой s2 и возвращает результат типа int: 0 –если строки эквивалентны, >0 – если s1<s2,  <0 — если s1>s2 Без учёта регистра |
| **Обработка символов** |
| [**isalnum(**c**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isalnum/) | возвращает значение true, если с является буквой или цифрой, и false в других случаях |
| [**isalpha(**c**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isalpha/) | возвращает значение true, если с является буквой,  и false в других случаях |
| [**isdigit(**c**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isdigit/) | возвращает значение true, если с является цифрой, и false в других случаях |
| [**islower(**c**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-islower/) | возвращает значение true, если с является буквой нижнего регистра, и false в других случаях |
| [**isupper(**c**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isupper/) | возвращает значение true, если с является буквой верхнего регистра, и false в других случаях |
| [**isspace(**c**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-isspace/) | возвращает значение true, если с является пробелом, и false в других случаях |
| [**toupper(**c**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cctupe-ctupe-h/funkciya-toupper/) | если символ с, является символом нижнего регистра, то функция возвращает преобразованный символ с в верхнем регистре, иначе символ возвращается без изменений. |
| **Функции поиска** |
| [**strchr(**s,c**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strchr/) | поиск первого вхождения символа св строке s**.**В случае удачного поиска возвращает указатель на место первого вхождения символа с**.**Если символ не найден, то возвращается ноль. |
| [**strcspn(**s1,s2**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strcspn/) | определяет длину начального сегмента строки s1, содержащего те символы, которые не входят в строку s2 |
| [**strspn(**s1,s2**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstring-string-h/funkciya-strspn/) | возвращает длину начального сегмента строки s1, содержащего только те символы, которые входят в строку s2 |
| **strprbk(**s1,s2**)** | Возвращает указатель  первого вхождения любого символа строки s2 в строке s1 |
| **Функции преобразования** |
| [**atof(**s1**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdlib-stdlib-h/funkciya-atof/) | преобразует строку s1 в тип double |
| [**atoi(**s1**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdlib-stdlib-h/funkciya-atoi/) | преобразует строку s1 в тип int |
| [**atol(**s1**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdlib-stdlib-h/funkciya-atol/) | преобразует строку s1 в тип long int |
| **Функции стандартной библиотеки ввода/вывода <stdio>** |
| [**getchar(**с**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdio-stdio-h/funkciya-getchar/) | считывает символ ссо стандартного потока ввода, возвращает символ в формате int |
| [**gets(**s**)**](http://cppstudio.com/spravochnik/standartnye-zagolovochnye-fajly-iz-si-v-s/zagolovochnyj-fajl-cstdio-stdio-h/funkciya-gets/) | считывает поток символов со стандартного устройства ввода в строку sдо тех пор, пока не будет нажата клавиша ENTER |

**3.Копирование строк**

|  |  |
| --- | --- |
| 1234567891011121314151617 | **// str\_cpy.cpp//: определяет точку входа для консольного приложения.** #include "stdafx.h"#include <iostream>using namespace std; int main(int argc, char\* argv[]){ **char s2[27] = "Counter-Strike 1.6****forever";          // инициализация строки s2****char s1[27];   // резервируем строку для функции strcpy()****cout <<"strcpy(s1,s2)=" << strcpy(s1,s2) << endl; // содержимое строки s2 скопировалось в строку s1, возвращается указатель на s1****cout << "s1=" << s1 << endl; // вывод содержимого строки s1****chars3[7];                                          // резервируем строку для следующей функции** **cout<<strncpy(s3,s2,7)<<endl;      // копируем 7 символов из строки s2 в строку s3**    system("pause");    return 0;} |

В **строках 9, 10**создаём строковые массивы на 27 символов, словосочетание "Counter-Strike 1.6 forever" содержит 26 символов, последнее место в массиве займет нуль-символ. В **строке 11** функция strcpy(s1,s2) копирует  значение строки  s2 в строку s1 и возвращает указатель на строку s1. Если строка s1 будет меньше строки s2, то скопируется то количество символов, которое вместится в  строку s2**. Строка 10**всего лишь показывает, что в строке s1содержится скопированное значение. В **строке 14**функция strncpy(s3, s2, sizeof(s3)) выполняет копирование 7 символов строки s2в строку s3 (см. Рисунок 2).

**Конкатенация строк**

**Использование функций strcat() и strncat(), для объединения строк, то есть для их** конкатенации.

|  |  |
| --- | --- |
| 12345678910111213141516 | // str\_cat.cpp: определяет точку входа для консольного приложения. #include "stdafx.h"#include <iostream>using namespace std; int main(int argc, char\* argv[]){    **char s1[30] = "I am ";**  **char s2[] = "programmer on the C++!!!!";****cout << strcat(s1,s2) << endl;               // объединяем строки s1 и s2, результат сохраняется в строке s1****char s3[23] = "I am a good ";****cout << strncat(s3,s2,10) <<  "!!!" << endl; // объединяем 10 символов строки s2 со строкой s3**    system("pause");    return 0;} |

В **строке 11**функция strcat(s1,s2)объединяет строки s1и s2, результат сохраняется в строке s1. По этому при объявлении строки s1 её размер установлен на 30 символов. В **строке 13**функция**strncat(s3,s2,10)**объединяет 10 символов из строки s2(как раз помещается слово programmer) со строкой s3, результат сохраняется в строке s3. И по этому размер строки s3также задан фиксировано (см. Рисунок 3).

**5.Сравнение строк**

**Рассмотрим работу функции strcmp(), остальные функции используются аналогично**, так что каждую рассматривать не будем.

|  |  |
| --- | --- |
| 1234567891011121314151617 | **// str\_cmp.cpp: определяет точку входа для консольного приложения**. #include "stdafx.h"#include <iostream>#include <iomanip>using namespace std; int main(int argc, char\* argv[]){    **char s1[] = "www.cppstudio.com";****char s2[] = "http://www.cppstudio.com";****cout << " s1 == s1 -> " << setw(2) << strcmp(s1,s1) << endl; // строка s1 = s1****cout << " s1 <  s2 -> " << setw(2) << strcmp(s1,s2) << endl; // строка s1 < s2****cout << " s2 >  s1 -> " << setw(2) << strcmp(s2,s1) << endl; // строка s2 > s1**    system("pause");    return 0;} |

В**строках 12, 13, 14**функция strcmp()выполняет сравнение строк, в различном порядке, таким образом были рассмотрены все возможные варианты работы данной функции. В первом случае строку s1сравнивали с самой собой соответственно и результат равен 0. Во втором случае строка s1 оказалась меньше строки s2поэтому результат равен **1**. В третьем случае строка s1по-прежнему меньше строки s2, но мы поменяли эти строки местами, таким образом, порядок параметров функции изменился и результат стал равен **-1**

**Обработка символов**

Функции из данной группы умеют различать, к какому типу знаков относятся те или иные символы, например буквы, цифры, специальные знаки.

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627 | // issss.cpp: определяет точку входа для консольного приложения. #include "stdafx.h"#include <iostream>using namespace std; int main(int argc, char\* argv[]){    char symbol    = 'd'; // буква    char digit     = '9'; // цифра    char space     = ' '; // пробел    char character = '#'; // знак    **// функция isalnum() проверяет является ли её аргумент буквой или цифрой****cout << symbol    << " - it is digit or alpha?: "; isalnum(symbol)    ? cout << "true\n": cout << "false\n";****// функция isalpha() проверяет является ли её аргумент буквой****cout << symbol    << " - it is alpha?:          "; isalpha(symbol)    ? cout << "true\n": cout << "false\n";****// функция isdigit() проверяет является ли её аргумент цифрой****cout << digit     << " - it is digit?:          "; isdigit(digit)     ? cout << "true\n": cout << "false\n";****// функция isspace() проверяет является ли её аргумент пробелом****cout << space     << " - it is space?:          "; isspace(space)     ? cout << "true\n": cout << "false\n";****// функция islower() проверяет является ли её аргумент буквой нижнего регистра****cout << symbol    << " - it is lower alpha?:    "; islower(symbol)    ? cout << "true\n": cout << "false\n";****// функция isupper() проверяет является ли её аргумент буквой верхнего регистра****cout << symbol    << " - it is upper alpha?:    "; isupper(symbol) ? cout << "true\n": cout << "false\n";**    system("pause");    return 0;} |

В данной программе по использованию функций вопросов возникать не должно. Функции сами по себе возвращают целочисленное значение, положительное – true, отрицательное или  ноль – false. В формировании результата работы функций участвовал тернарный оператор, то есть выводилось сообщение true или false без всяких числовых значений (см. Рисунок 5).