***Краткая справка по некоторым особенностям C++***

Оглавление

[Общее. 1](#_Toc493074776)

[Цвет фона и текста консоли. 3](#_Toc493074777)

[var++ и ++var. Разница. 4](#_Toc493074778)

[Узнать время работы алгоритма. 4](#_Toc493074779)

[Строки. Всё про string. 5](#_Toc493074780)

[Работа с текстовыми файлами. 7](#_Toc493074781)

[Работа с заголовочными (Header) и исходными (Source) файлами. 8](#_Toc493074782)

# Общее.

http://cppstudio.com/ // Сайт по C++

ios\_base::sync\_with\_stdio(false); /\* Ускорить работу проги с потоками,

если используется cin/cout. \*/

//#include <windows.h>

setlocale(LC\_ALL, "Russian"); // Корректный вывод рус. языка (Win 7)

#include <iomanip>

setw(n); // Выравнивает строку вывода по n-ому положению

system("pause>>void") /\* Чтобы в конце проги не выводилась надпись

про нажатие любой клавиши. \*/

#include <windows.h>

//#include <clocale> // Шрифт в настройках консоли "Lucida Console".

SetConsoleOutputCP(1251);

SetConsoleCP(1251); // Корректный ввод и вывод рус. языка (Win XP и 7).

#include <windows.h>

Sleep(кол-во милисек.); // Тайминг вывода.

# Цвет фона и текста консоли.

system("color [num]")

[num] - Две шестнадцатеричные цифры.

> 1-ая задаёт цвет фона.

> 2-ая задаёт цвет текста.

Значения цифр:

0 = Чёрный 8 = Серый

1 = Синий 9 = Светло-синий

2 = Зелёный A = Светло-зелёный

3 = Голубой B = Светло-голубой

4 = Красный C = Светло-красный

5 = Лиловый D = Светло-лиловый

6 = Жёлтый E = Светло-жёлтый

7 = Белый F = Ярко-белый

Если аргумент не был введён, то команда восстанавливает стандартные значения цветов.

# var++ и ++var. Разница.

++var имеет более высокий приоритет выполнения чем var++,

т.е. в выражениях сначала выполняется операция

инкремента (увелич. на 1), а затем уже другая операция.

Пример:

int x = 5;

int y;

y = ++x; // y = 6, x = 6: сначала увеличивается x на 1, потом y = x.

y = --x; // y = 5, x = 5: сначала уменьшается x на 1, потом y = x.

y = x++; // y = 5, x = 6: сначала присваивается значение y = x, потом x увеличивается на 1.

# Узнать время работы алгоритма.

#include <time.h>;

unsigned int start\_time = clock(); // Начальное время.

// Здесь должен быть фрагмент кода, время выполнения которого нужно измерить .

unsigned int end\_time = clock(); // Конечное время.

unsigned int search\_time = end\_time - start\_time; // Искомое время в миллисекундах.

# Строки. Всё про string.

#include <string>

string s; // Объявление строки.

string s = "someth"; // Можно так.

getline(cin, s); // Пользователь вводит строку в s.

s.empty(); // True - если s пуста.

// False - если не пуста.

s.size(); // int число. Кол-во символов в s.

s.at(2) == "m"; // 3-ий по счёту элем. Строки.

s.substr(2, 3) == "met"; // Вывести начиная с какого и сколько.

string s1 = "123";

s.insert(3, s1) == "som123eth"; // С какого элем. вставить s1 в s .

string s = "qwertyq", s1 = "er";

s.find(s1) == 2; // int число. Ищет 1-ое вхождение s1 в s с начала.

s.find("q", 2) == 6; // ... Ищет со 2-го элемента.

s.rfind("q") == 6; // ... Ищет с конца.

s.erase(2, 3) == "qwyq"; // С какого элем. и сколько удалить их.

string s = "125.61";

double a; // double a = 123; - на конечн. знач. не повлияло бы.

#include <sstream>

stringstream ss; /\* Всё, что вводится в такой поток, добавляется в конец строки,

а всё, что считыватся из потока — извлекается из начала строки. \*/

ss << s; // Вводится в поток.

ss >> a; /\* Выводится из потока. a принимает значение всего, что было в потоке

За 1 раз выводится только до пробела. \*/

a == 125.61;

cout << ss.str() << "\n"; // 125.61 Вывести на экран то, что в ss.

В обратную сторону:

{

string s; // string s = "831"; - на конечн. знач. не повлияло бы.

double a = 456;

ss << a;

ss >> s;

s == "456";

}

Ещё есть похожие функции:

{

stof(s); // string --> float(double)

stoi(s); // string --> int

to\_string(a); // int --> string

}

# Работа с текстовыми файлами.

#include <fstream>

fstream f; // Объявляется файловая переменная.

string s;

f.open("input.txt", ios::in); // Связывается переменная с файлом. in - в программу.

getline(f, s); // Считывается 1-а строка из файла в строку s.

f.close(); // Закрывается файл.

f.open("output.txt", ios::out); // Связывается переменная с файлом. out - из программы.

f << s; // Вводится в 1-ую строку файла то, что хранится в s.

f.close();

# Работа с заголовочными (Header) и исходными (Source) файлами.

К примеру нам нужен исходник с разнообразными полезными функциями отдельно от основного исходника, включающего в себя функцию «main».

Состав проекта:

• main.cpp – исходный файл с функцией «main»,

• UsefulFunctions.cpp – исходный файл с полезными функциями,

• UsefulFunctions.h – заголовочный файл для UsefulFunctions.cpp.

Обязательно нужно добавить в следующие файлы следующие строки кода, если их ещё нет:

• В main.cpp:

#include "UsefulFunctions.h"

• В UsefulFunctions.cpp (Show() – пример полезной функции):

#include "UsefulFunctions.h"

int Show()

{

return 228;

}

• В UsefulFunctions.h:

#pragma once

int Show();