**ЛЕКЦИЯ 9. СТРОКИ STRING**

Оглавление

[1 Строки string 1](#_Toc118654286)

[1.1 Функции работы со строками 2](#_Toc118654287)

[1.2 Преобразование строки в числовое значение 5](#_Toc118654288)

[1.3 Преобразование числового значения в строку 5](#_Toc118654289)

# Строки string

Тип данных **string** стандартной библиотеки лишен недостатков массивов символов, но может проигрывать им в эффективности.

Основные действия со строками **string** выполняются с помощью операций и методов, а длина строки изменяется динамически в соответствии с потребностями. Для использования класса необходимо подключить к программе заголовочный файл **<string>**.

**std::string** – это псевдоним для **std::basic\_string<char>**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main()**

**{**

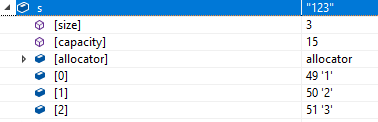
**string str{ "Some string" };**

**string copy = str;**

**string s = "123";**

**cout << str << endl << copy << endl<<s <<endl;**

**}**



* 1. Функции работы со строками

**Пример 1**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main() {**

**setlocale(LC\_ALL, "Russian");**

**string s1("прекрасная королева"), s2("лe"),s3("KopoBa");**

**cout << "s1= " << s1 << endl;**

**cout << "s2= " << s2 << endl;**

**cout << "s3= " << s3 << endl;**

**// Применение функции insert:**

**cout << "после insert:" << endl;**

**cout << "s3= " << s3.insert(4, s2) << endl;**

**cout << "s3= " << s3.insert(7, "к") << endl;**

**// Применение функции erase:**

**s1.erase(0, 3);**

**cout << "после erase:" << endl;**

**cout << "s1= " << s1.erase(12, 2) << endl;**

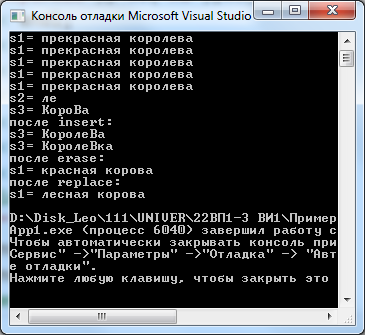
**// Применение функции replace:**

**cout << "после replace:" << endl;**

**cout << "s1= " << s1.replace(0, 3, s3, 4, 2) << endl;**

**return 0;**

**}**



**Пример 2**

**string s1("лecнaя королева"), s2("лe");**

**cout << "sl= " << s1 << endl;**

**cout << "s2= " << s2 << endl;**

**int i = s1.find(s2);**

**int j = s1.rfind(s2);**

**cout << "первое s2 в s1 " << i << endl;**

**cout << "последнее s2 в s1 " << j << endl;**

**cout << "первое 'о' в s1 " << s1.find('о') << endl;**

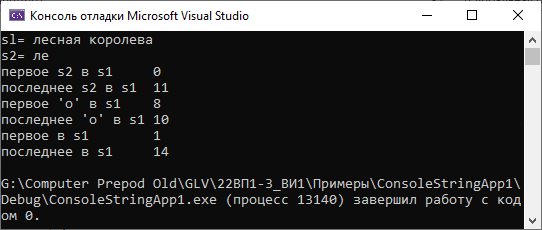
**cout << "последнее 'о' в s1 " << s1.rfind('о') << endl;**

**cout << "первое в s1 " << s1.find\_first\_of("aбвгдe")**

**<< endl;**

**cout << "последнее в s1 " << s1.find\_last\_of("a6вгдe")**

**<< endl;**



**Пример 3**

**string s1("лесная королева"), s2("лe"), s3("корова");**

**cout << "sl= " << s1 << endl;**

**cout << "s2= " << s2 << endl;**

**cout << "s3= " << s3 << endl;**

**if (s2.compare(s3) > 0)**

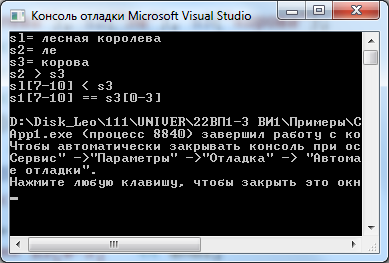
**cout << "s2 > s3 " << endl;**

**if (s1.compare(7, 4, s3) < 0)**

**cout << "sl[7-10] < s3 " << endl;**

**if(s1.compare(7, 4, s3, 0, 4) == 0)**

**cout << "s1[7-10] == s3[0-3] " << endl;**



**Пример 4**

**string s1("Quick! Send for Count Graystone.");**

**//Быстро!Пошлите за графом Грейстоуном**

**string s2("Lord");**

**string s3("Don't ");**

**s1.erase(0, 7); //remove "Quick! "**

**s1.replace(9, 5, s2); //replace "Count" with "Lord"**

**s1.replace(0, 1, "s"); //replace 'S' with 's'**

**s1.insert(0, s3); //insert "Don't " at beginning**

**s1.erase(s1.size() - 1, 1); //remove '.'**

**s1.append(3, '!'); //append "!!!"**

**int x = s1.find(' '); //find a space**

**while (x < s1.size()) //loop while spaces remain**

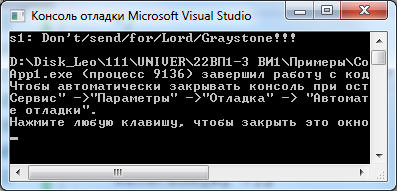
**{**

**s1.replace(x, 1, "/"); //replace with slash**

**x = s1.find(' '); //find next space**

**}**

**cout << "s1: " << s1 << endl;**



**Пример 5**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main()**

**{**

**string text; // Исходный текст**

**cout << "Enter some text: ";**

**getline(cin, text);**

**const string separators{ " ,;:.\"!?'\*\n" }; // разделители слов**

**string words[80]; // массив для хранения слов**

**size\_t start{ text.find\_first\_not\_of(separators) }; // начальный**

**//индекс первого слова**

**int numbers{};**

**// проходим, пока в строке не окажется других символов, кроме**

**//separators**

**while (start != string::npos)**

**{**

**// находим, где кончается слово**

**size\_t end = text.find\_first\_of(separators, start + 1);**

**// если НЕ найден ни один из символов-разделителей**

**if (end == string::npos)**

**// устанавливаем переменную end на конец текста**

**end = text.length();**

**// добавляем в массив слово**

**words[numbers++] = text.substr(start, end - start);**

**// находим начальный индекс следующего слова и переустанавливаем**

**//start**

**start = text.find\_first\_not\_of(separators, end + 1);**

**}**

**// выводим количество слов**

**cout << "\nText contains " << numbers << " words:" << endl;**

**// выводим все слова на консоль**

**for (const auto& word : words)**

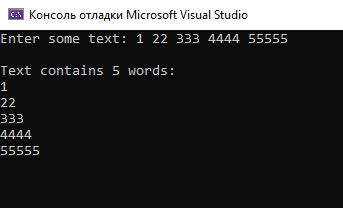
**{**

**cout << word << endl;**

**}**

**return 0;**

**}**



* 1. Преобразование строки в числовое значение

Начиная со стандарта C++11, преобразование строки в число и наоборот встроены в стандартную библиотеку. Все следующие функции присутствуют в <string>.

**float stof(const string& str, size\_t \*idx = 0);**

**double stod(const string& str, size\_t \*idx = 0);**

**long double stold(const string& str, size\_t \*idx = 0);**

**int stoi(const string& str, size\_t \*idx = 0, int base = 10);**

**long stol(const string& str, size\_t \*idx = 0, int base = 10);**

**unsigned long stoul(const string& str, size\_t \*idx = 0, int base = 10);**

**long long stoll(const string& str, size\_t \*idx = 0, int base = 10);**

**unsigned long long stoull(const string& str, size\_t \*idx = 0, int base = 10)**

**string s = "123";**

**int a = stoi(s);**

* 1. Преобразование числового значения в строку

**string to\_string(int val);**

**string to\_string(unsigned val);**

**string to\_string(long val);**

**string to\_string(unsigned long val);**

**string to\_string(long long val);**

**string to\_string(unsigned long long val);**

**string to\_string(float val);**

**string to\_string(double val);**

**string to\_string(long double val);**

**string s1;**

**int a = 123;**

**s1 = to\_string(a);**

**cout << s1 <<endl;**