

Ravesli [Ravesli](#)


- [Уроки по C++](#)
- [OpenGL](#)
- [SFML](#)
- [Qt5](#)
- [RegExр](#)
- [Ассемблер](#)
- [Купить .PDF](#)


Урок №57. Введение в std::string

 [Юрий](#) |

- [Уроки C++](#)

|

 Обновл. 14 Сен 2020 |

 132743

[↑](#)  66

Вашей первой программой на языке C++, вероятно, была всеми известная программа «Hello, world!»:

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hello, world!" << std::endl;
6     return 0;
7 }
```

Не так ли? Но что такое Hello, world!? Hello, world! — это последовательность символов или просто **строка** (англ. «*string*»). В языке C++ мы используем строки для представления текста (имен, адресов, слов и предложений). Строковые литералы (такие как Hello, world!) помещаются в двойные кавычки.

Поскольку их часто используют в программах, то большинство современных языков программирования имеют встроенный тип данных string. В языке C++ есть также этот тип, но не как часть основного языка, а как часть Стандартной библиотеки C++.

Оглавление:

1. [Тип данных string](#)
2. [Ввод/вывод строк](#)
3. [Использование std::getline\(\)](#)
4. [Использование std::getline\(\) с std::cin](#)

5. [Добавление строк](#)
6. [Длина строк](#)
7. [Тест](#)

Тип данных string

Чтобы иметь возможность использовать строки в C++, сначала нужно подключить [заголовочный файл](#) `string`. Как только это будет сделано, мы сможем определять переменные типа `string`:

```
1 #include <string>
2
3 // ...
4 std::string name;
5 // ...
```

Как и с обычными переменными, мы можем инициализировать переменные типа `string` или присваивать им значения:

```
1 std::string name("Sasha"); // инициализируем переменную name строковым литералом "Sasha"
2 name = "Masha"; // присваиваем переменной name строковый литерал "Masha"
```

Строки также могут содержать числа:

```
1 std::string myID("34"); // "34" здесь - это не целое число 34!
```

Стоит отметить, что присваиваемые числа тип `string` обрабатывает как текст, а не как числа, и, следовательно, ими нельзя манипулировать как обычными числами (например, вы не сможете выполнять с ними [арифметические операции](#)). Язык C++ [автоматически не преобразовывает](#) их в значения целочисленных типов или типов с плавающей точкой.

Ввод/вывод строк

Строки можно выводить с помощью `std::cout`:

```
1 #include <iostream>
2 #include <string>
3
4 int main()
5 {
6     std::string name("Sasha");
7     std::cout << "My name is " << name;
8
9     return 0;
10 }
```

Результат выполнения программы:

My name is Sasha

А вот с std::cin дела обстоят несколько иначе. Рассмотрим следующий пример:

```
1 #include <iostream>
2 #include <string>
3
4 int main()
5 {
6     std::cout << "Enter your full name: ";
7     std::string myName;
8     std::cin >> myName; // это будет работать не так, как ожидается, поскольку извлече
9
10    std::cout << "Enter your age: ";
11    std::string myAge;
12    std::cin >> myAge;
13
14    std::cout << "Your name is " << myName << " and your age is " << myAge;
15 }
```

Результат выполнения программы:

```
Enter your full name: Sasha Mak
Enter your age: Your name is Sasha and your age is Mak
```

Хм, что-то не так! Что же случилось? Оказывается, оператор извлечения (>>) возвращает символы из входного потока данных только до первого пробела. Все остальные символы остаются внутри cin, ожидая следующего извлечения.

Поэтому, когда мы использовали оператор >> для извлечения данных в переменную myName, только Sasha был извлечен, Mak остался внутри std::cin, ожидая следующего извлечения. Когда мы использовали оператор >> снова, чтобы извлечь данные в переменную myAge, мы получили Mak вместо 25. Если бы мы сделали третье извлечение, то получили бы 25.

Использование std::getline()

Чтобы извлечь полную строку из входного потока данных (вместе с пробелами), используйте **функцию std::getline()**. Она принимает два параметра: первый — std::cin, второй — переменная типа string.

Вот программа, приведенная выше, но уже с использованием std::getline():

```
1 #include <iostream>
2 #include <string>
3
4 int main()
5 {
6     std::cout << "Enter your full name: ";
7     std::string myName;
8     std::getline(std::cin, myName); // полностью извлекаем строку в переменную myName
```

```
9
10 std::cout << "Enter your age: ";
11 std::string myAge;
12 std::getline(std::cin, myAge); // полностью извлекаем строку в переменную myAge
13
14 std::cout << "Your name is " << myName << " and your age is " << myAge;
15 }
```

Теперь работает как надо:

```
Enter your full name: Sasha Mak
Enter your age: 25
Your name is Sasha Mak and your age is 25
```

Использование std::getline() с std::cin

Извлечение данных из std::cin с помощью std::getline() иногда может приводить к неожиданным результатам. Например, рассмотрим следующую программу:

```
1 #include <iostream>
2 #include <string>
3
4 int main()
5 {
6     std::cout << "Pick 1 or 2: ";
7     int choice;
8     std::cin >> choice;
9
10    std::cout << "Now enter your name: ";
11    std::string myName;
12    std::getline(std::cin, myName);
13
14    std::cout << "Hello, " << myName << ", you picked " << choice << '\n';
15
16    return 0;
17 }
```

Возможно, вы удивитесь, но когда вы запустите эту программу, и она попросит вас ввести ваше имя, она не будет ожидать вашего ввода, а сразу выведет результат (просто пробел вместо вашего имени)!

Пробный запуск программы:

```
Pick 1 or 2: 2
Now enter your name: Hello, , you picked 2
```

Почему так? Оказывается, когда вы вводите числовое значение, поток cin захватывает вместе с вашим числом и символ новой строки. Поэтому, когда мы ввели 2, cin фактически получил 2\n. Затем он извлек значение 2 в переменную, оставляя \n (символ новой строки) во входном потоке. Затем, когда std::getline()

извлекает данные для myName, он видит в потоке \n и думает, что мы, должно быть, ввели просто пустую строку! А это определенно не то, что мы хотим.

Хорошей практикой является удалять из входного потока данных символ новой строки. Это можно сделать следующим образом:

```
1 | std::cin.ignore(32767, '\n'); // игнорируем символы перевода строки "\n" во входящем потоке
```

Если мы вставим эту строку непосредственно после получения входных данных, то символ новой строки будет удален из входного потока, и программа будет работать должным образом:

```
1 | #include <iostream>
2 | #include <string>
3 |
4 | int main()
5 | {
6 |     std::cout << "Pick 1 or 2: ";
7 |     int choice;
8 |     std::cin >> choice;
9 |
10 |    std::cin.ignore(32767, '\n'); // удаляем символ новой строки из входного потока данных
11 |
12 |    std::cout << "Now enter your name: ";
13 |    std::string myName;
14 |    std::getline(std::cin, myName);
15 |
16 |    std::cout << "Hello, " << myName << ", you picked " << choice << '\n';
17 |
18 |    return 0;
19 | }
```

Правило: При вводе числовых значений не забывайте удалять символ новой строки из входного потока данных с помощью std::cin.ignore().

Добавление строк

Вы можете использовать оператор + для объединения двух строк или оператор += для добавления одной строки к другой.

В следующей программе мы протестируем эти два оператора, а также покажем, что произойдет, если вы попытаетесь использовать оператор + для соединения двух числовых строк:

```
1 | #include <iostream>
2 | #include <string>
3 |
4 | int main()
5 | {
6 |     std::string x("44");
7 |     std::string y("12");
```

```
8
9     std::cout << x + y << "\n"; // объединяем строки x и y (а не складываем числа)
10    x += " cats";
11    std::cout << x;
12
13    return 0;
14 }
```

Результат выполнения программы:

```
4412
44 cats
```

Обратите внимание, оператор `+` объединил две числовые строки в одну ($44 + 12 = 4412$). Он не складывал эти строки как числа.

Длина строк

Чтобы узнать длину строки, мы можем сделать следующее:

```
1 #include <iostream>
2 #include <string>
3
4 int main()
5 {
6     std::string myName("Sasha");
7     std::cout << myName << " has " << myName.length() << " characters\n";
8     return 0;
9 }
10
```

Результат выполнения программы:

```
Sasha has 5 characters
```

Обратите внимание, вместо запроса длины строки как `length(myName)`, мы пишем `myName.length()`.

Функция запроса длины строки не является обычной функцией как те, которые мы использовали ранее. Это особый тип функции класса `std::string`, который называется методом. Мы поговорим об этом детально, когда будем рассматривать классы.

Тест

Напишите программу, которая просит у пользователя ввести его имя, фамилию и возраст. В результате, укажите пользователю, сколько лет он прожил на каждую букву из его имени и фамилии (чтобы было проще, учитывайте также пробелы). Например:

Enter your full name: Tom Cats

Enter your age: 45

You've lived 5.625 years for each letter in your name.

Уточнение: Возраст 45 делится на длину имени и фамилии «Tom Cats» (8 букв, учитывая пробел), что равно 5.625.

Ответ

```
1 #include <iostream>
2 #include <string>
3
4 int main()
5 {
6     std::cout << "Enter your full name: ";
7     std::string myName;
8     std::getline(std::cin, myName); // извлекаем целую строку из входного потока в переменную myName
9
10    std::cout << "Enter your age: ";
11    int myAge; // переменная myAge должна быть типа int, а не типа string, чтобы мы могли использовать оператор >>
12    std::cin >> myAge;
13
14    int letters = myName.length(); // вычисляем длину переменной myName (учитывая пробел)
15    double agePerLetter = static_cast<double>(myAge) / letters; // используем оператор static_cast
16    std::cout << "You've lived " << agePerLetter << " years for each letter in your name." << endl;
17
18    return 0;
19 }
```

Оценить статью:

★★★★★ (344 оценок, среднее: 4,88 из 5)



[← Урок №56. Явное преобразование типов данных](#)

[Урок №58. Перечисления](#)



Комментариев: 66



1. Влад:

[7 ноября 2020 в 08:35](#)

Вот автор пишет по поводу первой программы :

Оказывается, оператор извлечения (>>) возвращает символы из входного потока данных только до первого пробела. Все остальные символы остаются внутри cin, ожидая следующего извлечения.

По моему не совсем так :

символы входного потока можно ввести до нажатия Эндера или пробел (равноценно) и следующие символы (возраст) уже не вводятся

[Ответить](#)



2. *Андрей:*

[16 октября 2020 в 21:17](#)

Оцените код

```
1 #include <iostream>
2 #include <string>
3
4
5 int main() {
6     using std::string;
7     using std::cin;
8     using std::cout;
9     using std::getline;
10
11     cout << "Enter your full name:";
12     string name;
13     getline(cin, name);
14
15     cout << "Enter your Age:";
16     int age;
17     cin >> age;
18     double letters = static_cast<double>(name.length());
19
20     cout << "years for each letter in your name\n" << letters / age;
21
22
23     return 0;
24 }
```

[Ответить](#)



3. *Пётр:*

[10 октября 2020 в 23:44](#)

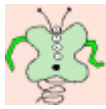
Вот такой вот код вышел.

Сначала написал, потом посмотрел ответ и понял, что несколько усложнил... Но зато функция main

очень наглядно читается 😊

```
1 #include <iostream>
2 #include <string>
3 #include <ios>
4 #include <limits>
5
6 std::string ask_for_fullname() {
7     std::string fullname;
8
9     std::cout << "Please type your full name: ";
10    std::getline(std::cin, fullname);
11    // std::cout << std::endl;
12
13    return fullname;
14 }
15
16 int ask_for_age() {
17     int age;
18
19     std::cout << "Please type your age: ";
20     std::cin >> age;
21     std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
22     // std::cout << std::endl;
23
24     return age;
25 }
26
27 double calculate_years_per_symbol(std::string fullname, int age) {
28     return static_cast<double>(age) / fullname.length();
29 }
30
31 int main() {
32     std::string fullname;
33     int age;
34     double years_per_symbol;
35
36     fullname = ask_for_fullname();
37     age = ask_for_age();
38     years_per_symbol = calculate_years_per_symbol(fullname, age);
39
40     std::cout << "You've lived " << years_per_symbol << " for each " <<
41         "symbol in your name." << std::endl;
42
43     return 0;
44 }
```

[ОТВЕТИТЬ](#)



4. *Каши́ровский:*
[14 августа 2020 в 00:16](#)

У нас есть карта

```
1  const int H = 14;
2  const int W = 195;
3
4  String TileMap[H] =
5
6  {"XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
7
8  ----- | | -----
9
10 "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX" }
```

Если делать длиннее чем 195 вылезает ошибка Out of range, как увеличить длину строки ? Может что-то в памяти как-то на это выделить ?

[Ответить](#)



5. *Руслан:*
[11 июля 2020 в 00:26](#)

Здравствуй! Что за магическое число 32767 используется в качестве параметра std::cin.ignore в данном уроке? Что лучше использовать с std::cin.ignore конкретный числовой литерал или же следующий стейтмент:

```
1  std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
```

?

[Ответить](#)



6. *Руслан:*
[11 июля 2020 в 00:22](#)

```
1  #include <iostream>
2  #include <string>
3
4  std::string input_name(std::string my_name) // функция ввода имени и фамилии
5  {
6      std::cout << "Enter your name and second name" << '\n';
7      std::getline(std::cin, my_name);
8      std::cout << '\a';
9      return my_name;
10 }
```

```
11
12 unsigned short input_age(unsigned short age) // функция ввода возраста
13 {
14     std::cout << "Enter your age" << '\n';
15     std::cin >> age;
16     std::cout << '\a';
17     std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); // на слу
18     return age;
19 }
20
21 int main()
22 {
23     std::string my_name {};
24     my_name = input_name(my_name);
25     unsigned short age {};
26     age = input_age(age);
27
28     std::cout << "Your full name:" << '\t' << my_name << '\n';
29     std::cout << "Your age:" << '\t' << age << '\n';
30     std::cout << "You've lived " << (static_cast<float>(age) / my_name.length());
31     std::cout << " years for each letter in your name" << std::endl;
32
33     return 0;
34 }
```

[Ответить](#)

7. Vadim:

[17 мая 2020 в 23:19](#)

Скажите, сколько может в себя вместить этот string из cin, ведь string объявлен на стеке, а с стек ограничен или я чего не понимаю? Как вообще разруливается эта проблема?

[Ответить](#)

1. Roman:

[8 июня 2020 в 13:15](#)

Со строками и выделением памяти под них и где именно эта память будет выделена не все тривиально. Как пишет Страуструп, выделение памяти на стеке работает для небольших строк, хотя понятие небольшой — относительное. Если у вас строка в 14 символов, то можно считать, что она не большая. Для длинных строк происходит выделение памяти в куче, тоже происходит если строка стала длинной. Но память снова может быть выделена на стеке, если строка стала достаточно короткой.

[Ответить](#)

8. *Богдан:*[19 марта 2020 в 00:42](#)

Скажите, какой размер имеет переменная типа string?

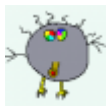
В этом коде результат 40. Это 40 байт?

```

1  #include<iostream>
2  #include<string>
3
4  int main()
5  {
6      std::string name = { " " };
7      std::cout << sizeof(name) << std::endl;
8
9      system("pause");
10     return 0;
11 }
```

[Ответить](#)1. *Roman:*[8 июня 2020 в 13:36](#)

Да, тут 40 байтов. Не очень понятно, что вы хотели получить: размер объекта в байтах или длину строки? Для длины строки нужно использовать функцию-член объекта string length(). А используя sizeof() вы можете получать разные результаты в зависимости от того, где выделена память под хранение массива символов: на стеке или в куче, и в зависимости от реализации самого объекта string. Есть ли вообще смысл использовать sizeof() для объекта string? Почитайте комментарий выше, про способы хранения символов в string. И да, у вас строка только из одного пробела, но памяти выделяется обычно больше, чтобы при изменении размера строки не приходилось постоянно перераспределять ее. Количество выделяемой памяти и в каком месте она выделяется отдано на откуп стандартной библиотеке и компилятору.

[Ответить](#)9. *Furxie Fluke:*[14 марта 2020 в 17:34](#)

Написал маленькую программку по заданию, и всё нормально, но, решил сюда скинуть свой код и спросить может кто знает о одной проблемке которая возникает

Если в переменной strangeNumber поменять тип данных double на int, происходит ошибка при запуске

То есть, да, int меньше double, и прочие вещи, переполнение там, но всё же это должно работать, или я что то не правильно понимаю?

```

1  int main()
```

```

2  {
3      SetConsoleOutputCP(1251);
4      SetConsoleCP(1251);
5
6
7      std::cout << "Введи своё имя тут - ";
8      std::string name{};
9      std::getline(std::cin, name);
10     std::cout << "А тут введи второе своё имя - ";
11     std::string secondName{};
12     std::getline(std::cin, secondName);
13     std::cout << "Ещё введи свой возраст - ";
14     short age{};
15     std::cin >> age;
16     std::cin.ignore(255, '\n');
17
18     double strangeNumber{ static_cast<double>(age) / (name + secondName).length() };
19     std::cout << "Молодец! Тебя зовут " << name << ", твоё второе имя " << secondName << " ";
20     std::cout << "Если разделить количество лет прожитых тобой на количество букв ";
21
22     return 0;
23 }

```

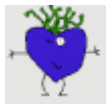
[Ответить](#)

1. Руслан:

[10 июля 2020 в 23:02](#)

если ещё актуально

error: narrowing conversion (компилятор g++ с включенными флагами) программа не собирается, аналогично с отключенными флагами программа собирается.

[Ответить](#)

10. Алексей:

[13 марта 2020 в 15:52](#)

```

1  #include <iostream>
2  #include <string>
3
4  int main()
5  {
6      std::cout << "Enter your full name ";
7      std::string myName;
8      std::getline(std::cin, myName); // Извлекаем целую строку вместе с пробелом
9      int x = myName.length();       // Считаем количество букв в имени с пробелом
10     std::cout << "Enter your age ";

```

```
11     int y;  
12     std::cin >> y;  
13  
14     std::cout << "You've lived " <<(float) y/x << " years for each letter in your  
15     return 0;  
16 }  
17 // Спасибо большое за Ваши Уроки! ;)
```

[Ответить](#)11. *mloborev:*[25 февраля 2020 в 17:03](#)

Сначала подумал, что сделал лишь хуже, вводя имя и фамилию по отдельности, зато так можно посчитать результат без пробела гораздо проще.

```
1  #include <iostream>  
2  #include <string>  
3  
4  int main()  
5  {  
6      std::string name, surname;  
7      double age;  
8  
9      std::cout << "Enter your name: ";  
10     std::getline(std::cin, name);  
11  
12     std::cout << "Enter your surname: ";  
13     std::getline(std::cin, surname);  
14  
15     std::cout << "Enter your age: ";  
16     std::cin >> age;  
17  
18     name += " " + surname;  
19  
20     std::cout << "You've lived " << age / name.length() << " years for each letter  
21  
22     return 0;  
23 }
```

[Ответить](#)12. *Inviser666:*[15 января 2020 в 15:52](#)

ну вроде так

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <string>
4
5 int main() {
6     std::string name;
7     double age;
8
9     std::cout << "Enter your full name: ";
10    std::getline(std::cin, name);
11
12    std::cout << "Enter your age: ";
13    std::cin >> age;
14
15    std::cout << "You've lived " << age/(name.length()) << " years for each letter";
16
17    system("pause");
18 }
```

[Ответить](#)13. *winnie:*[14 января 2020 в 21:32](#)

Столкнулся с проблемой вывода кириллических символов. Ранее во всех примерах использовал `setlocale(LC_ALL, "ru")` и вывод работал корректно. При использовании `getline ()` то, что вводится в переменную типа `string`, потом выводится в кривой кодировке. При использовании

```
1 SetConsoleCP(1251);
2 SetConsoleOutputCP(1251);
```

обратная ситуация — вывод идет некорректный, а то что на русском вводилось в `string` переменную потом отображается корректно.

Подскажите, пожалуйста, куда копать и как это победить?

Спасибо.

[Ответить](#)1. *Константин:*[27 мая 2020 в 00:27](#)

`#include <Windows.h>` легитимизирует все эти `SetConsole`-и

[Ответить](#)14. *armus1:*[13 октября 2019 в 17:42](#)

```
1 #include <iostream>
2 #include <string>
3
4 int main()
5 {
6     std::cout << "Enter your full name:";
7     std::string myName;
8     std::getline(std::cin, myName);
9
10    std::cin.ignore(32767, '\n');
11
12    std::cout << "Enter your age:";
13    double myAge;
14    std::cin >> myAge;
15
16    std::cout << "You've lived" << myAge/(myName.length()) << "years for each let
17
18    return 0;
19 }
```

Уважаемый Юрий, скажите пожалуйста, допустимо ли такое решение?

[Ответить](#)



1. *Артур:*

[24 декабря 2019 в 23:05](#)

Можете подсказать, что делать если мне вместо результата выдаёт inf в тесте?
Выдаёт вот это: You have lived inf years for each letter in your name

[Ответить](#)



1. *PraiseTheSun:*

[15 января 2020 в 20:57](#)

тип double может выдавать значение бесконечности при делении на 0

[Ответить](#)



15. *Alex:*

[25 сентября 2019 в 11:37](#)

Точнее использовать

```
1 std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
```

[Ответить](#)



16. *Игорь:*

[12 сентября 2019 в 11:31](#)

```
1 #include "pch.h"
2 #include <iostream>
3 #include <string>
4 using namespace std;
5
6 int main()
7 {
8     cin.ignore(32767, '\n');
9     cout << "You name:";
10    string name;
11    std::getline(std::cin, name);
12    cout << "You age:";
13    int age;
14    cin >> age;
15    int x;
16    x = name.length();
17    cout << "You've lived " << age / static_cast<double>(x) << " years for each l
18    return 0;
19 }
```

[Ответить](#)



17. *Алексей:*

[11 сентября 2019 в 19:55](#)

```
1 #include <bits/stdc++.h>
2 #include <string>
3 using namespace std;
4
5 string vvodFii()
6 {
7     cout << "Введите Ваше имя и фамилию: ";
8     string fii;
9     getline(cin, fii);
10    return fii;
11 }
12 int vvodVozrast()
13 {
14     cout << "Введите Ваш возраст в годах: ";
15     int vozrast;
16     cin >> vozrast;
17     return vozrast;
18 }
```

```
19 double raschet(string fii, int vozrast)
20 {
21     int x = fii.length()-1;
22     // в расчете убрали пробел между именем и фамилией
23     double y = double(vozrast) / double(x);
24     return y;
25 }
26
27 int main()
28 {
29     setlocale(LC_ALL, "rus");
30
31     string fii = vvodFii();
32     int vozrast = vvodVozrast();
33     double y = raschet(fii, vozrast);
34     cout << "Ваш возраст составляет: " << vozrast << endl;
35     cout << "Количество букв Вашего имени и фамилии составляе: " << fii.length()
36     cout << "Вы прожили: " << y << " лет на каждую букву Вашего имени и фамилии"
37
38     return 0;
39 }
```

Ответить



18. Markiyan:

[26 июля 2019 в 11:55](#)

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5
6
7 string enterName() {
8     cout << "Enter your full name: ";
9     string fullName;
10    getline(cin, fullName);
11    return fullName;
12 }
13
14
15 int countLength() {
16     string name = enterName();
17     int nameLength = name.length();
18     return nameLength;
19 }
20
21
```

```
22 int Age() {
23     cout << "Enter your age: ";
24     int age;
25     cin >> age;
26     return age;
27 }
28
29 void Count() {
30     int nameLength = countLength();
31     int age = Age();
32     cout << "You've lived " << float(age) / float(nameLength) << " years for each
33 }
34
35 int main()
36 {
37
38     Count();
39 }
```

[ОТВЕТИТЬ](#)

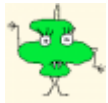


19. *Steve Dekart:*

[12 июля 2019 в 20:01](#)

```
1 #include "stdafx.h"
2 #include <iostream>
3 #include <string>
4 #include <iomanip>
5
6
7 double WorkingOb(std::string myName, int age)
8 {
9     double nameLetter = myName.length();
10    double age2 = age;
11
12
13    return age2 / nameLetter;
14 }
15
16 int main()
17 {
18     std::cout << "Enter your full name: " << std::endl;
19     std::string myName;
20     std::getline (std::cin, myName);
21
22     std::cout << "Enter your age: " << std::endl;
23     int age;
24     std::cin >> age;
```

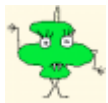
```
25
26     double agelived = WorkingOb(myName, age);
27
28     std::cout << std::setprecision(4);
29     std::cout << "You've lived " << agelived << "years for each letter in your name";
30
31     return 0;
32 }
```

[Ответить](#)

20. Алексей:

[11 июля 2019 в 17:59](#)

Каюсь, подглядываю как описать строку и прочее)
С опытом придет)

[Ответить](#)

21. Алексей:

[11 июля 2019 в 17:57](#)

Эм, написал за 4 минуты билеберду малек)
Работает, но видимо не понял, что именно просили в задании)

```
1  #include <iostream>
2  #include <string>
3
4  int main()
5  {
6
7      std::cout << "Enter name, last name:";
8      std::string nameLN;
9      std::getline(std::cin, nameLN);
10
11     std::cout << "Enter your age:";
12     int age;
13     std::cin >> age;
14
15     std::cout << static_cast<double>(age) / nameLN.length() << std::endl;
16 }
```

[Ответить](#)

22. havvk01:

[22 апреля 2019 в 07:23](#)

```

1 #include "pch.h"
2 #include <iostream>
3 #include <string>
4
5 int main()
6 {
7     std::cout << "Enter your full name: ";
8     std::string myName;
9     std::getline(std::cin, myName);
10
11     std::cout << "Enter your age: ";
12     int myAge;
13     std::cin >> myAge;
14
15     std::cout << "You've lived " << static_cast<float>(myAge)/myName.length() <<
16     system("pause");
17 }

```

[Ответить](#)



23. *Илья:*

[3 марта 2019 в 23:02](#)

Здравствуйте!

У меня три вопроса:

- 1) Почему Вы в своём ответе на тестовый вопрос не используете стейтмент `cin.ignore (32767, '\n');` ?
- 2) Что это вообще за магическое число такое 32767? откуда оно взялось? Почему не использовать меньшее, например 100 (врядли в мире есть имя и фамилия длиной больше ста символов. Ну если есть, то использовать 200, 300, 500 в конце концов=) ну не 32 тысячи же=)
- 3) Почему у меня программа после получения имени и фамилии после первого нажатия Enter просто переходит на следующую строку в консоли и только после повторного нажатия Enter выводит запрос возраста. Заранее большое спасибо за помощь!

Вот программа:

```

1 #include "pch.h"
2 #include <iostream>
3 #include <string>           // Для работы с переменными типа string
4 #include <iomanip>          // Для работы функции setprecision
5 using namespace std;
6
7 int main()
8 {
9     cout << "Enter your name and surname: ";
10     string myName;           // Вводим переменную типа string
11     getline(cin, myName);    // используем команду getline, что
12     cin.ignore(5, '\n');     // Удаляем из потока входных данных
13
14     cout << "Enter your age: ";

```

```

15     int x;
16     cin >> x;
17
18     int y = myName.length();           // Присваиваем переменной y значение
19
20
21
22     cout << "Hello, " << myName << "! Do you no, that you have lived " << setprec
23     return 0;
24 }
25
26
27 // Для справки (вдруг забыл=) ):
28 // setprecision(3) - количество символов, которое необходимо вывести на экран
29 // static_cast<double>(x) - преобразование типа данных переменной x из int в
30 // Переменная y преобразуется в double автоматически в следствие преобразовани

```

Ответить

1. Константин:

[12 марта 2019 в 06:37](#)

32767 — это (как я догадываюсь) число символов, которые можно разместить в одной строке; в конце компилятор сам (втихаря) вставляет \n

Ответить

2. Cerberus:

[27 марта 2019 в 13:45](#)

"Почему у меня программа после получения имени и фамилии после первого нажатия Enter просто переходит на следующую строку в консоли и только после повторного нажатия Enter выводит запрос возраста"

std::getline забирает строку целиком, вместе с символом перевода строки. А после этого Вы говорите, что cin должен проигнорировать следующий перевод строки, прежде чем что-то делать дальше. Вот он и ждал, пока этот самый перевод строки появится в потоке ввода — то есть, пока Вы не нажмёте Enter.

Ответить

24. Денис:

[27 февраля 2019 в 04:35](#)

По поводу getline разъяснил Георгий в комментариях (3 августа 2018), спасибо
Чуть-чуть в ступоре был, почему string русский выводил в виде символов, ведь setlocal был прописан, оказалось всё просто:) Ещё скрыл символы после точки double при выводе результата,

может кому-то будет полезно (если float писать вместо "%.2lf", нужно "%.2f", где 2 (кол-во символов после точки), но результат, как я понял, округляется. Будьте осторожней при необходимости вывести точный результат (или проверьте мои слова, я не уверен). И да, я решил без функций, с ними дольше, а программа маленькая.

```
1 #include "stdafx.h"
2 #include "windows.h"
3 #include <string>
4 #include <iostream>
5
6 int main()
7 {
8     SetConsoleCP(1251);
9     SetConsoleOutputCP(1251); //Подключаем русский в консоль (в частности для string,
10 using std::cout;
11 using std::cin;
12 using std::endl;
13 using std::string;
14 using std::getline;
15 setlocale(0, " ");
16
17 cout << "Введите ваше имя: ";
18 string name;
19 getline(cin, name); //Ввод имени пользователя.
20
21 cout << "Введите вашу фамилию: ";
22 string surname;
23 getline(cin, surname); //Ввод фамилии пользователя.
24
25 cout << "Введите ваш возраст: ";
26 int old;
27 cin >> old; // Ввод возраста пользователя.
28 std::cin.ignore(32767, '\n'); //очистка пробела после cin. (для образца, пробела не
29
30 cout << "Ваше имя: " << name << " " << surname << ", Возраст: " << old << endl; //
31
32 int x = name.length() + surname.length(); //Вычисляю кол-во символов в string name
33 double oldname = static_cast <double> (old) / (x); //Вычисляю количество лет на к
34
35 printf("На каждую букву вашего имени приходится %.1lf", oldname); // вывод лет. А
36 cout << " лет вашей жизни\n";
37 system("pause");
38 return 0;
39
40 }
```

[Ответить](#)



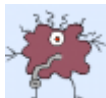
25. **Вячеслав:**

[23 февраля 2019 в 00:08](#)

у меня так получилось:

```
1 #include "pch.h"
2 #include <iostream>
3 #include <string> //подключаем директиву для string и функции getline()
4
5
6 int main()
7 {
8     using namespace std; //объявляем локальную область видимости namespace std
9     cout << "Enter your full name: "; //выводим в консоль запрос полного имени
10    string myName; //объявляем переменную типа string
11    getline(cin, myName); //функция считывания полной строки в переменную myName
12
13    cout << "Enter your age: "; //выводим в консоль запрос возраста
14    int myAge; //целочисленная переменная
15    cin >> myAge; //вводим значение переменной с консоли в переменную myAge
16    cin.ignore(32767, '\n'); //отключаем перенос строки
17    //объявляем переменную типа с плавающей точкой и преобразуем переменную myAge из
18    double myVozrast = static_cast<double>(myAge) / myName.length();
19    //выводим в консоль результаты деления возраста на количество букв полного имени
20    cout << "You've lived " << myVozrast << "years for each letter in your name.\n";
21
22    return 0;
23 }
```

[Ответить](#)



26. **Игорь:**

[20 февраля 2019 в 23:29](#)

У меня как-то размазано получилось по сравнению с ответом)))

```
1 #include <stdafx.h>
2 #include <iostream>
3 #include <string>
4 #include <locale.h>
5
6 using namespace std;
7
8 string enterFullName()
9 {
10     cout << "Введи Ваше имя и фамилию: ";
```



```
11     string fullName;
12     getline(cin, fullName);
13     return fullName;
14 }
15
16 int enterAge()
17 {
18     cout<<"Введите Ваш возраст: ";
19     int age;
20     cin>>age;
21     return age;
22 }
23
24 void printResult(double age,double x)
25 {
26     cout<<"В Вашем имени "<<x<<" букв. Вы прожили по "<<age/x<<" лет на каждую бу
27 }
28
29 int main()
30 {
31     setlocale(LC_ALL, "rus");
32
33     string fullName = enterFullName();
34     int age = enterAge();
35     int x = fullName.length();
36     printResult(age,x);
37
38     system("pause");
39     return 0;
40 }
```

[Ответить](#)

27. Alexey:

[4 февраля 2019 в 18:14](#)

Подскажите, насколько код написан красиво, вроде старался следовать всем рекомендациям.

```
1 #include "pch.h"
2 #include <iostream>
3 #include <windows.h>
4 #include <string>
5
6 int inputAge() {
7     using namespace std;
8     cout << "Введите ваш возраст: " << "\n";
9     int age{};
10    cin >> age;
```

```
11     cin.ignore(32767, '\n');
12     return age;
13 }
14
15 std::string inputName() {
16     using namespace std;
17     cout << "Введите ваше имя: " << "\n";
18     string name{};
19     getline(cin, name);
20     return name;
21 }
22
23 void agePerNamePrint() {
24     using namespace std;
25     string name = inputName();
26     double lenghtName = name.length() * 1.00;
27     cout << "Вы прожили " << inputAge()/lenghtName << " лет, на каждую букву свое
28 }
29
30 int main() {
31     using namespace std;
32     SetConsoleCP(1251);
33     SetConsoleOutputCP(1251);
34     agePerNamePrint();
35     return 0;
36 }
```

[Ответить](#)



28. *Алексей:*

[17 декабря 2018 в 21:19](#)

```
1 #include <iostream>
2 #include <string>
3
4 int main()
5 {
6     std::cout<<"Enter your full name: ";
7     std::string myName;
8     std::getline(std::cin, myName);
9
10    std::cout<<"Enter your age: ";
11    int myAge=0;
12    std::cin>>myAge;
13    std::cin.ignore(32767, '\n');
14
15    std::cout<<"Your name is "<<myName<<" and your age is "<<myAge<<std::endl;
16    long y = static_cast<long>(myAge)/myName.length();
```

```

17     std::cout<<"You've lived "<<y<<" years for each letter in your name.";
18     return 0;
19 }

```

[Ответить](#)


29. *somebox:*

[17 ноября 2018 в 22:36](#)

Что-то у меня не получается. Ввожу имя "Саша", возраст — "12" и получаю "1,5" на выходе, а должен же 4. Что не так? Вот код:

```

1  #include <stdio.h>
2  #include <iostream>
3  #include <string>
4
5  int main() {
6      std::cout << "Ведите ваше имя: ";
7      std::string myName;
8      std::getline(std::cin, myName);
9
10     std::cout << "Ведите ваш возраст: ";
11     int myAge(0);
12     std::cin >> myAge;
13     std::cin.ignore(32767, '\n');
14
15     double myYears = static_cast<double>(myAge) / myName.length();
16
17     std::cout << myName << ", вы прожили " << myYears << " года за каждую букву в
18
19     return 0;
20 }

```

[Ответить](#)


1. *Владимир:*

[29 ноября 2018 в 11:01](#)

Скопировал программу, у меня всё нормально выдаёт, но выдать он должен 3, а не 4 (12 / 4 = 3)

[Ответить](#)

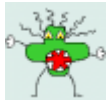

30. *Giveyn:*

[5 сентября 2018 в 22:31](#)

А почему сразу myAge не присвоить double, и обойтись без промежуточного agePerLetter, а деление выполнить в строке вывода?

```
1 #include <string>
2 #include <iostream>
3
4 int main()
5 {
6
7     std::cout << "Enter your full name: ";
8     std::string yourName;
9     std::getline(std::cin, yourName);
10
11     std::cout << "Enter your age:";
12     double choice;
13     std::cin >> choice;
14     std::cin.ignore(32767, '\n'); // удаляем символ новой строки из потока
15
16     int yourLife;
17     yourLife = yourName.length();
18
19     std::cout << "You've lived " << choice/ yourLife << "years for each letter in
20
21     system("pause");
22
23     return 0;
24 }
```

[Ответить](#)

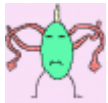


31. Ярослав:

[7 августа 2018 в 21:51](#)

```
1 #include "stdafx.h"
2 #include <iostream>
3 #include <string>
4
5 std::string getName()
6 {
7     std::cout << "Enter your full name: ";
8     std::string name;
9     std::getline(std::cin, name);
10     return name;
11 }
12
13 int getAge()
14 {
15     std::cout << "Enter your age: ";
```

```
16     int age;
17     std::cin >> age;
18     return age;
19 }
20
21 double calcYearsPerLetter(int age, std::string name)
22 {
23     double yearsPerLetter = static_cast<double>(age) / name.length();
24     return yearsPerLetter;
25 }
26
27 int main()
28 {
29     std::string name = getName();
30     int age = getAge();
31     double yearsPerLetter = calcYearsPerLetter(age, name);
32
33     std::cout << "You have lived " << yearsPerLetter << " years for each letter in " << name << "\n";
34
35     return 0;
36 }
```

[Ответить](#)32. *Katerina:*[2 августа 2018 в 05:39](#)

Здравствуйте, подскажите, пожалуйста, почему в приведенной функции выскакивает предупреждение? Я так понимаю linker не доволен... но программа компилируется и работает правильно вроде. Заранее благодарна.

```
1  #include <iostream>
2  #include <string>
3
4  int getNameLength()
5  {
6      std::cout << "Enter your full name: ";
7      std::string name;
8      std::getline(std::cin, name);
9      std::cout << name.length() << std::endl;
10     int nameLength = name.length();
11     return nameLength;
12 }
```

[Ответить](#)1. *Георгий:*

[3 августа 2018 в 00:09](#)

Функция "std::getline" получает от пользователя не цифру, а строку, так что число которое ты введёшь, будет просто строкой, а не числом (звучит странно), при этом ты пытаешься в тип данных int "впихнуть" строчку. Проще говоря, это тоже самое, что написать:

```
1 std::string name("Katerina");  
2 int x = name
```

Надеюсь, я объяснил понятно 😊

[Ответить](#)



1. *Георгий:*

[3 августа 2018 в 00:22](#)

А нет, я пересмотрел программу, и не знаю в чём проблема 😊

[Ответить](#)

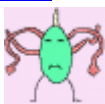


2. *Torgu:*

[3 августа 2018 в 10:25](#)

было бы неплохо увидеть само предупреждение

[Ответить](#)



1. *Katerina:*

[4 августа 2018 в 22:01](#)

Добрый вечер! Спасибо за feedback.

Ниже код и предупреждение:

Файл io.h

```
1 #ifndef IO_H  
2 #define IO_H  
3  
4 int getNameLength();  
5 int getAge();  
6 double calculation(double x, double y);  
7  
8 #endif // IO_H
```

Файл io.cpp

```
1 #include <iostream>  
2 #include <string>  
3
```

```
4 // -----
5 int getNameLength()
6 {
7     std::cout << "Enter your full name: ";
8     std::string name;
9     std::getline(std::cin, name);
10    std::cout << name.length() << std::endl;
11    int nameLength = name.length();
12    return nameLength;
13 }
14
15 // -----
16 int getAge()
17 {
18     std::cout << "Enter your age: ";
19     int age;
20     std::cin >> age;
21     return age;
22 }
23
24 // -----
25 double calculation(double x, double y) {
26     return x / y;
27 }
```

Файл main.cpp

```
1 #include <iostream>
2 #include "io.h"
3
4 int main()
5 {
6     double y = getNameLength();
7     double x = getAge();
8     std::cout << "You've lived " << calculation(x, y) << " years for each year." << std::endl;
9     return 0;
10 }
```

Предупреждение в файле io.cpp в строке `int nameLength = name.length();`

Implicit conversion loses integer precision: 'std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> >::size_type' to 'int'

Названия файлов просто лень переименовывать)))
Спасибо!

[Ответить](#)



1. *Get Leevz:*

[5 августа 2018 в 21:20](#)

Метод `length()` возвращает тип `unsigned int`, который вы присваиваете переменной `signed int`. Из-за этого происходит неявное преобразование с возможностью потери точности. Это можно решить с помощью изначального типа переменной `unsigned int`, но я бы просто забил, ведь чтобы произошло переполнение, пользователь должен ввести имя длиною более чем в 2 147 483 647 символов, а это...хम्म, как бы немного многовато



2. *Katerina:*

[7 августа 2018 в 00:52](#)

Get Leevz, почему-то не могу ответить на ваш комментарий, поэтому отвечаю на свой)) Спасибо вам большое за разъяснения, сама бы до этого не дошла!



33. *Sergey Groysman:*

[18 апреля 2018 в 19:40](#)

Юрий, доброго дня. Подскажите, в чём разница в данных записях, если работают обе, переводя число в `double` и мы получаем на выходе число с плавающей точкой.

```
1 double agePerLetter;
2 agePerLetter = (double)myAge / letters;
```

вместо

```
1 double agePerLetter = static_cast<double>(myAge) / letters;
```

можно ли обходиться в принципе без `static_cast`?
Спасибо.

[Ответить](#)



1. *Юрий:*

[21 апреля 2018 в 13:47](#)

Привет. Ответ на ваш вопрос находится в [уроке 56](#).

[Ответить](#)



1. *Torgu:*

[3 июля 2018 в 14:13](#)

А обязательно переводить вручную? Я просто дал переменной возраста тип float. Компилятор же сам преобразовывает типы

[Ответить](#)



34. *Серж:*

[7 марта 2018 в 09:05](#)

Перечитал урок 7.

Проблема 1

Как использовать русский язык в консоли C++?

Слово в слово повторено как в программе выше и где сказано что всё работает.

У меня в CodeBlocks не работает. Возможно в студии будет работать , не проверял.

Остается в уроке 7 прочитать совет в самом низу: Во-первых, спросите Google...

[Ответить](#)



1. *Юрий:*

[7 марта 2018 в 11:42](#)

Верно, если Google не поможет, спросите на StackOverflow. Код рабочий, если у вас не работает, значит у вас уже специфическая проблема: возможно кодировка в Code:Blocks у вас не та, что нужно. Возможно, у вас что-то подключено (плагин, расширение или что-либо другое), что блокирует корректное выполнение setlocale.

[Ответить](#)



1. *Серж:*

[7 марта 2018 в 11:50](#)

Ясно, спасибо. Подключена русификация интерфейса и все. Пока не искал решения проблемы, т.к. это не сильно влияет на изучение.

Огромное спасибо за такой нужный труд по переводу и поддержанию сайта с этим учебным курсом. Из всего что я находил ,это самое лучшее. Спасибо!!!

[Ответить](#)



1. *Юрий:*

[7 марта 2018 в 12:03](#)

Пожалуйста 😊

Попробуйте отключить русификацию интерфейса и посмотрите какая у вас кодировка стоит — сбросьте всё до значений по умолчанию. И если уже никак, то вы знаете, где искать.



35. Максим:

[6 марта 2018 в 04:29](#)

```
1 #include "stdafx.h"
2 #include <iostream>
3 #include <string>
4 double ageForChar(std::string fullName, int age) {
5     int nameLength = fullName.length();
6     return static_cast<double>(age) / static_cast<double>(nameLength);
7 }
8 void printResult(std::string fullName, int age) {
9     std::cout << fullName << " " << age << " years old" << std::endl;
10    std::cout << "You're lived " << ageForChar(fullName, age) << " year for every
11 }
12 int main()
13 {
14     std::cout << "Enter your full name: ";
15     std::string fullName;
16     std::getline(std::cin, fullName);
17
18     std::cout << "Enter your age: ";
19     int age;
20     std::cin >> age;
21     std::cin.ignore(32767, '\n');
22
23     printResult(fullName, age);
24
25     system("pause");
26     return 0;
27 }
```

[ОТВЕТИТЬ](#)

36. Ka4:

[24 февраля 2018 в 20:10](#)

```
1 #include "stdafx.h"
2 #include <iostream>
3 #include <string>
4 #include <Windows.h>
5
6 using namespace std;
7
8 int im9_famili9()
9 {
10
```

```
11     cout << "Введите Имя и Фамилию: ";
12     string name;
13     getline(cin, name);
14     int kolvo = name.length();
15     return kolvo;
16 }
17
18 int vozrast()
19 {
20     cout << "Введите ваш возраст: ";
21     int a;
22     cin >> a;
23     return a;
24 }
25
26 int result(int v, int t)
27 {
28     cout << "С каждой вашей буквой вы прожили: " << static_cast<double>(v) / stat
29     return v/t;
30 }
31
32
33 int main()
34 {
35     SetConsoleCP(1251);
36     SetConsoleOutputCP(1251);
37
38     int t = im9_famili9();
39     int v = vozrast();
40
41     double z = result(v, t);
42
43     system("pause");
44     return 0;
45 }
```

Скажите, пожалуйста, что я не так сделал!

[Ответить](#)

1.  Юрий:

[25 февраля 2018 в 18:43](#)

Ваш код рабочий. Результаты совпадают.

Что не так:

1. У вас функция `im9_famili9` выполняет сразу два задания: получает строку и высчитывает её длину. (Функция должна выполнять одно задание)
2. Функции `result` нужно было присвоить тип `void` и ничего не возвращать, а сразу выводить результат в консоль. А то у вас `result` выводит результат и возвращает его еще в `main()` — зачем и почему? В `main` вы с переменной `z` ничего не делаете, её объявление вообще не нужно.

Всё остальное гуд.

[Ответить](#)



1. *Ка4:*

[25 февраля 2018 в 21:55](#)

Точно..

Спасибо больше!)

Отличные уроки!!!

Догоняю программу в ВУЗе с помощью этого сайта!

[Ответить](#)



1. *Юрий:*

[26 февраля 2018 в 00:42](#)

Если честно, то я тоже свою программу из универа догнал и перегнал именно по этим урокам 😊



37. *Герман:*

[26 января 2018 в 11:59](#)

Спасибо!!!

[Ответить](#)



1. *Юрий:*

[26 января 2018 в 12:55](#)

Пожалуйста 😊

[Ответить](#)



38. *Герман:*

[25 января 2018 в 13:32](#)

```
1 | int main()  
2 | {
```

```
3  setlocale(0, "");
4  using namespace std;
5
6  cout << "Введите фамилию и имя: ";
7  string myName;
8  getline(cin, myName);
9
10 int a = myName.length();
11
12 cout << "Введите возраст: ";
13 int myAge;
14 cin >> myAge;
15 cin.ignore(1, '\n');
16
17 double z = static_cast<double>(myAge) / static_cast<double>(a);
18
19 cout << "Фамилия и имя: " << myName << ", Возраст: " << myAge << ", Результат
20
21
22 return 0;
23 }
```

Уважаемый автор, почему при выводе в консоль значение переменной myName, не поддерживается кириллица!

Ответить

1.  Юрий:

[26 января 2018 в 00:40](#)

Нужно подключить Windows.h (заголовочный файл) и прописать строчки:

```
1  SetConsoleCP(1251);
2  SetConsoleOutputCP(1251);
```

Вот всё работает:

```
1  #include "stdafx.h"
2  #include <iostream>
3  #include <string>
4  #include <Windows.h>
5
6  int main()
7  {
8      SetConsoleCP(1251);
9      SetConsoleOutputCP(1251);
10
11     using namespace std;
```

```
12
13     cout << "Введите фамилию и имя: ";
14     string myName;
15     getline(cin, myName);
16
17     int a = myName.length();
18
19     cout << "Введите возраст: ";
20     int myAge;
21     cin >> myAge;
22     cin.ignore(1, '\n');
23
24     double z = static_cast<double>(myAge) / static_cast<double>(a);
25
26     cout << "Фамилия и имя: " << myName << ", Возраст: " << myAge << ", Резу:
27
28     return 0;
29 }
```

[Ответить](#)



1. *Серж:*

[6 марта 2018 в 15:24](#)

Почему-то в CodeBlocks 17.12 выводит этим способом кракозябры как на ввод ,так и на вывод. Если использовать `setlocale(LC_ALL, "rus");` или `setlocale(LC_ALL, "");` вывод идет на русском , а на ввод кракозябры ,если использовать `SetConsoleCP(1251);` `SetConsoleOutputCP(1251);` то даже при подключенном `setlocale(LC_ALL, "rus");` все равно кракозябры.

[Ответить](#)



1. *Юрий:*

[6 марта 2018 в 21:23](#)

Ответ на ваш вопрос ищите в [Уроке 7. Самые распространенные проблемы и их решения в C++](#).



2. *GOGOLEK:*

[20 февраля 2019 в 07:04](#)

Разве `setlocale(LC_ALL, "rus")` уже не помогает?

[Ответить](#)

Добавить комментарий

Ваш E-mail не будет опубликован. Обязательные поля помечены *

Имя *

Email *

Комментарий






☐ Сохранить моё Имя и E-mail. Видеть комментарии, отправленные на модерацию

☐ Получать уведомления о новых комментариях по электронной почте. Вы можете [подписаться](#) без комментирования.

[TELEGRAM](#)  [КАНАЛ](#)

[ПАБЛИК](#) 

ТОП СТАТЬИ

-  [Словарь программиста. Сленг, который должен знать каждый кодер](#)
-  [Урок №1. Введение в программирование](#)
-  [70+ бесплатных ресурсов для изучения программирования](#)
-  [Урок №1: Введение в создание игры «SameGame» на C++/MFC](#)
-  [Урок №4. Установка IDE \(Интегрированной Среды Разработки\)](#)

- [Ravesli](#)
- - [О проекте/Контакты](#) -
- - [Пользовательское Соглашение](#) -
- - [Все статьи](#) -
- Copyright © 2015 - 2020