

Ravesli [Ravesli](#)


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


Урок №61. Структуры

 [Юрий](#) |

- [Уроки C++](#)

|

 Обновл. 14 Сен 2020 |

 113308

[↑](#)  37

В программировании есть много случаев, когда может понадобиться больше одной переменной для представления определенного объекта.

Оглавление:

1. [Зачем нужны структуры?](#)
2. [Объявление и определение структур](#)
3. [Доступ к членам структур](#)
4. [Инициализация структур](#)
5. [C++11/14: Инициализация нестатических членов структур](#)
6. [Присваивание значений членам структур](#)
7. [Структуры и функции](#)
8. [Вложенные структуры](#)
9. [Размер структур](#)
10. [Доступ к структурам из нескольких файлов](#)
11. [Заключение](#)
12. [Тест](#)

Зачем нужны структуры?

Например, для представления самого себя, вы, скорее всего, захотите указать свое имя, день рождения, рост, вес или любую другую информацию:

```
1 | std::string myName;  
2 | int myBirthDay;
```

```
3 | int myBirthMonth;  
4 | int myBirthYear;  
5 | int myHeight;  
6 | int myWeight;
```

Но теперь у вас есть 6 отдельных независимых переменных. Если вы захотите передать информацию о себе в функцию, то вам придется передавать каждую переменную по отдельности. Кроме того, если вы захотите хранить информацию о ком-то еще, то вам придется дополнительно объявить еще 6 переменных на каждого человека! Невооруженным глазом видно, что такая реализация не очень эффективна.

К счастью, язык C++ позволяет программистам создавать свои собственные **пользовательские типы данных** — типы, которые группируют несколько отдельных переменных вместе. Одним из простейших пользовательских типов данных является структура. **Структура позволяет сгруппировать переменные разных типов в единое целое.**

Объявление и определение структур

Поскольку структуры определяются программистом, то вначале мы должны сообщить компилятору, как она вообще будет выглядеть. Для этого используется **ключевое слово struct**:

```
1 | struct Employee  
2 | {  
3 |     short id;  
4 |     int age;  
5 |     double salary;  
6 | };
```

Мы определили структуру с именем Employee. Она содержит 3 переменные:

- id типа short;
- age типа int;
- salary типа double.

Эти переменные, которые являются частью структуры, называются **членами структуры** (или «*полями структуры*»). Employee — это простое объявление структуры. Хотя мы и указали компилятору, что она имеет переменные-члены, память под нее сейчас не выделяется. Имена структур принято писать с заглавной буквы, чтобы отличать их от имен переменных.

Предупреждение: Одна из самых простых ошибок в C++ — забыть точку с запятой в конце объявления структуры. Это приведет к ошибке компиляции в следующей строке кода. Современные компиляторы, такие как Visual Studio версии 2010, а также более новых версий, укажут вам, что вы забыли точку с запятой в конце, но более старые компиляторы могут этого и не сделать, из-за чего такую ошибку будет трудно найти. О том, как установить Visual Studio или какую выбрать IDE мы уже говорили на [уроке №4](#).

Чтобы использовать структуру Employee, нам нужно просто объявить переменную типа Employee:

```
1 | Employee john; // имя структуры Employee начинается с заглавной буквы, а переменная joh
```

Здесь мы определили переменную типа `Employee` с именем `john`. Как и в случае с обычными переменными, определение переменной, типом которой является структура, приведет к выделению памяти для этой переменной.

Объявить можно и несколько переменных одной структуры:

```
1 | Employee john; // создаем отдельную структуру Employee для John-a
2 | Employee james; // создаем отдельную структуру Employee для James-a
```

Доступ к членам структур

Когда мы объявляем переменную структуры, например, `Employee john`, то `john` ссылается на всю структуру. Для того, чтобы получить доступ к отдельным её членам, используется **оператор выбора члена** (`.`). Например, в коде, приведенном ниже, мы используем оператор выбора членов для инициализации каждого члена структуры:

```
1 | Employee john; // создаем отдельную структуру Employee для John-a
2 | john.id = 8; // присваиваем значение члену id структуры john
3 | john.age = 27; // присваиваем значение члену age структуры john
4 | john.salary = 32.17; // присваиваем значение члену salary структуры john
5 |
6 | Employee james; // создаем отдельную структуру Employee для James-a
7 | james.id = 9; // присваиваем значение члену id структуры james
8 | james.age = 30; // присваиваем значение члену age структуры james
9 | james.salary = 28.35; // присваиваем значение члену salary структуры james
```

Как и в случае с обычными переменными, переменные-члены структуры не инициализируются автоматически и обычно содержат мусор. Инициализировать их нужно вручную.

В примере, приведенном выше, легко определить, какая переменная относится к структуре `John`, а какая — к структуре `James`. Это обеспечивает гораздо более высокий уровень организации, чем в случае с обычными отдельными переменными.

Переменные-члены структуры работают так же, как и простые переменные, поэтому с ними можно выполнять обычные [арифметические операции](#) и [операции сравнения](#):

```
1 | int totalAge = john.age + james.age;
2 |
3 | if (john.salary > james.salary)
4 |     cout << "John makes more than James\n";
5 | else if (john.salary < james.salary)
6 |     cout << "John makes less than James\n";
7 | else
8 |     cout << "John and James make the same amount\n";
9 |
10 | // James получил повышение в должности
11 | james.salary += 3.75;
```

```
12 |  
13 | // Сегодня день рождения у John-а  
14 | ++john.age; // используем пре-инкремент для увеличения возраста John-а на 1 год
```

Инициализация структур

Инициализация структур путем присваивания значений каждому члену по порядку — занятие довольно громоздкое (особенно, если этих членов много), поэтому в языке C++ есть более быстрый способ инициализации структур — с помощью **списка инициализаторов**. Он позволяет инициализировать некоторые или все члены структуры во время объявления переменной типа struct:

```
1 struct Employee  
2 {  
3     short id;  
4     int age;  
5     double salary;  
6 };  
7  
8 Employee john = { 5, 27, 45000.0 }; // john.id = 5, john.age = 27, john.salary = 45000.0  
9 Employee james = { 6, 29 }; // james.id = 6, james.age = 29, james.salary = 0.0 (инициализация по умолчанию)
```

В C++11 также можно использовать [uniform-инициализацию](#):

```
1 Employee john { 5, 27, 45000.0 }; // john.id = 5, john.age = 27, john.salary = 45000.0  
2 Employee james { 6, 29 }; // james.id = 6, james.age = 29, james.salary = 0.0 (инициализация по умолчанию)
```

Если в списке инициализаторов не будет одного или нескольких элементов, то им присвоятся значения по умолчанию (обычно, 0). В примере, приведенном выше, члену `james.salary` присваивается значение по умолчанию `0.0`, так как мы сами не предоставили никакого значения во время инициализации.

C++11/14: Инициализация нестатических членов структур

В C++11 добавили возможность присваивать нестатическим (обычным) членам структуры значения по умолчанию. Например:

```
1 #include <iostream>  
2  
3 struct Triangle  
4 {  
5     double length = 2.0;  
6     double width = 2.0;  
7 };  
8  
9 int main()  
10 {
```

```
11 Triangle z; // длина = 2.0, ширина = 2.0
12
13 z.length = 3.0; // вы также можете присваивать членам структуры и другие значения
14
15 return 0;
16 }
```

К сожалению, в C++11 синтаксис инициализации нестатических членов структуры несовместим с синтаксисом списка инициализаторов или `uniform`-инициализацией. В C++11 следующая программа не скомпилируется:

```
1 #include <iostream>
2
3 struct Triangle
4 {
5     double length = 2.0; // нестатическая инициализация членов
6     double width = 2.0;
7 };
8
9 int main()
10 {
11     Triangle z{ 3.0, 3.0 }; // uniform-инициализация
12
13     return 0;
14 }
```

Следовательно, вам нужно будет решить, хотите ли вы использовать инициализацию нестатических полей или `uniform`-инициализацию. `uniform`-инициализация более гибкая, поэтому рекомендуется использовать именно её.

Однако в C++14 это ограничение было снято, и оба варианта можно использовать. Мы еще поговорим детально о статических членах структуры на соответствующем уроке.

Присваивание значений членам структур

До C++11, если бы мы захотели присвоить значения членам структуры, то нам бы пришлось это делать вручную каждому члену по отдельности:

```
1 struct Employee
2 {
3     short id;
4     int age;
5     double salary;
6 };
7
8 Employee john;
9 john.id = 5;
```

```
10 | john.age = 27;  
11 | john.salary = 45000.0;
```

Это боль, особенно когда членов в структуре много. В C++11 вы можете присваивать значения членам структур, используя список инициализаторов:

```
1 | struct Employee  
2 | {  
3 |     short id;  
4 |     int age;  
5 |     double salary;  
6 | };  
7 |  
8 | Employee john;  
9 | john = { 5, 27, 45000.0 }; // ТОЛЬКО C++11
```

Структуры и функции

Большим преимуществом использования структур, нежели отдельных переменных, является возможность передать всю структуру в функцию, которая должна работать с её членами:

```
1 | #include <iostream>  
2 |  
3 | struct Employee  
4 | {  
5 |     short id;  
6 |     int age;  
7 |     double salary;  
8 | };  
9 |  
10 | void printInformation(Employee employee)  
11 | {  
12 |     std::cout << "ID: " << employee.id << "\n";  
13 |     std::cout << "Age: " << employee.age << "\n";  
14 |     std::cout << "Salary: " << employee.salary << "\n";  
15 | }  
16 |  
17 | int main()  
18 | {  
19 |     Employee john = { 21, 27, 28.45 };  
20 |     Employee james = { 22, 29, 19.29 };  
21 |  
22 |     // Выводим информацию о John-е  
23 |     printInformation(john);  
24 |  
25 |     std::cout << "\n";  
26 |  
27 |     // Выводим информацию о James-е  
28 | }
```

```
29     printInformation(james);
30
31     return 0;
}
```

В примере, приведенном выше, мы передали структуру `Employee` в функцию `printInformation()`. Это позволило нам не передавать каждую переменную по отдельности. Более того, если мы когда-либо захотим добавить новых членов в структуру `Employee`, то нам не придется изменять объявление или вызов функции!

Результат выполнения программы:

```
ID: 21
Age: 27
Salary: 28.45
```

```
ID: 22
Age: 29
Salary: 19.29
```

Функция также может возвращать структуру (это один из тех немногих случаев, когда функция может возвращать несколько переменных). Например:

```
1  #include <iostream>
2
3  struct Point3d
4  {
5      double x;
6      double y;
7      double z;
8  };
9
10 Point3d getZeroPoint()
11 {
12     Point3d temp = { 0.0, 0.0, 0.0 };
13     return temp;
14 }
15
16 int main()
17 {
18     Point3d zero = getZeroPoint();
19
20     if (zero.x == 0.0 && zero.y == 0.0 && zero.z == 0.0)
21         std::cout << "The point is zero\n";
22     else
23         std::cout << "The point is not zero\n";
24
25     return 0;
26 }
```

Результат выполнения программы:

The point is zero

Вложенные структуры

Одни структуры могут содержать другие структуры. Например:

```
1 struct Employee
2 {
3     short id;
4     int age;
5     double salary;
6 };
7
8 struct Company
9 {
10     Employee CEO; // Employee - это структура внутри структуры Company
11     int numberOfEmployees;
12 };
13
14 Company myCompany;
```

В этом случае, если бы мы хотели узнать, какая зарплата у CEO (исполнительного директора), то нам бы пришлось использовать оператор выбора членов дважды:

```
1 myCompany.CEO.salary
```

Сначала мы выбираем поле CEO из структуры myCompany, а затем поле salary из структуры Employee.

Вы можете использовать вложенные списки инициализаторов с вложенными структурами:

```
1 struct Employee
2 {
3     short id;
4     int age;
5     float salary;
6 };
7
8 struct Company
9 {
10     Employee CEO; // Employee является структурой внутри структуры Company
11     int numberOfEmployees;
12 };
13
14 Company myCompany = {{ 3, 35, 55000.0f }, 7 };
```


Размер структур

Как правило, размер структуры — это сумма размеров всех её членов, но не всегда! Например, рассмотрим структуру `Employee`. На большинстве платформ [тип `short`](#) занимает 2 байта, тип `int` — 4 байта, а [тип `double`](#) — 8 байт. Следовательно, ожидается, что `Employee` будет занимать $2 + 4 + 8 = 14$ байт. Чтобы узнать точный размер `Employee`, мы можем воспользоваться [оператором `sizeof`](#):

```
1 #include <iostream>
2
3 struct Employee
4 {
5     short id;
6     int age;
7     double salary;
8 };
9
10 int main()
11 {
12     std::cout << "The size of Employee is " << sizeof(Employee) << "\n";
13
14     return 0;
15 }
```

Результат выполнения программы (на компьютере автора):

The size of Employee is 16

Оказывается, мы можем сказать только, что размер структуры будет, *по крайней мере*, не меньше суммы размеров всех её членов. Но он может быть и больше! По соображениям производительности компилятор иногда может добавлять «пробелы/промежутки» в структуры.

В структуре `Employee` компилятор неявно добавил 2 байта после члена `id`, увеличивая размер структуры до 16 байтов вместо 14. Описание причины, по которой это происходит, выходит за рамки этого урока, но если вы хотите знать больше, то можете прочитать о [выравнивании данных в Википедии](#).

Доступ к структурам из нескольких файлов

Поскольку объявление структуры не провоцирует выделение памяти, то использовать [предварительное объявление](#) для нее вы не сможете. Но есть обходной путь: если вы хотите использовать объявление структуры в нескольких файлах (чтобы иметь возможность создавать переменные этой структуры в нескольких файлах), поместите объявление структуры в [заголовочный файл](#) и подключайте этот файл везде, где необходимо использовать структуру.

Переменные типа `struct` подчиняются тем же правилам, что и обычные переменные. Следовательно, если вы хотите сделать переменную структуры доступной в нескольких файлах, то вы можете использовать [ключевое слово `extern`](#).

Заключение

Структуры очень важны в языке C++, поскольку их понимание — это первый большой шаг на пути к объектно-ориентированному программированию! Чуть позже мы рассмотрим другой пользовательский тип данных — класс (который является продолжением темы структур).

Тест

Задание №1

У вас есть веб-сайт и вы хотите отслеживать, сколько денег вы зарабатываете в день от размещенной на нем рекламы. Объявите структуру `Advertising`, которая будет отслеживать:

- ➔ сколько объявлений вы показали посетителям (1);
- ➔ сколько процентов посетителей нажали на объявления (2);
- ➔ сколько вы заработали в среднем за каждое нажатие на объявления (3).

Значения этих 3-х полей должен вводить пользователь. Передайте структуру `Advertising` в функцию, которая выведет каждое из этих значений, а затем подсчитает, сколько всего денег вы заработали за день (перемножьте все 3 поля).

Ответ №1

```
1  #include <iostream>
2
3  // Сначала объявляем структуру Advertising
4  struct Advertising
5  {
6      int adsShown;
7      double clickThroughRatePercentage;
8      double averageEarningsPerClick;
9  };
10
11 void printAdvertising(Advertising ad)
12 {
13     using namespace std;
14     cout << "Number of ads shown: " << ad.adsShown << endl;
15     cout << "Click through rate: " << ad.clickThroughRatePercentage << endl;
16     cout << "Average earnings per click: $" << ad.averageEarningsPerClick << endl;
17
18     // Следующая строка кода разбита из-за своей длины.
19     // Мы делим ad.clickThroughRatePercentage на 100, так как пользователь указывает п
20     cout << "Total Earnings: $" <<
21         (ad.adsShown * ad.clickThroughRatePercentage / 100 * ad.averageEarningsPerClick)
22 }
23
24 int main()
25 {
26     using namespace std;
```

```
27 // объявляем переменную структуры Advertising
28 Advertising ad;
29
30 cout << "How many ads were shown today? ";
31 cin >> ad.adsShown;
32 cout << "What percentage of users clicked on the ads? ";
33 cin >> ad.clickThroughRatePercentage;
34 cout << "What was the average earnings per click? ";
35 cin >> ad.averageEarningsPerClick;
36
37 printAdvertising(ad);
38
39 return 0;
40 }
```

Задание №2

Создайте структуру для хранения дробных чисел. Структура должна иметь 2 члена: целочисленный числитель и целочисленный знаменатель. Объявите две дробные переменные и получите их значения от пользователя. Напишите функцию `multiply()` (параметрами которой будут эти две переменные), которая будет перемножать эти числа и выводить результат в виде десятичного числа.

Ответ №2

```
1 #include <iostream>
2
3 struct Drob
4 {
5     int chislitel;
6     int znamenatel;
7 };
8
9 void multiply(Drob d1, Drob d2)
10 {
11     using namespace std;
12
13     // Не забываем об операторе static_cast, иначе компилятор выполнит целочисленное д
14     cout << static_cast<float>(d1.chislitel* d2.chislitel) /
15         (d1.znamenatel* d2.znamenatel);
16 }
17
18 int main()
19 {
20     using namespace std;
21
22     // Определяем первую переменную-дробь
23     Drob d1;
24     cout << "Input the first chislitel: ";
```

```
25     cin >> d1.chislitel;  
26     cout << "Input the first znamenatel: ";  
27     cin >> d1.znamenatel;  
28  
29     // Определяем вторую переменную-дробь  
30     Drob d2;  
31     cout << "Input the second chislitel: ";  
32     cin >> d2.chislitel;  
33     cout << "Input the second znamenatel: ";  
34     cin >> d2.znamenatel;  
35  
36     multiply(d1, d2);  
37  
38     return 0;  
39 }
```

Оценить статью:

★★★★★ (344 оценок, среднее: 4,90 из 5)



← [Урок №60. Псевдонимы типов: typedef и type alias](#)

[Урок №62. Вывод типов: ключевое слово auto](#) →



Комментариев: 37



1. [Юрий:](#)
[23 ноября 2020 в 15:25](#)

//задача 1 структуры

```
1  #include <iostream>  
2  using namespace std;  
3  struct Advertising  
4  {  
5      int ad_person;  
6      float push_persent;  
7      float averageProfit;  
8  };  
9  
10 void showAll(Advertising ad)  
11 {
```

```

12     cout << "Объявлений показано посетителям: " << ad.ad_person << endl;
13     cout << "Процен посетителей нажало на объявление: " << ad.push_persent << endl;
14     cout << "Заработано в среднем за каждое объявление: " << ad.averageProfit << endl;
15     cout << "Заработано в день: " << ad.averageProfit* ad.push_persent/100* ad.ad_person << endl;
16
17 }
18
19 int main()
20 {
21     setlocale(LC_ALL, "rus");
22     Advertising ad;
23     ad.ad_person=0;
24     ad.push_persent=0;
25     ad.averageProfit=0;
26
27     cout << "Объявлений показано посетителям: " << endl;
28     cin >> ad.ad_person;
29     cout << "Процен посетителей нажало на объявление: " << endl;
30     cin >> ad.push_persent;
31     cout << "Заработано в среднем за каждое объявление: " << endl;
32     cin >> ad.averageProfit;
33     showAll(ad);
34
35     return 0;
36 }

```

//задача 2 структуры

```

1  #include <iostream>
2
3  struct Digits
4  {
5      int numerator;
6      int denominator;
7  };
8  void multiply(Digits num1, Digits num2)
9  {
10     std::cout<<"Десятичная дробь:"<<(static_cast<float>(num1.numerator) / num1.denominator) << endl;
11 }
12 int main()
13 {
14     using namespace std;
15     setlocale(LC_ALL, "rus");
16     Digits num1;
17     Digits num2;
18
19     cout << "Число 1:\n";
20     cout << "Введите числитель: \n";
21     cin>> num1.numerator;

```

```
22     cout << "Введите знаменатель: \n";
23     cin >> num1.denominator;
24     cout << "Число 2:\n";
25     cout << "Введите числитель: \n";
26     cin >> num2.numerator;
27     cout << "Введите знаменатель: \n";
28     cin >> num2.denominator;
29
30     multiply(num1,num2);
31     return 0;
32 }
```

[Ответить](#)



2. Руслан:

[18 июля 2020 в 19:00](#)

Задание № 1

```
1  #include <iostream>
2
3  struct Advertising
4  {
5      int ads;
6      double percent;
7      double profit;
8  };
9
10 Advertising input()
11 {
12     std::cout << "Enter enter the number of: " << '\n';
13     std::cout << "ads have you shown to visitors" << '\n';
14     std::cout << "percentage of visitors clicked on ads" << '\n';
15     std::cout << "average revenue per click on ads" << '\n';
16     int ads;
17     double percent;
18     double profit;
19
20     std::cin >> ads >> percent >> profit;
21     Advertising temp {ads, percent, profit};
22     std::cout << '\a';
23
24     return temp;
25 }
26
27 void print(Advertising income)
28 {
29     std::cout << "How many ads have you shown to visitors" << '\t' << income.ads
```

```

30     std::cout << "What percentage of visitors clicked on ads" << '\t' << income.p
31     std::cout << "How much you earned on average per click on ads" << '\t' << inc
32     std::cout << "How much money did you make in a day" << '\t' ;
33     std::cout << income.ads * income.percent / 100 * income.profit << std::endl;
34 }
35
36 int main()
37 {
38     Advertising income = input();
39     print(income);
40
41     return 0;
42 }

```

Задание № 2

```

1  #include <iostream>
2
3  struct Fraction // структура дробного числа
4  {
5      int numerator; // числитель
6      int denominator; // знаменатель
7  };
8
9  Fraction input() // функция пользовательского ввода
10 {
11     std::cout << "Enter numerator and denominator fraction" << '\n';
12     int numerator;
13     int denominator;
14
15     std::cin >> numerator >> denominator;
16     Fraction temp {numerator, denominator};
17     std::cout << '\a';
18
19     return temp;
20 }
21
22 void multiply(Fraction one, Fraction two) // функция вывода результата
23 {
24     std::cout << "Your fraction one: " << one.numerator << "/" << one.denominator
25     std::cout << "Your fraction two: " << two.numerator << "/" << two.denominator
26     std::cout << "Composition of your fraction:" << '\t';
27     std::cout << static_cast<double>(one.numerator * two.numerator) / (one.denomi
28 }
29
30 int main()
31 {
32     Fraction one = input(); // инициализация переменных структуры
33

```

```
34     Fraction two = input();
35     multiply(one, two);
36
37     return 0;
}
```

[Ответить](#)



3. *Владимир:*

[10 июня 2020 в 01:26](#)

Случайно наткнулся на сайт, подумал: "Ну обычный самоучитель". Но спустя около часа проведения на сайте понял, что это самый лучший самоучитель, всё подробно и понятно описано, приятно читать) Удачи автору)

Вот, кстати, мои варианты решения задачи:

1)

```
1  #include <iostream>
2  #include <windows.h>
3
4  using namespace std;
5
6  struct Advertising {
7      int showedAd;
8      int percAd;
9      int salarAd;
10 };
11
12 int main() {
13     setlocale(0, "RU");
14     SetConsoleOutputCP(1251);
15     SetConsoleCP(1251);
16     int a, b, c;
17     cout << "Введите количество показанной рекламы: ";
18     cin >> a;
19     cout << "Введите сколько процентов людей нажали на объявление: ";
20     cin >> b;
21     cout << "Введите сколько Вы получаете за каждое просматриваемое объявление: ";
22     cin >> c;
23     Advertising ad;
24     ad = { a,b,c };
25     cout << "Итак, вы заработали " << ad.showedAd * ad.percAd * ad.salarAd << endl;
26     return 0;
27 }
```

2)


```
1 #include <iostream> // Извиняюсь за столь подробное описание каждого действия, сд
2
3 using namespace std;
4
5 // Объявление структуры
6 struct Fraction {
7     double numer;
8     double denomer;
9 };
10
11 // Объявление функции-члена
12 double multiply(double n, double d) {
13     return (n * d);
14 }
15
16 int main() {
17     setlocale(0, "RU");
18
19     // Определение объектов структуры
20     Fraction fr1;
21     Fraction fr2;
22
23     // Объявление переменных
24     double a1, b1, a2, b2;
25
26     // Получение значений первой дроби
27     cout << "Введите числитель и знаменатель первой дроби: ";
28     cin >> a1 >> b1;
29     fr1 = { a1, b1 };
30
31     // Получение значений второй дроби
32     cout << "Введите числитель и знаменатель второй дроби: ";
33     cin >> a2 >> b2;
34     fr2 = { a2, b2 };
35
36     // Вычисления
37     double first_fr, second_fr;
38     first_fr = a1 / b1;
39     second_fr = a2 / b2;
40
41     // Вывод результата
42     cout << multiply(first_fr, second_fr);
43     return 0;
44 }
```

Не судите строго 😊

[Ответить](#)

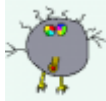


4. Яна:

[9 июня 2020 в 12:47](#)

Замечательный урок. Немного второе задание заставило посидеть в написании переменных, т.к. легко можно попутать из-за невнимательности.

```
1  #include <iostream>
2
3  using std::cout;
4  using std::cin;
5
6  struct Drob
7  {
8      int d1;
9      int d2;
10 };
11 Drob first;
12 Drob second;
13
14
15 void Result(Drob first, Drob second)
16 {
17     float x = static_cast<float>(first.d1 * second.d1) / (first.d2 * second.d2);
18     cout << "Если умножить две дроби между собой получим "<<x<<'\n';
19 }
20
21 int main(Drob first, Drob second)
22 {
23     setlocale(0, "");
24
25     cout << "Введите числитель первой дроби: ";
26     cin >> first.d1;
27     cout << "Введите знаменатель первой дроби: ";
28     cin >> first.d2;
29     cout << "Введите числитель второй дроби: ";
30     cin >> second.d1;
31     cout << "Введите знаменатель второй дроби: ";
32     cin >> second.d2;
33
34     Result(first, second);
35 }
```

[Ответить](#)

5. Furxie Fluke:

[15 марта 2020 в 17:00](#)

Спасибо огромное за всю эту информацию, которая ещё и так хорошо преподнесена)

Вот такое написал, в чём постарался ещё и совместить некоторые прошлые уроки, и проверить разные вещи в которых не был уверен

Так как сам обычно просматриваю все комментарии в поиске дополнительной информации, что бывает очень полезно, решил выложить и свой вариантик :3 Вдруг для кого то он будет полезен

```
1 struct Advertising
2 {
3     double watches;
4     double clicks;
5     double moneyEarnedByClicks;
6 };
7
8 enum class Message
9 {
10     FIRST_CALL,
11     SECOND_CALL,
12     THIRD_CALL,
13 };
14
15 void messages(int number) // С типом данных переменной
16 {
17     if (number == static_cast<int>(Message::FIRST_CALL)) // Тип данных перечисления
18         std::cout << "\nВведи количество просмотров рекламы ";
19     else if
20         (number == static_cast<int>(Message::SECOND_CALL))
21         std::cout << "\nВведи процент кликов относительно просмотров, 100.0% = 100";
22     else if
23         (number == static_cast<int>(Message::THIRD_CALL))
24         std::cout << "\nВведи средний заработок за каждое нажатие на объявления ";
25     else
26         std::cout << "\nОшибка вывода текста\n";
27 }
28
29 double getInfoFromUser()
30 {
31     static short timesCalled{ 0 }; // тест статической переменной, если я правильно
32     messages(timesCalled++); // подсчёт вызовов, что мне очень нужно. Но
33     if (timesCalled == 3)
34         timesCalled = 0;
35     double getNumber{};
36     std::cin >> getNumber;
37     std::cin.ignore(255, '\n');
38     return getNumber;
39 }
40
41
```

```

42
43 Advertising getInfo()    // по сути не нужная функция, лишь для задания нужна
44 {
45     Advertising info{getInfoFromUser(), getInfoFromUser(), getInfoFromUser()};
46
47     return info;
48     //    Альтернатива
49     //Advertising info;
50     std::cout << "Введи то всякое что нужно";
51     std::cin >> info.clicks;
52     std::cin >> info.watches;
53     std::cin >> info.moneyEarnedByClicks;
54     return info;
55     // так можно было бы закончить всё это что я начинаю из за этой функции прям
56 }
57
58
59
60 struct Calculation        // Структура для конечного подсчёта
61 {
62     double clicksPercent;
63     double totalEarning;
64 };
65
66 double getRawPercent(double percent)    // Для удобства нахожу процент который п
67 {
68     double rawPercent{percent / 100};
69     return rawPercent;
70 }
71
72 Calculation calculateAdvertising(Advertising incomeInfo)
73 {
74
75     double rawPercent{ getRawPercent(incomeInfo.clicks)};
76     double clicksCount{ incomeInfo.watches * rawPercent };
77     // Calculation result{ clicksCount, clicksCount * incomeInfo.moneyEarnedByClick
78
79     return { clicksCount, clicksCount * incomeInfo.moneyEarnedByClicks };    //
80
81 }
82
83
84 //std::string text
85
86 void printInfo(Calculation incomeInfo)
87 {
88     // std::string stringClickPercent(std::to_string(incomeInfo.clicksPercent));
89
90

```

```

91     std::cout << "\nКликов по объявлению примерно - " << incomeInfo.clicksPercent
92
93 }
94
95 int main()
96 {
97     SetConsoleOutputCP(1251);
98     SetConsoleCP(1251);
99
100    Advertising incomeInfo{getInfo()}; // ВЫЗОВ
101    Calculation calcIncomeInfo{ calculateAdvertising(incomeInfo) }; // ВЫЗОВ
102    printInfo(calcIncomeInfo); // ВЫЗОВ
103
104    // std::cout << "Sizeof long long watches " << sizeof(incomeInfo.watches) << "\
105    //      "\nType of watches " << typeid(incomeInfo.watches).name() << "\nType of
106    return 0;
    }

```

[Ответить](#)

1. Сергей:

[15 сентября 2020 в 22:52](#)

Ну наворотил то сколько всего!

[Ответить](#)

6. Дима:

[29 февраля 2020 в 18:17](#)

Огромное спасибо за этот самоучитель, это самое лучшее, что я читал, везде что то опускают, и таким как я очень тяжело, понять, где встречается, что то непонятное, здесь все по порядку не опуская ничего.

[Ответить](#)

7. Alex_1988:

[20 января 2020 в 21:30](#)

"Переменные типа struct подчиняются тем же правилам, что и обычные переменные. Следовательно, если вы хотите сделать переменную структуры доступной в нескольких файлах, то вы можете использовать ключевое слово extern."

Можно показать пример? У меня не получается подружить структуру со связью extern.

[Ответить](#)



1. *Onium:*

[21 июня 2020 в 21:18](#)

main.cpp

```
1  #include <iostream>
2  #include "Header.h"
3
4  int main()
5  {
6
7      extern Employee john;
8
9      std::cout << john.id << "\n";
10     std::cout << john.age << "\n";
11     std::cout << john.salary << "\n";
12
13     return 0;
14 }
```

header.h

```
1  #ifndef HEADER_H
2  #define HEADER_H
3
4  struct Employee
5  {
6      short id;
7      int age;
8      float salary;
9  };
10 #endif
```

source.cpp

```
1  #include "Header.h"
2
3  Employee john{ 1, 2, 3.3f };
```

[Ответить](#)



8. *Владимир:*

[8 декабря 2019 в 14:00](#)

Решил, что входные данные программка тоже должна принимать:

```
1  #include<iostream>
```

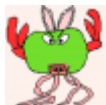


```

1 #include <iostream>
2
3 struct FractionalNumbers //структура для хранения дробных чисел
4 {
5     int integerNumerator; //числитель дробной переменной
6     int integerDenominator; //знаменатель дробной переменной
7 };
8
9 void multiply(FractionalNumbers fractionalVariable1, FractionalNumbers fractionalVariable2)
10 {
11     std::cout << "The first fractional variable :" << fractionalVariable1.integerNumerator;
12     std::cout << "The second fractional variable :" << fractionalVariable2.integerDenominator;
13     std::cout << "the result of the product of two fractional variables :" << fractionalVariable1.integerNumerator * fractionalVariable2.integerDenominator;
14 }
15
16 int main()
17 {
18     FractionalNumbers fractionalVariable1; //объявляем первую дробную переменную
19     FractionalNumbers fractionalVariable2; //объявляем вторую дробную переменную
20
21     std::cout << "Enter the numerator of the first fractional variable :";
22     std::cin >> fractionalVariable1.integerNumerator;
23     std::cout << "Enter the denominator of the first fractional variable :";
24     std::cin >> fractionalVariable1.integerDenominator;
25
26     std::cout << "Enter the numerator of the second fractional variable :";
27     std::cin >> fractionalVariable2.integerNumerator;
28     std::cout << "Enter the denominator of the second fractional variable :";
29     std::cin >> fractionalVariable2.integerDenominator;
30
31     multiply(fractionalVariable1, fractionalVariable2);
32
33     return 0;
34 }

```

[Ответить](#)



10. [armus1:](#)

[1 ноября 2019 в 22:36](#)

```

1 #include <iostream>
2
3 struct Advertising //Создаем структуру Advertising с тремя членами
4 {
5     int adsShown;
6     double percentageOfVisitors;
7     double earnings;
8 }

```



```

8  };
9
10 void dataOnEarnings(Advertising advertising) //определяем функцию для вывода введенных данных
11 {
12     std::cout << "Number of ads shown:" << advertising.adsShown << std::endl;
13     std::cout << "Percentage of visitors who clicked on ads:" << advertising.percentageOfVisitors << std::endl;
14     std::cout << "Average earnings per click on an ad: EUR" << advertising.earnings << std::endl;
15
16     std::cout << "Your earnings per day amounted to: EUR" << advertising.adsShown << std::endl;
17 }
18
19 int main()
20 {
21     Advertising advertising; //переменная структуры Advertising
22
23     std::cout << "Enter the number of ads shown:";
24     std::cin >> advertising.adsShown;
25     std::cout << "Enter what percentage of visitors clicked on the ads:";
26     std::cin >> advertising.percentageOfVisitors;
27     std::cout << "Enter the average earnings for each click on the ad: EUR";
28     std::cin >> advertising.earnings;
29
30     dataOnEarnings(advertising);
31
32     return 0;
33 }
34

```

[Ответить](#)



11. *Игорь:*

[13 сентября 2019 в 16:31](#)

задача №2

```

1  #include "pch.h"
2  #include <iostream>
3  using namespace std;
4  //создаем структуру
5  struct Intrigers
6  {
7      //члены структуры
8      int chislitel;
9      int znamenatel;
10 };
11
12 void multiplay(Intrigers d1, Intrigers d2)
13 {

```

```

14     setlocale(LC_CTYPE, "rus");
15     //выводим наши дроби введенные от пользователя
16     cout << "\nПервая дробь: " << d1.chislitel<<"/"<<d1.znamenatel<<"\n";
17     cout << "Вторая дробь: " << d2.chislitel<<"/" <<d2.znamenatel<<"\n\n";
18     //перемножаем эти дроби(незабываем что тип нужно изменить на тип с плавающей
19     cout << "Результат умножения: " <<static_cast<float>(d1.chislitel*d2.chislitel/znamenatel);
20 }
21
22 int main()
23 {
24     setlocale(LC_CTYPE, "rus");
25     //объявляем переменные для членов структуры
26     Intrigers d1;
27     Intrigers d2;
28     //вводим переменные
29     cout << "Введите числитель 1-й дроби: ";
30     cin >> d1.chislitel;
31     cout << "Введите знаменатель 1-й дроби: ";
32     cin >> d1.znamenatel;
33     cout << "Введите числитель 2-й дроби: ";
34     cin >> d2.chislitel;
35     cout << "Введите знаменатель 2-й дроби: ";
36     cin >> d2.znamenatel;
37
38     multiplay(d1, d2);
39 }

```

Ответить



12. **Игорь:**

[13 сентября 2019 в 11:52](#)

```

1  #include "pch.h"
2  #include <iostream>
3  using namespace std;
4
5  struct Advertising
6  {
7      int kolichPokazov;
8      double procentPosetiteley;
9      double zarabotok;
10 };
11
12 void printInfo(Advertising adv)
13 {
14     setlocale(LC_CTYPE, "rus");
15     cout << "\nКоличество показов: " << adv.kolichPokazov;
16     cout << "\nПроцент посетителей: " << adv.procentPosetiteley;

```

```

17     cout << "\nСредний заработок: " << adv.zarabotok << " руб.\n\n";
18
19     cout << "Сумма заработка за день: " << adv.kolichPokazov*(adv.procentPosetiteley);
20 }
21
22 int main()
23 {
24     setlocale(LC_CTYPE, "rus");
25     Advertising adv = { 0, 0, 0 };
26     cout << "Введите количество показов: ";
27     cin >> adv.kolichPokazov;
28     cout << "Введите процент посетителей: ";
29     cin >> adv.procentPosetiteley;
30     cout << "Введите средний заработок: ";
31     cin >> adv.zarabotok;
32
33     printInfo(adv);
34     return 0;
35 }

```

Ответить



13. *Ruslan:*

[17 июля 2019 в 11:19](#)

Немного не подумал, что вы хотели во втором задании просто перемножить два числа, и сделал так:

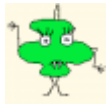
```

1  #include <iostream>
2
3  struct FractNum
4  {
5      int numerator;
6      int denominator;
7  };
8
9  FractNum reduction(FractNum num);
10 FractNum multiply(FractNum num1, FractNum num2);
11 void ShowNumber(FractNum num);
12
13 int main()
14 {
15     system("cls");
16
17     FractNum firstNumber;
18     FractNum secondNumber;
19
20     std::cout << "Enter a numerator of first fractional number: ";

```

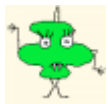
```
21     std::cin >> firstNumber.numerator;
22     std::cout << "Enter a denominator of first fractional number: ";
23     std::cin >> firstNumber.denominator;
24
25     std::cout << "\nEnter a numerator of second fractional number: ";
26     std::cin >> secondNumber.numerator;
27     std::cout << "Enter a denominator of second fractional number: ";
28     std::cin >> secondNumber.denominator;
29
30     firstNumber = reduction(firstNumber);
31     secondNumber = reduction(secondNumber);
32
33     FractNum ans = multiply(firstNumber, secondNumber);
34     ans = reduction(ans);
35
36     std::cout << "\n";
37     ShowNumber(firstNumber);
38     std::cout << " * ";
39     ShowNumber(secondNumber);
40     std::cout << " = ";
41     ShowNumber(ans);
42
43     std::cin.ignore();
44 }
45
46 FractNum reduction(FractNum num)
47 {
48     while (num.numerator % 2 == 0 && num.denominator % 2 == 0)
49     {
50         num.numerator /= 2;
51         num.denominator /= 2;
52     }
53     while (num.numerator % 3 == 0 && num.denominator % 3 == 0)
54     {
55         num.numerator /= 3;
56         num.denominator /= 3;
57     }
58     while (num.numerator % 5 == 0 && num.denominator % 5 == 0)
59     {
60         num.numerator /= 5;
61         num.denominator /= 5;
62     }
63     return num;
64 }
65
66 FractNum multiply(FractNum num1, FractNum num2)
67 {
68     FractNum answer = {num1.numerator * num2.numerator, num1.denominator * num2.d
```

```
70     return answer;  
71 }  
72  
73 void ShowNumber(FractNum num)  
74 {  
75     std::cout << num.numerator << "/" << num.denominator;  
76 }
```

[Ответить](#)14. *Алексей:*[16 июля 2019 в 16:20](#)

Вот честно, не понял я второго задания.

Тоже ничего сложного по структуре, но написано одно — делать другое.

[Ответить](#)15. *Алексей:*[16 июля 2019 в 15:28](#)

Сильный был "завтык" с объявлением структуры в функцию.

Сделал муторно, потом подправил, но думаю это все равно сложно написано. Задание не сложное, все те же переменные и функции, добавление данных, но вот...

```
1  #include <iostream>  
2  
3  struct Advertising { int amountViews; double percentClicks; double avarageCost; }  
4  
5  void result(Advertising add)  
6  {  
7      std::cout << add.amountViews << "\n";  
8      std::cout << add.percentClicks << "\n";  
9      std::cout << add.avarageCost << "\n";  
10  
11     std::cout << (add.amountViews*add.percentClicks/100*add.avarageCost) << std::endl  
12 }  
13  
14 int main()  
15 {  
16     Advertising add;  
17  
18     std::cout << "Enter views:\n";  
19     std::cin >> add.amountViews;  
20     std::cout << "Enter clicks:\n";  
21     std::cin >> add.percentClicks;  
22     std::cout << "Enter cost:\n";
```

```
23 | std::cin >> add.avarageCost;  
24 |  
25 | result(add);
```

[Ответить](#)16. *Алексей:*[16 июля 2019 в 15:06](#)

Структуры сильно напоминают массивы.

Тут конечно же предоставляются имена для каждой, но все же.

[Ответить](#)17. *Екатерина:*[6 марта 2019 в 22:33](#)

Добрый день. Нравятся ваши уроки. Не думаете выпустить печатный вариант?) Легко даётся материал по вашим урокам..и было бы удобно иметь подобного рода настольную книгу)

[Ответить](#)**R**1. *Юрий:*[7 марта 2019 в 06:43](#)

Привет, чуть позднее будет .pdf версия всех уроков по C++.

[Ответить](#)18. *somebox:*[20 ноября 2018 в 03:06](#)

Только посмотрев решение, понял, что нужно указать не дробные, а "дробные" переменные. А то не мог понять, причем тут double. Из-за этого не решил второе задание. Мне кажется, чуть внятнее нужно давать условия задачи.

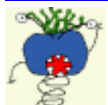
[Ответить](#)19. *Дмитрий:*[7 ноября 2018 в 15:05](#)

Решение на первый тест:

```
1 | #include <iostream>  
2 |  
3 | using namespace std;  
4 |
```

```
5
6 struct Advertising
7 {
8     int pokaz;
9     double procent_users;
10    double salary;
11 };
12
13 void bablo(Advertising z)
14 {
15     cout << "\nПоказано объявлений: " << z.pokaz;
16     cout << "\nПроцент просмотров: " << z.procent_users;
17     cout << "\nЗаработано: " << z.salary;
18     cout << "\n\nВсего заработано: " << z.pokaz * z.procent_users / 100 * z.salary;
19 }
20
21
22 int main (int argc, char* argv[])
23 {
24     setlocale (LC_ALL, "RUS");
25
26     Advertising users_vvod;
27     cout << "\nУкажите кол-во объявлений: ";
28     cin >> users_vvod.pokaz;
29     cin.ignore(32767, '\n');
30     cout << "\nПроцент просмотров: ";
31     cin >> users_vvod.procent_users;
32     cin.ignore(32767, '\n');
33     cout << "\nЗаработано: ";
34     cin >> users_vvod.salary;
35     cin.ignore(32767, '\n');
36     system("cls");
37     bablo(users_vvod);
38
39
40
41     cout << "\n\nВремя выполнения программы: " << clock() / 1000.0 << endl;
42     return 0;
43 }
```

Ответить



20. Алексей:

1 октября 2018 в 09:33

Ответ к заданию номер 2:

```
1 #include <iostream>
```

```
2
3 using namespace std;
4
5 struct Fraction
6 {
7     int numerator;
8     int denominator;
9 };
10
11 Fraction getFraction () //запрос числителя и знаменателя новой дроби
12 {
13     cout << "Lets create a new fraction. " << endl;
14     Fraction fract; //объявление переменной структуры Fraction
15     cout << "Enter numerator pls: ";
16     cin >> fract.numerator; //запрос числителя
17     cout << "Enter denominator pls: ";
18     cin >> fract.denominator; //запрос знаменателя
19     cout << "-----" << endl; //разделительная черта
20     return fract;
21 }
22
23 double fractMult(Fraction first, Fraction second) //вычисление результата
24 {
25     double multRes;
26     //явное преобразование int в double
27     multRes = static_cast <double> (first.numerator * second.numerator) / (first.denominator * second.denominator);
28     return multRes;
29 }
30
31
32 void multShow (Fraction first, Fraction second, double multRes) //вывод результата
33 {
34     cout << "Multiplication result of " << first.numerator << "/" << first.denominator << " and " << second.numerator << "/" << second.denominator << " is: " << multRes << endl;
35 }
36
37
38 int main()
39 {
40     Fraction first = getFraction ();
41     Fraction second = getFraction ();
42     double multRes = fractMult (first, second);
43     multShow (first, second, multRes);
44
45     return 0;
46 }
```

[Ответить](#)

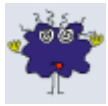


21. Максим:

18 сентября 2018 в 11:56

```
1  #include <iostream>
2
3  struct Advertising
4  {
5      int adsNumber;
6      double percentClickAd;
7      double averageEarningClick;
8  };
9
10 Advertising getClick()
11 {
12     using std::cout;
13     using std::cin;
14     Advertising temp={0,0.0,0.0};
15     cout<<"How many ads have you shown to your visitors? \n";
16     cin>>temp.adsNumber;
17     cout<<"How many percent of visitors clicked on ads? \n";
18     cin>>temp.percentClickAd;
19     cout<<"How much you earned on average for each click on ads? \n";
20     cin>>temp.averageEarningClick;
21     cout<<"\n";
22     return temp;
23 }
24
25 void printAdvertising(int a, double b, double c, double d)
26 {
27     using std::cout;
28     cout<<"Number of ads shown to visitors "<<a<<"\n" ;
29     cout<<"Percentage of visitors who clicked on the ad "<<b<<"%\n" ;
30     cout<<"Average earnings per click on ads "<<c<<"\n" ;
31     cout<<"\n";
32     cout<<"Earning money per day "<<d<<"\n" ;
33 }
34
35 double calculateAdvertising(int a, double b, double c)
36 {
37     return a*b/100*c;
38 }
39
40 int main()
41 {
42     Advertising click = getClick();
43     double sumAdv=calculateAdvertising(click.adsNumber, click.percentClickAd, click.averageEarningClick);
44     printAdvertising(click.adsNumber, click.percentClickAd, click.averageEarningClick, sumAdv);
45 }
```

```
45     return 0;  
46 }
```

[Ответить](#)

22. Эдуард:

[29 июля 2018 в 17:54](#)

```
1  #include <iostream>  
2  #include <locale>  
3  
4  struct Fraction  
5  {  
6      int numerator;  
7      int denominator;  
8  };  
9  
10 Fraction fractionEntering(Fraction fr, int a)  
11 {  
12     std::cout << "\nВведите числитель " << a << " дроби: ";  
13     std::cin >> fr.numerator;  
14  
15     std::cout << "Введите знаменатель " << a << " дроби: ";  
16     std::cin >> fr.denominator;  
17  
18     return fr;  
19 }  
20  
21 void multiply(Fraction fr1, Fraction fr2)  
22 {  
23     std::cout << "Деление двух дробей = " << static_cast<double>(fr1.numerator *  
24         (fr2.denominator * fr2.denominator) << "\n";  
25 }  
26  
27 int main()  
28 {  
29     setlocale(LC_ALL, "Rus");  
30  
31     Fraction fr1 = { 0, 0 };  
32     fr1 = fractionEntering(fr1, 1);  
33  
34     Fraction fr2 = { 0, 0 };  
35     fr2 = fractionEntering(fr2, 2);  
36  
37     multiply(fr1, fr2);  
38  
39     return 0;  
40 }
```

Такой код уместен ко второму заданию? (Да, знаю что используются "магические числа", а чем их можно заменить, если не использовать цикл?

[Ответить](#)



23. *Torgu:*

[3 июля 2018 в 17:15](#)

Здравствуйте, увидев снятие ограничения на инициализацию структур в C++14, я решил узнать, используется ли у меня C++14. В интернете везде один и тот же способ — написать программу:

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << __cplusplus;
6 }
```

Если программа вывела 199711, то C++98/03, если 201103 — C++11, а если 201300 или 201402, то C++14. У меня вывело 199711, то есть даже C++11 нет. Но тем не менее работает уникальная инициализация, свойственная для C++11. В чем проблема? Как точно определить версию C++? Использую Microsoft Visual Studio Community 2017, все по гайду

[Ответить](#)



24. *master114:*

[7 мая 2018 в 13:18](#)

Мне кажется, что в первом задании Значению количества просмотров лучше не `int` задавать, а что-нибудь побольше (я `long` присвоил).

Ведь обычно количество просмотров рекламы больше, чем 127.

Иначе будет переполнение и результат мусорный???

[Ответить](#)



1. *Юрий:*

[7 мая 2018 в 18:28](#)

Переменная `int`, скорее всего, на вашем компьютере занимает 4 байта (у меня 4). Диапазон значений для 4 байтов — от -2 147 483 648 до 2 147 483 647 — всего должно хватать.

[Ответить](#)



25. *Серж:*

[12 марта 2018 в 15:23](#)

Пример программы к заданию номер 2 (проверил ,работает)

```
1 #include <iostream>
2
3 struct drob
4 {
5     int cislit;
6     int znamenat;
7 };
8
9 drob vvod()
10 {
11     drob x{};
12     std::cout << "введите числитель " ;
13     std::cin >> x.cislit;
14     std::cout << "введите знаменатель ";
15     std::cin >> x.znamenat;
16     return x;
17 }
18 void multiply(drob x, drob y)
19 {
20     double z=0.0;
21     z=((double)x.cislit*y.cislit/(x.znamenat*y.znamenat));
22     std::cout << "произведение дробей " << z << "\n";
23 }
24
25 int main()
26 {
27     setlocale(LC_ALL, "rus");
28     drob a{};
29     drob b{};
30     std::cout << "Введите первую дробь: " << "\n" ;
31     a=vvod();
32
33     std::cout << "Введите вторую дробь: " << "\n" ;
34     b=vvod();
35
36     multiply(a, b);
37     //ожидание закрытия окна
38     std::cin.clear();
39     std::cin.ignore(32767, '\n');
40     std::cin.get();
41
42     return 0;
43 }
```

[Ответить](#)

R

1. Юрий:

[12 марта 2018 в 18:58](#)

Как вариант да, может быть.

[Ответить](#)



26. *Максим:*

[10 марта 2018 в 12:18](#)

Действительно. Спасибо за Вашу работу. Мои варианты решения.

Первая:

```
1 #include "stdafx.h"
2 #include <iostream>
3 struct Adversiting {
4     int x;
5     int y;
6     int z;
7 };
8 Adversiting getValue() {
9     std::cout << "enter the number of shows: ";
10    int x;
11    std::cin >> x;
12    std::cout << "enter the nubers click's: ";
13    int y;
14    std::cin >> y;
15    std::cout << "enter how many money you get from click's: ";
16    int z;
17    std::cin >> z;
18    return { x, y, z };
19 }
20 int accountant(Adversiting value) {
21     return value.x * value.y * value.z;
22 }
23 int main() {
24     Adversiting webSite = getValue();
25     std::cout << "your money today is " << accountant(webSite) << " dollars" << s
26     system("PAUSE");
27     return 0;
28 }
```

Вторая:

```
1 #include "stdafx.h"
2 #include <iostream>
3 struct Partitive {
4     int numerator;
5     int denominator;
6 };
7 Partitive getValue() {
```

```

8     int numerator;
9     int denominator;
10    std::cout << "Enter the numerator: ";
11    std::cin >> numerator;
12    std::cout << "Enter the denominator: ";
13    std::cin >> denominator;
14    return { numerator, denominator };
15 }
16 double multiply(Partitive first, Partitive second) {
17     int firstOperation = first.numerator * second.numerator;
18     int secondOperation = first.denominator * second.denominator;
19     return (static_cast<double>(firstOperation)) / (static_cast<double>(secondOperation));
20 }
21 int main() {
22     Partitive first = getValue();
23     Partitive second = getValue();
24     std::cout << first.numerator << "/" << first.denominator << " * " << second.numerator << "/" << second.denominator << "\n";
25     system("PAUSE");
26     return 0;
27 }

```

И для себя лично хотелось бы спросить/попросить примеры реальных рабочих задач, какие действительно нужно уметь решать на момент собеседования и на начальных этапах работы.

[Ответить](#)

1.  Юрий:

[12 марта 2018 в 19:18](#)

У вас как в первом, так и во втором задании используется возврат сразу нескольких значений (через оператор return). Это неверно, оператор return должен возвращать одно значение.

Насчет задачек — поищите в Интернете или в Ютубе. Предоставить вам реальные задачи, которые используются на собеседовании я не могу, так как сам на собеседования по C++ не ходил.

[Ответить](#)

1.  Максим:

[16 марта 2018 в 03:56](#)

А почему такое ограничение на return? Код ведь рабочий и значения передаются как положено.

[Ответить](#)

1.  Юрий:

[17 марта 2018 в 01:14](#)

Например:

```
1 #include "stdafx.h"
2 #include <iostream>
3
4
5
6 int mult(int x, int y, int z)
7 {
8     return x, y, z;
9 }
10
11 int main() {
12
13     int x, y, z;
14
15     std::cout << "Enter x: ";
16     std::cin >> x;
17
18     std::cout << "Enter y: ";
19     std::cin >> y;
20
21     std::cout << "Enter z: ";
22     std::cin >> z;
23
24     std::cout << mult(x, y, z);
25
26     return 0;
27 }
```

Выведется только одно последнее значение, которое вы введете, так как return может возвращать только одно значение. Когда вам нужно будет использовать возврат значений из функции, как в примере выше, то что тогда будете делать? Есть другие способы возврата сразу нескольких значений из функции, не нарушая при этом правил (смотрите [урок 98](#)).



27. Санжар:

[29 августа 2017 в 12:54](#)

```
/*Спасибо за Вашу работу!
```

```
Подскажите пожалуйста, а такой вариант приемлем в первом задании?*/
```

```
1 #include
2
3 struct Advertising
```

```
4 {
5     int quantityShown;
6     double percentClick;
7     double moneyPerClick;
8 };
9
10 int main()
11 {
12     int x;
13     double y,z;
14     Advertising actual{x,y,z};
15     std::cin >> x;
16     std::cin >> y;
17     std::cin >> z;
18     std::cout << "You gained " << x*y*z << " dollars.";
19     return 0;
20 }
```

[Ответить](#)

1.  Юрий:
[31 августа 2017 в 17:06](#)

Пожалуйста 😊

Если судить только по результату, то ваш вариант подходит также. Но если с пользовательской точки зрения, то не очень: желательнее всё разбить на функции и описывать действия в программе которые выполняются (например, что значат переменные x, y, z — какие значения пользователю следует вводить).

[Ответить](#)

1. Санжар:
[5 сентября 2017 в 14:24](#)

Спасибо за ответ, понял.
Успехов Вам!

[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