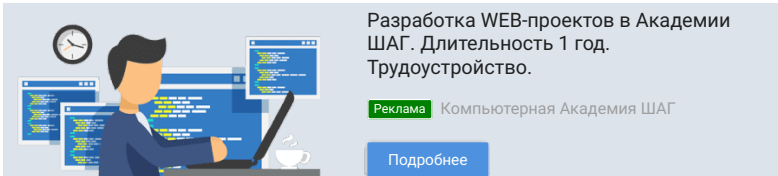


- [Учиться здесь](#)
- [Упражнения здесь](#)
- [Список книг](#)
- [Хороший стиль программирования](#)
- [обо мне](#)

« [C++ для начинающих как повернуть отрезок вокруг его середины](#)  
[C++ для начинающих Динамическая структура и бинарный файл](#) »

## C++ для начинающих. Запись структуры в файл и чтение структуры из файла

Опубликовано 26.08.2012 | Автор: [admin](#)



На этой странице моих материалов по C++ вы можете почитать про то, **как записывать структуру в файл и как её считывать**. В свое время я потратил много времени на поиски вразумительных материалов, но тогда это всё было без толку и первые мои попытки описать эту тему были ошибочны. Сейчас такая информация ищется на просторах интернета, но немного тяжеловато. Не буду мусолить эту тему, перейду к делу.

Я буду исходить из того, что **в полях структуры не будет ни одного указателя**. В полях структуры будут находиться только примитивные типы данных. Будет там и массив символов (куда же без него). Дело в том, что указатели в полях структуры являются большим источником проблем и многих заблуждений. Это тот случай когда двойственность указателей паразитирует и ведет себя наиболее яростно по отношению к программисту. Начинающий программист легко вводится в заблуждение тем, что программа может выдавать результат за правду, но на самом деле всё не так как ему кажется. Очень часто происходит так, что в файл записывается не значение указателя, а сам указатель. Значение остается в памяти и поэтому при попытке прочитать файл, очень часто значение получается правильное. Из файла считывается указатель, с указателя берется адрес для вытаскивания значения и значение вытаскивается не из файла, а из памяти. Так как значение располагается в памяти, то нет вероятности того, что оно там лежит в целости и сохранности. Файл — это одно, а память другое. Из-за такого рода проблемы совет избегать указателей очень даже актуален.

Я признаюсь: «Понятия не имею как записывать в файл структуру, если внутри ее полей находятся указатели». Я могу записывать каждое поле по отдельности, но это не то, чего бы мне хотелось на самом деле. На самом деле мне хотелось бы записывать структуру сразу целиком и такая возможность для структур существует. (Единственная оговорка, что мое: «существует» относится к структурам без указателей. С указателями я не знаю (думается, что для структур с указателями обрабатывать поля нужно только по отдельности)).

Итак, чтобы записать структуру в файл нужно, чтобы была такая структура, которую мы хотим записать. Логично? — логично. Поэтому Начинаем с написания простой структуры

```

1 struct mystruct //Структура MyStruct с тремя полями (i,buf,d)
2 {
3     long i;
4     char buf[255];
5     double d;
6 };
7
8 int main()
9 {
10     mystruct X; //Объявляем структуру X, тип которой mystruct
11     return 0;
12 }
```

Наверное это достаточно просто и понятно. Создали структуру и переменную типа созданной структуры. Теперь осталась самая важная часть — записать структуру в файл. Чтобы записать структуру в файл, нужно знать размеры структуры.

Снова поговорю об указателях. Если в полях структуры есть указатели, то правильный размер структуры узнать не получится. Размер объекта из указателя вытащить нельзя. Существует возможность узнать размер самого указателя, но не объекта на который он указывает. Это порождает некоторые заметные трудности. Указатель — это все-таки указатель, сложная гадость.

Возвращаемся к записи несложной структуры в файл. Чтобы записать структуру в файл, нужно сообщить компилятору адрес структуры, приведенный к типу указатель на `char` и размер записываемой структуры. Делается это так

```

1 #include <string.h>
2 #include <iostream.h>
```

```

3 #include <fstream.h>
4
5 struct mystruct {
6     long i;
7     char buf[255];
8     double d;
9 };
10
11 int main()
12 {
13     const char *FName="d:\\txt"; //Путь к файлу. Вписывайте свой.
14     mystruct X;
15     //memset(&ms,0,sizeof ms); //можно проинициализировать элементы структуры нулями так
16     X.i = 10; //записали значение в поле i
17     X.d = 2; //записали значение в поле d
18     strcpy(X.buf,"ci-plus-plus-snachala.ru"); //Записали значение в поле buf
19
20     // открываем файл для записи
21     ofstream f(FName,ios::binary|ios::out);
22     f.write((char*)&X,sizeof X); //Записали всю структуру целиком
23     f.close();
24
25     strcpy(X.buf,"0"); //я изменяю поле buf структуры X
26     X.i=0; //я изменяю поле i структуры X
27
28     //открываем файл для чтения
29     ifstream in(FName,ios::binary|ios::in);
30     in.read((char*)&X,sizeof X); //Читаем структуру целиком сразу
31     in.close();
32
33     cout<<X.buf<<"\n"; //Выводим значение из поля структуры. Оно равно ci-plus-plus-snachala.ru
34     return 0;
35 }

```

Очень хорошо строку с записью в файл я объяснить не смогу. Этот синтаксис проще запомнить, главное, что в первом параметре используется указатель на char из-за чего используется приведение типов к такому указателю, а после запятой идет второй параметр. Вторым параметром сообщается размер, который нужно использовать для записи в файл. Те кто знакомы с приведениями типов должны понимать, что происходит, кто не знаком имеет смысл привыкать. Понимание еще придет.

- (char\*)&X → Неудобная запись. Читать ее нужно справа налево. Адрес структуры **X** приводится к типу указатель на char. Это необходимо, чтобы сообщить методу **write (read)** адрес структуры в нужном ему виде. Внимательно прочитайте справа-налево, единственное: &X = адрес структуры **X**, и имейте в виду, что такой вид обратного чтения при работе с указателями бывает часто удобен.

Вот такой пример записи структуры в файл и чтения данных из файла в структуру для начинающих. Этот пример относительно прост. Главное понимать, что когда вы открываете файл для обработки, то после обработки обязательно его закрывать. Что при чтении или записи структуры из файла по описанному алгоритму, нужно привести адрес структуры к типу указатель на char и сообщить размер структуры. Следует помнить, что указатели: «Сложная гадость» и использовать их может или ввести в заблуждение или заставить написать больше сложного кода. Также хочу лишний раз обратить внимание на то, что в символьный массив нельзя присвоить строку и чтобы записать в него значение, нужно использовать приемы копирования.

статья полностью переписана 07 января 2014г.

если увидите похожие статьи, но с указателями, то знайте — они ошибочны и мне их нужно поправить.  
Просьба о них сообщить. Сам я все отследить не могу, но и в заблуждения вводить не хочу



Рубрика: [Без рубрики](#)

Все комментарии на сайте проверяются, поэтому ваш комментарий может появиться не сразу. Для вставки кода в комментарий используйте теги: **[php]** ВАШ\_КОД **[/php]**

12 комментариев: [C++ для начинающих. Запись структуры в файл и чтение структуры из файла](#)



Sergio говорит:

[09.09.2013 в 4:14 пп](#)

Что-то мне подсказывает, что в предпоследнем примере (33 строка) и последнем (39 строка) косяк...



Для **записи** в файл должно быть **ios::out**.

[Автор сайта отвечает:](#)

Истинно так. Спасибо.

Sergio говорит:

А вот в последнем примере в 47 строке зря исправили: нужно **in** 😊



[Ответить](#)



Максим говорит:

[07.01.2014 в 1:55 дп](#)

никакой строки тут в файл не сохраняется, туда сохраняются структура, т.е поля структуры X, а именно: указатель на строку, и сайз соответственно, т.е. не больше 8 байт. сама строка остаётся в памяти. если она будет очищена и перезаписана то обращение к игреку вызовет ошибку, да и в данном случае при использовании минимально вменяемого отладчика ругнулся бы на попытку очищения игрека после икса, так как к этому моменту память уже очищена.

а память выделенная игреку как-раз безвозвратно потеряна и очистится только с выходом из программы.

[Автор сайта отвечает:](#)

позднее посмотрю. скорее всего это так и есть.

Исправлено

👍 0 👎 0

[Ответить](#)



магомед говорит:

[09.04.2015 в 2:11 пп](#)

использовал ваш код в своей программе как выше сказано так и есть как только программу закрываю она выводит какие то левые символы а это не очень хорошо. как именно со всеми значениями сохранить данные не знаете?

👍 0 👎 0

[Ответить](#)



[admin](#) говорит:

[09.04.2015 в 2:48 пп](#)

Выше сказано про другое. То о чем говорилось тогда я исправил тогда.

На строку выделено 255 символов.

В строку вводится строка

Последний символ в строке — признак окончания строки

За признаком окончания строки все что угодно. Это мусор.

Добавить строчку

```
1 strcpy(X.buf,"ci-plus-plus-snachala.ru"); //Записали значение в поле buf
2 for (int i=strlen(X.buf);i<255;i++) X.buf[i]='\0'; //Вот эту
```

и мусора не будет видно в файле.

👍 0 👎 0

[Ответить](#)



магомед говорит:

[09.04.2015 в 2:54 пп](#)

вот короче помогите с этим отрезком кода для структуры. мне надо значения в файл сохранить и потом считать для обработки я сделал вот так если программу не закрывать то все норм а если закрыть и открыть то уже мусор выводит.

```
1 struct january{
2     int kol_os[30];
3     int veter[30];
4     int temp[30];
5 };
6 system("cls");
7 const char *FName = "january.txt"; //Путь к файлу. Вписывайте свой.
8 january X;
9 cout << "введите день месяца который следует заполнить:";
10 cin >> a;
11 cout << "введите количество осадков:"; cin >> X.kol_os[a];
12 cout << "введите направление ветра:"; cin >> X.veter[a];
13 cout << "введите температуру:"; cin >> X.temp[a];
14
15
16 ofstream f(FName, ios::binary | ios::out); // открываем файл для записи
17 f.write((char*)&X, sizeof(january)*3); //Записали всю структуру целиком
18 f.close();vibor();
```

👍 0 👎 0

[Ответить](#)



магомед говорит:

[09.04.2015 в 2:55 пп](#)

случайно в описание написал вопрос. прочитайте там

0 0

[Ответить](#)



[admin](#) говорит:

[09.04.2015 в 4:05 пп](#)

Я не очень понимаю

я и из программы выходил и перезагружался и все норм. и в разных компиляторах проверял

ничего и не изменил.

```
1 #include <iostream>
2 #include <fstream>
3
4 using namespace std;
5
6 struct Month{
7     int kol_os[30];
8     int veter[30];
9     int temp[30];
10 };
11
12 int main(){
13     setlocale(LC_ALL, "");
14     //system("cls");
15     const char *FName = "d:\\month.txt"; //Путь к файлу. Вписывайте свой.
16     Month X;
17
18     for (int i=0; i<30; i++){
19         X.kol_os[i]=0;
20         X.veter[i]=0;
21         X.temp[i]=0;
22     }
23
24     int a=0;
25     cout << "введите день месяца который следует заполнить:";
26     cin >> a;
27     cout << "введите количество осадков:"; cin >> X.kol_os[a];
28     cout << "введите направление ветра:"; cin >> X.veter[a];
29     cout << "введите температуру:"; cin >> X.temp[a];
30
31     /*ofstream f(FName, ios::binary | ios::out); // открываем файл для записи
32     f.write((char*)&X, sizeof(X)); //Записали всю структуру целиком
33     f.close();
34 */
35     ifstream in(FName, ios::binary | ios::in);
36     in.read((char*)&X, sizeof(X)); //Читаем структуру целиком сразу
37     in.close();
38
39     for (int i=0; i<30; i++){
40         cout << i+1 << ": " << X.kol_os[i] << " " << X.veter[i] << " " << X.temp[i] << "\n";
41     }
42
43
44
45
46 }
```

Вероятно проблема в том, что в структуре массивы и значит целиком ее не запишешь. Я просто в упор не вижу когда ошибка случается.

0 0

[Ответить](#)



магомед говорит:

[09.04.2015 в 11:21 пп](#)

в общем я скопировал ваш код и вставил себе все отлично заработало если запустить заполнение через цикл а когда вручную ввожу оно сохраняет одно только последнее значение почему то

0 0

[Ответить](#)



[admin](#) говорит:

[09.04.2015 в 11:31 пп](#)

Я не заполняю в цикле структуру. Я в цикле присваиваю нули не инициализированным переменным. Без

этого в них хранится любой мусор. А только после этого работаю с этой структурой.

Я заполняю только одно значение, цикл для этого не требуется.

Я Ваш код не видел, поэтому догадаться о проблеме у меня не выйдет.

Также я понятия не имею как у Вас происходит ручное заполнение.

Мой левел экстрасенсорика — нуб

👍 0 👎 0

[Ответить](#)



RomixERR говорит:

[13.05.2015 в 4:02 nn](#)

А теперь я объясню подробнее чем такое (char\*)&X .

```
1 #include <iostream>
2 #include <fstream>
3 #include <ios>
4 #include <conio.h>
5 using namespace std;
6 struct vol
7 {
8     int A;
9     int B;
10 };
11
12 void main()
13 {
14     vol X; //объявили переменную X типа структуры vol
15     X.A=82; //определили (задали поля структ. переменной)
16     X.B=111;
17
18     ofstream FILE("1.txt",ios::binary); //создаём объект класса ofstream при помощи конструктора
19
20     // для записи в файл используется метод экземпляра класса .write - первый его аргумент требует char * - то есть указатель на
21     // если бы у нас был массив char X[] то такой указатель получить было бы просто: X - это и есть указатель на первый элемент
22     // а у нас структура, по этому нужно сделать преобразование типов указателей. Если писать подробно, то:
23
24     vol * UkVol; //объявляем переменную типа "указатель vol"
25     char * UkChar; //объявляем переменную типа "указатель char"
26
27     UkVol = &X; //присваиваем ссылку на X - указателю
28     // далее следует сделать приведение типов
29     UkChar = reinterpret_cast<char*>(UkVol);
30     // можно и так, но это устаревшая форма записи:
31     //UkChar=(char*)UkVol;
32     FILE.write(UkChar,sizeof X); //Записываем наконец-то в файл используя в методе те аргументы, которые требуются
33     FILE.close();
34     //в папке проекта создается файл с именем 1.txt - там будет строка "R o ".
35     //открыть структуру можно аналогично, используя метод .read
36
37 }
```

Действительно, (char\*)&X не удобная и устаревшая запись. Следует писать reinterpret\_cast&X

👍 0 👎 0

[Ответить](#)



о

admin говорит:

[13.05.2015 в 5:27 nn](#)

reinterpret\_cast(UkVol); гораздо неудобнее и выглядит уродливо.

То не устаревшая форма записи, то форма записи в стиле Си. Да, считается, что преобразования делать лучше не в стиле Си.

А еще лучше использовать std::string и не использовать указатели

👍 0 👎 0

[Ответить](#)



Влад говорит:

[10.11.2016 в 12:37 nn](#)

Спасибо большое, работает!

👍 0 👎 0

[Ответить](#)

**Добавить комментарий**

Ваш e-mail не будет опубликован.



Имя

E-mail

Сайт

## Поиск

☒ на сайте ☐ в интернете

## Последние темы

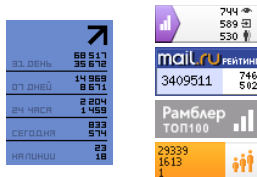
- [Спецификатор вывода типа: auto. C++11](#)
- [C++. Знакомство с decltype, по следам Скотт Мейерса](#)
- [C++ Паттерн проектирования "Фабричный метод". \(Продолжение\)](#)

## Случайная книга в электронном формате



## Последние комментарии

- [admin](#) к записи [C++ для начинающих. Как открыть файл в нужном режиме](#)
- [Игорь](#) к записи [C++ для начинающих. Как открыть файл в нужном режиме](#)
- [admin](#) к записи [Работа с текстовым файлом в C++ для начинающих. Запись и чтение](#)



## Мета

- [Войти](#)
- [RSS записей](#)
- [RSS комментариев](#)
- [WordPress.org](#)

## НАГРАДИ АВТОРА САЙТА

WEBMONEY

R375024497470

U251140483387

Z301246203264

E149319127674

Работа программиста и шамана имеет много общего - оба бормочут непонятные слова, совершают непонятные действия, и не могут объяснить, как оно работает

## Выражаю свою признательность

- **Максиму** очень признателен за указание на мои ошибки и неточности.
- **Sergio** ==> за оказание помощи в исправлении моих ошибок
- **Gen** ==> за правильное стремление помочь другим новичкам и выявления моих ошибок

связь с автором [daslex@yandex.ru](mailto:daslex@yandex.ru)



Сайт располагается на хостинге <http://oxnull.net/>

<http://ci-plus-plus-snachala.ru>

Designed by [das](#).