

# Работа с файлами

**№ урока:** 11 **Курс:** C++ Advanced

**Средства обучения:** Qt Creator

## Обзор, цель и назначение урока

Научить студентов понимать и применять на практике базовые механизмы работы с файлами, а именно чтение и запись, показать, какие проблемы могут возникнуть и пути их решения.

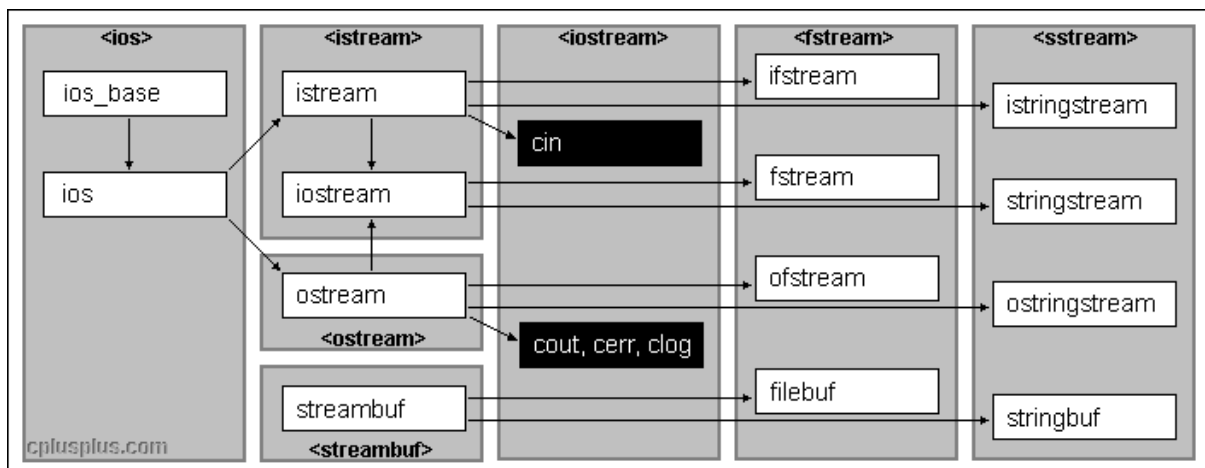
## Изучив материал данного занятия, учащийся сможет

- Понимать, что такое работа с файлами.
- Уметь объяснить разницу между форматированным и неформатированным процессом обработки потока ввода/вывода.
- Понимать, что такое rdbuf.
- Знать особенности работы с файлами, какие негативные тесты стоит применять к обработке.
- Знать о таком понятии, как локализация, фасеты, бинарные файлы.

## Содержание урока

1. base input/output stream classes
2. fstream, ofstream, ifstream
3. Filebuf
4. Базовые операции при работе с файлами

## Резюме



ios_base::in	открыть файл для чтения
ios_base::out	открыть файл для записи
ios_base::ate	при открытии переместить указатель в конец файла
ios_base::app	открыть файл для записи в конец файла

<code>ios_base::trunc</code>	удалить содержимое файла, если он существует
<code>ios_base::binary</code>	открытие файла в двоичном режиме

Файл – именованная область внешней памяти, выделенная для хранения массива данных. Данные, содержащиеся в файлах, имеют самый разнообразный характер: программы на алгоритмическом или машинном языке; исходные данные для работы программ или результаты выполнения программ; произвольные тексты; графические изображения и т. п.

`std :: basic_filebuf` - это `std :: basic_streambuf`, чья связанная последовательность символов представляет собой файл. И входная последовательность, и выходная последовательность связаны с одним и тем же файлом, и для обеих операций сохраняется позиция объединенного файла.

Функции `underflow ()` и `overflow ()` / `sync ()` выполняют фактический ввод-вывод между файлом и областями получения и размещения буфера. Когда `CharT` не является `char`, большинство реализаций хранит много байтовые символы в файле, а фасет `std :: codecvt` используется для преобразования широких / много байтовых символов.

### Закрепление материала

- Чем отличается `get()` от `getline()`?
- Чем отличается `std::failbit` от `std::badbit`?
- Для чего необходим `rdbuf`?
- Какие виды файлов вы знаете?

### Дополнительное задание

Задание

Изучите особенности фасетов, локалей, `wchar_t`.

### Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Реализуйте запись и считывание файлов разных локалей, разных расширений, бинарных и обычных, произведите замеры по выполнению, сделайте соответствующие выводы.

Задание 3

Зайдите на сайт MSDN.

Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

### Рекомендуемые ресурсы

<http://stdcxx.apache.org/doc/stdlibug/34-3.html>

[https://en.cppreference.com/w/cpp/io/basic\\_ios/rdbuf](https://en.cppreference.com/w/cpp/io/basic_ios/rdbuf)

<http://www.cplusplus.com/reference/sstream/stringstream/rdbuf/>

<https://stackoverflow.com/questions/524591/performance-of-creating-a-c-stdstring-from-an-input-iterator/524843#524843>

<https://gehrcke.de/2011/06/reading-files-in-c-using-ifstream-dealing-correctly-with-badbit-failbit-eofbit-and-perror/#ideal>

<https://lemire.me/blog/2012/06/26/which-is-fastest-read-fread-ifstream-or-mmap/>

<http://umich.edu/~eecs381/handouts/filestreams.pdf>

<http://www.drdobbs.com/the-standard-librarian-defining-a-facet/184403785>