



Урок 5. C++: Запись и обработка строк файла

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 5 из 34 · 45 минут · Модуль «Файлы»

Уровень: начинающий программист; C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

В отчёте надо сохранить исходный номер и длину непустой строки. Будет ли строка из двух пробелов пустой и что произойдёт при повторном запуске?

Результат урока: Записать отчёт с исходными номерами и длинами строк, проверить результат после закрытия потока и отличить перезапись от накопления журнала.

01. Проблемная ситуация

0–5 минут. Вход `A\n\n B \n` содержит три записи. Запишите непустые в формате номер|длина|текст. Пробелы сохраняются; `\n` в условии обозначает LF.

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Контракт определяет пустоту и длину

Формат входа: до 8 строк по 40 ASCII-знаков, LF или CRLF. Пустая строка имеет длину 0, строка из пробелов непуста. В ASCII `size()` соответствует числу знаков. Сначала читаем все строки в `vector`, удаляя только завершающий CR; затем открываем выход. Исходные номера идут по всем строкам, включая пропущенные.

Формат `output.txt`: номер|длина|текст и LF для каждой непустой строки. Нумерация с 1. `ios::trunc` заменяет старое содержимое, `ios::app` дописывает в конец. Бинарный режим вывода делает LF одинаковым на разных системах. Вход и выход имеют разные имена; запускайте в своей учебной папке.

Успешное открытие ещё не доказывает завершение записи. Закрываем поток и проверяем его состояние; при ошибке выдаём `WRITE_ERROR`. Этот урок гарантирует сохранность старого отчёта при ошибке чтения до открытия выхода, но не обещает восстановить файл после сбоя самой записи. Проверку содержимого добавим в следующем уроке.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

`A\n\n B \n` преобразуется в `1|1|A\n3|3| B \n`. Номер 2 пропущен, пробелы вокруг `B` сохранены, длина равна 3. `written=2`. При втором запуске с тем же входом файл побайтно тот же.

```
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
int main() {
    std::ifstream f("input.txt", std::ios::binary);
    if (!f) { std::cout<<"READ_ERROR\n"; return 0; }
    std::vector<std::string> rows;
    std::string s;
    while (std::getline(f, s)) {
        if (!s.empty() && s.back()=='\r') s.pop_back();
        rows.push_back(s);
    }
    if (!f.eof()) { std::cout<<"READ_ERROR\n"; return 0; }
    std::ofstream out("output.txt", std::ios::binary|std::ios::trunc);
    if (!out) { std::cout<<"WRITE_ERROR\n"; return 0; }
    int written=0;
    for (std::size_t i=0; i<rows.size(); ++i) {
        if (!rows[i].empty()) {
            out<<i+1<<' '|<<rows[i].size()<<' '|<<rows[i]<<'\n';
            ++written;
        }
    }
    out.close();
    if (!out) { std::cout<<"WRITE_ERROR\n"; return 0; }
    std::cout<<"written="<<written<<'\n';
}
```



Входная запись 1	A → длина 1 → 1 1 A
Входная запись 2	пустая → пропустить
Входная запись 3	пробел B пробел → длина 3
Режим trunc	повтор запуска заменяет, не удваивает

Номер строки относится к входу, а длина включает пробелы; пустые записи не перенумеровывают последующие.

03. Практика 1. Проверяем материал

Перезапись даёт воспроизводимый отчёт

14–25 минут. Сравните прогнозы по пяти строкам таблицы с файлами, созданными программой. Каждую строку результата проверяйте вместе с пробелами и LF.

Вход	output.txt; written
A\n\n B \n	1 1 A\n3 3 B \n; 2
Пустой / один LF	Пустой выход; 0
ху без LF	1 2 ху\n; 1
Два пробела и LF	1 2 \n; 1
A\r\nB	1 1 A\n2 1 B\n; 2

2 минуты — прогноз; 4 — проверка файлов; 3 — повторный запуск одного входа и сравнение; 2 — объяснение номера 3. Критерии: точный формат, длины, отсутствие удвоения, правильное закрытие потока.

04. Практика 2. Перенос способа

Журнал хранит несколько запусков

25–33 минуты. Второй самостоятельный исходник ведёт log.txt: lines=N;nonempty=K;chars=C, где C — сумма длин строк без CR/LF. За 2 минуты предскажите строку для A\n\n B \n, за 4 выполните два запуска, за 2 сравните log.txt и output.txt. Перед опытом создайте отдельную папку без старого log.txt. Критерии: 3,2,4 и ровно две одинаковые записи журнала.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты выберите режим для окончательного отчёта и для истории запусков, за 3 докажите выбор повторным запуском, за 2 укажите ограничение ASCII.

06. Индивидуальная проверка

40–45 минут. Вход Z\n\nqq без последнего LF. Запишите весь отчёт, written, строку журнала и эффект двух запусков отчёта. По баллу за пункт.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Основной отчёт содержит две записи с номерами 1 и 3. Пустые входы очищают прежний результат до пустого файла — это корректная перезапись. Пробелы считаются знаками. Повторный запуск в `trunc` не увеличивает файл; замена на `app` без изменения контракта ошибочно удвоила бы отчёт.

Практика 2: разбор и критерии

На каждый запуск добавляется `lines=3;nonempty=2;chars=4` и `LF`; после двух запусков две строки. Пустой файл добавляет `lines=0;nonempty=0;chars=0`. Журнал намеренно накопительный, поэтому его размер растёт. Вход и `output.txt` второй исходник не меняет.

```
#include <fstream>
#include <iostream>
#include <string>
int main() {
    std::ifstream f("input.txt",std::ios::binary);
    if (!f) { std::cout<<"READ_ERROR\n"; return 0; }
    int n=0,k=0,c=0;
    std::string s;
    while (std::getline(f,s)) {
        if (!s.empty() && s.back()=='\r') s.pop_back();
        ++n; k+=!s.empty(); c+=static_cast<int>(s.size());
    }
    if (!f.eof()) { std::cout<<"READ_ERROR\n"; return 0; }
    std::ofstream out("log.txt",std::ios::binary|std::ios::app);
    if (!out) { std::cout<<"WRITE_ERROR\n"; return 0; }
    out<<"lines="<<n<<"nonempty="<<k<<"chars="<<c<<"\n";
    out.close();
    std::cout<<(out?"logged":"WRITE_ERROR")<<"\n";
}
```

Обсуждение: возможный вывод

Отчёт описывает текущий вход, поэтому заменяется. История хранит события, поэтому дополняется. `size()` для UTF-8 считает байты, а не количество видимых букв; в данном уроке ASCII исключает это расхождение. Режим записи определяется назначением файла.

Индивидуальная проверка: ответ и критерии

`1|1|Z\n3|2|qq\n`; `written=2; lines=3;nonempty=2;chars=3` и `LF`; после двух запусков отчёт остаётся тем же. Журнал при двух запусках второго исходника содержит две записи.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Компилятор G++ с поддержкой C++17 и редактор. Сборка: `g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson`. Запуск `./lesson` в Linux/macOS или `.\lesson.exe` в Windows (при сборке задайте `-o lesson.exe`). Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 9 класс. Предпосылки: `getline`, `string`, `vector`, файловые потоки. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. Матрица программы предназначена учителю.



Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ

07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Файлы	Записать отчёт с исходными номерами и длинами строк, проверить результат после закрытия потока и отличить перезапись от накопления журнала.	Исходник, журнал тестов и объяснение результата
Типы файлов; чтение, запись и обработка данных	Журнал хранит несколько запусков	Самостоятельная доработка, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

C++: файловые потоки

<https://eel.is/c++draft/fstream.syn>

C++: открытие и закрытие выходного потока

<https://eel.is/c++draft/ofstream.members>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 18.10.2026.