



Урок 2. Python: Запись и обработка строк файла

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 2 из 34 · 45 минут · Модуль «Файлы»

Уровень: начинающий программист; Python 3.

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

После обработки файла номера строк стали 1 и 2 вместо 1 и 3. Где исчезла информация о пустой записи и почему повторный запуск иногда удваивает отчёт?

Результат урока: Преобразовать строки файла в отчёт с исходными номерами и различить перезапись w и добавление a.

01. Проблемная ситуация

0-5 минут. Даны строки oak, пустая, затем " elm " с пробелами. Нужно пропустить только пустую строку и записать исходные номера. Составьте две строки результата до запуска.

02. Понятия и разобранный пример

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

Сначала прочитать, затем записать

Обрабатываем до 8 строк по 40 символов в UTF-8 без BOM. Удаляем только окончание строки; `line!=""` исключает именно пустую запись. Строка из пробелов не пуста по нашему контракту. `enumerate(...,1)` идёт по всем исходным строкам, поэтому после пропуска сохраняется номер 3.

Режим `w` создаёт файл или очищает прежний при открытии. Режим `a` добавляет запись в конец. Для воспроизводимого отчёта выбираем `w`; для журнала событий — `a`. В учебной папке вход `input.txt` и выход `output.txt` различаются, чтобы не потерять источник. Файл закрывается до чтения результата.

Формат `output.txt`: номер, двоеточие без пробела, исходная непустая строка, LF. Пробелы самой строки сохраняем. Сначала собираем текст, потом открываем выход. Здесь успешное чтение предполагается; проверка ошибок и сохранность прежнего результата при неверном вводе рассматриваются отдельно.

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

`input.txt`: `oak\n\n elm \n`. Результат `output.txt`: `1:oak\n3: elm \n`. Вторая строка вывода начинается цифрой 3, а не 2; пробелы вокруг `elm` остаются. Запускайте в отдельной учебной папке, где `output.txt` можно перезаписывать.

```
with open("input.txt", encoding="utf-8") as src:
    rows = [raw.rstrip("\n") for raw in src]
result = []
for i, line in enumerate(rows, 1):
    if line != "":
        result.append(f"{i}:{line}\n")
with open("output.txt", "w", encoding="utf-8",
        newline="\n") as dst:
    dst.write("".join(result))
print("written=", len(result), sep="")
```



Исходные строки	1: oak 2: пусто 3: ·elm·
Условие	оставить пропустить оставить
Выход	1:oak\n3:·elm·\n
Повтор с w	тот же файл, без удвоения

Знак · в схеме обозначает обычный пробел; номера сохраняют позицию в исходнике.

03. Практика 1. Проверяем материал

Номер относится к исходному файлу

14–25 минут. Создайте четыре варианта входа и сравните точный выход, в том числе пробелы. Для основного входа запустите программу дважды.

Вход; \n означает LF	Выход output.txt
oak\n\n elm \n	1:oak\n3: elm \n
Пустой файл / \n	Пустой файл в обоих случаях
x без LF	1:x\n
Два пробела и LF	1: \n; пробелы сохранены

2 минуты — прогноз; 4 — запуски; 3 — замените w на a, выполните два запуска в новом пустом output.txt и сравните; 2 — восстановите w. Критерии: номера, пробелы, пустой результат и отсутствие удвоения при w.

04. Практика 2. Перенос способа

Добавление журнала запусков

25–33 минуты. Сделайте отдельный журнал log.txt: за каждый запуск добавлять lines=N;nonempty=K и LF, где N — все строки, K — непустые без удаления пробелов. За 2 минуты рассчитайте основной вход, за 4 реализуйте, за 2 проверьте два запуска и пустой файл. Критерии: a, два счётчика, одна запись на запуск.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты выберите режим для отчёта и для журнала, за 3 объясните риск совпадения пути источника и результата, за 2 обоснуйте необходимость повторного чтения результата после закрытия файла.

06. Индивидуальная проверка

40–45 минут. Для input.txt со строками A, пустая, B укажите точный output.txt, written, строку журнала и режим для накопления журнала. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

При `w` основной отчёт каждый раз одинаков: две строки с номерами 1 и 3. При `a` два запуска создадут четыре строки 1,3,1,3. Пустой вход с `w` очищает прежний отчёт до нуля байтов. Строка из двух пробелов сохраняется: проверка `strip()` изменила бы контракт.

Практика 2: разбор и критерии

Для основного входа каждый запуск добавляет `lines=3;nonempty=2`. После двух запусков изначально пустой `log.txt` содержит эту строку дважды. Для пустого входа добавляется `lines=0;nonempty=0`. Полный отдельный исходник:

```
with open("input.txt", encoding="utf-8") as src:
    rows = [raw.rstrip("\n") for raw in src]
k = sum(line != "" for line in rows)
with open("log.txt", "a", encoding="utf-8",
        newline="\n") as dst:
    dst.write(f"lines={len(rows)};nonempty={k}\n")
print("logged")
```

Обсуждение: возможный вывод

Отчёт отражает текущее состояние — `w`; журнал накапливает события — `a`. Открытие входа на запись может уничтожить данные до чтения. Проверять следует фактическое содержимое файла, а не только сообщение `written`: сообщение не заменяет сравнение.

Индивидуальная проверка: ответ и критерии

1:A\n3:B\n; `written=2`; `lines=3;nonempty=2` с LF; режим `a`. По баллу за пункт; разделители и номера входят в контракт.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Python 3.11+ и IDLE или редактор; запуск `python lesson.py` (либо `python3 lesson.py`). Внешние библиотеки не нужны. 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: `open`, `with`, UTF-8, `enumerate`, чтение строк. 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	Преобразовать строки файла в отчёт с исходными номерами и различить перезапись w и добавление a.	Исходник, журнал тестов и объяснение результата
Типы файлов; чтение, запись и обработка данных	Добавление журнала запусков	Самостоятельная доработка, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

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

Python: чтение и запись файлов

<https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files>

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.