



Урок 3. Python: Ошибки и надёжный файловый конвейер

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 3 из 34 · 45 минут · Модуль «Файлы»

Уровень: начинающий программист; Python 3.

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

Во второй строке вместо числа стоит 7x. Должна ли программа сохранить частичную сумму и затереть вчерашний правильный отчёт?

Результат урока: Проверить весь вход до записи результата, различить ошибки чтения и данных и доказать сохранность прежнего отчёта при неверном вводе.

01. Проблемная ситуация

0–5 минут. Файл input.txt содержит 5\n7x\n2\n; output.txt содержит OLD. Укажите строку ошибки и решите, что должно остаться в output.txt. Нельзя молча считать 7x числом 7.

02. Понятия и разобранный пример

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

Чтение, проверка, расчёт, запись

Учебный формат: от 0 до 8 строк; каждая — необязательный минус и 1–3 ASCII-цифры, без пробелов, диапазон -999...999. Пустой файл корректен, пустая строка внутри файла — ошибка. Последний LF необязателен. Пример -007 допустим и означает -7.

Конвейер сначала полностью читает и проверяет строки, затем рассчитывает count и sum, и лишь после этого открывает выход в режиме w. При ошибке данных возвращаем DATA_ERROR, при проблеме чтения — READ_ERROR; прежний output.txt остаётся нетронутым. Файлы разные и лежат в отдельной учебной папке.

Это защита от неверного ввода, а не обещание атомарной записи. Если ошибка возникла уже при записи или закрытии выхода, результат может оказаться частичным: возвращаем WRITE_ERROR. Для полноценной устойчивости нужны временный файл, контролируемая замена и дополнительные гарантии среды; здесь их не имитируем.

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

5\n7x\n2\n отклоняется на строке 2 до открытия output.txt. После исправления на 7 получаем count=3\nsum=10\n. Пустой файл даёт count=0\nsum=0\n, что отличается от DATA_ERROR.

```
def convert(src, dst):
    try:
        with open(src, encoding="utf-8") as f:
            rows = [x.rstrip("\n") for x in f]
    except (OSError, UnicodeError):
        return "READ_ERROR"
    if len(rows) > 8:
        return "DATA_ERROR"
    values = []
    for s in rows:
        digits = s[1:] if s.startswith("-") else s
        if not 1 <= len(digits) <= 3:
            return "DATA_ERROR"
        if any(c < "0" or c > "9" for c in digits):
            return "DATA_ERROR"
        values.append(int(s))
    text = f"count={len(values)}\nsum={sum(values)}\n"
    try:
        with open(dst, "w", encoding="utf-8", newline="\n") as f:
            f.write(text)
    except (OSError, UnicodeError):
        return "WRITE_ERROR"
    return "OK"

if __name__ == "__main__":
    print(convert("input.txt", "output.txt"))
```



Чтение	сбой → READ_ERROR; выход не открыт
Проверка всех строк	сбой → DATA_ERROR; выход не открыт
Расчёт	count и sum по проверенным значениям
Запись	OK или WRITE_ERROR; частичный файл возможен

Запись начинается только после полной проверки входа. Ошибки записи выделены отдельно.

03. Практика 1. Проверяем материал

Ошибка не является нулевым результатом

14–25 минут. Сохраните программу как lesson.py. Для каждого входа запишите ожидание, запустите и перечитайте output.txt после завершения. Каждый отказ сравните с сообщением об успешном пустом файле.

input.txt	Статус; результат при успехе
5\n7\n-2\n	OK; count=3, sum=10
Пустой файл	OK; count=0, sum=0
-007\n0\n999	OK; count=3, sum=992
7x / +7 / пробел7 / пустая строка	DATA_ERROR
1000 / девять строк по 1	DATA_ERROR

2 минуты — прогноз; 4 — обычные и граничные запуски; 3 — объясните, почему int без предварительной проверки не соблюдает формат; 2 — журнал результатов. Критерии: полный формат, ограничение 8 строк, правильная сумма, разные статусы.

04. Практика 2. Перенос способа

Проверяем сохранность прежнего файла

25–33 минуты. Проверяющий заранее сохраняет в output.txt ровно OLD. За 2 минуты прогнозирует три случая: 1\n2\nBAD, отсутствующий input.txt, корректные восемь строк по 999. За 4 минуты автор выполняет каждый запуск, заново восстанавливая OLD перед случаем; за 2 сравнивают байты результата. Критерии: OLD сохранён при первых двух ошибках; при успехе count=8, sum=7992. Используется тот же полный исходник, дописывать код не требуется.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты сравните «не записывать при неверном вводе» и «всегда сохранять старый файл», за 3 укажите момент потери второй гарантии, за 2 сформулируйте ограничение программы.

06. Индивидуальная проверка

40–45 минут. Для -2\n-2\n0 укажите статус и две строки отчёта; для 2\n\n3 назовите статус; объясните, что останется от OLD во втором случае. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

int допускает некоторые записи, запрещённые контрактом, например +7 и пробелы. Наша проверка их отсекает. -007+0+999=992. Девятая строка отклоняет весь вход. Пустой файл допустим; файл из одного LF содержит пустую недопустимую запись.

Практика 2: разбор и критерии

Первый статус DATA_ERROR, второй READ_ERROR: output.txt должен содержать ровно OLD в обоих случаях. Третий — OK и две строки count=8 и sum=7992 с LF. Дополнительный перенос: задайте dst="missing_dir/output.txt" при отсутствующей папке — WRITE_ERROR. Проверяйте в отдельной учебной папке; не создавайте путь к чужим файлам.

Обсуждение: возможный вывод

До открытия выхода ошибки чтения и данных не меняют его. При открытии w прежний файл очищается; последующий сбой записи может оставить частичный результат. Поэтому программа не обещает защиту от отключения питания или заполнения диска. Сообщение OK означает успешное завершение операций в данном запуске.

Индивидуальная проверка: ответ и критерии

OK; count=3 и sum=-4 с LF; DATA_ERROR из-за пустой второй строки; OLD сохранится, потому что выход ещё не открывался. По баллу за пункт.

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: чтение и запись файлов, try/except, функции, списки. 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	Проверить весь вход до записи результата, различить ошибки чтения и данных и доказать сохранность прежнего отчёта при неверном вводе.	Исходник, журнал тестов и объяснение результата
Типы файлов; чтение, запись и обработка данных	Проверяем сохранность прежнего файла	Самостоятельная доработка, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

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

Python: обработка исключений

<https://docs.python.org/3/tutorial/errors.html>

Python: файловый ввод и вывод

<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.