



Урок 11. Python: поток данных и признак конца

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 11 из 34 · 45 минут · Модуль «Циклы»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Показания приходят по одному, а их количество заранее неизвестно. Как посчитать сумму и среднее, не сохраняя всю последовательность?

Результат урока: Обработать поток положительных целых до нуля, посчитать количество и сумму, корректно обработать пустой поток.

01. Проблемная ситуация

Ввод: 4,7,2,0 — каждое число на отдельной строке. Ноль означает конец, а не измерение. Сравните среднее $(4+7+2)/3$ и $(4+7+2+0)/4$. Какой ответ соответствует договору ввода? Что делать, если первая строка сразу 0?

02. Понятия и разобранный пример

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

Данные приходят по одному

Поток в этой задаче — последовательный ввод: храним только текущее x , сумму $total$ и количество $count$. Начальные значения суммы и количества равны нулю. Список всех чисел не требуется.

Договор: каждое данное — положительное целое, специальный ноль завершает поток и в расчёт не входит. Первый `input` находится до `while`; следующий — в конце тела. Если забыть обновление x , положительное первое значение даст бесконечный цикл.

Среднее $total/count$ имеет смысл только при $count > 0$. Для пустого потока выводим «нет данных». Отрицательные числа и отсутствие завершающего нуля вне договора; программа будет ждать следующий ввод, поэтому все тесты содержат 0. Настоящие показания учеников не собираются: вводятся учебные числа.

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

Для 4,7,2,0 после обработки данных пары $(count, total)$: (1,4),(2,11),(3,13). Ноль останавливает цикл, пара остаётся (3,13); среднее $13/3$, в формате с двумя знаками 4.33. Код выводит количество и сумму одной строкой, среднее или сообщение второй.

```
total = 0
count = 0
x = int(input())
while x != 0:
    total = total + x
    count = count + 1
    x = int(input())
print(count, total)
if count > 0:
    print(f"{total / count:.2f}")
else:
    print("нет данных")
```



Python: поток данных и признак конца

Вход	4 → 7 → 2 → 0
Накопление	total: 4 → 11 → 13
Количество	count: 1 → 2 → 3
Маркер 0	Не входит в сумму и количество

Поток 4 → 7 → 2 → 0: сохраняются только текущий элемент, сумма и счётчик; ноль завершает ввод.

03. Практика 1. Проверяем материал

Отделяем ноль-маркер от данных

Все три потока учебные и завершаются нулём. Числа вводятся без точки с запятой: каждое в отдельной строке.

Ввод по строкам	Первая строка; вторая строка вывода
4;7;2;0	3 13; 4.33
5;0	1 5; 5.00
0	0 0; нет данных

Наберите `stream.py`. До запуска заполните пары `count,total` после каждого положительного числа. Проверьте три потока и отметьте строку кода, которая не допускает деления на ноль. Результат — программа и три протокола. Критерии: маркер не посчитан, ввод обновляется, сумма/количество верны, пустой поток обработан.

04. Практика 2. Перенос способа

Пустой поток без деления на ноль

Новая задача для того же договора потока: считать не среднее, а количество чётных положительных чисел и их сумму. Напишите `even_stream.py`; вывод — два целых одной строкой. Тесты: 4,7,2,0 → 2 6; 3,5,0 → 0 0; 0 → 0 0. Ноль не должен увеличить число чётных, хотя арифметически он чётный.

05. Обсуждение и границы вывода

Можно ли использовать 0 как признак конца, если реальные данные тоже могут быть нулевыми? Предложите альтернативу и объясните, почему молча сохранять старый договор нельзя.

06. Индивидуальная проверка

Для потока 2,8,0 запишите количество, сумму, среднее и число выполнений тела. Для пустого потока объясните, почему нельзя выводить среднее 0. Критерии: три значения, число повторений, смысл пустоты.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Ответы приведены в таблице; для 5,0 единственная пара (1,5), для 0 нет ни одного выполнения тела. Ветка `if count>0` защищает деление. Полный код практики 1 совпадает с примером. По 1 баллу за четыре критерия.

Практика 2: разбор и критерии

Сначала `while` исключает маркер, затем `if x%2==0` выбирает данные. Критерии: правильное расположение проверки, сумма только чётных, три теста.

```
count = 0
total = 0
x = int(input())
while x != 0:
    if x % 2 == 0:
        count = count + 1
        total = total + x
    x = int(input())
print(count, total)
```

Обсуждение: возможный вывод

Тогда невозможно отличить данные от маркера. Нужно заранее изменить протокол: например, сначала вводить количество элементов, либо использовать отдельную команду конца. Здесь договор положительных данных делает ноль однозначным; изменение протокола требует изменения программы.

Индивидуальная проверка: ответ и критерии

Количество 2, сумма 10, среднее 5.00; тело выполняется дважды. Пустой поток не имеет среднего: 0 было бы выдуманным измерением, а 0/0 недопустимо. 3 балла.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, локальную папку с правом записи, Python/IDLE, распечатку ученической части PDF, бумагу и ручку. Внешние библиотеки не нужны: только стандартный Python 3.11 или новее. Все программы, входные данные, таблицы и ключи находятся в статье и PDF. Детские аккаунты, сеть и регистрация не нужны.

Уровень — начинающий, 7 класс. Предпосылки: `while`, `if`, сумма и счётчик. Работа в паре: автор прогнозирует и вводит код, проверяющий ведёт журнал; затем роли меняются. На бумаге трассируйте готовый код и входные данные; для графики используйте координатную сетку и схему. Учитель может показать заранее полученный вывод/PNG. Бумажный вариант проверяет понимание, но не выдаётся за реальный запуск учеником. Чтение, ввод и проверка включены в план. Матрица программы — материал учителя.

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
Обработка потока данных	Обновляет сумму и счётчик без списка	Правильная статистика
Циклы с условием. Бесконечные циклы	Обновляет ввод и исключает маркер	Конечный и пустой потоки

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Официальное руководство Python 3.11: управление выполнением

<https://docs.python.org/3.11/tutorial/controlflow.html>

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: 17.10.2026.