



Урок 25. Python: Сумма и количество по условию

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 25 из 33 · 45 минут · Модуль «Массивы и списки»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

В пяти результатах есть ноль, отрицательное число и повторяющиеся значения. Почему число всех записей не равно числу записей, прошедших порог?

Результат урока: Одним перебором найти сумму и количество подходящих элементов, корректно обработав пустой набор.

01. Проблемная ситуация

0-5 минут. Даны учебные значения `[-2,0,3,3,7]`, порог 3. Отметьте подходящие записи по условию `x >= 3`. Сосчитайте записи, а не различные значения; объясните, почему обе тройки учитываются.

02. Понятия и разобранный пример

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

Два накопителя и одно условие

Сумма и количество — разные величины. Начинаем с `total=0` и `count=0`. Для каждого `x`, удовлетворяющего условию, прибавляем `x` к `total` и 1 к `count`. Не увеличивайте `count` вне `if`: тогда получится размер всего списка.

После обработки первых `k` элементов накопители описывают только подходящие элементы этого префикса. Это утверждение верно до цикла для пустого префикса и сохраняется каждым шагом. Поэтому после завершения оно относится ко всему списку.

Контракт: список целых и целый порог. Значения могут повторяться и быть отрицательными. Пустой список и непустой список без подходящих элементов оба дают `(0,0)`; это корректные результаты. Для среднего дополнительно требуется `count > 0`, иначе деление не выполняют.

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 `[-2,0,3,3,7]`, `t=3` первые две записи пропущены. Далее `(total,count): (3,1),(6,2),(13,3)`. При замене `>=` на `>` обе тройки исчезают: станет `(7,1)`.

```
def report(a, threshold):
    total = 0
    count = 0
    for x in a:
        if x >= threshold:
            total += x
            count += 1
    return total, count

for a, t in [([-2,0,3,3,7],3), ([],3), ([1,2],3),
             ([3,3],3), ([-5,-2],-3), ([0],0)]:
    print(a, t, report(a, t))
```



Значения	-2 0 3 3 7
Условие $x \geq 3$	нет нет да да да
Сумма	0 0 3 6 13
Количество	0 0 1 2 3

Два накопителя после каждого элемента; повторные значения являются отдельными записями.

03. Практика 1. Проверяем материал

Граница порога и повторения

14–25 минут. Для шести тестов предскажите пары. В первом тесте запишите состояния после каждого шага. Выполните предоставленный исходник.

Список; порог	(сумма, количество)
<code>[-2,0,3,3,7]; 3</code>	(13,3)
<code>[]; 3</code> и <code>[1,2]; 3</code>	(0,0) для обоих
<code>[3,3]; 3</code>	(6,2)
<code>[-5,-2]; -3</code> и <code>[0]; 0</code>	(-2,1) и (0,1)

2 минуты — расчёт, 4 — запуск, 3 — перенесите `count+=1` за пределы `if` и найдите вход, отличающий ошибочный результат от правильного, 2 — восстановите код. Критерии: пары, префиксная трассировка, граница и объяснение ошибки.

04. Практика 2. Перенос способа

Среднее только при непустой выборке

25–33 минуты. Выберите элементы в замкнутом диапазоне `lo<=x<=hi` и верните сумму, количество и среднее; при отсутствии выбранных среднее `None`. Считайте `lo<=hi` заданным условием. За 2 минуты разберите `[2,4,6,8]` для `3..7`, за 4 напишите функцию, за 2 проверьте `[]`, `[3,3]` для `3..3` и `[-2,2]` для `-2..2`. Критерии: обе границы, два накопителя, проверка нулевого количества, исходник не меняется.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты сопоставьте (0,0), (0,1), (0,2). За 3 приведите вход для каждой пары. За 2 объясните, почему удаление повторов перед подсчётом меняет задачу.

06. Индивидуальная проверка

40–45 минут. Для `[-3,3,3,8]` найдите `report` с порогом 3, `interval_report` для `3..3`, назовите пропущенные элементы во втором случае и объясните условие вычисления среднего. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Контрпример [1,2], порог 3: ошибка даёт (0,2), правильно (0,0). [0], порог 0 даёт сумму 0, но количество 1: нулевая сумма не означает отсутствия выбранных элементов. Все шесть ключей приведены в таблице.

Практика 2: разбор и критерии

Ключи: (10,2,5.0), (0,0,None), (6,2,3.0), (0,2,0.0). В последнем случае сумма нулевая, но среднее существует. Пустую выборку отличаем по количеству, а не по сумме.

```
def interval_report(a, lo, hi):
    total = 0
    count = 0
    for x in a:
        if lo <= x <= hi:
            total += x
            count += 1
    mean = total / count if count > 0 else None
    return total, count, mean
```

```
for a, lo, hi in [([2,4,6,8],3,7),([],3,7),
                 ([3,3],3,3),([-2,2],-2,2)]:
    print(interval_report(a, lo, hi))
```

Обсуждение: возможный вывод

Примеры при пороге 0: [] даёт (0,0), [0] даёт (0,1), [0,0] даёт (0,2). Повторяющиеся записи учитываются отдельно; удаление повторов считает различные значения. Среднее пустой выборки не определено в нашем контракте.

Индивидуальная проверка: ответ и критерии

report: (14,3); interval_report: (6,2,3.0); пропущены -3 и 8; делить можно только при count>0. По одному 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, 8 класс. Предпосылки: списки целых, for, if, функции и сравнения. 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
Массивы и списки	Одним перебором найти сумму и количество подходящих элементов, корректно обработав пустой набор.	Исходник, трассировка и таблица проверок
Создание, ввод и обработка массивов; перебор элементов	Среднее только при непустой выборке	Самостоятельный алгоритм, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Python: структуры данных

<https://docs.python.org/3/tutorial/datastructures.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: 18.10.2026.