



Урок 32. Текстовый отчёт по данным

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 32 из 33 · 45 минут · Модуль «Марафон по строкам и массивам»

Уровень: начинающий программист; Python 3 и C++17.

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

Две программы получили одинаковые числа, но одна написала `mean=0` для пустой серии. Как задать отчёт так, чтобы отсутствие среднего не выдавалось за измеренный ноль?

Результат урока: Собрать детерминированный текстовый отчёт из чисел и условий и сопоставить его на Python и C++.

01. Проблемная ситуация

0–5 минут. Для `[-2,0,3,3]` определите число записей, сумму и число неотрицательных. Сравните с `[]` и `[0]`. Какая строка отчёта позволит различить все три серии?

02. Понятия и разобранный пример

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

Формат отчёта — часть контракта

Отчёт имеет ровно три поля в этом порядке: `count=N;sum=S;nonnegative=K`. Пробелов нет. `count` — длина, `sum` — сумма всех значений, `nonnegative` — количество `x ≥ 0`. Повторы учитываются отдельно. Контракт: не более 100 целых от -1000 до 1000.

Сначала вычисляем числа, затем форматируем. Такая граница позволяет отдельно проверять арифметику и текст. Для `[]` три числа равны 0; для `[0]` `count=1` и `nonnegative=1`. Среднее здесь намеренно не входит в формат; его нельзя подменять нулём при отсутствии данных.

Python f-строка и C++ `std::to_string` решают одну задачу форматирования. В C++ сумма `long long` даёт запас, хотя в нашем диапазоне модуль суммы не превышает 100000. Точный порядок полей делает сравнение результатов простым и воспроизводимым.

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]` `count=4`; сумма `-2+0+3+3=4`; подходящие индексы 1,2,3, поэтому `nonnegative=3`. Итог: `count=4;sum=4;nonnegative=3`. Нулевой элемент включён в счётчик.

```
def report(a):
    total = 0
    count_nonnegative = 0
    for x in a:
        total += x
        if x >= 0:
            count_nonnegative += 1
    return (f"count={len(a)};sum={total};"
            f"nonnegative={count_nonnegative}")
```

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

Данные	-2 0 3 3
Сумма	-2 → -2 → 1 → 4
Счётчик ≥ 0	0 → 1 → 2 → 3
Строка	count=4;sum=4;nonnegative=3

Расчёт и форматирование разделены: одна входная запись даёт один шаг обработки.



03. Практика 1. Проверяем материал

Считаем по всем записям

14–25 минут. До запуска составьте четыре строки вручную, затем сравните символ в символ. Отдельно проверьте число разделителей: два.

Вход	Точный вывод
<code>[-2,0,3,3]</code>	<code>count=4;sum=4;nonnegative=3</code>
<code>[]</code>	<code>count=0;sum=0;nonnegative=0</code>
<code>[0]</code>	<code>count=1;sum=0;nonnegative=1</code>
<code>[-5,-1]</code>	<code>count=2;sum=-6;nonnegative=0</code>

2 минуты — расчёт; 4 — запуск; 3 — замените `>=0` на `>0` и покажите различающий тест; 2 — восстановите код. Критерии: три числа, ноль, порядок полей, точная строка.

04. Практика 2. Перенос способа

Одинаковая строка в двух языках

25–33 минуты. Реализуйте тот же формат на C++. За 2 минуты рассчитайте `[2,-2,2]`, за 4 сопоставьте код и запустите, за 2 сравните вывод двух языков для `[2,-2,2]`, `[]`, `[-1]` и `[0,0]`. Критерии: точное совпадение строк, отрицательная сумма, пустота и повторения.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты сравните числовую правильность с точностью формата, за 3 объясните, почему отчёт не доказывает равенство исходных серий, за 2 найдите разные серии с одинаковым отчётом.

06. Индивидуальная проверка

40–45 минут. Для `[-3,0,1,1]` напишите полный отчёт, назовите включённые в `nonnegative` индексы, объясните отличие от `[]` и приведите перестановку с тем же отчётом. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Ошибка проявится на [0]: nonnegative станет 0 вместо 1. В основной серии получится 2 вместо 3. Нулевая сумма не означает пустоту. В каждой строке ровно два символа ;, знаки минуса сохраняются.

Практика 2: разбор и критерии

Ответы: count=3;sum=2;nonnegative=2; для [] все нули; для [-1] count=1;sum=-1;nonnegative=0; для [0,0] count=2;sum=0;nonnegative=2. В строках-ответах финальная точка не входит в вывод.

```
#include <iostream>
#include <string>
#include <vector>
std::string report(const std::vector<int>& a) {
    long long sum=0;
    std::size_t k=0;
    for (int x : a) { sum+=x; if (x>=0) { ++k; } }
    return "count="+std::to_string(a.size())+
        ";sum="+std::to_string(sum)+
        ";nonnegative="+std::to_string(k);
}
int main() {
    for (auto a : std::vector<std::vector<int>>
        {{2,-2,2},{},{-1},{0,0}}) {
        std::cout << report(a) << "\n";
    }
}
```

Обсуждение: возможный вывод

[1,2] и [2,1] дают count=2;sum=3;nonnegative=2, но порядок различается. Отчёт — сводка, а не полная копия данных. Перестановка полей может быть понятна человеку, но нарушает согласованный формат для автоматической проверки.

Индивидуальная проверка: ответ и критерии

count=4;sum=-1;nonnegative=3; индексы [1,2,3]; у [] count=0; например [1,-3,1,0] даёт ту же сводку. 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. Для второй практики также подготовьте G++: g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson; запуск ./lesson. 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: строки, суммы, счётчики, сравнения, списки и vector. 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
Марафон по строкам и массивам	Собрать детерминированный текстовый отчёт из чисел и условий и сопоставить его на Python и C++.	Исходник, журнал тестов и объяснение результата
Повторение и обобщение; строки, условия и коллекции	Одинаковая строка в двух языках	Самостоятельная доработка, 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>

C++: преобразование чисел в строку

<https://eel.is/c++draft/string.conversions>

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.