



Урок 28. C++: поток данных и признак конца

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 28 из 34 · 45 минут · Модуль «Циклы: обработка потока»

Уровень: начинающий программист; C++17.

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

Заранее неизвестно, сколько чисел пришлёт датчик. Ноль — настоящее измерение, а `-1` — команда остановки. Как не включить её в сумму?

Результат урока: Обработать целочисленный поток без массива, отличить данные от стоп-сигнала и учесть пустую серию.

01. Проблемная ситуация

Поток `4 0 7 -1 99`. За 5 минут запишите, какие числа являются данными и какие строки вывода ожидаются: количество и сумма. Объясните, почему `99` не должно обрабатываться после `-1`.

02. Понятия и разобранный пример

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

Чтение, стоп-сигнал и обновление состояния

Условие `while(std::cin >> x)` продолжает работу, пока успешно читается очередное значение. В наших файлах только целые токены; конец файла завершает чтение. Некорректный текстовый токен прекращает извлечение, а не считается измерением. Контроль формата файла остаётся отдельной задачей.

Стоп-сигнал проверяют сразу после чтения и до накопления. Команда `break` выходит из ближайшего цикла. В задаче число `-1` зарезервировано, поэтому данные ограничены `0...1000`. Счётчик и сумма обновляются только для принятых данных. В памяти хранят три числа, а не весь поток.

В файле допускается максимум `100000` данных до конца. Сумма не больше 10^8 , хранится в `long long`. Пустой файл и немедленный `-1` дают `count=0, sum=0`; ноль внутри серии увеличивает `count`, но не `sum`. В терминале без стоп-сигнала программа ждёт следующего числа: это ожидание ввода, не вычислительный бесконечный цикл.

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 0 7 -1 99` вывод `3 11`. После `-1` чтение серии закончено по договорённости, `99` остаётся непрочитанным. Для файла `4 0 7` без `-1` тот же ответ получается при достижении конца файла.

```
#include <iostream>
```

```
int main() {
    long long x, count = 0, sum = 0;
    while (std::cin >> x) {
        if (x == -1) break;
        if (x < 0 || x > 1000 || count == 100000) {
            std::cout << "error\n"; return 0;
        }
        ++count;
        sum += x;
    }
    std::cout << count << " " << sum << "\n";
}
```



C++: поток данных и признак конца

Старт	count=0; sum=0
Чтение	успех? иначе конец
Стоп -1	выход до накопления
Данные	count++; sum+=x

Поток 4,0,7,-1,99: приняты 4,0,7; count=3, sum=11; -1 и хвост не включены.

03. Практика 1. Проверяем материал

Пустой поток и нулевые значения

14–25 минут. Реализуйте чтение до -1 или конца файла. Тестовые строки запускайте как отдельные серии.

Поток / конец файла	Вывод
пустой файл	0 0
-1	0 0
0 -1	1 0
4 0 7 -1 99	3 11
1000 1000 (конец)	2 2000
-2 -1	error

2 минуты — прогнозы; 4 — код; 3 — шесть проверок; 2 — отметьте, где стоит проверка -1. Продукт: программа и журнал, включая пустой файл. Критерии: ноль учтён, сигнал не суммируется, хвост после сигнала не читается, нет массива и деления на пустой счётчик.

04. Практика 2. Перенос способа

Минимум и максимум без фиктивного нуля

25–33 минуты. Новый поток — температуры: целые -50...50; стоп-сигнал 999 или конец файла. До 100000 значений. Для непустой серии выведите count, min, max, для пустой empty. За 2 минуты определите начальное состояние; за 4 напишите код; за 2 проверьте -8 -3 -5 999, 7 12 999, 999 и 51 999. Критерии: первый элемент задаёт минимум и максимум, отрицательные температуры допустимы, сигнал не участвует.

05. Обсуждение и границы вывода

33–40 минут. Когда допустимо кодировать отсутствие данных нулём? Сравните пустую серию и 0 -1. Можно ли восстановить их различие по одной сумме?

06. Индивидуальная проверка

40–45 минут. Для 0 0 5 -1 7 назовите count и sum. Для температур -4 -2 999 назовите минимум и максимум. Объясните, зачем отдельный ответ empty. Четыре пункта по баллу.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Ответы даны в таблице. Если суммировать до проверки, немедленный -1 испортит сумму. Если использовать `while(x!=−1)` до первого чтения, начальное `x` ещё не имеет данных. Конструкция с чтением в условии устраняет эту ошибку.

```
#include <iostream>

int main() {
    long long x, count = 0, sum = 0;
    while (std::cin >> x) {
        if (x == -1) break;
        if (x < 0 || x > 1000 || count == 100000) {
            std::cout << "error\n"; return 0;
        }
        ++count;
        sum += x;
    }
    std::cout << count << " " << sum << "\n";
}
```

Практика 2: разбор и критерии

Ответы: 3 −8 −3; 2 7 12; empty; error. Начальное `lo=hi=0` не используют как измерение: условие `count==0` заменяет обе границы первым `x`. Иначе серия только отрицательных чисел получила бы ложный максимум 0.

```
#include <iostream>

int main() {
    int x, count = 0, lo = 0, hi = 0;
    while (std::cin >> x) {
        if (x == 999) break;
        if (x < -50 || x > 50 || count == 100000) {
            std::cout << "error\n"; return 0;
        }
        if (count == 0 || x < lo) lo = x;
        if (count == 0 || x > hi) hi = x;
        ++count;
    }
    if (count == 0) std::cout << "empty\n";
    else std::cout << count << " " << lo << " " << hi << "\n";
}
```

Обсуждение: возможный вывод

По сумме обе дают 0; различие сохраняет `count`. Стоп-сигнал должен лежать вне допустимых данных или иметь отдельный формат. Если разрешить -1 как температуру, выбирать его сигналом нельзя; поэтому новая задача использует 999.

Индивидуальная проверка: ответ и критерии

`count=3,sum=5; min=−4,max=−2`. Empty отделяет отсутствие измерений от настоящих нулевых или отрицательных данных.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, рабочую папку с правом записи, редактор, распечатку ученической части PDF, бумагу и ручку. Локальный компилятор G++ с C++17 или настроенная учителем IDE; сборка `g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson`. В Windows задайте `lesson.exe`. Запуск `.lesson` в Linux/macOS, `.\lesson.exe` в Windows. Для урока 31 дополнительно обычный браузер: открывать локальный SVG, не C++-исходник. Все входы и ожидаемые результаты даны. Установку и проверку запуска учитель выполняет до занятия; детские аккаунты и интернет не нужны.

Уровень — начинающий программист, 7 класс. Предпосылки: `while`, `if`, накопитель и счётчик. В паре автор вводит и объясняет, проверяющий сопоставляет с прогнозом, затем роли меняются. Без компьютера трассируйте полный код на бумаге, отмечайте координаты по схеме и проверяйте



готовые результаты; учитель демонстрирует заранее собранный пример. Бумажная трассировка не выдаётся за реальную компиляцию или запуск ученика. План 45 минут включает чтение, ввод, сборку и обсуждение. В графических задачах сначала прогноз, затем сравнение. 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
Обработка потока данных	Обработать целочисленный поток без массива, отличить данные от стоп-сигнала и учесть пустую серию.	Самостоятельный продукт и тесты, описанные в практиках
Составлять и отлаживать простые диалоговые программы	Сверяет прогноз с фактическим запуском; объясняет ошибочный вариант	Журнал тестов и индивидуальный ответ по критериям

Источники и права на изображения

Исходная программа курса

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

C++: форматированное чтение чисел

<https://eel.is/c++draft/istream.formatted.arithmetic>

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.