



Урок 29. C++: цикл со счётчиком и шагом

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 29 из 34 · 45 минут · Модуль «Циклы: for и границы»

Уровень: начинающий программист; C++17.

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

В списке 3,6,9,12 последнее число должно остаться. Почему замена `<=` на `<` незаметно удаляет его?

Результат урока: Выбрать начало, условие и шаг `for`, сравнить с `while` и проверить возрастающий и убывающий перебор.

01. Проблемная ситуация

За 5 минут запишите начало, проверку и шаг для 3,6,9,12. Сравните четыре числа с тремя промежутками между ними. Укажите, сколько раз выполнится тело при условии `i<12`.

02. Понятия и разобранный пример

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

Начало, условие и шаг выполняются в разное время

В `for(начало;условие;шаг)` начало выполняется один раз, затем перед каждым телом проверяется условие, после тела выполняется шаг. Для 3...12 с шагом 3 это `i=3`, `i<=12`, `i+=3`. После печати 12 шаг даёт 15, и проверка завершает цикл.

Такой `for` можно переписать через `while`: сначала `i=3`, затем `while(i<=12){тело; i+=3;}`. Счётчик, объявленный в `for` как `int i`, доступен в его области, но не после цикла. Для общего количества и суммы заранее вводят отдельные переменные.

Нулевой шаг не приближает состояние к выходу. Для отрицательного шага применяется `i>=end`, а не `<=`. Если знак направлен от границы, наш контракт определяет пустой перебор. Малый диапазон `-100...100` гарантирует, что следующий шаг тоже помещается в используемый `int`; суммы здесь не превосходят по модулю 20100.

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

Программа печатает 3 6 9 12, затем на новой строке 4 30. Тело четыре раза, сумма 30. Замена `<=` на `<` дала бы 3 6 9 и 3 18: ошибка ровно на конечной точке.

```
#include <iostream>
```

```
int main() {
    int count = 0, sum = 0;
    for (int i = 3; i <= 12; i += 3) {
        std::cout << i << " ";
        ++count;
        sum += i;
    }
    std::cout << "\n" << count << " " << sum << "\n";
}
```

C++: цикл со счётчиком и шагом

Начало	<code>i=3</code>
Проверка	<code>i<=12</code>
Тело	печать; <code>count++</code> ; <code>sum+=i</code>
Шаг	<code>i+=3</code> → снова проверка

`for`: начало один раз; условие до тела; шаг после тела. Перебор 3,6,9,12 содержит четыре значения.



03. Практика 1. Проверяем материал

Таблица квадратов с обеими границами

14–25 минут. Для целого $0 \leq n \leq 10$ напечатайте пары i, i^2 для всех i от 0 до n включительно, каждая на своей строке.

n	Строки i и i^2
0	0 0
1	0 0; 1 1
3	0 0; 1 1; 2 4; 3 9
10	11 строк, последняя 10 100
-1 или 11	error

2 минуты — прогноз количества строк; 4 — код; 3 — тесты таблицы; 2 — эквивалентная запись через `while` на бумаге. Продукт: программа и два способа перебора. Критерии: 0 и n включены, $n=0$ даёт одну строку, умножение $i*i$, выход за диапазон отвергнут.

04. Практика 2. Перенос способа

Разный знак шага требует разного сравнения

25–33 минуты. Ввод `start, end, step` — целые $-100 \dots 100$, $step \neq 0$. Печатайте последовательность с включённой достижимой границей, затем `count, sum`. При направлении от границы список пуст и итог 0. За 2 минуты прогноз 7 $-2 -3$, за 4 код, за 2 тесты 1 6 2; 3 1 1; 5 5 -2 ; 0 3 0. Критерии: знак шага, граница, пустой случай, отказ при нуле.

05. Обсуждение и границы вывода

33–40 минут. Что понятнее для неизвестного потока и для заданного числа шагов: `while` или `for`? Защитите выбор без утверждения, что один цикл всегда лучше. Предложите тест, который отличает включённую границу от недостижимой.

06. Индивидуальная проверка

40–45 минут. Перечислите значения $8 \dots 0$ с шагом -3 , их количество и сумму. Объясните, почему `<=0` с таким стартом даёт пустой цикл. Четыре элемента по баллу.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Число строк $n+1$, а не n . Для $n=3$ последняя пара 3 9. Для $n=0$ условие $0 \leq 0$ истинно один раз. В варианте while i объявляется и обнуляется до цикла, увеличение стоит внутри тела.

```
#include <iostream>
```

```
int main() {
    int n;
    if (!(std::cin >> n) || n < 0 || n > 10) {
        std::cout << "error\n"; return 0;
    }
    for (int i = 0; i <= n; ++i)
        std::cout << i << " " << i * i << "\n";
}
```

Практика 2: разбор и критерии

7 -2 -3 → 7 4 1 -2, затем 4 10. 1 6 2 → 1 3 5, затем 3 9: число 6 недостижимо шагом. 3 1 1 → пустая строка и 0 0. 5 5 -2 → 5 и 1 5. Нулевой шаг даёт error. Полный код разделяет два направления.

```
#include <iostream>
```

```
int main() {
    int start, end, step;
    if (!(std::cin >> start >> end >> step) || start < -100
        || start > 100 || end < -100 || end > 100
        || step == 0 || step < -100 || step > 100) {
        std::cout << "error\n"; return 0;
    }
    int count = 0, sum = 0;
    if (step > 0) {
        for (int i = start; i <= end; i += step) {
            std::cout << i << " "; ++count; sum += i;
        }
    } else {
        for (int i = start; i >= end; i += step) {
            std::cout << i << " "; ++count; sum += i;
        }
    }
    std::cout << "\n" << count << " " << sum << "\n";
}
```

Обсуждение: возможный вывод

For удобно делает видимыми начало и шаг; while естественно связывается с чтением или состоянием. Оба могут реализовать многие одинаковые алгоритмы. Тест 1 7 2 включает 7, а 1 6 2 заканчивается на 5: условие не обязано создавать недостижимую точку.

Индивидуальная проверка: ответ и критерии

8,5,2; count=3,sum=15. Начальное $8 \leq 0$ ложно, поэтому тело не начнётся. Для убывания нужна проверка ≥ 0 .

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, сравнения и изменение переменной. 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
Циклы по переменной. Шаг изменения переменной цикла	Выбрать начало, условие и шаг for, сравнить с while и проверить возрастающий и убывающий перебор.	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++: цикл for

<https://eel.is/c++draft/stmt.for>

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.