



Урок 23. C++: целые числа и арифметика

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 23 из 34 · 45 минут · Модуль «Числа и выражения на C++»

Уровень: начинающий программист; C++17.

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

Сбор за упаковку берут с каждого набора, а программа прибавляет его только один раз. Как проверить смысл скобок до запуска?

Результат урока: Составить целочисленное выражение с правильным порядком операций, выбрать достаточный тип и проверить расчёт независимо.

01. Проблемная ситуация

Учебный заказ: 3 набора по 40 условных единиц и упаковка 5 с каждого набора. Сравните $3 \cdot 40 + 5$ и $3 \cdot (40 + 5)$. Выпишите все три одинаковых слагаемых и выберите выражение по условию. Это вымышленная арифметическая модель, не действующие цены.

02. Понятия и разобранный пример

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

Скобки, тип long long и границы результата

Умножение выполняется раньше сложения; скобки меняют порядок. Если сбор относится к каждому из n наборов, стоимость одного $\text{price} + \text{fee}$, а всей партии $n \cdot (\text{price} + \text{fee})$. Выражение $n \cdot \text{price} + \text{fee}$ описывает другой заказ: один сбор со всей партией. Компилятор может принять оба, но условию соответствует только одно.

Вход — три целых n , price , fee , каждое от 0 до 100000. Все переменные `long long`; в данной среде он обеспечивает диапазон как минимум 64-битного знакового целого. Максимум результата $100000 \cdot (100000 + 100000) = 20000000000$, что помещается. В обычный 32-битный `int` такое число не помещается. Даже широкий `total` не исправляет переполнение, если промежуточные множители были узкими: поэтому тип выбран у входных переменных.

Работаем только в заданном диапазоне; проверка неверных строк и чисел вне контракта здесь не добавлена. Все значения — целые условные единицы, поэтому вещественное округление не требуется. Сначала считайте вручную, затем сравнивайте машинный вывод. Порядок чтения n , price , fee важен.

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 3 40 5 результат 135, а ошибочная формула даёт 125. Для 0 40 5 — 0: ни одного набора, ни одной упаковки. Для 1 40 5 обе формулы случайно совпадут 45, поэтому такой тест один недостаточен.

```
#include <iostream>
int main() {
    long long n, price, fee;
    std::cin >> n >> price >> fee;
    long long total = n * (price + fee);
    std::cout << total << '\n';
    return 0;
}
```



C++: целые числа и арифметика

Цена одного	$40+5=45$
Три набора	$45+45+45=135$
Ошибочная модель	$3 \times 40+5=125$
Верхняя граница	$100000 \times 200000=20000000000$

Поштучный сбор: $3 \times (40+5)=135$. Один сбор со всей партии: $3 \times 40+5=125$. Разница возникает в модели, а не в синтаксисе C++.

03. Практика 1. Проверяем материал

Расчёт партии с поштучной упаковкой

14–25 минут. Запишите прогноз для всех пяти строк и объясните, какие тесты различают правильную и ошибочную формулы.

Ввод n price fee	Ожидаемый вывод
3 40 5	135
0 40 5	0
2 0 7	14
1 40 5	45
100000 100000 100000	20000000000

2 минуты — независимый расчёт, 4 — код и сборка, 3 — запуски, 2 — исправление ошибочного выражения в копии. Продукт — исходник и таблица пяти фактических ответов. Критерии: формула по смыслу, типы всех промежуточных величин, нулевая партия, большой граничный тест.

04. Практика 2. Перенос способа

Периметр и площадь как разные выражения

25–33 минуты. Новый предметный случай: прямоугольник со сторонами a , b от 1 до 100000 в сантиметрах. Программа читает два числа и выводит периметр в см и площадь в см^2 через пробел. Самостоятельно выберите выражения и типы. Проверьте 3 5, 1 1, 100000 100000. Критерии: скобки периметра, площадь, единицы, безопасный тип.

05. Обсуждение и границы вывода

33–40 минут. Достаточно ли сменить `int total` на `long long total`? Сравните тип результата и тип вычисления. Предложите тест, который обнаружит узкое промежуточное умножение, и объясните, почему успешная сборка не доказывает выполнение контракта.

06. Индивидуальная проверка

40–45 минут. Для 5 наборов по 60 и упаковки 4 с каждого найдите итог и ошибочный результат без нужных скобок. Для сторон 2 и 7 найдите периметр и площадь с единицами. Четыре результата по 1 баллу.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Прогнозы даны в таблице. Тест 1 набора не различает формулы; 3 набора и нулевая партия различают. Слишком узкий тип может испортить большой результат ещё до присваивания. Полное решение:

```
#include <iostream>
int main() {
    long long n, price, fee;
    std::cin >> n >> price >> fee;
    long long total = n * (price + fee);
    std::cout << total << '\n';
    return 0;
}
```

Практика 2: разбор и критерии

Ответы: 16 15; 4 1; 400000 100000000000. Площадь измеряется в квадратных сантиметрах, периметр — в сантиметрах. Выражение $2 \cdot a + b$ не является полным периметром. Рабочая программа:

```
#include <iostream>
int main() {
    long long a, b;
    std::cin >> a >> b;
    long long perimeter = 2 * (a + b);
    long long area = a * b;
    std::cout << perimeter << ' ' << area << '\n';
    return 0;
}
```

Обсуждение: возможный вывод

Тип выражения определяется его операндами. Если `n`, `price`, `fee` узкие, присваивание широкому `total` происходит слишком поздно. В решении все три входа `long long`. Граничный тест 200000000000 и ручная оценка максимума дополняют обычные малые проверки. Переполнение знакового целого не используют как способ получить правильный ответ.

Индивидуальная проверка: ответ и критерии

Правильно 320, ошибочно 304. Периметр 18 см, площадь 14 см². Четыре результата — 4 балла.

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 20–22 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: C++: переменные, `cin`, `cout`, сборка и точный вывод. 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 целых чисел и арифметические выражения	Переводит условие в выражение со скобками	Независимый расчёт совпадает с C++
Особенности машинных вычислений	Оценивает верхнюю границу и выбирает тип	Проверен большой вход без переполнения

Источники и права на изображения

Исходная программа курса

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

Рабочий текст стандарта C++: ввод и вывод

<https://eel.is/c++draft/input.output>

C++: минимальная ширина целочисленных типов

<https://eel.is/c++draft/basic.fundamental>

C++: результат вне диапазона типа

<https://eel.is/c++draft/expr.pre>

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.