



Урок 19. C++: ввод, переменные и диалог

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 19 из 34 · 45 минут · Модуль «Первые программы на C++»

Уровень: начинающий программист; C++17.

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

Программа всегда приветствует Mira, даже если введено Oleg. Как связать вывод с реально прочитанным значением?

Результат урока: Объявить строковую и целочисленную переменные, прочитать данные и проверить диалог на разных вводах.

01. Проблемная ситуация

Условный пользователь вводит Oleg и 0, а программа печатает Hello, Mira! и Cards: 3. Определите, какие части ответа постоянны, какие должны зависеть от ввода. За 5 минут составьте две разные входные карточки и ожидаемые результаты.

02. Понятия и разобранный пример

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

Тип переменной и извлечение из потока

Переменная хранит текущее значение под именем. `std::string` подходит для текста, `int` — для небольших целых этого урока. Заголовок `string` даёт соответствующий строковый тип, `iostream` — потоки. Объявление `int count;` ещё не считывает число. До использования `count` в выводе программа должна успешно его прочитать.

`std::cin >> name >> count` читает два значения по порядку. Пробелы и переводы строки разделяют их; `name` здесь одно слово латиницей, без пробелов, `count` — целое 0...20. Это контракт учебного ввода. Проверка неверного типа и многословные имена пока не рассматриваются; не подавайте буквы вместо числа. Реальные имена детей не нужны.

В `<< name` кавычек нет: печатается значение переменной. В `<< "name"` печаталось бы само слово `name`. Ввод, который терминал показывает при наборе, не является выводом программы. Для точной проверки сравнивайте только две сформированные программой строки. После каждого изменения исходник собирают заново командой из урока 18, изменив имя исполняемого файла на `lesson19`.

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

Ввод Mira 3 даёт Hello, Mira! и Cards: 3. Ввод Oleg 0 даёт Hello, Oleg! и Cards: 0. Ноль является допустимым количеством, не пропуском и не ошибкой.

```
#include <iostream>
#include <string>
int main() {
    std::string name;
    int count;
    std::cin >> name >> count;
    std::cout << "Hello, " << name << "!\n";
    std::cout << "Cards: " << count << "!\n";
    return 0;
}
```



C++: ввод, переменные и диалог

Ввод	Mira 3
name	Mira, тип std::string
count	3, тип int
Вывод	Hello, Mira! / Cards: 3

Два значения вводятся по порядку: name получает текст, count — целое. В выводе переменные подставляются без кавычек.

03. Практика 1. Проверяем материал

Приветствие и число карточек

14–25 минут. Создайте диалог из примера и проверьте все три входные карточки. Знак / в таблице означает перевод строки и не печатается.

Ввод	Две ожидаемые строки
Mira 3	Hello, Mira! / Cards: 3
Oleg 0	Hello, Oleg! / Cards: 0
Li 20	Hello, Li! / Cards: 20

2 минуты — прогноз, 4 — программа и сборка, 3 — три запуска, 2 — объяснение строки с переменными. Дополнительно в копии заключите name в кавычки и покажите, какой тест обнаруживает ошибку. Продукт — код, журнал трёх выводов и объяснение ошибочной версии. Критерии: два типа, правильный порядок чтения, зависимый от ввода текст, точный формат.

04. Практика 2. Перенос способа

Диалог с двумя числовыми полями

25–33 минуты. Другой диалог читает только два целых 0...20: число красных, затем синих карточек. Требуются строки Red: число и Blue: число. Самостоятельно выберите имена, объявите переменные и напишите полный исходник. Проверьте 0 5, 7 0, 20 20. Критерии: оба поля читаются в порядке условия, ноль сохраняется, значения не перепутаны.

05. Обсуждение и границы вывода

33–40 минут. Если оба числа равны 20, обнаружит ли тест перестановку переменных? Обоснуйте полезность несимметричного ввода 0 5 и сравните проверку формата с проверкой смысла.

06. Индивидуальная проверка

40–45 минут. Предскажите вывод исходного примера для Nika 12. Объясните разницу между count и "count", назовите контракт name и причину чтения до вывода. Четыре критерия по 1 баллу.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Три вывода соответствуют таблице. Вариант `<< "name"` всегда печатает `name` и теряет введённое значение. Замена приветствия готовой строкой `Mira` не решает задачу для других входов. Полный рабочий вариант:

```
#include <iostream>
#include <string>
int main() {
    std::string name;
    int count;
    std::cin >> name >> count;
    std::cout << "Hello, " << name << "!\n";
    std::cout << "Cards: " << count << '\n';
    return 0;
}
```

Практика 2: разбор и критерии

Для 0 5 строки `Red: 0` и `Blue: 5`; для 7 0 — `Red: 7` и `Blue: 0`; для 20 20 — обе 20. Сумму считать не требуется. Код нового диалога:

```
#include <iostream>
int main() {
    int red, blue;
    std::cin >> red >> blue;
    std::cout << "Red: " << red << '\n';
    std::cout << "Blue: " << blue << '\n';
    return 0;
}
```

Обсуждение: возможный вывод

При равных значениях перестановка незаметна. Разные значения показывают, какое поле попало в какой вывод. Хороший набор включает 0, верхнюю границу и различающиеся поля. Успешная сборка сама по себе не проверяет порядок.

Индивидуальная проверка: ответ и критерии

`Hello, Nika!` затем `Cards: 12`. `count` — значение переменной, `"count"` — постоянный текст. Имя одно латинское слово без пробелов. Чтение должно дать значение `count` до его использования.

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: сборка и запуск C++, `main`, `cout` и строковые литералы. 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
Диалоговые программы, переменные, консольный ввод и вывод	Читает два разных типа и формирует ответ	Три проверенных диалога
Структура программы и include	Подключает необходимые заголовки	Сборка без отсутствующих объявлений

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>

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.