



Урок 18. C++: среда, программа и первый вывод

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 18 из 34 · 45 минут · Модуль «Первые программы на C++»

Уровень: начинающий программист; C++17.

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

Файл `main.cpp` сохранён, но программа не запустилась. Чем исходный текст отличается от готовой исполняемой программы?

Результат урока: Скомпилировать и запустить первую программу C++, объяснить её строки и отличить ошибку компиляции от неверного вывода.

01. Проблемная ситуация

На столе три объекта: `main.cpp`, команда компиляции и исполняемый файл. Ученик запускает старый исполняемый файл после ошибки в новом исходнике и видит прежний результат. За 5 минут выстройте порядок действий и объясните, почему старый вывод не доказывает исправление кода.

02. Понятия и разобранный пример

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

Исходник, компиляция и выполнение

Исходник C++ — текст с расширением `.cpp`. До выполнения его переводят и собирают в исполняемую программу. В нашей среде этим управляет `g++`. Препроцессор обрабатывает `#include`; заголовок `iostream` даёт объявления для стандартного потокового ввода и вывода. `#include` не печатает строку и не требует точки с запятой.

`int main()` — начало выполнения; фигурные скобки ограничивают тело. `std::cout` обозначает стандартный выходной поток, `<<` передаёт в него текст. Двойные прямые кавычки ограничивают строку. Последовательность `\n` внутри строки означает перевод строки. Точка с запятой завершает инструкцию; `return 0` сообщает об обычном успешном завершении `main`.

Сохраните `main.cpp` в рабочей папке. В терминале этой папки выполните `g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson18`. При отсутствии ошибок запустите `./lesson18` в Linux/macOS или `.\lesson18.exe` в Windows с установленным G++. В Windows при необходимости задайте имя результата `lesson18.exe`. Альтернатива — заранее настроенная учителем локальная IDE с командами сборки и запуска, без аккаунтов. Наличие исполняемого файла после неудачной сборки не подтверждает, что он новый.

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

После успешной сборки пример печатает ровно две строки: `Hello!` и `Ready`. Ввод не нужен. Окно редактора хранит исходник, терминал показывает сообщения сборки и результат запуска; эти сообщения не являются частью ожидаемого вывода программы.

```
#include <iostream>
int main() {
    std::cout << "Hello!\n";
    std::cout << "Ready\n";
    return 0;
}
```



C++: среда, программа и первый вывод

1. Исходник	main.cpp: текст программы
2. Сборка	g++: ошибки исправляют до запуска
3. Выполнение	запуск нового исполняемого файла
4. Проверка	Hello! затем Ready

Путь C++: исходник main.cpp → успешная сборка → запуск нового исполняемого файла → проверка двух строк.

03. Практика 1. Проверяем материал

Первая проверяемая программа из трёх строк вывода

14–25 минут. Сначала соберите пример, затем сделайте свою версию из трёх строк: Start, Step 1, Ready. Все символы латинские, регистр и пробел существенны.

Проверка	Ожидаемый результат
Обычная сборка	Без сообщений об ошибке
Обычный запуск	Hello! и Ready на отдельных строках
Нет ; после первого вывода	Сборка завершается ошибкой
Восстановлена ;	Новая успешная сборка обязательна

2 минуты — порядок действий, 4 — ввод и сборка, 3 — проверка вывода и одна намеренная ошибка с ;, 2 — исправление и повторная сборка. Работайте с копией исходника, не удаляйте системные файлы. Продукт — полный исходник, три строки вывода и запись сообщения об ошибке своими словами. Критерии: корректная структура, точный вывод, успешная повторная сборка, различение сборки и запуска.

04. Практика 2. Перенос способа

Переводы строки и точный формат

25–33 минуты. Новый формат: первая строка Room 7, вторая Begin; пустой строки между ними нет. Ученик поставил два перевода строки после Room 7. Самостоятельно исправьте полный исходник, соберите и проверьте число строк. Критерии: два точных текста, один перевод между ними, успешная сборка.

05. Обсуждение и границы вывода

33–40 минут. Что доказывает успешная компиляция? Сравните программы с выводами Ready и Raedy: обе могут собраться, но одна нарушает условие. Предложите проверку, которую компилятор не выполнит вместо автора.

06. Индивидуальная проверка

40–45 минут. Объясните роли #include, main, std::cout и return 0. Затем назовите порядок «исправить исходник → ... → ...». Четыре роли оцениваются по 0,5 балла; правильный порядок — 1 балл, различение синтаксической и смысловой ошибки — 1.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Три инструкции вывода располагаются внутри `main`. Пропуск ; — ошибка синтаксиса, а не условие для вывода другой строки. После исправления обязательно собрать заново. Пример полного решения:

```
#include <iostream>
int main() {
    std::cout << "Start\n";
    std::cout << "Step 1\n";
    std::cout << "Ready\n";
    return 0;
}
```

Практика 2: разбор и критерии

Нужно по одному `\n` после каждой строки, а не два подряд после первой. Пустая строка — часть вывода и нарушает данный контракт. Ниже полный исходник нового случая.

```
#include <iostream>
int main() {
    std::cout << "Room 7\n";
    std::cout << "Begin\n";
    return 0;
}
```

Обсуждение: возможный вывод

Компилятор проверяет правила языка, но не знает, какую надпись требует задача. Нужны успешная сборка и сравнение фактического вывода с условием, включая регистр, пробелы и переводы строк. Старый файл не используют как доказательство исправления нового исходника.

Индивидуальная проверка: ответ и критерии

Заголовок подключает объявления; `main` начинает выполнение; `cout` выводит; `return 0` завершает `main` успешно. После исправления — сборка, затем запуск и проверка нового результата. Ошибка ; мешает сборке, опечатка в тексте может не мешать.

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: понятие алгоритма и опыт запуска программ Python; знание C++ не требуется. 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
Структура C++, компиляция, препроцессор и include	Собирает исходник и объясняет строки	Новый исполняемый файл и точный вывод
Простейшие программы	Разделяет синтаксическую и смысловую ошибку	Журнал исправления и повторной сборки

Источники и права на изображения

Исходная программа курса

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

GNU GCC: запуск компилятора G++

https://gcc.gnu.org/onlinedocs/gcc/Invoking-G_002b_002b.html

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.