



Урок 10. C++: Вложенные ветвления

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 10 из 33 · 45 минут · Модуль «Ветвления. Вложенные условия и границы»

Уровень: начинающий программист; C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Робот стоит в клетке (x,y). На диагонали $x=y$ он должен остановиться, а в остальных клетках выбрать направление. Как записать правило так, чтобы `else` относился именно к нужной проверке?

Результат урока: Построить дерево решений для клетки, реализовать вложенные `if` с фигурными скобками и проверить каждый конечный ответ.

01. Проблемная ситуация

0–5 минут. Учебное правило: на $x=y$ ответ `stay`; выше диагонали $y>x$ ответ `north`, ниже `south`. Для (2,2), (2,5), (5,2) запишите ответы. Один ученик проверяет только $x>0$. Найдите две точки с одинаковым x , но разными правильными ответами.

02. Понятия и разобранный пример

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

Каждая внутренняя проверка имеет свой контекст

Вложенный `if` выполняется только после выбора внешней ветви. Здесь сначала отделяем $x=y$. В ветви `else` уже известно, что координаты различны: остаётся решить, какая больше. Три ответа взаимно исключают друг друга и покрывают все допустимые входы.

В C++ `else` относится к ближайшему предшествующему `if` без собственного `else`. Отступы помогают человеку, но не меняют синтаксис. Фигурные скобки явно задают границы блоков; ставьте их у каждого условного оператора. Условие $x==y$ сравнивает, а $x=y$ присваивает: это разные действия.

Контракт примера: целые x, y от -20 до 20 . Ошибка чтения или выход за диапазон дают `error`. После проверки ввод не меняется. Для проверки дерева нужен хотя бы один вход на каждый лист и дополнительные граничные входы; один пример не доказывает все случаи.

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 $(-2, 3)$ равенство ложно, затем $y>x$ истинно: `north`. Для $(3, -2)$ обе проверки ложны: `south`. Для $(0, 0)$ внутренняя проверка не выполняется: `stay`. Оси здесь не особый случай, важна именно диагональ.

```
#include <iostream>
const char* direction(int x, int y) {
    if (x == y) {
        return "stay";
    } else {
        if (y > x) { return "north"; }
        else { return "south"; }
    }
}

int main() {
    int x, y;
    if (!(std::cin >> x >> y) || x < -20 || x > 20
        || y < -20 || y > 20) {
        std::cout << "error\n"; return 0;
    }
    std::cout << direction(x, y) << "\n";
}
```



$x = y$	да $\rightarrow$ stay
$x \neq y$ и $y > x$	north
$x \neq y$ и $y < x$	south
Проверка границ	$-20 \leq x, y \leq 20$ перед деревом

Дерево направлений: равенство проверяется первым; при неравенстве остаются два порядка координат.

03. Практика 1. Проверяем материал

Три листа дерева и привязка else

14–25 минут. Перенесите таблицу в журнал, закройте ключ и запустите полный пример на каждом входе.

Вход $x\ y$	Прогноз и путь
0 0	stay; внутренняя проверка пропущена
–20 20	north; равенства нет, $y > x$
20 –20	south; равенства нет, $y < x$
–3 –2	north
21 0; $x\ 0$	error; дерево не запускается

2 минуты — прогноз; 4 — сборка и запуски; 3 — замените $y > x$ на $y < x$ и найдите самый короткий тест, выявляющий ошибку; 2 — верните код и объясните путь. Продукт: пять строк журнала и контрпример. Критерии: все ответы, путь, обнаружение ошибки, восстановленный исходник.

04. Практика 2. Перенос способа

Дерево для двух независимых границ

25–33 минуты. Новое правило для учебного склада: целые w, h от 1 до 10; $w \leq 4$ и $h \leq 3 \rightarrow$ small; $w \leq 4$ и $h > 3 \rightarrow$ tall; $w > 4$ и $h \leq 3 \rightarrow$ wide; иначе large. Сначала проверяйте w , внутри каждой ветви h . За 2 минуты нарисуйте четыре листа, за 4 напишите программу, за 2 проверьте (4,3), (4,4), (5,3), (5,4), (0,3). Критерии: четыре ответа, включённые границы, вложенность, отказ недопустимому входу.

05. Обсуждение и границы вывода

33–40 минут. Одна пара предлагает вложенность, другая четыре независимых if. За 2 минуты найдите риск нескольких ответов, за 3 сравните на (4,3), за 2 сформулируйте условие корректности плоской записи. Должны ли отступы заменять фигурные скобки?

06. Индивидуальная проверка

40–45 минут. Для склада определите ответ (5,2), назовите две пройденные проверки, придумайте допустимый соседний вход с другим ответом и объясните роль скобок. По одному баллу за каждый пункт.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Ответы указаны в таблице. Контрпример (0,1): правильно north, после замены south. Равенство (0,0) ошибку внутренней ветви не обнаружит. Полный правильный код дан в разобранным примере; при исправлении меняется только знак сравнения.

Практика 2: разбор и критерии

Ответы small,tall,wide,large,error. Здесь h проверяется внутри обеих ветвей: общий else после внутреннего if не должен случайно относиться к внешнему условию.

```
#include <iostream>
int main() {
    int w, h;
    if (!(std::cin >> w >> h) || w<1 || w>10
        || h<1 || h>10) {
        std::cout << "error\n"; return 0;
    }
    if (w <= 4) {
        if (h <= 3) { std::cout << "small\n"; }
        else { std::cout << "tall\n"; }
    } else {
        if (h <= 3) { std::cout << "wide\n"; }
        else { std::cout << "large\n"; }
    }
}
```

Обсуждение: возможный вывод

Независимые if допустимы, если условия попарно несовместимы и покрывают контракт, например четыре полных сочетания w и h. Пропуск части условия создаёт пересечение. Скобки определяют блоки C++, отступы лишь показывают их человеку.

Индивидуальная проверка: ответ и критерии

wide; w≤4 ложно, h≤3 истинно; соседняя клетка (4,2) даёт small. Скобки задают границы блоков и принадлежность else. По баллу за ответ, путь, соседний пример и объяснение скобок.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Компилятор G++ с поддержкой C++17 и редактор. Сборка: g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson. Запуск ./lesson в Linux/macOS или .\lesson.exe в Windows (при сборке задайте -o lesson.exe). Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 8 класс. Предпосылки: bool, if/else, сравнения, чтение целых чисел. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. 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
Ветвления. Вложенные условия и границы	Построить дерево решений для клетки, реализовать вложенные if с фигурными скобками и проверить каждый конечный ответ.	Исходник, таблица прогнозов и фактических результатов
Составлять и отлаживать программы; использовать условные операторы	Дерево для двух независимых границ	Самостоятельное решение и индивидуальная проверка по четырём критериям

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Рабочий текст стандарта C++: условный оператор

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

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: 18.10.2026.