



Урок 11. C++: Границы условий и таблица тестов

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 11 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>

Учебный датчик принимает целые значения от -30 до 50 . Программа верно отвечает на 10 и 20 , но ошибается ровно на ноль. Проверим границы, а не только удобные числа.

Результат урока: Разбить целочисленный диапазон на непересекающиеся части и составить тесты до, на и после каждой границы.

01. Проблемная ситуация

0-5 минут. Правило: $-30 \dots -1 \rightarrow \text{cold}$; $0 \dots 25 \rightarrow \text{normal}$; $26 \dots 50 \rightarrow \text{hot}$; всё остальное $\rightarrow \text{error}$. В коде стоит $x \leq 0$ для cold. Какие ответы будут для $-1, 0, 1$ и какой единственный вход сразу выявляет ошибку?

02. Понятия и разобранный пример

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

Граница принадлежит ровно одному диапазону

Для целых чисел соседями границы b являются $b-1$ и $b+1$. Набор тестов строим из условия: около $-30, 0, 26, 50$ и снаружи контракта. Ожидаемый ответ записываем до запуска, иначе легко принять ошибку кода за правило.

Сначала отсекаем всё вне $-30 \dots 50$. Затем $x < 0$ означает cold. В следующем else уже известно $x \geq 0$; проверка $x \leq 25$ выделяет normal, оставшаяся часть даёт hot. Поэтому не обязательно повторять обе границы в каждом условии.

Мутация — намеренная малая ошибка для проверки силы набора тестов. Изменяйте один знак за раз, ищите вход, на котором правильная и ошибочная версии расходятся, затем восстанавливайте код. Тесты не заменяют доказательство покрытия диапазона.

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 $x=26$ внешняя защита пропускает число, $x < 0$ ложно, $x \leq 25$ ложно: hot. Для $x=50$ путь тот же. Если ошибочно написать $x \geq 50$ в защите, значение 50 станет error, хотя относится к hot.

```
#include <iostream>
const char* band(int x) {
    if (x < -30 || x > 50) { return "error"; }
    if (x < 0) { return "cold"; }
    else if (x <= 25) { return "normal"; }
    else { return "hot"; }
}
int main() {
    int x;
    if (!(std::cin >> x)) {
        std::cout << "error\n"; return 0;
    }
    std::cout << band(x) << "\n";
}
```



−31 −30 −29	error cold cold
−1 0 1	cold normal normal
25 26 27	normal hot hot
49 50 51	hot hot error

Четыре тройки тестов контролируют нижнюю, внутренние и верхнюю границы учебного датчика; значения целые.

03. Практика 1. Проверяем материал

Две мутации и минимальные контрпримеры

14–25 минут. Выполните 12 тестов таблицы, каждый вводите отдельным запуском.

Входы	Правильные ответы
−31, −30, −29	error, cold, cold
−1, 0, 1	cold, normal, normal
24, 25, 26	normal, normal, hot
49, 50, 51	hot, hot, error

2 минуты — прогноз; 4 — исходные запуски; 3 — отдельно замените $x < 0$ на $x \leq 0$, затем $x > 50$ на $x \geq 50$; 2 — восстановите и объясните ошибки. Продукт: журнал и по одному минимальному контрпримеру для каждой мутации. Критерии: 12 ответов, два точных входа, восстановленный код.

04. Практика 2. Перенос способа

Другое разбиение с нижней границей

25–33 минуты. Перенесите приём на число страниц: $1 \dots 8 \rightarrow \text{booklet}$; $9 \dots 32 \rightarrow \text{notebook}$; $33 \dots 96 \rightarrow \text{album}$; иначе error. Это учебные категории. За 2 минуты выпишите соседей 1, 9, 33, 96; за 4 напишите функцию и ввод; за 2 проверьте 0, 1, 8, 9, 32, 33, 96, 97. Критерии: точные границы, один ответ, защита и журнал.

05. Обсуждение и границы вывода

33–40 минут. Сравните набор 10, 15, 20 с набором −31, −30, 0, 25, 26, 50, 51. Защитите полезность второго, но укажите его ограничение. За 2 минуты подготовьте аргументы, за 3 обменяйтесь, за 2 запишите вывод о тестировании и доказательстве.

06. Индивидуальная проверка

40–45 минут. Для страниц укажите результат 32 и 33, вход, различающий $p \leq 8$ и $p < 8$, и правильный результат на нём. Четыре ответа — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Первая мутация обнаруживается на 0: cold вместо normal. Вторая на 50: error вместо hot. Программа для нормальных запусков дана выше. Проверка только 10 и 20 не затрагивает ни одной изменённой границы.

Практика 2: разбор и критерии

Ответы error,booklet,booklet,notebook,notebook,album,album,error. Верхний конец первого диапазона равен 8, а не 9.

```
#include <iostream>
const char* format(int p) {
    if (p<1 || p>96) { return "error"; }
    if (p<=8) { return "booklet"; }
    else if (p<=32) { return "notebook"; }
    else { return "album"; }
}
int main() {
    int p;
    if (!(std::cin >> p)) {
        std::cout << "error\n"; return 0;
    }
    std::cout << format(p) << "\n";
}
```

Обсуждение: возможный вывод

Второй набор касается каждой смены поведения и выхода из контракта. Но конечный набор сам по себе не гарантирует отсутствие ошибки на непроверенных значениях. Дополняем его анализом условий и, для небольшого целого диапазона, полным перебором.

Индивидуальная проверка: ответ и критерии

32 → notebook; 33 → album; различающий вход 8; правильный результат booklet, ошибочный notebook.

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: if/else, целые числа, сравнения и таблица проверок. 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	Разбить целочисленный диапазон на непересекающиеся части и составить тесты до, на и после каждой границы.	Исходник, таблица прогнозов и фактических результатов
Составлять и отлаживать программы; использовать условные операторы	Другое разбиение с нижней границей	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.