

Switch/Case в C++

Методичка для студентов: понятное объяснение, типовые ошибки, шаблоны, разбор примеров и практические задания.

Дата: 17 февраля 2026

Что вы получите

- Понимание, когда использовать switch, а когда if/else.
- Чёткая модель выполнения: case, break, default и «проваливание» (fallthrough).
- Как continue связан со switch (спойлер: continue относится к циклам).
- 10 интересных полностью разобранных примеров с готовым кодом.
- 15 задач для практики + 5 задач как домашнее задание.

1. Зачем нужен switch

switch - это конструкция множественного выбора по равенству. Она делает код короче и читабельнее, когда у вас много проверок вида $x == 1$, $x == 2$, $x == 3$ и так далее.

1.1. Когда switch подходит идеально

- Меню (команда 1/2/3/4).
- Состояния (режим 0/1/2).
- Символьные команды (w/a/s/d).
- Разбор кода ошибки или статуса (200/404/500).

1.2. Когда лучше if/else

- Диапазоны: $0 \leq x \leq 100$, $x > 10$, $x < 0$.
- Сложная логика: $(a > b \ \&\& \ c \neq 0) \ || \ \text{flag}$.
- Сравнение строк (string) и дробных чисел (double) напрямую.

Ограничения типа выражения switch

Можно	Нельзя (в базовом switch)
int, char, enum, short, long	string, double, float (как ключи case)

2. Как выполняется switch

Алгоритм работы:

- Вычисляется выражение в switch(...).
- Переход к case, который равен этому значению.
- Выполняется код начиная с найденного case.
- Остановка по break (или по return, throw).
- Если case не найден - выполняется default (если он есть).

2.1. break - стоп-кран

break завершает выполнение switch. Если break забыть - код «провалится» в следующий case (fallthrough).

2.2. fallthrough - проваливание

Иногда fallthrough используют специально, чтобы объединять несколько значений, но чаще это ошибка новичка.

2.3. default - страховка

default обрабатывает все неожиданные значения. В учебных задачах default нужен почти всегда.

3. Про continue: честно и понятно

Ключевой факт

`continue` относится к циклам (`for/while/do-while`), а не к `switch`. Его нельзя «использовать вместо `break`» внутри `switch` как отдельную идею - он влияет на ближайший цикл.

Если `switch` находится внутри цикла, то `continue` перескочит на следующую итерацию цикла, пропуская код после `switch`. Это часто путают.

4. Шаблоны, которые стоит запомнить

4.1. Базовый шаблон

```
int x;
cin >> x;

switch (x) {
    case 1:
        // ...
        break;
    case 2:
        // ...
        break;
    default:
        // ...
}
```

4.2. Группировка нескольких case

```
switch (month) {
    case 12: case 1: case 2:
        cout << "Winter";
        break;
    default:
        cout << "Other";
}
```

4.3. Вложенный switch для меню

```
switch (menu) {
    case 1: {
        switch (sub) {
            case 1: /* ... */ break;
            default: /* ... */
        }
        break;
    }
    default:
        /* ... */
}
```

Типовые ошибки

1) забыли break; 2) нет default; 3) в case пишут условие (case x>5: так нельзя); 4) используют switch для диапазонов; 5) путают break в цикле и break в switch.

5. 10 интересных примеров с разбором

Формат: условие → идея → код → мини-тест.

Пример 1. День недели

Условие: Ввод: число day (1..7). Вывод: название дня или Invalid.

Идея: switch по int, 7 case + default.

```
#include <iostream>
using namespace std;

int main() {
    int day;
    cin >> day;

    switch (day) {
        case 1: cout << "Monday"; break;
        case 2: cout << "Tuesday"; break;
        case 3: cout << "Wednesday"; break;
        case 4: cout << "Thursday"; break;
        case 5: cout << "Friday"; break;
        case 6: cout << "Saturday"; break;
        case 7: cout << "Sunday"; break;
        default: cout << "Invalid";
    }
    return 0;
}
```

Мини-тест: Тест: ввод 6 → вывод Saturday; ввод 9 → Invalid.

Пример 2. Калькулятор а op b

Условие: Ввод: а op b, где op один из + - * /. Вывод: результат или сообщение об ошибке.

Идея: switch по символу op. Для деления нужен if (b == 0).

```
#include <iostream>
using namespace std;

int main() {
    double a, b;
    char op;
    cin >> a >> op >> b;

    switch (op) {
        case '+': cout << a + b; break;
        case '-': cout << a - b; break;
        case '*': cout << a * b; break;
        case '/':
            if (b == 0) cout << "Division by zero";
            else cout << a / b;
            break;
        default:
            cout << "Invalid operator";
    }
    return 0;
}
```

Мини-тест: Тест: 10 / 2 → 5; 10 / 0 → Division by zero.

Пример 3. Гласная/согласная

Условие: Ввод: один символ. Вывод: Vowel/Consonant/Not a letter.

Идея: switch по char. Сначала приводим A..Z к a..z, чтобы не писать 10 case.

```
#include <iostream>
using namespace std;

int main() {
    char c;
    cin >> c;

    if (c >= 'A' && c <= 'Z') c = char(c - 'A' + 'a');

    switch (c) {
        case 'a': case 'e': case 'i': case 'o': case 'u':
            cout << "Vowel";
            break;
        default:
            if (c >= 'a' && c <= 'z') cout << "Consonant";
            else cout << "Not a letter";
    }
    return 0;
}
```

Мини-тест: Тест: E → Vowel; z → Consonant; 5 → Not a letter.

Пример 4. Месяц → количество дней

Условие: Ввод: month (1..12). Вывод: 28/30/31 или Invalid.

Идея: Группируем месяцы с одинаковым ответом через несколько case подряд.

```
#include <iostream>
using namespace std;

int main() {
    int m;
    cin >> m;

    switch (m) {
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            cout << 31; break;
        case 4: case 6: case 9: case 11:
            cout << 30; break;
        case 2:
            cout << 28; break;
        default:
            cout << "Invalid month";
    }
    return 0;
}
```

Мини-тест: Тест: 2 → 28; 11 → 30; 13 → Invalid month.

Пример 5. Сезон по месяцу (группировка case)

Условие: Ввод: month (1..12). Вывод: Winter/Spring/Summer/Autumn.

Идея: Группировка case - самый чистый способ.

```
#include <iostream>
using namespace std;

int main() {
    int m;
    cin >> m;

    switch (m) {
        case 12: case 1: case 2: cout << "Winter"; break;
        case 3: case 4: case 5: cout << "Spring"; break;
        case 6: case 7: case 8: cout << "Summer"; break;
        case 9: case 10: case 11: cout << "Autumn"; break;
        default: cout << "Invalid month";
    }
    return 0;
}
```

Мини-тест: Тест: 12 → Winter; 4 → Spring.

Пример 6. Меню приложения (команды 1..4)

Условие: Ввод: cmd. 1-Profile, 2-Settings, 3-Help, 4-Exit.

Идея: switch по int. Паттерн меню встречается постоянно.

```
#include <iostream>
using namespace std;

int main() {
    int cmd;
    cin >> cmd;

    switch (cmd) {
        case 1: cout << "Open Profile"; break;
        case 2: cout << "Open Settings"; break;
        case 3: cout << "Open Help"; break;
        case 4: cout << "Exit"; break;
        default: cout << "Unknown command";
    }
    return 0;
}
```

Мини-тест: Тест: 3 → Open Help; 0 → Unknown command.

Пример 7. WASD управление (switch по char)

Условие: Ввод: символ команды. Вывод: Up/Left/Down/Right/Unknown.

Идея: switch по char - коротко и понятно.

```
#include <iostream>
using namespace std;

int main() {
    char k;
    cin >> k;

    switch (k) {
        case 'w': cout << "Up"; break;
        case 'a': cout << "Left"; break;
        case 's': cout << "Down"; break;
        case 'd': cout << "Right"; break;
    }
```

```
        default: cout << "Unknown";
    }
    return 0;
}
```

Мини-тест: Тест: w → Up; x → Unknown.

Пример 8. Конвертер единиц (cm/m/km) через 2 switch

Условие: Ввод: value from to, где 1=cm,2=m,3=km. Вывод: число или ошибка.

Идея: Правильная стратегия: сначала переводим всё в метры, потом из метров в целевую единицу.

```
#include <iostream>
using namespace std;

int main() {
    double value;
    int from, to;
    cin >> value >> from >> to;

    double meters;

    switch (from) {
        case 1: meters = value / 100.0; break;    // cm -> m
        case 2: meters = value; break;           // m -> m
        case 3: meters = value * 1000.0; break;   // km -> m
        default: cout << "Invalid FROM"; return 0;
    }

    double result;
    switch (to) {
        case 1: result = meters * 100.0; break;   // m -> cm
        case 2: result = meters; break;           // m -> m
        case 3: result = meters / 1000.0; break;   // m -> km
        default: cout << "Invalid TO"; return 0;
    }

    cout << result;
    return 0;
}
```

Мини-тест: Тест: 150 1 2 → 1.5 (150 cm -> 1.5 m).

Пример 9. Вложенное меню (nested switch)

Условие: Ввод: menu. Если menu==1, ввод sub. Вывод: действие или ошибка.

Идея: Показывает, как делать подменю без if/else-лестниц.

```
#include <iostream>
using namespace std;

int main() {
    int menu;
    cin >> menu;

    switch (menu) {
        case 1: {
            int sub;
            cin >> sub;
```



```

        switch (sub) {
            case 1: cout << "Login..."; break;
            case 2: cout << "Logout..."; break;
            default: cout << "Invalid sub";
        }
        break;
    }
    case 2:
        cout << "Help opened";
        break;
    case 3:
        cout << "Bye";
        break;
    default:
        cout << "Invalid menu";
    }
    return 0;
}

```

Мини-тест: Тест: 1 2 → Logout...; 2 → Help opened.

Пример 10. switch внутри цикла + continue

Условие: Ввод: 5 команд. Команда 0 означает «пропустить». Для остальных выводим действие.

Идея: continue пропускает итерацию цикла целиком. Это не «замена break в switch».

```

#include <iostream>
using namespace std;

int main() {
    for (int i = 0; i < 5; i++) {
        int cmd;
        cin >> cmd;

        if (cmd == 0) continue; // пропускаем итерацию, switch не выполняем

        switch (cmd) {
            case 1: cout << "Ping\n"; break;
            case 2: cout << "Pong\n"; break;
            default: cout << "?\n";
        }
    }
    return 0;
}

```

Мини-тест: Тест: ввод 0 1 2 9 0 → (пропуск) Ping Pong ? (пропуск)

6. Практика на уроке: 15 задач

Эти задачи решаются студентами самостоятельно. Требование: использовать switch там, где это уместно. Если нужно обработать диапазон - добавляйте if.

- 1) День недели: day (1..7) -> название, иначе Invalid.
- 2) Месяц -> сезон (Winter/Spring/Summer/Autumn).
- 3) Месяц -> число дней (28/30/31).
- 4) Светофор: 1=Red,2=Yellow,3=Green -> STOP/READY/GO.
- 5) Рабочий/выходной: day (1..7) -> Workday/Weekend.
- 6) Роль: 1 Admin, 2 Moderator, 3 User -> описание доступа.
- 7) Меню кафе: 1 Tea(300), 2 Coffee(500), 3 Cola(450) -> чек.
- 8) Оценка: 1..5 -> текстовая оценка.
- 9) Команды игры: w/a/s/d/q -> действие.
- 10) Квартал года: month (1..12) -> Q1..Q4.
- 11) ASCII-категория: символ -> digit/letter/other.
- 12) Вложенное меню Настройки: Sound/Graphics/Exit + подменю.
- 13) Простой конвертер температуры: mode 1(C->F),2(F->C).
- 14) Статус HTTP: 200/201/400/401/403/404/500 -> текст.
- 15) Скидка по коду: code 1=5%,2=10%,3=20%, else 0%.

7. Домашнее задание: 5 задач

Сдача: один файл main.cpp. Каждая задача может быть отдельной программой (можно 5 файлов), или одной программой с меню выбора задачи.

ДЗ 1 - Калькулятор

Ввод: a op b. Вывод: результат или Invalid operator или Division by zero. Обязательно switch(op).

ДЗ 2 - Дни месяца

Ввод: month. Вывод: 28/30/31 или Invalid month. Используйте группировку case.

ДЗ 3 - WASD + Quit

Ввод: char. w/a/s/d/q -> Up/Left/Down/Right/Quit, иначе Unknown.

ДЗ 4 - Вложенное меню

Ввод: menu (1..3). Если menu==1, ввод sub (1..2). Реализовать nested switch.

ДЗ 5 - Скидка

Ввод: price code. code: 1=5%,2=10%,3=20%, else 0%. Вывести итоговую цену.

Чек-лист перед сдачей

- В каждом switch есть default (или вы объяснили, почему он не нужен).
- Нет случайного fallthrough: после case есть break (если проваливание не задумано).
- Деление на 0 обработано.
- Ввод/вывод соответствует условию.
- Код компилируется без ошибок.