

C++ 逻辑操作符详解

目录

- 逻辑操作符简介
- 逻辑与 (&&)
- 逻辑或 (||)
- 逻辑非 (!)
- 短路求值特性
- 真值表
- 示例代码
- 常见错误

1. 逻辑操作符简介

用于组合/修改布尔表达式，常用于条件判断。C++提供三个逻辑操作符：

- `&&` (逻辑与)
- `||` (逻辑或)
- `!` (逻辑非)

2. 逻辑与 (&&)

- 语法：表达式1 `&&` 表达式2
- 规则：仅当两个操作数都为 `true` 时返回 `true`
- 示例：

```
1 (age >= 18) && (hasID == true) // 判断是否成年且持有身份证
```

3. 逻辑或 (||)

- 语法：表达式1 `||` 表达式2
- 规则：任意一个操作数为 `true` 即返回 `true`
- 示例：

```
1 (temperature < 0) || (temperature > 100) // 判断温度是否异常
```

4. 逻辑非 (!)

- 语法：`!`表达式
- 规则：反转布尔值（真变假，假变真）

- 示例:

```
1 | !isDoorLocked // 门未锁定时为true
```

5. 短路求值特性

- 重要特性：当表达式结果可以确定时，停止后续计算
 - `&&`：遇到第一个 `false` 即停止
 - `||`：遇到第一个 `true` 即停止
- 优势：
 - 提升性能
 - 避免无效操作（如指针判空后访问）

6. 真值表

A	B	A && B	A B	!A
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

7. 示例代码

```
1 | #include <iostream>
2 | using namespace std;
3 |
4 | bool check(int value)
5 | {
6 |     cout << "Checking " << value << "... ";
7 |     return value > 10;
8 | }
9 |
10 | int main()
11 | {
12 |     int x = 5, y = 20;
13 |
14 |     // 逻辑与 (&&) 示例
15 |     cout << "Logical AND (&&):\n";
16 |     if (check(x) && check(y)) {
17 |         cout << "Both values are valid\n";
18 |     } else {
```

```

19         cout << "At least one value is invalid\n";
20     }
21
22     // 逻辑或 (||) 示例
23     cout << "\nLogical OR (||):\n";
24     if (check(x) || check(y)) {
25         cout << "At least one value is valid\n";
26     }
27
28     // 逻辑非 (!) 示例
29     cout << "\nLogical NOT (!):\n";
30     bool isOpen = false;
31     if (!isOpen) {
32         cout << "The door is closed\n";
33     }
34
35     // 短路求值演示
36     cout << "\nShort-circuit evaluation:\n";
37     int* ptr = nullptr;
38     if (ptr != nullptr && *ptr > 0) { // 安全访问
39         cout << "Valid pointer value";
40     } else {
41         cout << "Invalid pointer or value";
42     }
43
44     return 0;
45 }

```

输出结果：

```

1 Logical AND (&&):
2 Checking 5... At least one value is invalid
3
4 Logical OR (||):
5 Checking 5... Checking 20... At least one value is valid
6
7 Logical NOT (!):
8 The door is closed
9
10 Short-circuit evaluation:
11 Invalid pointer or value

```

8. 常见错误

1. 混淆位操作符：

```

1 | if (a & b) // 位与操作 (通常需要写成 a && b)

```

2. 错误优先级：

```
1 | if (a || b && c)    // && 优先级高于||, 实际为 a || (b && c)
```

3. 冗余判断:

```
1 | if (isValid == true) → 应简化为 if (isValid)
```

4. 误用逻辑非:

```
1 | if (!x > 5)        // 实际解析为 (!x) > 5
```

总结

- 使用逻辑操作符构建复杂条件判断
- 利用短路特性优化代码安全性和效率
- 注意操作符优先级（建议使用括号明确优先级）
- 区分逻辑操作符和位操作符

通过合理使用逻辑操作符，可以编写出更清晰、高效的C++代码。