

## Логика

Программистам часто приходится иметь дело с логическими задачами. Освоив формальную логику можно успешно справляться с элементарными задачами.

Для решения логических задач могу использоваться различные операторы (+, \*, -, !=, &&, ...). В формальной логике переменные и операторы указываются на достоверность, т. е. Выражают не числа а **истинность** (true) или **ложность** (false). Например «если вода в бассейне тёплая, то я буду плавать». Эту задачу можно преобразовать в логические переменные:

А: Вода в бассейне тёплая

В: Я плаваю

Они либо истинные, либо ложные. Переменная В не может быть наполовину истинной, потому, что я не способен плавать лишь отчасти. Зависимость между А и В выражает идею, что А = true влечёт за собой В = true. Иначе никак. Т.е. А == В означает, что я буду плавать, если вода в бассейне тёплая.

Пример 1:

```
// При каких условиях можно поплавать?
bool a = false; // Вода холодная
bool b = false; // Я плаваю
cout << "Вода холодная? (1/0)" << endl;
cin >> a;
if (!a) { // Если a == 1
    b = !b; // То b - true
}
else {
    b = b; // Иначе b == false
}
cout << "Плаваем или нет? - " << b << endl;
```

Пример 2:

```
// При каких условиях сервер будет исправен?
cout.setf(ios::boolalpha);
bool a = true; // Сервер неизбежно нагревается во время работы
bool b = false; // Состояние кондиционера
bool c = false; // Состояние кулера процессора
bool d = false; // Состояние сервера
cout << "Кондиционирование отключено (1/0)?" << endl;
cin >> b;
cout << "Охлаждение не работает (1/0)?" << endl;
cin >> c;
```

```
// d = (a && b) || (a && c); // Как вариант проверки условий
```

```
d = a && (b || c); // Проверка условий
cout << "Компьютер вышел из строя?: " << d << endl;
```

## Законы логических операций

### Ассоциативность

1.  $(a \parallel b) \parallel c$  **это**  $a \parallel (b \parallel c)$
2.  $(a \&\& b) \&\& c$  **это**  $a \&\& (b \&\& c)$
3.  $(a \wedge b) \wedge c$  **это**  $a \wedge (b \wedge c)$

### Коммутативность

1.  $a \parallel b$  **это**  $b \parallel a$
2.  $a \&\& b$  **это**  $b \&\& a$
3.  $a \wedge b$  **это**  $b \wedge a$
4.  $a == b$  **это**  $b == a$

### Дистрибутивность

1.  $a \parallel (b \&\& c)$  **это**  $(a \parallel b) \&\& (a \parallel c)$
2.  $a \&\& (b \parallel c)$  **это**  $(a \&\& b) \parallel (a \&\& c)$
3.  $a \parallel (b \wedge c)$  **это**  $(a \parallel b) \wedge (a \parallel c)$

### Двойное отрицание

1.  $!!a$  **это**  $a$

### Законы Де Моргана

1.  $!(a \parallel b)$  **это**  $(!a) \&\& (!b)$
2.  $!(a \&\& b)$  **это**  $(!a) \parallel (!b)$

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

```
int main() {
    cout << boolalpha;
    bool a(true), b(true), c(true);
    cout << "Ассоциативность\n";
    cout << "(a || b) || c - " << ((a || b) || c) << " - тоже самое, что a || (b || c) - " << (a || (b || c)) <<
endl;
    cout << "(a && b) && c - " << ((a && b) && c) << " - тоже самое, что a && (b && c) - " <<
(a && (b && c)) << endl;
    cout << "(a ^ b) ^ c - " << ((a ^ b) ^ c) << " - тоже самое, что " << " a ^ (b ^ c) - " << (a ^ (b ^
c)) << endl;
    cout << "Коммутативность\n";
    cout << "a && b - " << (a && b) << " - тоже самое, что b && a - " << (b && a) << endl;
    cout << "a || b - " << (a || b) << " - тоже самое, что b || a - " << (b || a) << endl;
    cout << "a ^ b - " << (a ^ b) << " - тоже самое, что b ^ a - " << (b ^ a) << endl;
    cout << "a == b - " << (a == b) << " - тоже самое, что b == a - " << (b == a) << endl;
    cout << "Дистрибутивность\n";
    cout << "a && (b || c) - " << (a && (b || c)) << " - тоже самое, что (a && b) || (a && c) - " <<
((a && b) || (a && c)) << endl;
```

```

    cout << "a || (b && c) - " << (a || (b && c)) << " - тоже самое, что (a || b) && (a && c) - " <<
((a || b) && (a && c)) << endl;
    cout << "a && (b ^ c) - " << (a && (b ^ c)) << " - тоже самое, что (a && b) ^ (b && c) - " <<
((a && b) ^ (b && c)) << endl;
    cout << "Двойное отрицание\n";
    cout << "!!a это " << !!a << endl;
    cout << "Законные Де Моргана\n";
    cout << "!(a && b) - " << !(a && b) << " это тоже самое, что (!a) || (!b) - " << ((!a) || (!b)) <<
endl;
    cout << "!(a || b) - " << !(a || b) << " это тоже самое, что (!a) && (!b) - " << ((!a) && (!b)) <<
endl;
    return 0;
}
// Output:
/*
Ассоциативность
(a || b) || c - true - тоже самое, что a || (b || c) - true
(a && b) && c - true - тоже самое, что a && (b && c) - true
(a ^ b) ^ c - 1 - тоже самое, что a ^ (b ^ c) - 1
Коммутативность
a && b - true - тоже самое, что b && a - true
a || b - true - тоже самое, что b || a - true
a ^ b - 0 - тоже самое, что b ^ a - 0
a == b - true - тоже самое, что b == a - true
Дистрибутивность
a && (b || c) - true - тоже самое, что (a && b) || (a && c) - true
a || (b && c) - true - тоже самое, что (a || b) && (a || c) - true
a && (b ^ c) - false - тоже самое, что (a && b) ^ (a && c) - 0
Двойное отрицание
!!a это true
Законные Де Моргана
!(a && b) - false это тоже самое, что (!a) || (!b) - false
!(a || b) - false это тоже самое, что (!a) && (!b) - false
*/

```

### Задание 1

Курьер должен доставить заказ вовремя. Но в городе могут быть **пробки**. К тому же автомобиль курьера не должен **сломаться** или попасть в **дтп**.  
При каких условиях курьер доставит заказ вовремя?

### Задание 2

Человек должен утром сесть в автобус и доехать до работы. Но ему могут помешать — **дождь**, отсутствие **зонта** или **автобуса**.  
При каких условиях человек попадёт на работу?

Задания для самостоятельной работы

Тест

Какой результат следующих выражений?

Выражение №1: (true && true) || false

Выражение №2: (false && true) || true

Выражение №3: (false && true) || false || true

Выражение №4: (5 > 6 || 4 > 3) && (7 > 8)

Выражение №5: !(7 > 6 || 3 > 4)

Задание 3

Проставить результаты в ячейках:

<b>a</b>	<b>b</b>	<b>a &amp;&amp; b</b>	<b>a    b</b>	<b>a == b</b>	<b>a != b</b>
0	0				
0	1				
1	0				
1	1				