Лекция №5
по дисциплине
«ТЕОРИЯ АЛГОРИТМОВ»

ПЕРЕМЕННЫЕ ЧАСТЬ2

Преподаватель:
Золотоверх Д.О.

ЧТО ЭТО

- Переменная — поименованная, адресуемая область памяти;
- При осуществлении доступа к переменной производится доступ к данным;
- Данные, находящиеся в переменной, называются значением этой переменной;
- Существуют простые и сложные.

ОСНОВНЫЕ ПРОСТЫЕ ТИПЫ ДАННЫХ В C++

Не имеют внутреннюю структуру.

Распространённые простые типы:

- int — целочисленный тип данных.
- float — тип данных с плавающей запятой.
- double — тип данных с плавающей запятой двойной точности.
- char — символьный тип данных.
- bool — логический тип данных.

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

int main() {
    bool status;
    status = true;
    int answer = 42;
    double pi = 3.14;
    char d = 'd';
}
```

BOOL

Логический тип данных

Может иметь два значения (true или false)

Занимает один байт памяти

Синтаксис:

```
bool var = true;
bool locked = false;
bool casted = 1;
```

В памяти хранится:

true 0000 0001

false 0000 0000

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

int main() {
    bool isCodingFun = true;
    bool isFishTasty = false;

    // Выводит 1 (true)
    cout << isCodingFun;
    // Выводит 0 (false)
    cout << isFishTasty;
}
```

BOOL

Логический тип данных

Может иметь два значения (true или false)

Занимает один байт памяти

Синтаксис:

```
bool var = true;
bool locked = false;
bool casted = 1;
```

- Операторы:
  - ! отрицание
  - & оператор И
  - | оператор ИЛИ
  - ^ Исключ. ИЛИ

x	y	x&y	x|y	x^y
true	true	true	true	false
true	false	false	true	true
false	true	false	true	true
false	false	false	false	false

INT (ЦЕЛОЕ ЧИСЛО)

Тип данных для хранения числа

Может хранить значение от

−2 147 483 648 до 2 147 483 647

или от -2^{31} до $2^{31} - 1$

Первый бит отображает знак

Синтаксис:

```
int amount = -416;
int num = 1000 - 7;
```

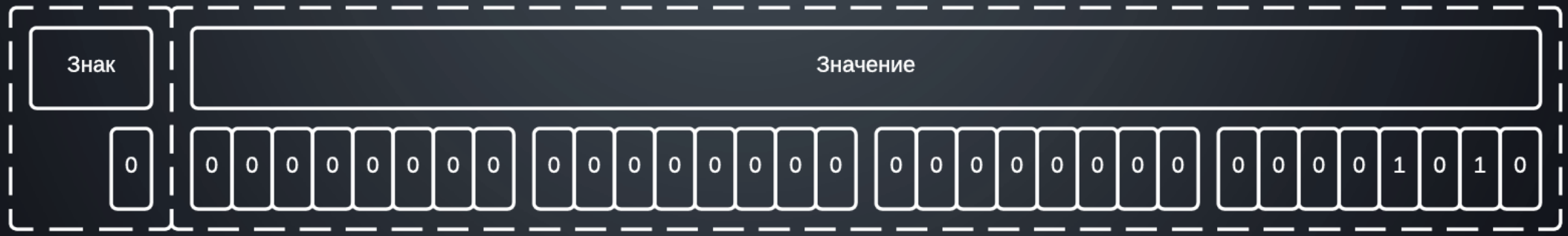
Арифметические операторы:

- + Сложение
- Вычитание
- * Умножение
- / Деление
- % Деление по модулю

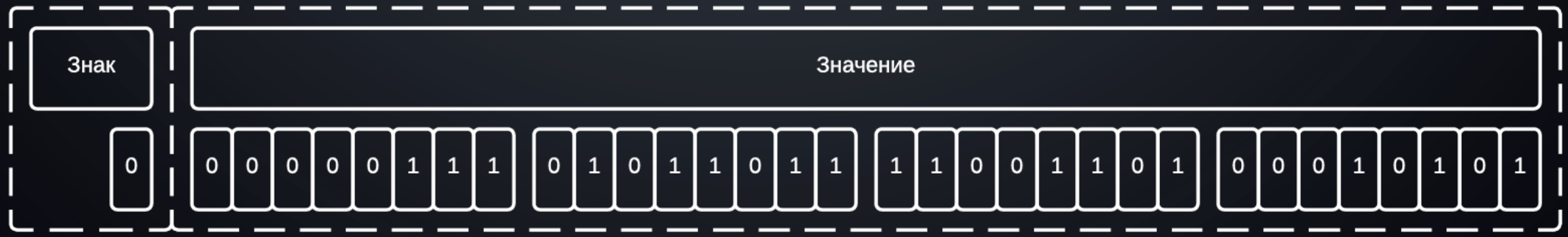
```
#include <iostream>
using namespace std;

int main() {
    int quantity;
    int previous = 0;
    int next = 1;
    int buffer = next;
    cin >> quantity;
    while (quantity > 0) {
        cout << next << endl;
        buffer = next;
        next += previous;
        previous = buffer;
        quantity--;
    }
}
```

INT (ЦЕЛОЕ ЧИСЛО)

10



123456789



INT (ЦЕЛОЕ ЧИСЛО)

Помимо арифметических операторов, также можно использовать логические и побитовые

Логические операторы:

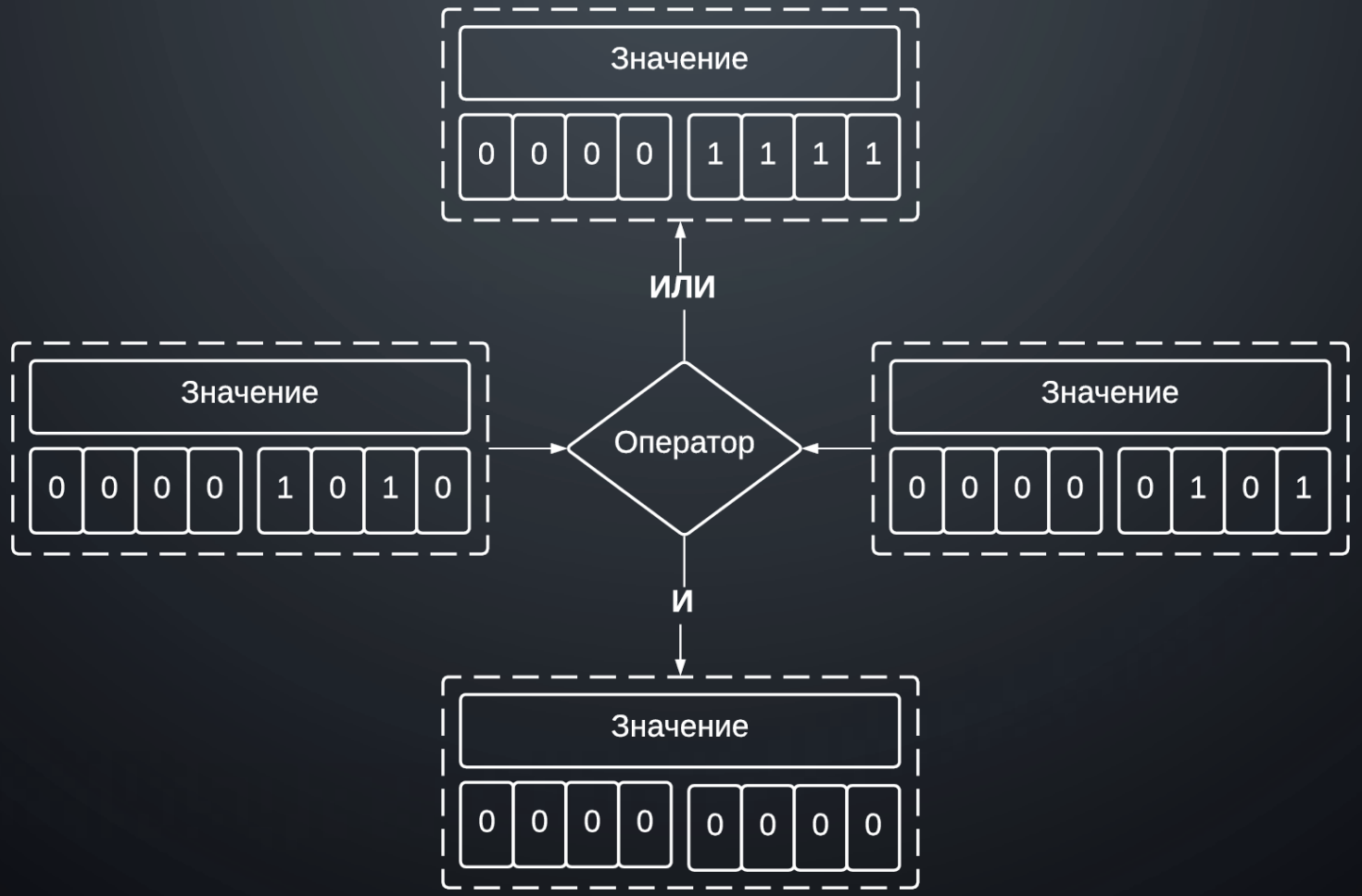
- > Больше
- < Меньше
- == Равно
- >= Больше равно
- <= Меньше равно

Побитовые операторы:

- & Битовый И
- | Битовый ИЛИ
- ^ Битовый Искл. И
- << Битовый сдвиг Влево
- >> Битовый сдвиг Вправо

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

int main() {
    int quantity;
    int previous = 0;
    int next = 1;
    int buffer = next;
    cin >> quantity;
    while (quantity > 0) {
        cout << next << endl;
        buffer = next;
        next += previous;
        previous = buffer;
        quantity--;
    }
}
```

INT ПОБИТОВЫЕ ОПЕРАТОРЫ



FLOAT (ЧИСЛО С ПЛАВ. ЗАПЯТОЙ)

Тип данных для хранения числа

- Может хранить значение
- от -3.4028235E+38
- до 3.4028235E+38
- с точностью до 3.4028235E-38

Синтаксис:

```
float one = 1.0;
float result = 3.0 / 2;
```

Операторы:

- Арифметические
- Логические
- Побитовые???

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

int main() {
    float num1 = 3.0f;
    float num2 = 3.5f;
    float num3 = 3E-5f;

    double num4 = 3.0;
    double num5 = 3.5;
    double num6 = 3E-5;
}
```

FLOAT (ЧИСЛО С ПЛАВ. ЗАПЯТОЙ)

1.2

14660154687500

ПРЕОБРАЗОВАНИЯ

В C++ есть возможность конвертации типов данных.

Способы конвертации:

- конвертация C-style;
- применение оператора static_cast;
- применение оператора const_cast;
- применение оператора dynamic_cast;
- применение оператора reinterpret_cast.

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


int main() {
    char c = 97;
    cout << static_cast<int>(c) << endl;
    // в результате выведется 97, а не 'a'
}
```

СПАСИБО ЗА ВНИМАНИЕ!