Лекция №4
по дисциплине
«ТЕОРИЯ АЛГОРИТМОВ»

ПЕРЕМЕННЫЕ

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.

ТИПИЗАЦИЯ

Бывает динамическая и статическая:

- Статическая – типы данных значения переменной не может меняться при выполнении программы.

Как следствие – простой машинный код, но сложнее процесс программирование.

- Динамическая – наоборот, значения могут менять тип данных.
- C++ имеет статическую типизацию

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.

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

int main() {
    bool 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;
}
```

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;
```

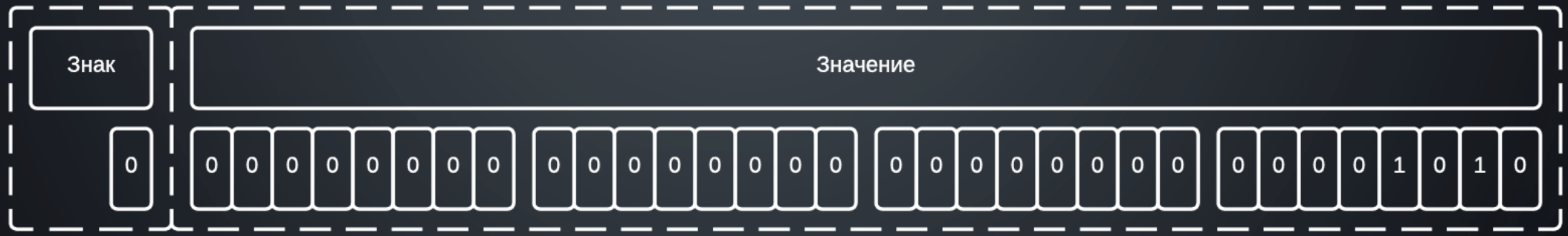
В памяти хранится:

1	0000 0000 0000 0001
-1	1111 1111 1111 1111
993	0000 0011 1110 0001
-32768	1000 0000 0000 0000

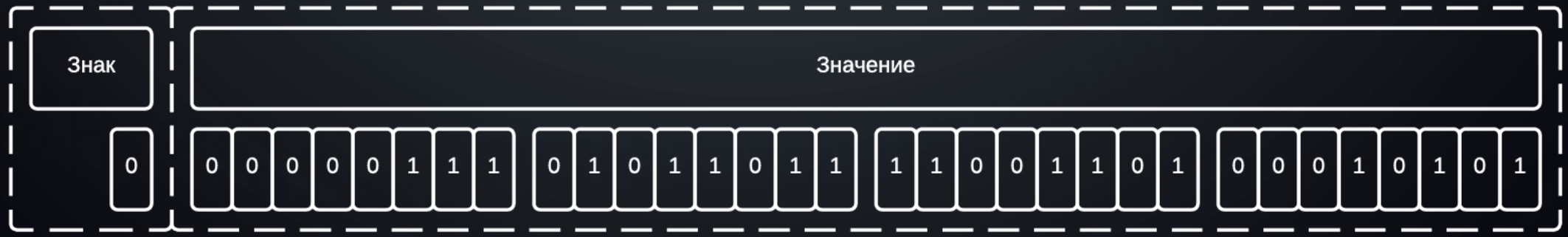
```
#include <iostream>
using namespace std;

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



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;
```

В памяти хранится:

0	0000 0000 0000 0000 0000 0000 0000 0000
1.0	0011 1111 1000 0000 0000 0000 0000 0000
123.321	0100 0010 1111 0110 1010 0100 0101 1010

```
#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

CHAR (БУКОВКА)

- Тип данных для хранения символов

  Может хранить целочисленное значение от 0 до 255

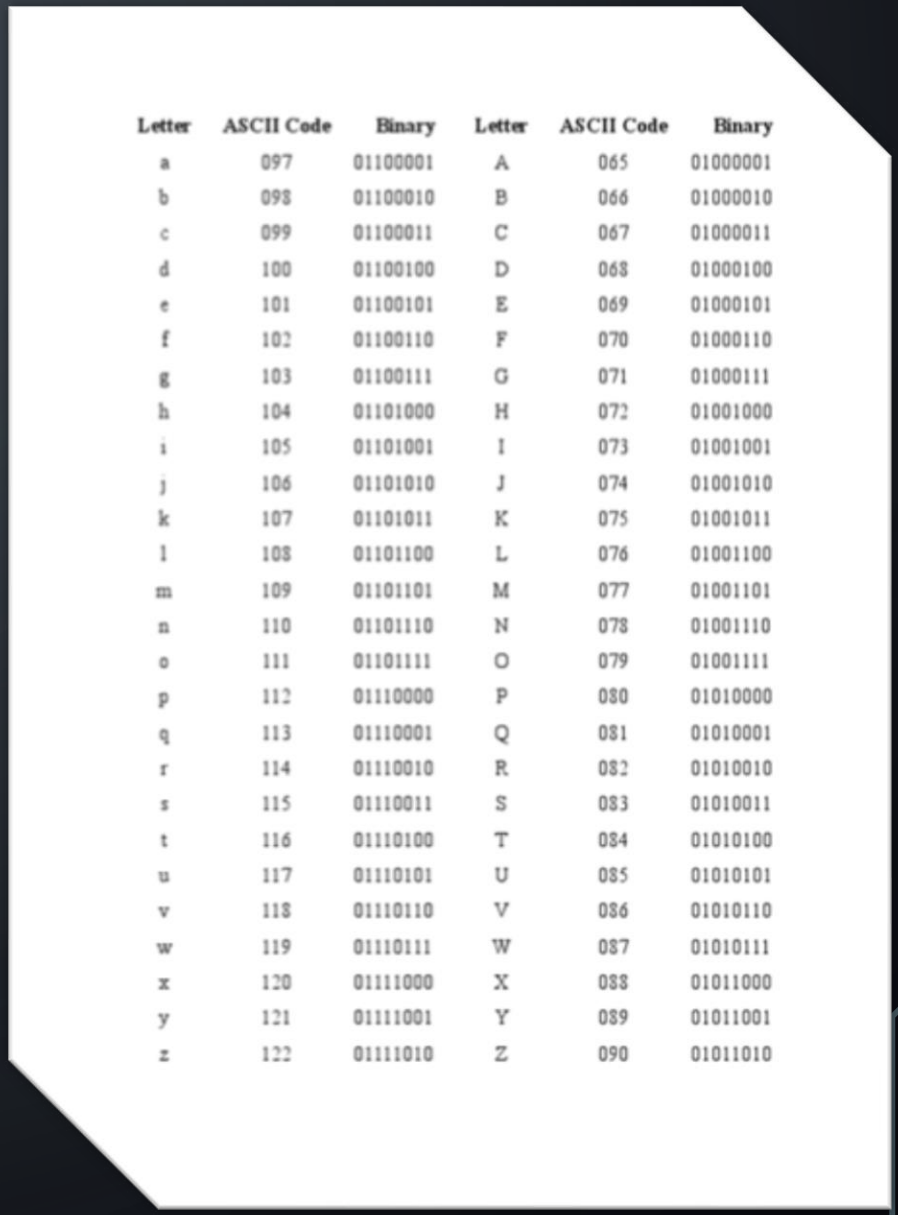
  или от 0 до $2^8 - 1$

- Занимает один байт памяти
- Синтаксис:

```
char letterA = 'a';
char tabSymb = '\t';
```

- В памяти хранится:

'a'	0110 0001
'A'	0100 0001
'1'	0011 0000



Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

СПАСИБО ЗА ВНИМАНИЕ!