Лекция №8
по дисциплине
«ТЕОРИЯ АЛГОРИТМОВ»

ПЕРЕМЕННЫЕ ЧАСТЬ3

Преподаватель:
Золотоверх Д.О.

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';
}
```

ОСНОВНЫЕ СЛОЖНЫЕ ТИПЫ ДАННЫХ В C++

Имеют внутреннюю структуру.

Распространённые сложные типы:

- array — массив элементов заданного типа;
- string — произвольная последовательность символов;
- struct — композитный тип данных;
- class — определяемый пользователем структура данных.

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

int main() {
    int array[] = {1, 2, 3, 4};
    string str = "Hello";

    struct Coor {
        float x;
        float y;
    };
}
```

ХРАНЕНИЕ ДАННЫХ В (ОЗУ)

В каждой ячейке хранится 8 бит

8 бит = 1 байт

МАССИВ (ARRAY)

Набор однотипных данных

это область памяти, где могут последовательно храниться несколько значений.

Синтаксис:

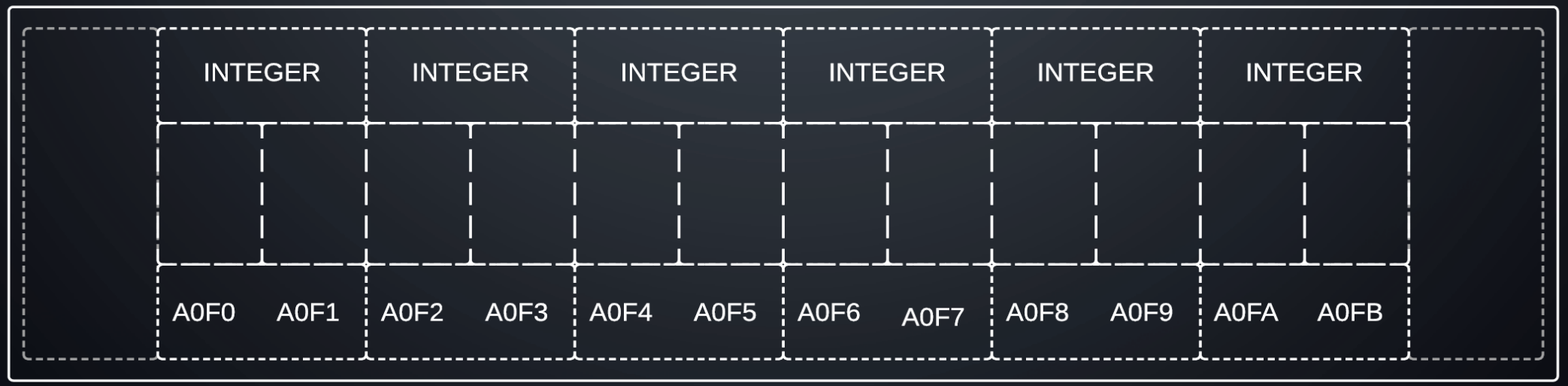
```
int myInts[6];
int myPins[] = {2, 4, 8, 3, 6};
int myVals[6] = {2, 4, -8, 3, 2};
char message[6] = "hello";
```

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


int main() {
    int array[] = {3, 2, 1};
    for (int i = 0; i < 3; i++)
{
        printf("%d\n",
array[i]);
    }
}
```

МАССИВ В ПАМЯТИ

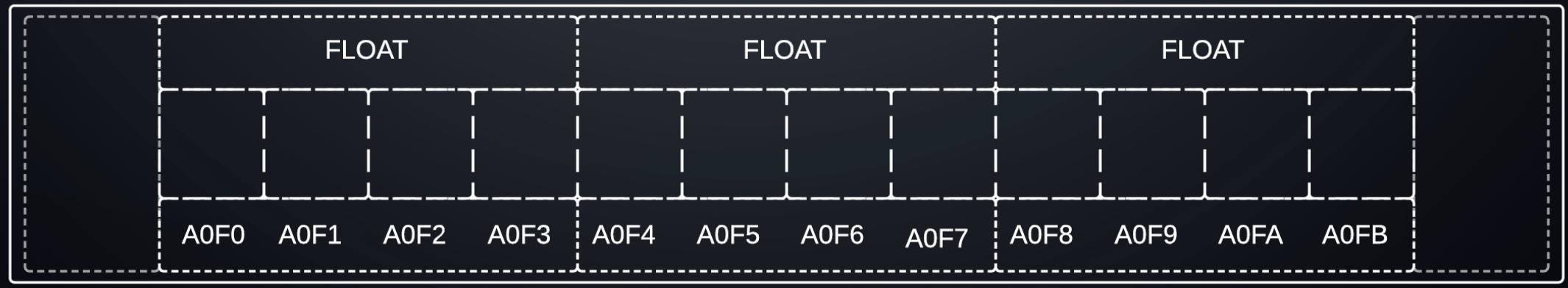
Хранится в виде последовательности данных указанного типа

Есть возможность обратится к каждому элементу



МАССИВ В ПАМЯТИ

12 char`a == 3 float`a



АДРЕСАЦИЯ МАССИВА

НАЧИНАЕТСЯ С НУЛЯ

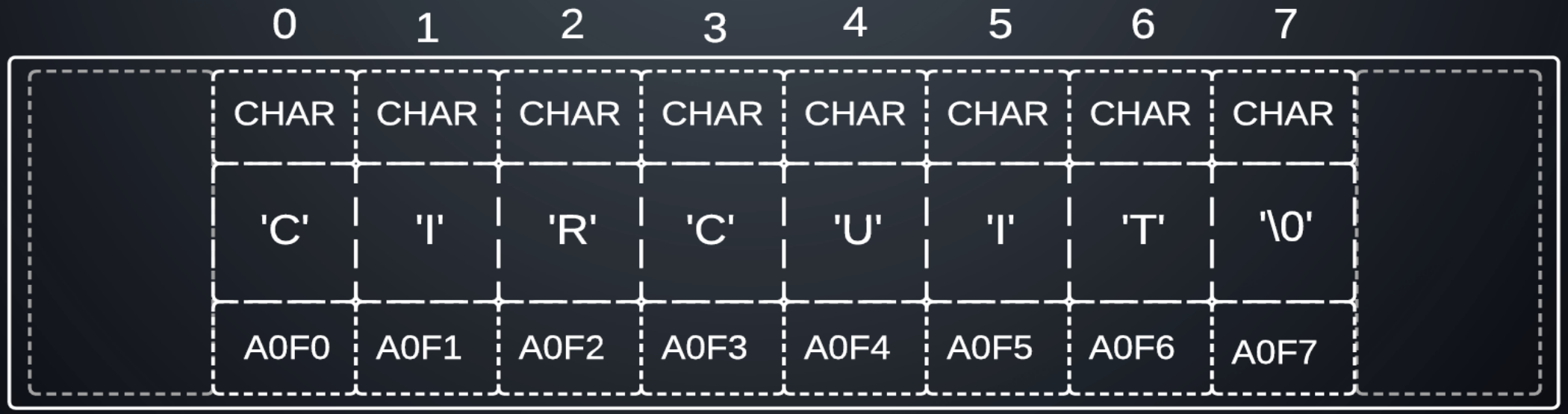
Синтаксис:

```
int myPins[] = {2, 4, 8, 3, 6};
int myElement = myPins[2];
char message[6] = "hello";
char letter = message[0];
```

```
#include <iostream>

using namespace std;


int main() {
    int array[] = {42, 997, 1337, 420};

    int firstValue = array[0];
    int secondValue = array[1];
    int lastValue = array[3];

    printf("First value: %d\n", firstValue);
    printf("Second value: %d\n", secondValue);
    printf("Last value: %d\n", lastValue);
}
```

МАССИВ В ПАМЯТИ

```
char message[] = "CIRCUIT";
```



ДВУМЕРНЫЙ МАССИВ

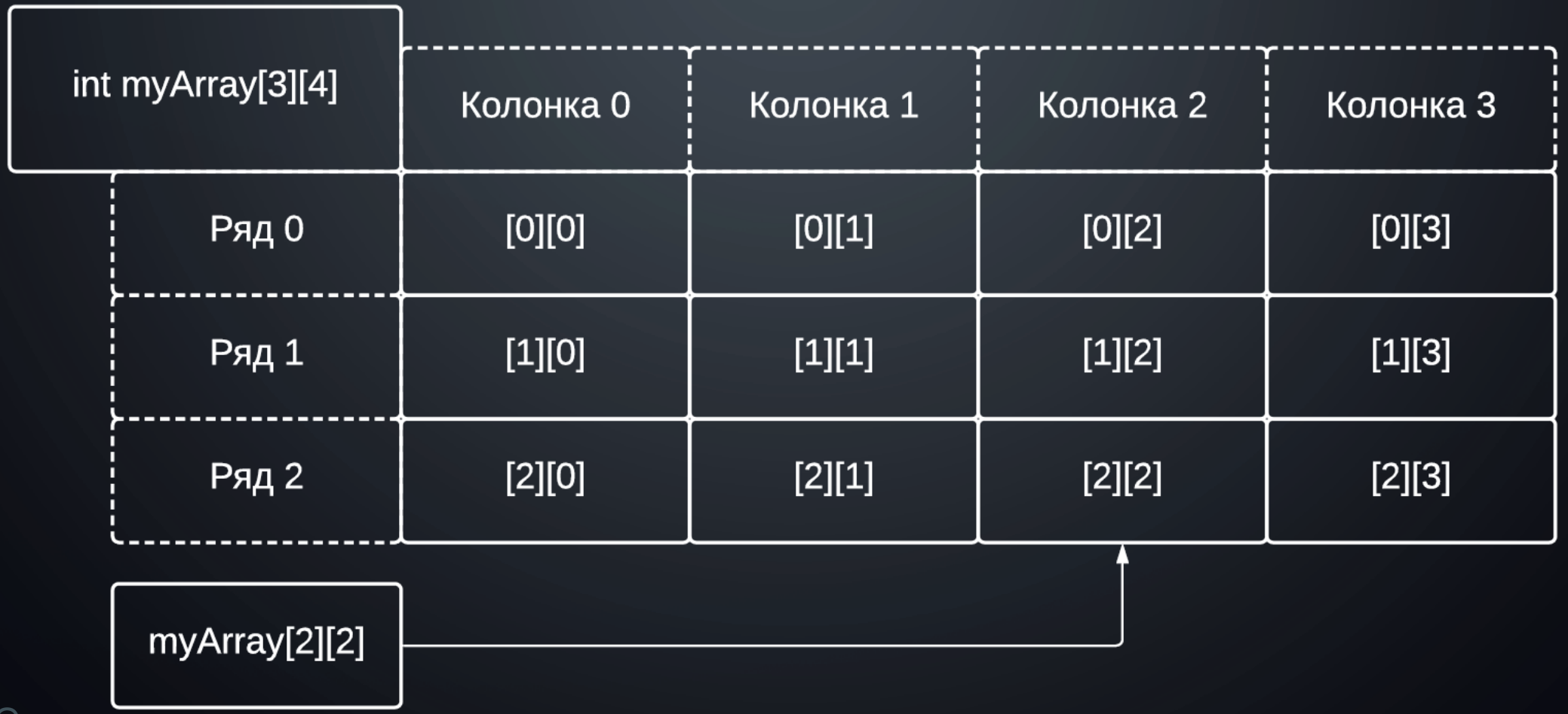
МАССИВ, КОТОРЫЙ ХРАНИТ СЕБЕ МАССИВЫ

Синтаксис:

```
int a[2][2] = {{1, 2}, {3, 4}};
int b[2][2] = {{1}, {3, 4}};
int numberA = a[1][1];
int numberB = b[0][3];
```

```
#include <iostream>

using namespace std;

int main() {
    int array1[2][2] = {
            {1, 2}, {3, 4}
        };
    int array2[2][2] = {
            {1}, {3, 4}
        };

    int numberA = array1[1][1];
    int numberB = array2[0][3];

    printf("%d\n", numberA);
    printf("%d\n", numberB);
}
```

АДРЕСАЦИЯ ДВУМЕРНОГО МАССИВА



СТРОКА (STRING)

Могут быть представлены двумя способами char-массивом или std:string

Если задавать первым способом, необходимо указать или оставить место для нулевого конца /0.

Синтаксис:

```
char message[6] = "hello";
string stringOne = "Hello String";
```

```
#include <iostream>

using namespace std;

int main() {
    string str = "Hello!";
    printf("%s\n", str.c_str());

    char char_array[] = "Hello!";
    printf("%s\n", char_array);
}
```

СПАСИБО ЗА ВНИМАНИЕ!