

# **ЗАНЯТИЕ №2**

---

# МАССИВЫ

Массив – структура данных, представленная в виде группы ячеек одного типа, объединенных под одним единым именем.

«Массив (array) – это последовательность элементов, в которой доступ к каждому элементу осуществляется с помощью его индекса (позиции).»

5	-12	-12	9	10	0	-9	-12	-1	23	65	64	11	43	39	-15
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]

# МАССИВЫ

- **СТАТИЧЕСКИЕ**

- **ДИНАМИЧЕСКИЕ**

# МАССИВЫ

```
int a[10];
```

```
int a[]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

```
int a[16] = { 5, -12, 64, 11, 43, 39, -15 };
```

```
int a[10][10];
```

```
int a[][]={{10,9},{8, 7}};
```

# ДИНАМИЧЕСКИЕ МАССИВЫ.

- Динамические массивы создаются в качестве указателя на некоторую область памяти.

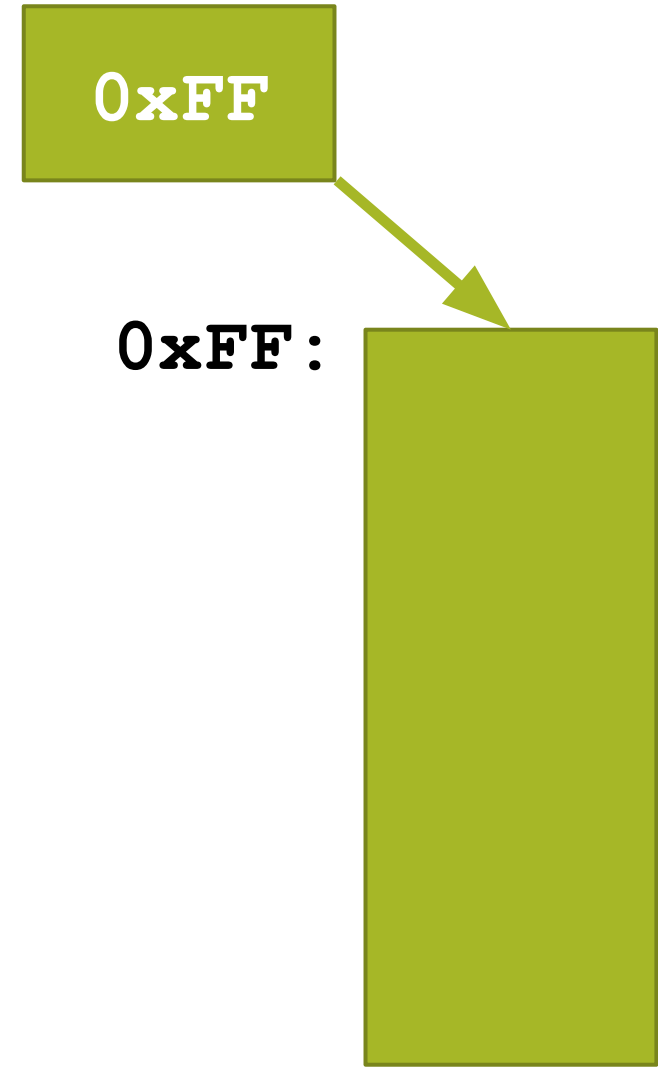
```
int *a=new int [10]; /*это указатель  
на область памяти, где хранится 10  
элементов типа int*/
```

# ДИНАМИЧЕСКИЕ МАССИВЫ.

```
int **a=new int *[n];  
for (int i=0;i<n;i++)  
    a[i]=new int [m];
```

# УКАЗАТЕЛЬ

- Массив – это именованная область памяти, где имя массива является указателем на эту область.
- Указатель показывает на место в памяти, где хранится элемент.



# УКАЗАТЕЛИ

Значением указателя является первый байт выделенной памяти

Указатель не знает, на какое число элементов он указывает


Указатель указывает на объект определенного типа



# УКАЗАТЕЛЬ

- При работе с динамическим массивом возможно выходить за пределы изначально объявленной длины.

```
int *y=new int[1];  
    y[0]=1;  
    y[1]=2;  
    y[2]=3;
```



Так делать не  
надо

# VECTOR.

- vector – реализация динамического массива переменного типа

```
#include <vector>
```

```
int main()
```

```
{
```

```
    vector <int> vec = new vector();
```

```
    vector <vector<int>> vec2=new vector();
```

```
    vector <int> vec3={1, 2, 3, 4, 5, 6};
```

```
Return 0;
```

```
}
```

# VECTOR.

Операции над векторами:

`size`, `max_size`, `capacity` – определение размера, максимального размера и емкости вектора

`insert`, `push_back` – вставка в указанное место одного\нескольких элементов и вставка одного элемента в конец вектора

`pop_back`, `erase` – удаление последнего элемента из конца и удаление нескольких элементов из вектора

`resize` – изменение размера вектора

`swap` – обмен содержимого векторов

`clear` – очистка (удаление всех элементов) вектора