



ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
**ВЫСШАЯ ШКОЛА ЭКОНОМИКИ**

# Учебная практика ОСНОВЫ C++

**День 1**  
**Массивы**

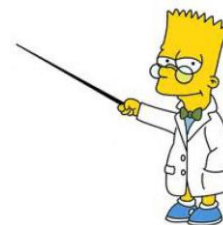


# Массивы

**Массив** – это упорядоченная последовательность переменных одного типа. Индексация – с нуля.

Выход за пределы массива НЕ контролируется компиляторами!!!

Проверьте на примере



Массивы могут быть статические и динамические.

Различаются в т.ч. расположением в памяти и главное – очисткой памяти после использования.



# Статические массивы

Одномерные массивы. Описание и инициализация

```
int n = 20;  
int mas[100];  
int mas1[3] = {5, 15, 7};
```

Многомерные массивы. Описание и инициализация

```
int mas[10][20];  
int mas1[3][2] = {5, 15, 7, 2, 4, 8};
```

$$mas1 = \begin{bmatrix} 5 & 15 \\ 7 & 2 \\ 4 & 8 \end{bmatrix}$$



# Статические массивы.

## Примеры

```
int n = 5;  
int d[3];  
int a[3] = { 1, 2, 3 };  
int c[3 + 1];  
int e[3 + 1] = { 5, 6, 7, 8 };
```

**ДА**

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

**ДА**

```
//int b[n];
```

**НЕТ**



# Динамические массивы. Освобождение памяти

```
int n = 5;  
int *mas;           // Объявление динамического массива  
mas = new int [n];  // выделение памяти в куче  
int * a = new [10];  
...  
delete [] mas;      // память обязательно освободить  
delete [] a;
```



Опасность – утечка памяти!

Если написать

```
delete mas;
```

освободится только память нулевого  
элемента массива.



Имя массива без индекса = указатель на нулевой элемент массива



# Динамические массивы.

## Примеры

```
int n = 5;  
int *arr = new int[n];  
int *att = new int[n+1];  
int *arr1 = new int[n] {1,2,3};
```

**ДА**

```
//int *arr1 = new int[n] = {1,2,3};
```

**НЕТ**



# Массивы и указатели

Имя массива без индекса = указатель на нулевой элемент массива

```
double a[5], b;  
double *da = a;  
//указатель da направлен на нулевой элемент массива  
...  
b = a[3];  
b = *(da + 3); // указатель на 3-й элемент массива  
b = da[3];     // индексированный указатель
```



Очень полезно на будущее

# Двумерные динамические массивы

```
int rows = 3, cols = 7;
int **mas = new int*[rows];
for (int i = 0; i < rows; i++) {
    mas[i] = new int[cols];
}
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        mas[i][j] = rand() % 200;
        cout << mas[i][j] << "\t";
    }
    cout << endl;
}
for (int i = 0; i < rows; i++)
    delete[] mas[i];
delete[] mas;
```

**\*\*mas** - указатель на массив указателей  
размерностью rows.

Занимаем память размером  
`sizeof(int*) * rows`.

**mas[i]** - указатель на массив  
интов размерностью cols.  
Занимаем память размером  
`sizeof(int) * cols`.

Заполнение случайными целыми

Форматирование вывода

Очистка памяти

Сначала каждая строка  
Затем «массив массивов»





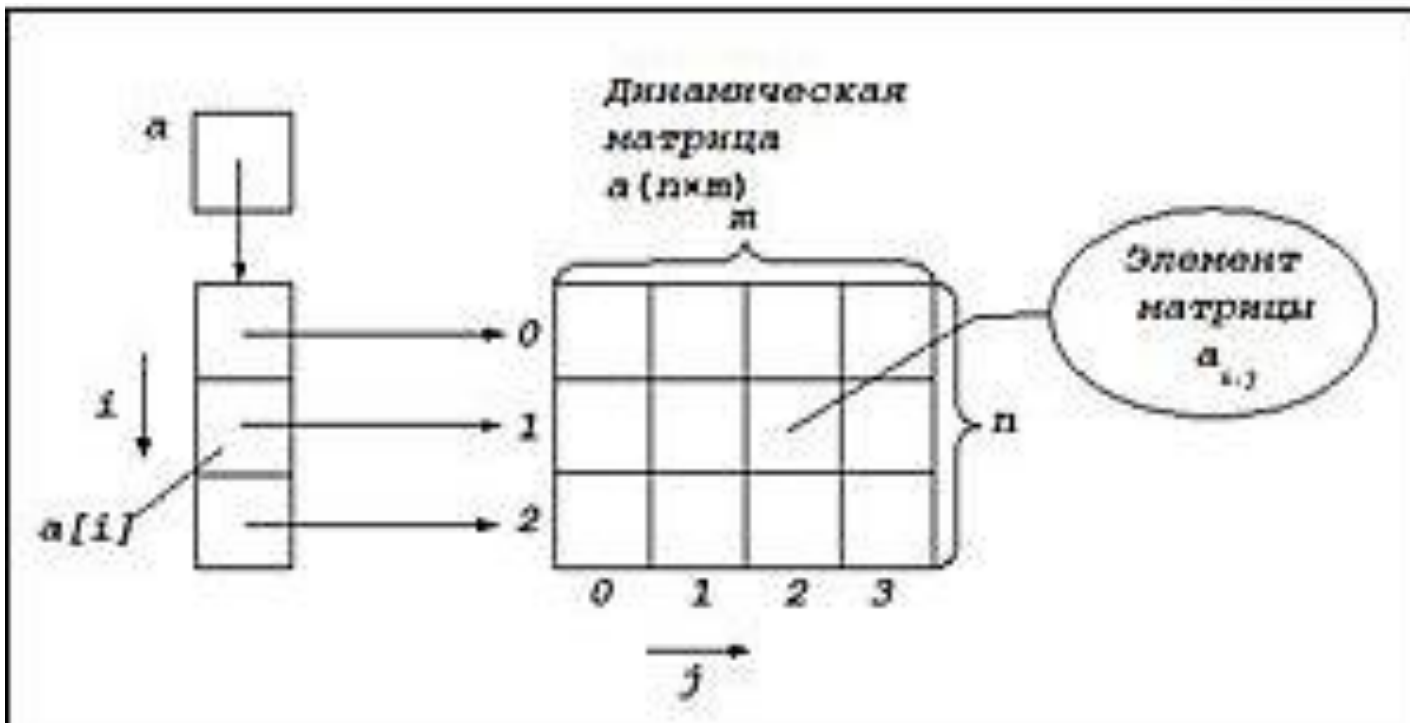
Очень полезно на будущее

# Двумерные динамические массивы

```
1 2 3 4 5 6 7
41    67    134    100    169    124    78
158    162    64    105    145    81    27
161    91    195    142    27    36    191
```

Для продолжения нажмите любую клавишу


Форматирование вывода





# Массивы

## Два типа массивов в C++

| Статический (локальный)                                   | Динамический   |
|---|--|
| Может быть только размера определенной константы          | Размер может зависеть от какой-либо переменной   |
| Нельзя удалить из памяти «руками», удалится автоматически | Нужно удалять из памяти «руками», иначе останется в памяти навсегда -> утечка памяти   |
| <pre>int arr[3] = {1, 2, 3};</pre>                        | <pre>int n=5; int* arr1 = new int[n]; int* arr2 = new int[n] {5,3,2,6,7}; //int* arr2 = new int[n] = {5,3,2,6,7};ошибка</pre>  |
| Удаление не нужно   | <pre>delete[] arr1; delete[] arr2; // delete[] arr1, arr2; // не освободится память под arr2</pre>  |