

# Семинар №4

ФАКИ 2017

---

Бирюков В. А.

October 9, 2017

# Массивы

---

- Массив – набор элементов, расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по индексу
- Объявление:

```
type arrayName [ arraySize ];
```

- Доступ к элементу (Нумерация в массиве начинается с 0):

```
arrayName [ index ];
```

Объявление:

```
int array[10];  
float average_temperature[12];
```

Доступ к элементу:

Нумерация в массиве начинается с 0

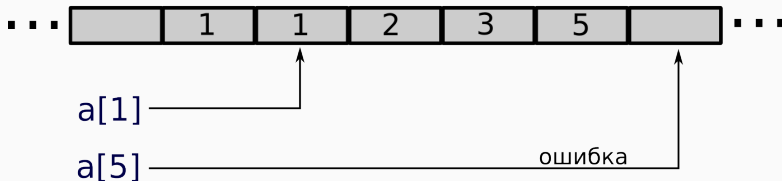
```
printf("%d\n", array[9]);  
average_temperature[2] = 5.2;
```

```
int array[10];  
for (int i = 0; i < 10; ++i) {  
    array[i] = /* something */;  
}
```

Или так:

```
int array[5] = {1, 1, 2, 3, 5};
```

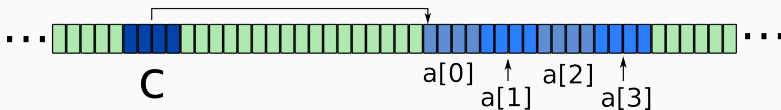
```
int a[5] = {1, 1, 2 ,3, 5};
```



```
int a[4] = {1, 2, 3, 4};
```

```
int * c = &a[0];
```

```
*c = 5;
```



# Многомерные массивы

---



```
type name[size1][size2]...[sizeN];
```

например объявление двумерного массива:

```
int array[5][10];
```

задание значений:

```
for (int i = 0; i < 5; ++i)
    for (int j = 0; j < 10; ++j)
        array[i][j] = something;
```

```
/* массив из 5-ти рядов и 2-х столбцов */  
int array[5][2] = { {0,0}, {1,2}, {2,4}, {3,6},{4,8}};  
int i, j;  
  
/* вывод всех элементов массива */  
for ( i = 0; i < 5; i++ )  
{  
    for ( j = 0; j < 2; j++ )  
    {  
        printf("%d\n", array[i][j] );  
    }  
}
```

# Введение в алгоритмы

---

- Алгоритм – это формально описанная вычислительная процедура, получающая исходные данные, и выдающая результат вычислений на выход (Кормен и др. "Алгоритмы: построение и анализ")

- Задана последовательность чисел
- Нужно найти такую перестановку исходной последовательности, чтобы элементы были расположены по возрастанию
- 5 2 4 6 1 3 2 9  $\rightarrow$  1 2 2 3 4 5 6 9

- Сортировка вставками
- Сортировка выбором
- Сортировка пузырьком

- Обычно изучают зависимость времени работы от размера входа
- Размер входа – зависит от конкретной задачи
- Для сортировки, размер входа – это количество элементов, которые нужно отсортировать
- Время работы – число элементарных шагов, которые выполняет алгоритм

```
for (int j = 0; j < length-1; j++)  
    for (int i = 0; i < length-1; i++)  
        if (a[i] > a[i+1])  
            swap(&a[i], &a[i+1]);
```

функция `swap(int, int)` меняет значения 2-х переменных

- Число операций, требуемых на один проход:  $a * n$
- Число проходов:  $n$
- Значит, время работы  $\sim n^2$